SingCERT
Singapore Cyber Emergency Response Team

Security Bulletin 23 September 2026

Generated on 23 September 2026

SingCERT's Security Bulletin summarises the list of vulnerabilities collated from the National Institute of Standards and Technology (NIST)'s National Vulnerability Database (NVD) in the past week.

The vulnerabilities are tabled based on severity, in accordance to their CVSSv3 base scores:

Critical	vulnerabilities with a base score of 9.0 to 10.0
High	vulnerabilities with a base score of 7.0 to 8.9
Medium	vulnerabilities with a base score of 4.0 to 6.9
Low	vulnerabilities with a base score of 0.1 to 3.9
None	vulnerabilities with a base score of 0.0

For those vulnerabilities without assigned CVSS scores, please visit NVD for the updated CVSS vulnerability entries.

CRITICAL VULNERABILITIES

CVE Number	Description	Base Score	Reference
CVE-2026-20192	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20192 are related to improper access control issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-284.	10.0	More Details
CVE-2026-85889	Missing authentication for critical function in Azure AI Foundry allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-76460	A vulnerability in an API of Cisco Identity Services Engine (ISE) could allow an unauthenticated, remote attacker to bypass authentication. This vulnerability is due to insufficient authentication control on an API endpoint. An attacker could exploit this vulnerability by sending a crafted request to an affected API endpoint. A successful exploit could allow the attacker to gain unauthorized access to the affected device by bypassing the web-based management interface.	10.0	More Details
CVE-2026-70200	Improper limitation of a pathname to a restricted directory ('path traversal') in Azure Logic Apps allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-69399	Azure Arc Elevation of Privilege Vulnerability	10.0	More Details
CVE-2026-70416	Dell ObjectScale, versions prior to 4.4.0.0, contains a Deserialization of Untrusted Data vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Remote execution.	10.0	More Details
CVE-2026-83944	Improper access control in Azure Logic Apps allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-54734	Prebid Server Java is the Java version of Prebid Server. Prior to 3.43.0, certain bidder adapters interpolate user-supplied parameters into outbound request URLs without using HttpUtil to validate the resulting domain or path segment. A malicious actor who can supply bid-request parameters can cause the server to send HTTP requests to unintended destinations, potentially reaching internal network services, metadata endpoints, or other sensitive server endpoints with the server's network access. This issue is fixed in version 3.43.0.	10.0	More Details
CVE-2026-20130	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC), engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20130 are related to improper neutralization of special elements issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-74.	10.0	More Details
CVE-2026-76423	A vulnerability in the REST API of Cisco ISE and Cisco ISE-PIC could allow an unauthenticated, remote attacker to gain administrative access to an affected device. This vulnerability is due to the REST API web service being exposed with insufficient authorization checks. An attacker could exploit this vulnerability by sending a crafted HTTP request to the exposed REST API port. A successful exploit could allow the attacker to read and modify ISE configuration and identity data with administrative privileges.	10.0	More Details
	Under certain circumstances on affected platforms running Arista EOS with gRPC Network Packet Sampling Interface		

CVE-2026-73456	(gNPSI) enabled, an unauthenticated gNPSI client can craft a malicious request to allow arbitrary code execution, granting an attacker full administrative control over the compromised switch.	10.0	More Details
CVE-2026-62104	Unauthenticated Remote Code Execution (RCE) in Migratico Lite <= 2.6.8 versions.	10.0	More Details
CVE-2026-92937	vm2 3.11.6 is vulnerable to a sandbox escape leading to remote code execution in the host Node.js process. The fix for GHSA-m283-3h24-438v is incomplete: the bridge gate at lib/bridge.js:1624 identity-checks only the direct call target when deciding whether to rebuild/sanitise a rejected host Promise value. Registering the rejection handler through Function.prototype.call or .apply indirection (e.g., p.then.call(p, undefined, cb)) makes the intercepted target host Function.prototype.call, so the sanitiser never runs and the raw host error reaches sandbox code with its own properties intact. If an embedder exposes a host-realm Promise to the sandbox (an async host function bridged via the sandbox option, or a NodeVM external module's async method) and that Promise rejects with an Error carrying a non-primitive own property referencing a host object (for example err.detail = process), untrusted code in the sandbox obtains a fully functional proxy to that host object and can execute arbitrary commands with the privileges of the host process (e.g., e.detail.mainModule.require('child_process').execSync(...)). The direct p.then(undefined, cb), bind, and Reflect.apply forms are correctly sanitised. Fixed in vm2 3.11.7.	10.0	More Details
CVE-2026-92941	vm2 versions from 3.11.3 before 3.11.7 expose the host tls module to NodeVM sandbox code, allowing attackers to call tls.setDefaultCACertificates() and replace process-wide certificate authorities. Attackers with access to allowed tls and url builtins can use URLSearchParams to create host-realm arrays and manipulate the TLS trust store, enabling subsequent host HTTPS clients to accept attacker-controlled certificates.	10.0	More Details
CVE-2026-92946	vm2 before 3.11.7 contains a remote code execution vulnerability when require.external is enabled without an explicit require.root that excludes node_modules. Sandboxed code can require vm2's own package, instantiate an unrestricted NodeVM instance, and execute arbitrary host OS commands via child_process.	10.0	More Details
CVE-2026-92947	vm2 before 3.11.7 exposes Node's shared Buffer pool to sandboxed code, allowing disclosure of host memory used by Buffer.from, Buffer.concat, and related allocations. Sandboxed code can read and write to host-realm buffers by acquiring ArrayBuffers from small allocations, leading to sensitive data exposure and potential denial-of-service.	10.0	More Details
CVE-2026-92940	vm2 versions 3.11.3 through 3.11.6 expose the host process's real https.globalAgent to sandboxed code when a NodeVM is explicitly configured to allow require('https'). The builtin loader wraps host modules in a read-only proxy, but method calls such as Agent.prototype.on() are forwarded to the underlying host object, so sandbox code can register a listener for the agent's 'free' event. When an unrelated host HTTPS request releases a pooled connection, the listener receives the live host request options and the host TLSSocket, allowing sandboxed code to read the host's Authorization header and private destination host/port, attach a data listener to the released socket and read subsequent host response bodies in plaintext, and issue attacker-chosen authenticated requests using the stolen credentials. The issue is fixed in 3.11.7.	10.0	More Details
CVE-2026-69865	Authorization bypass through user-controlled key in Microsoft Container Registry allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-92953	vm2 versions from 3.11.0 before 3.11.8 fail to protect host TypedArray and ArrayBuffer prototypes from sandbox mutation. Attackers can use prototype-walking primitives to reach and modify host Uint8Array.prototype, %TypedArray%.prototype, and ArrayBuffer.prototype, causing host-created typed arrays to observe attacker-controlled properties after VM.run() returns.	10.0	More Details
CVE-2026-73453	An unauthenticated P4Runtime (Programming Protocol-Independent Packet Processors Runtime) client can achieve arbitrary code execution under certain conditions on affected platforms running Arista EOS configured with P4Runtime. P4Runtime is disabled by default in Arista EOS. By crafting a malicious packet during the initiation of a P4Runtime session, an attacker can obtain complete administrative control over the compromised switch. This issue was discovered internally by Arista, and the company is not aware of any malicious exploitation of this vulnerability in customer networks.	10.0	More Details
CVE-2026-92955	vm2 before 3.11.8 contains a sandbox escape vulnerability in NodeVM that allows attackers to access the host __proto__ getter/setter through console._stdout and console._stderr. Attackers can overwrite EventEmitter.prototype.emit and trigger process events to execute code with process context, bypassing code generation restrictions.	10.0	More Details
CVE-2026-62874	Insufficient verification of data authenticity in Azure Billing allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-92956	vm2 versions 3.10.1 through 3.11.6 contain a sandbox escape reachable from a default `new VM()` sandbox when running on Node.js 26. WebAssembly.compileStreaming and WebAssembly.instantiateStreaming can produce a raw host-realm Promise that rejects with a host-realm error object; by controlling Symbol.species via Promise.prototype.finally, sandbox code receives that raw host error, walks from the host error constructor to the host Function constructor, and recovers the real host `process` object, gaining host Node.js capabilities (e.g. access to host modules such as fs) in the context of the process running the sandbox. No NodeVM, require permission, host object injection, or otherwise unsafe configuration is required. This is a bypass of the fix for GHSA-6j2x-vhqr-qr7q, which removed the JSPI entry points WebAssembly.promising and WebAssembly.Suspending. The issue is fixed in 3.11.7.	10.0	More Details
CVE-2026-92960	vm2 before 3.11.6 fails to restrict access to os and dns builtins under the builtin: ['*'] configuration, allowing sandbox code to read host process identity and network topology. Attackers can invoke dns.setServers() to hijack the host process DNS resolver globally, redirecting all subsequent host DNS queries through an attacker-controlled resolver.	10.0	More Details
CVE-2026-69843	Authentication bypass by spoofing in Microsoft Fabric allows an unauthorized attacker to elevate privileges over a network.	10.0	More Details
CVE-2026-	vm2 before 3.11.7 contains an incorrect authorization vulnerability in the external package allowlist check that uses non-exact substring matching instead of full package-name boundary validation. Attackers can bypass the allowlist by		

92951	requiring a colliding package name that contains an allowlisted package substring, causing vm2 to load and execute unauthorized host packages in the host context.	9.9	More Details
CVE-2026-20234	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20234 are related to insufficiently protected credentials issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-522.	9.9	More Details
CVE-2026-20322	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20322 are related to improper access control issues that are grouped under the Common Weakness Enumeration (CWE) CWE-284.	9.9	More Details
CVE-2026-20324	A vulnerability in the sftunnel inter-device communication protocol of Cisco Secure Firewall Management Center (FMC) Software could allow an authenticated, remote attacker to execute arbitrary commands as root. This vulnerability exists because a registered sftunnel peer has incorrect permissions to write an arbitrary file to any location on the device. An attacker could exploit this vulnerability by hijacking the sftunnel communication connection or being a valid registered sftunnel peer and sending an sftunnel command to write a malicious file to the disk of an affected device. A successful exploit could allow the attacker to write a file to the device that is executed with root privileges. To exploit this vulnerability, the attacker must have valid user credentials on the affected device.	9.9	More Details
CVE-2026-20325	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20325 are related to improper neutralization of special elements used in a command issue that are grouped under the Common Weakness Enumeration (CWE) CWE-77.	9.9	More Details
CVE-2026-20332	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20332 are related to improper access control issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-284.	9.9	More Details
CVE-2026-85878	Improper authorization in Azure Database for PostgreSQL allows an authorized attacker to elevate privileges over a network.	9.9	More Details
CVE-2026-92948	vm2 versions >= 3.9.6 and <= 3.11.6 are affected by a NodeVM builtin allowlist bypass that permits a sandbox escape on Node.js 24 and newer when the embedder explicitly allows the node:test builtin (e.g. require: { builtin: ['node:test'] }). On Node.js 24+, module.builtinModules exposes the scheme-only key node:test, which is not covered by vm2's family-based DANGEROUS_BUILTINS protection, so it is stored in the generic host-passthrough loader. Because requireImpl() in lib/setup-node-sandbox.js strips a single 'node:' prefix before the builtin lookup, sandbox code calling require('node:node:test') resolves to the stored node:test key and receives a readonly proxy to the host module. Calls to node:test.run() are forwarded to the host implementation, which spawns a separate Node process for process-isolated test execution and passes through attacker-controlled execArgv values; supplying --eval=<JavaScript> therefore executes arbitrary JavaScript in an unrestricted host Node process outside the NodeVM sandbox. Fixed in vm2 3.11.7.	9.9	More Details
CVE-2026-85885	Improper neutralization of special elements used in a command ('command injection') in M365 Copilot allows an authorized attacker to elevate privileges over a network.	9.9	More Details
CVE-2026-20330	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20330 are related to improper neutralization issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-707.	9.9	More Details
CVE-2026-92938	vm2 versions 3.11.3 through 3.11.6 expose Node.js's host node:sqlite module to code running in NodeVM when that builtin is permitted, either explicitly or through builtin: ['*']. The module is wrapped with vm.readonly(), which prevents property assignment but leaves host-authority callables reachable; in addition, the resolver treats any request starting with 'node:' as a core-module request and the runtime strips only one 'node:' prefix, so a sandbox request for 'node:node:sqlite' resolves to the configured node:sqlite entry. Sandboxed code can therefore create an in-memory DatabaseSync with extension loading enabled and call DatabaseSync.loadExtension() on a native library bundled in the untrusted plugin package (path derived from __dirname). SQLite loads the library into the Node.js host process and invokes its native entry point, giving the sandboxed plugin arbitrary native code execution outside the sandbox with the host process's privileges. The issue is fixed in vm2 3.11.7.	9.9	More Details
CVE-2026-92939	vm2 3.11.3 through 3.11.6 exposes the host Node.js crypto module to a NodeVM sandbox when the crypto builtin is allowed. The module is presented via a recursive read-only proxy, but its callable exports still execute with host-process authority. Sandboxed JavaScript can therefore call crypto.setEngine() with a filesystem path to an attacker-supplied native library (for example, one bundled in an untrusted plugin package already written to disk); OpenSSL asks the operating-system dynamic loader to load the file, and the library's constructor executes native code in the host process before engine-symbol validation rejects it. Exploitation requires only the crypto builtin and does not require fs, process, module, child_process, worker_threads, vm, or inspector access, resulting in a sandbox escape and arbitrary native code execution. Fixed in 3.11.7.	9.9	More Details
	A vulnerability in the web-based management interface of Cisco ISE could allow an authenticated, remote attacker to		

CVE-2026-20307	execute arbitrary commands on the underlying operating system of an affected device. To exploit this vulnerability, the attacker must have at least low-privileged administrative credentials. This vulnerability is due to insecure deserialization of a user-supplied Java byte stream. An attacker could exploit this vulnerability by sending a crafted serialized Java object to the web-based management interface of an affected device. A successful exploit could allow the attacker to execute arbitrary code on the device and elevate privileges to root. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a denial of service (DoS) condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored.	9.9	More Details
CVE-2026-20329	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20329 are related to issues concerning improper handling of exceptional conditions that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-703.	9.9	More Details
CVE-2026-92957	vm2 through 3.11.6 does not normalize `node:`-prefixed builtin specifiers when evaluating user-supplied negative (deny) entries in a NodeVM wildcard require policy. Although NodeVM strips the `node:` prefix during require() resolution, negative wildcard entries are matched by exact string comparison against the canonical builtin names, so a policy such as `new NodeVM({ require: { builtin: ['*', '-node:child_process'] } })` fails to deny the canonical `child_process` module. Sandboxed code can therefore obtain the host `child_process` builtin via `require('child_process')` or `require('node:child_process')`, gaining references to process-spawning APIs such as execSync and spawn, which is equivalent to host command-execution capability for untrusted sandbox code. Fixed in vm2 3.11.7. (Suggested title: "vm2 before 3.11.7: NodeVM builtin deny-list bypass via node:-prefixed specifiers exposes child_process")	9.9	More Details
CVE-2026-14349	The TrueBooker – Appointment Booking and Scheduler System plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 1.2.3. This is due to the plugin not properly verifying that a user is authorized to perform an action. This makes it possible for unauthenticated attackers to modify the email address of arbitrary user accounts, including administrators, which can be leveraged to reset the account's password and gain access to it.	9.8	More Details
CVE-2026-54460	OpenReception's appointment booking software provides an end-to-end encrypted appointment booking platform. Prior to 1.1.1, POST /api/auth/passkeys accepts a request-body userId and attacker-supplied passkey without an authenticated session, does not call WebAuthnService.verifyRegistration, and does not bind enrollment to locals.user.id. An unauthenticated attacker who knows the public tenant ID and the target staff email can use the public booking bootstrap and GET /api/tenants/[id]/appointments/staff-public-keys to obtain candidate userId values. The attacker first causes UserService.addAdditionalPasskey to store a controlled public key for a candidate userId, then attempts login with the target email; the login check compares verificationResult.userId with the email-resolved account and reveals whether the injected credential belongs to that target. Repeating this injection-before-login sequence identifies the matching userId, and the normal login endpoint accepts the attacker's assertion for the stored key and creates a STAFF session. The session can expose tenant data and reveal TENANT_ADMIN identifiers for further takeover; GLOBAL_ADMIN accounts are not reachable through this tenant-scoped path. A hijacked TENANT_ADMIN can modify or delete tenant resources and key shares, potentially making appointment data permanently undecryptable and taking booking services offline. This issue is fixed in version 1.1.1.	9.8	More Details
CVE-2025-56563	A Server-Side Request Forgery vulnerability exists in sat_proxy.php in Zenith Satellite Tracker 1.0. The script accepts an attacker-controlled address URL parameter and passes it to curl_setopt(CURLOPT_URL) without host or scheme validation. An unauthenticated remote attacker can leverage this to make arbitrary HTTP and HTTPS requests from the server to internal networks or cloud metadata services, potentially obtaining sensitive information or pivoting to further attacks.	9.8	More Details
CVE-2026-45140	Chamilo LMS is an open-source learning management system. Prior to 2.0.1, Chamilo LMS allows an unauthenticated remote attacker to execute arbitrary code on the server. The authoritative advisory does not identify the affected endpoint, component, input, or exploitation mechanism. This issue is fixed in version 2.0.1.	9.8	More Details
CVE-2026-20326	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20326 are related to missing authentication for critical function issues that are grouped under the Common Weakness Enumeration (CWE) CWE-306.	9.8	More Details
CVE-2025-55787	In MailData Email Archiving System v4.2 and earlier, a SQL injection vulnerability exists.	9.8	More Details
CVE-2026-20242	A vulnerability in the External Database Access feature of Cisco Secure Firewall Management Center (FMC) Software could allow an unauthenticated, remote attacker to execute arbitrary commands as root on an affected device. This vulnerability is due to insecure deserialization of a user-supplied Java byte stream from a host that is configured in the external database access list. An attacker could exploit this vulnerability by sending a crafted, serialized Java byte stream to a specific TCP port of an affected device. A successful exploit could allow the attacker to execute arbitrary commands on the device and elevate privileges to root. Notes: This vulnerability can be exploited only by an attacker who has control of a host in the external database access list. If the FMC management interface does not have public internet access, the attack surface that is associated with this vulnerability is reduced.	9.8	More Details
CVE-2026-12793	The JetFormBuilder — Dynamic Blocks Form Builder plugin for WordPress is vulnerable to Privilege Escalation in all versions up to, and including, 3.6.2. This is due to the plugin not validating that a submitted form ID belongs to a JetFormBuilder form before parsing the referenced post's content as form schema and executing an Advanced Validation server-side callback. This makes it possible for unauthenticated attackers to create a new administrator-level user account.	9.8	More Details

CVE-2026-92787	Feast through 0.66.0 fails to verify JWT token signatures before establishing user identity, allowing attackers to bypass all role-based access control by presenting an unverified token with a hardcoded claim value. Attackers can obtain trusted internal identity and gain unchecked read and write access to all entities, feature views, data sources, and permission policies on the server.	9.8	More Details
CVE-2026-62108	Unauthenticated Broken Authentication in Headless Single Sign On <= 1.7.0 versions.	9.8	More Details
CVE-2026-92489	In the Linux kernel, the following vulnerability has been resolved: xfrm: Fix skb double-free in xfrm_dev_direct_output() A return value other than 1 from local_out() means that the skb has been consumed or its ownership was transferred. xfrm_dev_direct_output() nevertheless frees the skb on this path, causing a double-free when netfilter drops the packet and invalidating any other owner. Return the local_out() result directly, matching the ownership handling in xfrm_output_resume().	9.8	More Details
CVE-2026-90173	In the Linux kernel, the following vulnerability has been resolved: smb: smbdirect: free completion queues with ib_free_cq() smbdirect_connection_destroy_qp() creates the send and receive completion queues with ib_alloc_cq_any(), which for IB_POLL_WORKQUEUE arms an internal completion handler that runs ib_cq_poll_work() on a workqueue. Tearing those CQs down with ib_destroy_cq() frees them without first cancelling that poll work. If the provider posts a completion late -- for example Soft-RoCE (rxe) posting an RNR error from rxe_receiver() after rdma_destroy_qp() -- the handler re-queues ib_cq_poll_work() on the already-freed CQ, and a follow-on access faults in rxe_req_notify_cq(). Use ib_free_cq(), which cancel_work_sync()es the poll work before freeing the CQ, so no completion handler can run against a freed queue. [1236.599526] ========== [1236.602142] BUG: KASAN: slab-use-after-free in ib_cq_poll_work+0xd0/0x1a0 [1236.605524] Read of size 8 at addr ffff888111865800 by task kworker/4:1H/82 [1236.609017] [1236.609270] CPU: 4 UID: 0 PID: 82 Comm: kworker/4:1H Not tainted 7.2.0-rc3-next-20260717-virtme #110 PREEMPT(lazy) [1236.609287] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 [1236.609498] Workqueue: ib-comp-wq ib_cq_poll_work [1236.609525] Call Trace: [1236.609536] <TASK> [1236.609545] __dump_stack+0x21/0x60 [1236.609562] dump_stack_lvl+0xc2/0x100 [1236.609573] print_address_description+0x77/0x200 [1236.609587] ? ib_cq_poll_work+0xd0/0x1a0 [1236.609597] print_report+0x58/0x70 [1236.609607] kasan_report+0x117/0x150 [1236.609623] ? ib_cq_poll_work+0xd0/0x1a0 [1236.609636] ? process_scheduled_works+0x954/0x1600 [1236.609650] ib_cq_poll_work+0xd0/0x1a0 [1236.609662] ? process_scheduled_works+0x954/0x1600 [1236.609674] process_scheduled_works+0xc22/0x1600 [1236.609698] ? __pfx_process_scheduled_works+0x10/0x10 [1236.609713] ? __pfx_assign_work+0x10/0x10 [1236.609726] ? lock_is_held_type+0x7b/0x110 [1236.609741] worker_thread+0x975/0xee0 [1236.609757] ? __pfx_do_raw_spin_lock+0x10/0x10 [1236.609775] ? __kthread_parkme+0x21e/0x260 [1236.609789] kthread+0x3a6/0x490 [1236.609800] ? __pfx_worker_thread+0x10/0x10 [1236.609809] ? __pfx_kthread+0x10/0x10 [1236.609820] ret_from_fork+0x55a/0xa20 [1236.609835] ? __pfx_ret_from_fork+0x10/0x10 [1236.609850] ? __pfx_kthread+0x10/0x10 [1236.609861] ret_from_fork_asm+0x1a/0x30 [1236.609880] </TASK> [1236.609886] [1236.661292] Allocated by task 5076: [1236.662640] kasan_save_track+0x3e/0x80 [1236.663842] __kasan_kmalloc+0x72/0x90 [1236.664763] __kmalloc_noprof+0x2b0/0x5d0 [1236.665356] __ib_alloc_cq+0x284/0x1000 [1236.666573] __ib_alloc_cq_any+0x23e/0x340 [1236.668654] smbdirect_connection_create_qp+0x6f7/0x1070 [1236.669757] smbdirect_accept_connect_request+0x500/0x1ca0 [1236.672625] smbdirect_listen_rdma_event_handler+0x1655/0x1c50 [1236.673930] cma_listen_handler+0x1bf/0x260 [1236.674923] cma_cm_event_handler+0x128/0x380 [1236.676926] cma_ib_req_handler+0x2d3d/0x4de0 [1236.678368] cm_process_work+0xb0/0x530 [1236.680454] cm_queue_work_unlock+0xb1/0x230 [1236.681673] cm_work_handler+0x969f/0xdca0 [1236.682704] process_scheduled_works+0xc22/0x1600 [1236.683447] worker_thread+0x975/0xee0 [1236.685901] kthread+0x3a6/0x490 [1236.688164] ret_from_fork+0x55a/0xa20 [1236.689522] ret_from_fork_asm+0x1a/0x30 [1236.690073] [1236.690378] Freed by task 5137: [1236.692242] kasan_save_track+0x3e/0x80 [1236.694272] kasan_save_free_info+0x40/0x50 [1236.695514] __kasan_slab_free+0x3a/0x60 [1236.696773] kfree+0x14e/0x4e0 [1236.697216] ib_destroy_cq_user+0x18d/0x250 [1236.699817] smbdirect_connection_destroy_qp+0xf2/0x280 [1236.702115] smbdirect_socket_destroy_sync+0x1607/0x2720 [1236.704062] smbdirect_socket_release+0x140/0x280 [1236.705286] smb_direct_free_transpor ---truncated---	9.8	More Details
CVE-2026-90151	In the Linux kernel, the following vulnerability has been resolved: NFSv4: remove callback IDR entry on client allocation failure nfs4_alloc_client() allocates an NFSv4.0 callback identifier before it finishes setting up the client. If any later initialization step fails, the error path frees the nfs_client directly with nfs_free_client(). That bypasses nfs_put_client(), which is where the callback IDR entry is removed during normal teardown. A failed allocation can therefore leave cb_ident_idr pointing at a freed nfs_client. A later NFSv4.0 callback lookup by cb_ident would find the stale pointer and take a reference to it. Make the callback IDR removal helper callable by the allocation failure path, and remove the callback identifier before freeing the client. This was found by a local static-analysis checker for publish-before-free lifetime bugs and confirmed by manual inspection.	9.8	More Details
CVE-2026-90104	In the Linux kernel, the following vulnerability has been resolved: NFSv4.1: zero referring call lists before decoding decode_cb_sequence_args() allocates csa_rclists with kmalloc_objs(), so each referring_call_list starts uninitialized. decode_rc_list() assigns rcl_refcalls only when rcl_nrefcalls is nonzero. A valid list with zero referring calls therefore leaves the pointer uninitialized, and nfs4_callback_sequence() later passes stale slab contents to kfree(). Allocate csa_rclists with kzalloc_objs() so every rcl_refcalls member is NULL from the beginning, including valid empty referring call lists.	9.8	More Details
	pgAdmin 4's Webserver authentication source is intended to accept an identity asserted by the web server or reverse proxy in front of pgAdmin, delivered through the WSGI/CGI environment. WebserverAuthentication.get_user() read config.WEBSERVER_REMOTE_USER from request.environ and, when that returned nothing, fell back to reading the same name directly from the inbound HTTP request headers via request.headers.get(). An inbound HTTP header is written by whoever sends the request, so any client able to reach pgAdmin could supply that header itself and be authenticated as any username it named, including an existing Administrator, without presenting a password or any other credential. The environment lookup could also be satisfied by a client-supplied header whenever WEBSERVER_REMOTE_USER was configured to an HTTP_-prefixed or hyphenated name such as HTTP_X_FORWARDED_USER or X-Forwarded-User, since WSGI servers place inbound headers into the environment		

CVE-2026-86863	under exactly those names. Deployments are affected only when 'webserver' is enabled in AUTHENTICATION_SOURCES. The fix distinguishes a genuine CGI/WSGI variable from a header-derived one and implicitly trusts only the former. A header-asserted identity is now accepted only when the operator explicitly opts in via WEBSERVER_REMOTE_USER_FROM_HEADER, the request arrives from a peer listed in WEBSERVER_TRUSTED_PROXIES, and, when configured, a shared secret supplied in WEBSERVER_SHARED_SECRET_HEADER matches WEBSERVER_SHARED_SECRET under a constant-time comparison. The trusted-peer check deliberately reads the real socket peer address rather than request.remote_addr, because ProxyFix rewrites the latter from the client-controlled X-Forwarded-For header and would otherwise allow an attacker to claim to be the trusted proxy. As defence in depth, login() now refuses any account whose auth_source is not 'webserver', so a misconfigured trust gate cannot be used to assume an internal or LDAP account. This issue affects pgAdmin 4: from 6.2 before 9.18.	9.8	More Details
CVE-2026-92944	vm2 versions 3.10.2 through 3.11.6 contain a sandbox escape vulnerability on Node.js 26 where Promise.prototype.finally() bypasses vm2's wrapper protections due to a stale PromiseThenLookupChain protector in V8 14.6. Attackers can exploit this by creating an async function that returns a Promise with an attacker-controlled constructor Symbol.species, allowing them to reach the host Function constructor and process object for arbitrary code execution.	9.8	More Details
CVE-2026-54617	GravitLauncher is an open-source Minecraft launcher based on sashok724's v3. Prior to 5.7.12, an unauthenticated remote actor can send a raw HTTP request target without a leading slash to the default LaunchServer file server on port 9274. FileServerHandler.channelRead0 in components/launchserver/src/main/java/pro/gravit/launchserver/socket/handlers/fileserver/FileServerHandler.java strips the first request-target character and resolves the remaining path against updatesDir without re-normalizing and verifying containment. This leaves parent-directory components in a no-leading-slash request and allows reading any file accessible to the LaunchServer process, including .keys/ecdsa_id, .keys/legacySalt, and LaunchServer.json. Disclosure of those files can expose signing keys, refresh-token material, and database credentials, enabling forged administrative access tokens and full authentication bypass. A normalizing L7 proxy may block the primary request form, but direct exposure and L4/TCP proxies remain affected, and netty.fileServerEnabled is enabled by default. This issue is fixed in 5.7.12.	9.8	More Details
CVE-2026-62101	Unauthenticated Broken Authentication in EduAdmin Booking <= 5.4.2 versions.	9.8	More Details
CVE-2026-92805	UVdesk Community Skeleton through 1.1.8 fails to authenticate or validate installation state on wizard endpoints in ConfigureHelpdesk controller actions. Unauthenticated attackers can repoint the database and create super administrator accounts by submitting crafted requests to wizard endpoints, gaining full control of the instance.	9.8	More Details
CVE-2026-54626	SAIL is a cross-platform library for loading and saving images with support for animation, metadata, and ICC profiles. In 0.9.10 and earlier, the TGA_INDEXED_RLE path selected by image_type == 9 allocates an image buffer using the one-byte-per-pixel SAIL_PIXEL_FORMAT_BPP8_INDEXED format returned by tga_private_sail_pixel_format() in src/sail-codecs/tga/helpers.c, while sail_codec_load_frame_v8_tga() in src/sail-codecs/tga/tga.c derives a two-to-four-byte pixel_size from an attacker-controlled header bpp value from 9 through 32. Loading a crafted color-mapped run-length-encoded TGA through sail_load_from_file() or sail_load_from_memory() therefore writes attacker-controlled bytes beyond the heap pixel buffer. The pixel-count clamp added for CVE-2026-40494 does not constrain the per-pixel write width, so this issue is an incomplete fix of that vulnerability and can cause heap corruption, a reliable crash, or potential code execution. This issue is fixed in version 1.0.0.	9.8	More Details
CVE-2026-90822	FatPipe MPVPN, WARP, and IPVPN appliances running the end-of-life firmware version 10.1.2r60p100 contain an OS command injection vulnerability in the xtremed daemon. An unauthenticated remote attacker with access to the affected management interface can submit crafted input to the AuthFormServlet endpoint, causing authentication data to be processed by a shell and allowing arbitrary commands to execute as root. The affected management interface is disabled by default and must be affirmatively enabled by the customer before the endpoint becomes reachable. FatPipe recommends restricting management access to trusted administrative networks and using WAN access control lists to limit access to trusted sources. Customers running the affected end-of-life firmware can contact FatPipe Support for help confirming their firmware version and upgrading to a current supported release at https://www.fatpipeinc.com/support/support, support@fatpipeinc.com, or +1 800-724-8521 (option 3).	9.8	More Details
CVE-2026-54627	SAIL is a cross-platform library for loading and saving images with support for animation, metadata, and ICC profiles. In 0.9.10 and earlier, psd_private_sail_pixel_format() in src/sail-codecs/psd/helpers.c resolves a one-channel PSD in Bitmap color mode to SAIL_PIXEL_FORMAT_BPP1_INDEXED without requiring the file depth to be one, so the pixel buffer uses one-bit rows while sail_codec_load_frame_v8_psd() in src/sail-codecs/psd/psd.c accepts depth == 8 and writes one attacker-controlled byte per pixel. Loading a crafted PSD through sail_load_from_file() or sail_load_from_memory() therefore writes beyond each heap row, causing memory corruption, a reliable crash, or potential code execution. This mode/depth mismatch is distinct from GHSA-rcqx-gc76-r9mv and GHSA-wcj8-hxxf-pq2c. This issue is fixed in version 1.0.0.	9.8	More Details
CVE-2026-86710	The Login with QR WordPress plugin through 1.0.0 does not verify that the code used to log a user in is one it issued, matching any stored user metadata value instead, which allows unauthenticated attackers to log in as any user, including administrators.	9.8	More Details
CVE-2026-86709	The Pressengine WordPress plugin through 1.0 does not stop its login handler from issuing a session when authentication fails, allowing unauthenticated attackers to log in as any user, including administrators.	9.8	More Details
CVE-2026-86707	The Private Feed Key WordPress plugin through 0.1 does not verify that the key used to authenticate a feed request is one it issued, matching any stored user metadata value instead, which allows unauthenticated attackers to log in as any user, including administrators.	9.8	More Details
CVE-2026-87796	The Multi Uploader for Gravity Forms plugin for WordPress is vulnerable to Arbitrary File Upload in all versions up to, and including, 1.1.9 via the move_file function. This is due to insufficient file type validation during chunked upload handling. This makes it possible for unauthenticated attackers to upload arbitrary files on the affected site's server	9.8	More Details

	which may make remote code execution possible.		
CVE-2026-90823	FatPipe MPVPN, WARP, and IPVPN appliances running the end-of-life firmware version 10.1.2r60p100 contain a stack-based buffer overflow in /usr/sbin/auth_user_pass. An unauthenticated remote attacker with access to the affected management interface can submit a crafted authentication request that reaches an unchecked copy into a fixed-size stack buffer, potentially allowing arbitrary code execution as root. The affected management interface is disabled by default and must be affirmatively enabled by the customer before the endpoint becomes reachable. FatPipe recommends restricting management access to trusted administrative networks and using WAN access control lists to limit access to trusted sources. Customers running the affected end-of-life firmware can contact FatPipe Support for help confirming their firmware version and upgrading to a current supported release at https://www.fatpipeinc.com/support/support, support@fatpipeinc.com, or +1 800-724-8521 (option 3).	9.8	More Details
CVE-2026-91103	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91106	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-89857	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Hold qpair lock when sending NVMe LS reject qla_nvme_ls_reject_iocb() allocates from and advances the request ring through __qla2x00_alloc_iocbs() (which assumes the hardware_lock is held) and qla2x00_start_iocbs() (which advances the ring and rings the request-in doorbell), but takes no lock itself. Two of its callers invoke it without the producer lock held: - qla_nvme_xmt_ls_rsp(), the NVMe-FC .xmt_ls_rsp transport callback, on its error path, and - qla2xxx_process_purls_pkt(), run from the purex work/DPC context. Both use ha->base_qpair, whose qp_lock_ptr is hardware_lock, so they can run concurrently with normal I/O submission on the base ring and corrupt the ring producer state, leading to duplicated or dropped commands. The third caller, qla2xxx_process_purls_iocb(), runs inside qla24xx_process_response_queue() with the qpair lock already held and is safe; that is also why the lock cannot be taken inside the helper itself (it would recursively re-acquire hardware_lock on the response path). Take qp_lock_ptr around the two unlocked callers and document the helper as caller-locked. Both run in process context, so spin_lock_irqsave() is used and nothing in the locked region sleeps.	9.8	More Details
CVE-2026-90048	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: fix slab-out-of-bounds write in ni_create_attr_list() ni_create_attr_list() allocates a fixed buffer of al_aligned(record_size) (== record_size) bytes and then walks every attribute of the primary MFT record, writing one ATTR_LIST_ENTRY per attribute and advancing the cursor by le_size(name_len), with no check against the end of the buffer; the total size is only computed after the loop. A minimum-size resident attribute occupies SIZEOF_RESIDENT (0x18 = 24) bytes on disk, but an unnamed attribute expands to le_size(0) (0x20 = 32) bytes in the list. Because the number of attributes in a record is not bounded (mi_enum_attr() accepts arbitrarily many equal-type, nameless minimum-size attributes), a crafted record packed with such attributes produces a list larger than record_size and overflows the heap buffer. This is reachable from a crafted, loop-mounted NTFS image: opening the file and adding an attribute (e.g. via setxattr) drives ntfs_set_ea() -> ni_insert_resident() -> ni_insert_attr() -> ni_ins_attr_ext() -> ni_create_attr_list(). BUG: KASAN: slab-out-of-bounds in ni_create_attr_list+0xc48/0x1058 Write of size 4 at addr ffff000008984c00 by task setfattr/345 ni_create_attr_list+0xc48/0x1058 ni_ins_attr_ext+0x510/0x7c0 ni_insert_attr+0x3f8/0x70c ni_insert_resident+0xc8/0x3b0 ntfs_set_ea+0x66c/0xd28 ntfs_setxattr+0x4d8/0x5b0 __arm64_sys_setxattr+0xa4/0x124 Allocated by task 345: ni_create_attr_list+0x188/0x1058 The buggy address belongs to the cache kmalloc-1k of size 1024 (the write lands at object+1024). Size the buffer from the actual attributes instead of assuming a single record_size is always enough.	9.8	More Details
CVE-2026-90042	In the Linux kernel, the following vulnerability has been resolved: ceph: properly decrypt filenames in vmalloc() buffers The fscrypt subsystem uses the scatterlist crypto API, inheriting its requirement that any buffers are in the linear mapping region. However, the messenger client uses kvmalloc() to create buffers for messages, which will occasionally place those buffers in the vmalloc() region when physical memory fragmentation doesn't permit a large enough kmalloc(). The various callers of ceph_fname_to_usr() directly pass (slices of) raw messages from the MDS without considering that the messages may be in vmalloc() buffers, resulting in oopses especially on non-x86 platforms (see 'Closes:' for more details and a reproducer). Make ceph_fname_to_usr() explicitly tolerant of vmalloc()-allocated fname->ctext, fname->name, and/or oname->name buffers, using `tname` (which, when non-null, must be a linear address; when null, is briefly allocated as necessary) as a bounce buffer to avoid passing any inappropriate addresses to fscrypt_fname_disk_to_usr(). Additionally change parse_reply_info_readdir() -- the only function to supply its own `tname` -- to follow the new "tname must never come from vmalloc()" rule by passing NULL when the message is not in the linear region. Though this causes a per-dentry kmalloc()+kfree(), this overhead exists only when processing the minority of messages that spill into vmalloc(). My (crude) testing puts this at only about 1 in 8,000 readdir messages. Still, if the overhead proves unreasonable in the future, it is easy enough to mitigate: a future change could allocate a bounce buffer in parse_reply_info_readdir() and use that as `tname` instead.	9.8	More Details
CVE-2026-90038	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent client use-after-free during export state revocation nfsd4_revoke_export_states() has the same use-after-free as nfsd4_revoke_states(): it drops nn->client_lock across revoke_one_stid() and the following read of clp->cl_minorversion, but the stateid reference it holds does not pin the client. A teardown racing the dropped lock can free the client while revoke_one_stid() still dereferences it. exportfs -u drives this path through NFSD_CMD_UNLOCK_EXPORT, so an administrator removing an export can race a client expiry. Skip a client that is already expiring and otherwise pin it with cl_rpc_users under client_lock before dropping the lock, matching nfsd4_revoke_states().	9.8	More Details
CVE-2026-90036	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent client use-after-free during blocked-lock reaping A bare lock owner -- its only remaining reference a blocked lock on nn->blocked_locks_lru -- holds a raw pointer to its nfs4_client but no reference keeping the client alive. When the per-net laundromat reaps such a lock, freeing the nbl drops the owner reference held through flc_owner, and the final nfs4_put_stateowner() takes the client's cl_lock. Because the laundromat detaches the nbl first, __destroy_client() no longer finds it, so a concurrent force_expire_client() can free the client before nfs4_put_stateowner() runs, dereferencing cl_lock in freed memory. Pin	9.8	More Details

	the client with cl_rpc_users before dropping nn->blocked_locks_lock, and skip clients already expiring, whose blocked locks __destroy_client() frees while holding an owner reference. Take nn->client_lock outside nn->blocked_locks_lock. Every other site holds nn->blocked_locks_lock as a leaf, acquiring no further lock, so placing nn->client_lock outside it cannot form a lock-order cycle.		
CVE-2026-90012	In the Linux kernel, the following vulnerability has been resolved: spi: Fix DMA mapping ownership on partial map failure If RX mapping fails after TX mapping succeeds, __spi_map_msg() unmaps TX but leaves tx_sg_mapped set. If TX mapping fails on a later transfer, mappings created for earlier transfers remain active. In both cases, cur_{tx,rx}_dma_dev have not yet been updated because they are assigned only after every transfer has been mapped. The subsequent spi_unmap_msg() may therefore unmap the TX mapping again or release earlier mappings using a NULL or stale device. Using a NULL device can trigger an oops. An empty SG table does not prevent the NULL dereference because dma_unmap_sg_attrs() accesses the device before checking the entry count. Publish both mapping devices before mapping starts and unwind all failures through __spi_unmap_msg(). This clears the mapping flags and releases each mapping once with the device that created it. Publishing the devices before the loop also refreshes them when no transfer needs mapping. No mapping flag is set in that case, so current users do not use the pointers as mapping owners.	9.8	More Details
CVE-2026-89990	In the Linux kernel, the following vulnerability has been resolved: ceph: lock mutex in ceph_mds_check_access() MDS session OPEN handling replaces mdsc->s_cap_auths under mdsc->mutex, freeing the previous array and its strings. ceph_mds_check_access() traverses this array without holding the mutex. A concurrent session reopen can therefore free the array while it is being inspected, resulting in a use-after-free like this: Unable to handle kernel paging request at virtual address 003aaad64b2c8bb9 [...] Internal error: Oops: 0000000096000004 [#1] SMP Modules linked in: CPU: 56 UID: 2953037534 PID: 1253231 Comm: php-cgi8.4 Not tainted 6.18.45-i2-ampere #1146 NONE [..] pc : ceph_mds_check_access+0xd4/0x550 lr : ceph_mds_check_access+0xc8/0x550 [...] Call trace: ceph_mds_check_access+0xd4/0x550 (P) ceph_atomic_open+0x138/0xbe8 path_openat+0xa24/0xfa8 do_filp_open+0x94/0x158 do_sys_openat2+0x88/0xf8	9.8	More Details
CVE-2026-89972	In the Linux kernel, the following vulnerability has been resolved: nvme: add missing SRCU grace period in error path nvme_alloc_ns() error path at out_unlink_ns removes ns from the namespace head siblings list with list_del_rcu(&ns->siblings) but does not wait for SRCU readers before freeing the namespace struct. Multipath code iterates the head->list under srcu_read_lock() in nvme_find_path() and nvme_mpath_revalidate_paths(), so a concurrent reader can still hold a reference to ns when kfree(ns) runs. The normal removal path in nvme_ns_remove() correctly calls synchronize_srcu(&ns->head->srcu) after list_del_rcu() to wait for in-progress readers. Add the same grace period in the error path.	9.8	More Details
CVE-2026-89970	In the Linux kernel, the following vulnerability has been resolved: nvmet-auth: Synchronize timeout work during SQ teardown nvmet_auth_sq_free() cancels auth_expired_work with cancel_delayed_work(). If the work has already started, cancellation does not wait for the callback. Transport teardown can consequently free or reuse the queue containing struct nvmet_sq while nvmet_auth_expired_work() still accesses that SQ. Add a teardown-specific helper that synchronously drains the delayed work before freeing authentication state, and use it from nvmet_sq_destroy(). Keep the non-synchronous helper for in-band authentication state cleanup, where the SQ owner remains alive.	9.8	More Details
CVE-2026-89969	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: fix out-of-bounds write when receiving an over-long PDU nvmet_tcp_try_recv_pdu() reads a PDU header into the fixed 128-byte queue->pdu union, then computes the remaining payload length as queue->left = hdr->hlen - queue->offset + hdgst; and reads that many more bytes into &queue->pdu + queue->offset, without ever bounding the result against sizeof(queue->pdu). A struct nvme_tcp_icreq_pdu is itself 128 bytes, exactly the size of the union. Once a header digest has been negotiated (hdgst = 4), a second ICReq passes the hlen == nvmet_tcp_pdu_size() check but yields queue->left = 128 - 8 + 4 = 124, so bytes 8..132 are written into the 128-byte buffer -- 4 bytes past its end, over queue->hdr_digest and queue->data_digest. Those bytes are attacker-controlled (an ICReq carries no digest), and the duplicate ICReq is only rejected later, after the overflow. A remote unauthenticated host can thus corrupt kernel memory adjacent to the receive buffer. Reject any PDU whose declared length would read past the end of queue->pdu before the second recv.	9.8	More Details
CVE-2026-89847	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Avoid double completion in async IOCB timeout qla2x00_async_iocb_timeout() tries to abort a timed-out async IOCB. When qla24xx_async_abort_cmd() fails, both the SRB_LOGIN_CMD path and the SRB_CTRL_VP/default path scan outstanding_cmds[] for the SRB and then call sp->done(sp, QLA_FUNCTION_TIMEOUT) unconditionally, without checking whether the SRB was actually found and removed. If the response ISR completes the same handle first, it removes the SRB under qp_lock_ptr and runs sp->done() -> complete(sp->comp). The submitter qla24xx_control_vp() wakes from wait_for_completion(), clears sp->comp, drops its reference and returns, reclaiming the on-stack completion. The timer reference keeps the SRB alive across the timeout handler, but not the submitter's stack. The timeout then issues a second sp->done() -> qla_ctrlvp_sp_done(), which evaluates "if (sp->comp) complete(sp->comp)"; with the pointer loaded before the submitter's NULL store, complete() writes into the freed stack frame, a use-after-free. Track whether this path removed the SRB from outstanding_cmds and only call sp->done() when it did, so the command is completed exactly once by whichever path owns it. This mirrors the sp_found guard already used in qla24xx_abort_iocb_timeout().	9.8	More Details
CVE-2025-59953	LMDeploy is a toolkit for compressing, deploying, and serving large language models. Starting in version 0.9.1 and prior to version 0.10.2, the LMdeploy implements an rpc server (AsyncRPCServer in zmq_rpc.py) for supporting the RPC communications. In its core functionality call_and_response(), I found it will directly use the pickles.loads() to deserialize the received messages without any sanitization, hence resulting in a remote code execution vulnerability by this RPC server. Version 0.10.2 contains a patch.	9.8	More Details
CVE-2026-76187	Apache Airflow Keycloak provider: the unauthenticated token endpoint accepts a client-credentials grant for any confidential client registered in the Keycloak realm, not only the client configured for Airflow. No allowlist restricts which client ids may authenticate, so the credentials of an unrelated application that happens to share the realm are valid Airflow login credentials, and Airflow mints a signed session token for that application's service account. The endpoint also answers unauthenticated credential guesses against Keycloak under Airflow's identity. Affects deployments using the Keycloak auth manager whose realm is shared with other confidential clients. The attacker needs valid credentials for any one of those clients, not for Airflow. Resource authorization is still evaluated per subject, so the access gained is whatever that service account holds, plus any endpoint gated only on being authenticated.	9.8	More Details

	Users of apache-airflow-providers-keycloak are recommended to upgrade to version 0.10.0 or later, which accepts only the configured client on that grant.		
CVE-2026-89788	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix tree connection use-after-free in smb2_tree_connect() ksmbd_tree_conn_connect() publishes a new tree connection in sess->tree_conns with a single reference and returns its pointer to smb2_tree_connect(). The handler continues to initialize the object and build the response after publication. A concurrent session logoff can erase the connection and drop that reference, freeing the object while the handler still uses it. BUG: KASAN: slab-use-after-free in smb2_tree_connect+0xe3d/0xf90 smb2_tree_connect (fs/smb/server/smb2pdu.c:2872) handle_ksmbd_work process_one_work worker_thread kthread After xa_store() succeeds, take a second reference before releasing tree_conns_lock. The original reference belongs to the xarray entry and the second belongs to the creating smb2_tree_connect() handler. Keep the references balanced in every path: - On normal exit or an error after publication, smb2_tree_connect() drops its creator reference. Error cleanup also calls ksmbd_tree_conn_disconnect(), which drops the xarray reference only if it removes the exact entry. - SMB2 TREE_DISCONNECT uses the same helper to remove the entry and drop its xarray reference. The request's existing lookup reference remains owned by the request and is released by the existing cleanup. - Session LOGOFF removes each entry and drops its xarray reference. If it wins the race, later cleanup sees that the entry is gone and does not drop that reference again. To enforce this ownership, claim the disconnected state and erase the exact entry atomically under tree_conns_lock. This guarantees one drop for the xarray reference and one drop by each in-flight user, regardless of which teardown path wins. If logoff removes the entry before initialization completes, fail the connect instead of marking the detached object TREE_CONNECTED.	9.8	More Details
CVE-2026-89783	In the Linux kernel, the following vulnerability has been resolved: xfrm6: fix out-of-bounds write in xfrm6_input_addr() when secpath is full The depth check in xfrm6_input_addr() is off by one: if (1 + sp->len == XFRM_MAX_DEPTH) goto drop; ... sp->xvec[sp->len++] = x; xfrm_input() can leave sp->len == XFRM_MAX_DEPTH, and the transport-mode receive path re-enters IPv6 input via xfrm_trans_reinject() with that secpath preserved. If the inner packet carries a destination-options HAO option or a type-2 routing header, xfrm6_input_addr() is called with sp->len == XFRM_MAX_DEPTH; the check (1 + 6 == 6) is false, so sp->xvec[sp->len++] writes one slot past the 6-element xvec[]. The write stays within the sec_path allocation (invisible to KASAN); UBSAN_BOUNDS flags it and panics under panic_on_warn. Use "sp->len >= XFRM_MAX_DEPTH", matching xfrm_input(). This also restores one chain level the old check rejected at sp->len == 5. UBSAN: array-index-out-of-bounds in net/ipv6/xfrm6_input.c:309:10 index 6 is out of range for type 'xfrm_state *[6]'	9.8	More Details
CVE-2026-89778	In the Linux kernel, the following vulnerability has been resolved: isofs: fix out-of-bounds page array access on empty zisofs block zisofs_uncompress_block()'s empty-block fast path returns pcount << PAGE_SHIFT, ignoring the incoming poffset, unlike the decompression path which returns bytes produced relative to poffset. zisofs_fill_pages() uses that return to advance its page cursor, so when the zisofs block size is below PAGE_SIZE and a sub-page block leaves poffset partway into a page, a following empty block over-counts and advances pages[] one element past its end, after which "if (poffset && *pages)" reads pages[1] out of bounds. rock.c only rejects a block-size shift > 17, so a crafted "ZF" Rock Ridge record can set it below PAGE_SHIFT; the bug is reached by an ordinary read() of a compressed file on such a mounted ISO9660 image. Return the byte count relative to poffset and zero only [poffset, PAGE_SIZE) of the first page, matching the decompression path. The page-aligned case (poffset == 0) is unaffected. BUG: KASAN: slab-out-of-bounds in zisofs_read_folio (fs/isofs/compress.c:290) Read of size 8 at addr ffff88800f5eac48 by task exploit/142 zisofs_read_folio (fs/isofs/compress.c:290) read_pages (mm/readahead.c:184) ... filemap_read (mm/filemap.c:2814) vfs_read (fs/read_write.c:574) __x64_sys_pread64 (fs/read_write.c:769) do_syscall_64 (arch/x86/entry/syscall_64.c:94) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:121) The buggy address is located 0 bytes to the right of the allocated 8-byte region in the kmalloc-8 cache	9.8	More Details
CVE-2026-82311	Apache Airflow FAB provider: resetting a user's password does not delete that user's existing database-backed sessions, despite documented behaviour that it does. The cleanup compares the string identifier Flask-Login stores in the session against the user's integer database identifier, so the comparison never matches and no session is removed. An attacker who already holds a copy of the victim's session cookie keeps access as that user after the password change, so the reset does not evict them. Affects deployments using the FAB auth manager with `[fab] session_backend=database`. The trigger is an administrator (or the user) running the supported password-reset command as a containment action after a session cookie has been compromised; the secure-cookie backend is out of scope, as it documents that it cannot centrally delete sessions. apache-airflow-providers-fab 3.9.0 also fixes CVE-2026-86462, a second, independent route to the same outcome via the Admin user-edit endpoint; a single upgrade closes both. Users of apache-airflow-providers-fab are recommended to upgrade to version 3.9.0 or later, which compares the identifiers consistently.	9.8	More Details
CVE-2026-81642	In NLnet Labs Unbound up to and including 1.26.0, a vulnerability was found in the DNSSEC validator that enables denial of service and possible remote code execution as a result of digesting DNSKEYs. A DNSKEY with an owner compression pointer to its own RDATA can overflow the digest buffer. Remote code execution is possible through attacker controlled data. An adversary can exploit the vulnerability by controlling a malicious zone and querying a vulnerable Unbound.	9.8	More Details
CVE-2026-27565	An unauthenticated remote attacker can upload a malicious IODD file that places and executes a shell script with root privileges. The shell script remains active even after a reboot.	9.8	More Details
CVE-2026-27546	An unauthenticated remote attacker can exploit an authentication bypass in the _account_log function to log in as an admin, even when accounts are properly configured.	9.8	More Details
CVE-2026-93467	The OAKlouds developed by HGiga has a Insecure Deserialization vulnerability. Unauthenticated remote attackers can execute arbitrary code on the server by sending maliciously crafted serialized content.	9.8	More Details
CVE-2026-91843	A stack overflow during the unauthenticated login process may allow an attacker to run arbitrary code remotely with root privileges.	9.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent client use-after-free during close_lru reaping An nfs4_openowner left on nn->close_lru after its final CLOSE keeps its last closed stateid in oo_last_closed_stid, holding only a raw pointer to its nfs4_client. The laundromat reaps timed-out entries, drops nn-		

CVE-2026-90037	>client_lock, and calls nfs4_put_stid(), which dereferences the client through cl_lock. Nothing pins the client across that window, so a concurrent force_expire_client() can free it and nfs4_put_stid() reads freed memory. __destroy_client() hits the same race, walking clp->cl_openowners without cl_lock. Pin the client with cl_rpc_users before dropping client_lock, and skip clients already expiring. __destroy_client() then cleans up its own close_lru entries through release_last_closed_stateid(), so teardown no longer races the laundromat.	9.8	More Details
CVE-2026-90235	In the Linux kernel, the following vulnerability has been resolved: sunrpc: xprtsock: annotate shared socket callbacks with READ_ONCE/WRITE_ONCE xprtsock replaces and restores sk->sk_data_ready and sk->sk_write_space on live sockets with plain stores, and xs_udp_do_set_buffer_size() invokes sk->sk_write_space via a plain load. These callback pointers are shared with generic socket and protocol paths that may read or invoke them concurrently, so xprtsock needs the same READ_ONCE()/WRITE_ONCE() callback visibility contract that the validated 4022 family applied elsewhere. When SUNRPC takes over an AF_LOCAL, UDP, or TCP socket and later restores the lower-socket callbacks during teardown, another CPU may still hold an earlier callback snapshot. The plain replace/restore pattern leaves the same visibility hole as the validated 4022 family, so a stale snapshot can still invoke xs_data_ready() or xs_udp_write_space() after the live callback fields have already been restored to the lower-socket handlers. Use WRITE_ONCE() for the shared sk_data_ready and sk_write_space stores in xs_local_finish_connecting(), xs_udp_finish_connecting(), xs_tcp_finish_connecting(), and xs_restore_old_callbacks(). Use READ_ONCE() for the direct sk_write_space invocation in xs_udp_do_set_buffer_size(). This matches the required callback visibility contract while leaving adjacent sk_state_change and sk_error_report handling unchanged.	9.8	More Details
CVE-2026-51990	An issue in Sogou Sogou Input Method < 16.3.0.3498 (fixed in 16.3.0.3498) allows a remote attacker to execute arbitrary code via the biz_helper.exe component	9.8	More Details
CVE-2026-91099	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91100	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91097	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-68536	Server-Side Request Forgery / Local File Inclusion in Apache MyFace Core. Older unsupported versions may also be affected. Users are recommended to upgrade to versions 2.3.12, 2.3-next-M9, 3.0.4, 4.0.4, or 4.1.4, which fix this issue.	9.8	More Details
CVE-2026-91101	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-90999	Sentry Seer is vulnerable to a multi-stage trust-boundary violation that allows unauthenticated attacker-controlled telemetry to become code that is executed by an agent in a privileged automation environment. An external attacker can submit fabricated Sentry events without having access to the victim’s Sentry account, source repository, or infrastructure.	9.8	More Details
CVE-2026-91102	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91104	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91105	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-91098	HP has identified and remediated multiple externally reported vulnerabilities within HPLIP. The findings affect several software components that could potentially enable remote code execution, privilege escalation, denial of service, information disclosure, or unauthorized file modification under certain conditions.	9.8	More Details
CVE-2026-54053	Many Notes is a Markdown note-taking web application designed for simplicity. Prior to 0.16.0, the ZIP vault import implemented in app/Actions/ProcessImportedVault.php accepts archive filenames containing parent-directory traversal segments. An authenticated user can write arbitrary files outside the importing user's vault and into other users' vaults, including overwriting existing files. Disguised SVG content can be placed in another user's vault and execute stored cross-site scripting when the victim opens that vault. This issue is fixed in version 0.16.0.	9.6	More Details
CVE-2026-87701	Improper neutralization of special elements in output used by a downstream component ('injection') in Azure Cosmos DB allows an authorized attacker to elevate privileges over a network.	9.6	More Details
CVE-2026-93372	Buffer overflow in WebGL in Google Chrome on on Android prior to 153.0.8010.52 allowed a remote attacker to execute arbitrary code outside the sandbox via a crafted HTML page. (Chromium security severity: Critical)	9.6	More Details
CVE-2026-93373	Use after free in Extensions in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to execute arbitrary code outside the sandbox via a crafted Chrome extension. (Chromium security severity: High)	9.6	More Details
CVE-2026-93374	Use after free in Dawn in Google Chrome on on Android prior to 153.0.8010.52 allowed a remote attacker to potentially execute arbitrary code outside the sandbox via a crafted HTML page. (Chromium security severity: Critical)	9.6	More Details

CVE-2026-86106	An unauthenticated actor with network access to the private HA interconnect may trigger sensitive HA peer functions without verification. This could result in elevated command execution on Edge units where HA is enabled.	9.6	More Details
CVE-2026-54752	NetBox Device Type Library is a collection of community-sourced device type definitions for import into NetBox. The validation test harness can deserialize pull-request-controlled tracked pickle cache files through pickle.load in the read_pickle_data function in tests/pickle_operations.py. An unauthenticated contributor can change USE_LOCAL_KNOWN_SLUGS in tests/test_configuration.py and supply a crafted tests/known-modules.pickle or tests/known-racks.pickle file that tests/definitions_test.py loads when pytest runs. Deserialization invokes attacker-controlled object reduction behavior, allowing arbitrary code execution in the GitHub Actions runner or in a maintainer process that runs the tests, with the confidentiality, integrity, and availability of reachable resources at risk. This vulnerability is fixed with commit 1c6f7e2b93589b965318c6e67ac3504831f0e71e.	9.6	More Details
CVE-2026-85385	Concrete CMS below 9.5.4 did not validate the user timezone value (uTimezone) on write and rendered it without output encoding on the Dashboard user management page, where Date::getTimezoneDisplayName() returns any non-IANA value unchanged. A stored cross-site scripting payload saved in this field executed in an administrator's browser when they viewed the affected user in the Dashboard, running script in the admin session (for example to read CSRF tokens, create administrator accounts, or change site settings). In Concrete CMS 9.5.3 the field became reachable by unauthenticated visitors through public registration; in Concrete CMS below 9.5.3, the same field was reachable by any authenticated user through the account profile editor. Exploitation required concrete.misc.user_timezones to be enabled (off by default), and the unauthenticated path additionally required public registration to be enabled. The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 7.7 with vector CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:P/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N. Thanks Suraj Bhosale for reporting.	9.6	More Details
CVE-2026-20331	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20331 are related to the failure of protection mechanisms issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-693.	9.6	More Details
CVE-2026-92716	Shuffle through 2.2.1 contains a cross-tenant privilege escalation vulnerability in the HandleApiGeneration endpoint that allows administrators to reset and read API keys of non-administrator users in other organizations. Attackers with admin privileges in one organization can supply arbitrary user IDs to generate valid API keys for users in different organizations, enabling account takeover across tenant boundaries.	9.6	More Details
CVE-2026-54618	Obsidian Web MCP is a secure remote MCP server for Obsidian vaults. Prior to 0.2.0, /oauth/authorize issues an authorization code without a login, consent, or session check, and /oauth/token can exchange that code for the static VAULT_MCP_TOKEN without authenticating a client. An unauthenticated remote caller who can reach the intended tunnel deployment can therefore call /mcp and use vault_read, vault_write, vault_search, vault_list, vault_move, and vault_delete against the entire vault. Optional PKCE does not prevent an attacker-initiated flow, and unauthenticated /oauth/register also exposes a client_credentials path by returning the configured VAULT_OAUTH_CLIENT_SECRET. This issue is fixed in version 0.2.0.	9.4	More Details
CVE-2026-90110	In the Linux kernel, the following vulnerability has been resolved: inetpeer: randomize RB-tree node comparison using SipHash The inetpeer rate limiting system stores peer entries in a Red-Black tree keyed deterministically on the remote IP address. Because tree lookups walk the RB-tree using standard lexicographical comparisons (inetpeer_addr_cmp), an off-path adversary can predict the exact topology of the tree and the sequence of nodes traversed during lookups (the gc_stack candidate list). By combining deterministic tree traversal with aggressive garbage collection (triggered when tree size exceeds inet_peer_threshold), an attacker can selectively force the eviction of targeted inet_peer nodes. When an evicted node is subsequently re-created upon receiving a new packet, its rate-limiting token bucket (rate_tokens, rate_last) is reset to full capacity. This creates a side-channel primitive allowing off-path attackers to bypass IP-keyed ICMP rate limits and infer open UDP ports (similar to SAD DNS style attacks). Mitigate this by randomizing the RB-tree node comparison logic using SipHash with a secret key (inetpeer_hash_key) initialized via net_get_random_once(). Nodes are ordered in the tree by SipHash(addr, key) rather than raw IP addresses. Because the secret key is unknown to external entities, the tree layout and lookup traversal paths are unpredictable to off-path adversaries, breaking the deterministic eviction gadget. Cache the computed 64-bit SipHash (hash) in struct inet_peer and compute the target hash (dhash) once at the beginning of inet_getpeer() to avoid recomputing SipHash at every step of the RB-tree walk.	9.4	More Details
CVE-2026-89915	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Remove VM-wide VNCR mapping counter The global VNCR mapping counter is used to decide whether an L1 provided VNCR page is mapped in L0 on any CPU at the point of dealing with a TLB invalidation. It is incremented when a mapping is made in the fixmap, and decremented when unmapped. As it turns out, this tracking has several flaws: - we are trying to invalidate TLBs, and the mapping is only an opportunistic consequence of the TLB. Checking this counter to decide whether a TLB needs to be invalidated may result in missed invalidations. - an L1 vcpu invalidating its own TLB (a very likely case) will not succeed in invalidating the VNCR pseudo TLB because that page is not mapped in L0 at this stage. Given that this tracking fails at delivering the minimum guarantees that are required and is only a performance optimisation, remove it completely.	9.3	More Details
CVE-2026-90049	In the Linux kernel, the following vulnerability has been resolved: net: skbuff: don't skb_tx_error() the source skb in skb_zerocopy() skb_zerocopy() copies frags from @from into @to. On an skb_orphan_frags() failure it calls skb_tx_error(@from), a destructive operation on the source skb the copy helper does not own. That completes @from's zerocopy uarg and clears SKBFL_ALL_ZEROCOPY, including the SKBFL_SHARED_FRAG page-ownership marker. Both callers already report the failure on their own drop path. nfnetlink_queue does it at nla_put_failure, and Open vSwitch does it in the flow-miss drop arm of ovs_dp_process_packet(), so nothing is lost by dropping it here. On Open vSwitch's OVS_ACTION_ATTR_USERSPACE path the skb is not freed on this error: do_execute_actions() ignores output_userspace()'s return value and, unless the upcall was the last action, keeps forwarding the same skb through the flow's remaining actions. The uarg is completed while that skb is still in flight, telling the producer its buffers are free, and SKBFL_SHARED_FRAG is cleared on an skb the rest of the stack still handles. That flag is what makes esp_input() call skb_cow_data() instead of decrypting in place, so a later local ESP delivery can decrypt over frags the	9.3	More Details

	skb does not own privately. Leave error reporting to the callers.		
CVE-2026-70009	Improper limitation of a pathname to a restricted directory ('path traversal') in Azure Arc allows an unauthorized attacker to elevate privileges over a network.	9.3	More Details
CVE-2026-89775	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Handle negative S1 walk levels in VNCR TLB size evaluation Computing the effects of a TLB invalidation involves looking at the size of the mapping cached by the TLB. For S1 mappings such as VNCR, this is deducted from the combination of the base granule size and the mapping level. However, this implies that the S1 MMU is *on*. When the MMU is off, we indicate this with the level being set to a "creative" value of -127 (S1_MMU_DISABLED). This ends-up being misinterpreted by pgshift_level_to_ttl() as it doesn't handle negative levels at all (the level is immediately cast to a u8 and only the bottom two bits considered), leading to an invalidation size of 0. Not helpful. Tidy-up pgshift_level_to_ttl() to handle these negative levels, and ttl_to_size() to always return SZ_1G when no valid TTL is present. This allows the removal of open-coded checks for similar situations. Note that the check for a negative value not explicitely checking for S1_MMU_DISABLED is deliberate, so that actual negative levels introduced with LVA2 and D128 can take the same path if we ever support them.	9.3	More Details
CVE-2026-89916	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Make VNCR invalidation participate in MMU invalidation retry A VNCR TLB invalidation can occur on one vcpu while another vcpu is faulting in this same page. Without correctly handling this, we can end up with the following scenario: - vcpu A walks the PTs to translate VNCR - before vcpu A is able to grab the MMU lock to insert the TLB, vcpu B updates the S1 PTs with an invalid entry, and issues a TLBI S1E2 for this VA - vcpu A inserts the TLB for something that is now invalid This isn't a new problem, and we manage S2 by having the MMU notifier to bump up mmu_invalidate_seq on invalidation so that the fault can be replayed. We can perform something similar here, and extend invalidate_vncr_va() to update the same counter, clearly indicating that the context has changed under our feet. This is safe as the invalidation always happen while holding the MMU lock for write, and that we sample the sequence number before walking S1.	9.3	More Details
CVE-2026-89930	In the Linux kernel, the following vulnerability has been resolved: KVM: nVMX: Service local TLB flushes on failed nested VM-Enter KVM services local TLB flushes on "full" nested VM-Exits (through __nested_vmx_vmexit()), but not if a nested VM-Enter fails (e.g. due to failed VMCS checks in nested_vmx_enter_non_root_mode()). However, it is possible that KVM had queued TLB flushes that need to be performed, even if the nested VM-Enter was not successful. For example, if VPID is disabled for L2 (via nested_vmx_transition_tlb_flush(), or if via the MSR load lists, as the SDM says: If any MSR is being loaded in such a way that would architecturally require a TLB flush, the TLBs are updated so that, after VM entry, the logical processor will not use any translations that were cached before the transition. The SDM is unclear about when the TLB flush should occur, and whether or not a failed VM entry would flush the TLB, so it is safer to always do the TLB flush in this case. More concretely, KVM also updates the last VPID L1 used for L2 in nested_vmx_transition_tlb_flush() (i.e. last_vpid), even if the VM entry ultimately fails. With the current code, KVM could miss a TLB flush if L1 changes L2's VPID, then does a failed VM entry followed by a successful one, as the failed VM entry would update last_vpid but not actually flush the TLB. Servicing local TLB flushes on failed VM entries makes sure that the TLB is always flushed when last_vpid is updated.	9.3	More Details
CVE-2026-89918	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Correctly handle end of VA space TLBI invalidation Our TLB invalidation by VA code is based on comparing two ranges, one defined by the TLB, and one defined by the TLBI instruction. Each range is defined by a start and a size. However, the way the comparison is done doesn't account for address rollover, as it compares an address with (base + size). This works nicely until this expression represent the last page/block in the TTBR1 VA space, as the result is a big fat 0. And a failed TLB invalidation. Rewrite the comparison in a way that is immune to the address rollover (making the end address inclusive instead of exclusive), and move this into a common helper that is used by both VA and IPA invalidations, as suggested by Hyunwoo Kim (although the IPA version didn't suffer from this particular problem, obviously).	9.3	More Details
CVE-2026-89914	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Sign-extend VA for range-based TLBI invalidation When the decode_range_tlbi() helper was moved to be used for S1 TLBIs, the required sign extension was omitted. Add it. As a result, special care must be taken to not overflow PA bits when this is used for S2 invalidation.	9.3	More Details
CVE-2026-73639	Imager::File::PNG versions from 1.003 before 1.004 for Perl write past the end of the row buffer reading a PNG with a tRNS transparency chunk in read_direct8. With a tRNS chunk, read_direct8() adds an alpha channel to the image it creates but still sizes the row buffer from the original channel count. libpng expands the transparency into that extra channel, so png_read_row() fills one channel more than the buffer holds, at one byte per sample, and writes width bytes past the end of the allocation. Palette images go to read_paletted() and 16-bit images to read_direct16(), which sizes its buffer from png_get_rowbytes() and allocates enough for the expanded row. The same reader ships bundled in the Imager distribution. Reading an attacker-supplied PNG through Imager->read() corrupts the heap, which can crash the process.	9.1	More Details
CVE-2026-54670	WeGIA is a web manager for charitable institutions. Prior to 3.8.5, the contribution request dispatcher in web/html/contribuicao/controller/control.php accepts attacker-controlled nomeClasse and metodo values without a complete controller and method allowlist, exempts sensitive ContribuicaoLogController operations from authentication, and constructs a controller include path without canonical directory containment. An unauthenticated remote attacker can invoke getContribuicoesLogJSON, sincronizarStatus, registrarFaturas, and other sensitive methods to disclose contribution and donation records or trigger financial workflow operations. A traversal-shaped nomeClasse value can also cause require_once to include an accessible PHP or configuration file outside the intended controller directory, exposing source code, credentials, or other sensitive local data. This issue is fixed in version 3.8.5.	9.1	More Details
	In the Linux kernel, the following vulnerability has been resolved: IB/isert: reject PDUs declaring more data than was received isert_recv_done() hands each received PDU to the opcode handlers without ever looking at wc->byte_len, the number of bytes the HCA actually placed in the receive descriptor. The handlers then copy that many bytes - the data-segment length the initiator declared in the BHS (ntoh24(hdr->dlength), via the derived unsol_data_len / imm_data_len) - out of the fixed-size descriptor: isert_handle_iscsi_dataout(): sg_copy_from_buffer(sg_start, sg_nents, isert_get_data(rx_desc), unsol_data_len); isert_handle_scsi_cmd(): sg_copy_from_buffer(cmd->se_cmd.t_data_sg, sg_nents, isert_get_data(rx_desc), imm_data_len); Because the declared length is never checked against wc->byte_len, an initiator can declare a data segment larger than the bytes it actually sent (and larger than the descriptor) and cause an out-of-bounds read of the receive buffer. Nothing upstream of isert closes this door: - __iscsit_check_dataout_hdr()		

CVE-2026-90414	bounds the inbound payload against conn_ops->MaxXmitDataSegmentLength (MXDSL) - a transmit parameter, used here for the inbound check. - iscsi_set_connection_parameters() sets ops->MaxXmitDataSegmentLength = ops->TargetRecvDataSegmentLength; and TARGETRECVDATASEGMENTLENGTH is absent from the min()-clamp list in iscsi_check_acceptor_state(), so the value the initiator declares is adopted verbatim (type range 512..16777215). The initiator effectively raises its own ceiling. - isert never clamps the negotiated value to its own fixed receive descriptor (ISER_RX_SIZE, 9216 bytes), so the target core's bound and the descriptor size are unrelated. The imm_data_len == data_len path is more than an over-read: it aliases the receive descriptor via sg_set_buf() and passes it to the backend as the data source for the SCSI WRITE, so an over-declared length causes heap contents past the descriptor to be written through the backend to the backing store. The backend is the victim of the oversized scatterlist isert hands it, not the cause; no read-back of the written bytes was demonstrated. Trigger: after login completes (full feature phase), an initiator that has declared a large TargetRecvDataSegmentLength and a FirstBurstLength that permits unsolicited/immediate data sends a PDU whose declared data-segment length exceeds what was received. With KASAN: BUG: KASAN: slab-out-of-bounds in sg_copy_buffer+0x150/0x1c0 Read of size 4096 at addr ffff888109720800 by task kworker/1:0H/25 Workqueue: ib-comp-wq ib_cq_poll_work Call Trace: sg_copy_buffer+0x150/0x1c0 isert_recv_done+0xba6/0x2390 __ib_process_cq+0xe1/0x390 ib_cq_poll_work+0x46/0x150 isert_recv_done+0xba6 resolves to isert_handle_iscsi_dataout() (ib_isert.c:1160), inlined through isert_rx_opcode(). Validate wc->byte_len against the framing in isert_recv_done() before the PDU reaches any handler, and reinstate the connection if it is short. Because the test compares without subtracting the header length, it also rejects PDUs shorter than the iSER and iSCSI headers, which would otherwise be parsed out of stale descriptor contents. The login handler rejects PDUs shorter than ISER_HEADERS_LEN (commit 29e7b925ae6d ("IB/isert: Reject login PDUs shorter than ISER_HEADERS_LEN")) but does not bound the declared length either; that is fixed in the next patch. The data handlers had no length check at all. isert reads the data segment from a fixed offset: isert_get_data() returns the iSER header plus ISER_HEADERS_LEN and makes no adjustment for an AHS. The bytes the handlers touch are therefore exactly [ISER_HEADERS_LEN, ISER_HEADERS_LEN + dlength), and comparing that sum against wc->byte_len bounds precisely the region that is read. An AHS term would only make the test stricter without bounding anything furth ---truncated---	9.1	More Details
CVE-2026-54767	WeGIA is a web manager for charitable institutions. Prior to 3.8.5, web/html/socio/sistema/controller/deletar_socios.php exposes an unauthenticated GET endpoint whose chave parameter is checked only against a hardcoded chave_correta value embedded in the public source repository. A remote attacker who obtains that value can reach the endpoint's TRUNCATE TABLE operations for the endereco, pessoafisica, pessoajuridica, and socio tables without an administrative session or application authorization, permanently destroying member and contributor records. The attack requires the affected tables to exist and the web process database account to possess truncation privileges. This issue is fixed in version 3.8.5.	9.1	More Details
CVE-2026-20341	A vulnerability in the sftunnel inter-device communication protocol of Cisco Secure FMC Software could allow an authenticated, remote attacker to obtain root privileges. This vulnerability is due to unsecured deserialization of untrusted data over the sftunnel management connection. An attacker could exploit this vulnerability by sending crafted sftunnel remote procedure calls (RPCs). A successful exploit could allow the attacker to gain root privileges on a device that is running Cisco Secure FMC Software and its high-availability peer. To exploit this vulnerability, the attacker must have valid administrative credentials on a managed Cisco FTD device.	9.1	More Details
CVE-2026-90413	In the Linux kernel, the following vulnerability has been resolved: IB/isert: reject login PDUs declaring more data than was received isert_login_recv_done() records how many bytes the HCA actually placed in the login buffer, but nothing compares that against the length the login PDU's BHS declares. isert_rx_login_req() copies min(login_req_len, MAX_KEY_VALUE_PAIRS) bytes into login->req_buf, and the login code then reads the declared length back out of that buffer - for the first PDU in iscsi_target_locate_portal(), payload_length = ntoh24(login_req->dlength); tmpbuf = kmemdup_nul(login->req_buf, payload_length, GFP_KERNEL); and for the ones after it in iscsi_decode_text_input(), reached from iscsi_target_do_login(). login->req_buf is a fixed MAX_KEY_VALUE_PAIRS (8192) byte allocation, so an initiator that declares more than it sends reads off the end of it, before authentication and with the length under its control: BUG: KASAN: slab-out-of-bounds in kmemdup_nul+0x43/0x80 Read of size 8193 at addr ffff8881056a8000 by task iscsi_np/167 __asan_memcpy+0x23/0x60 kmemdup_nul+0x43/0x80 iscsi_target_locate_portal+0x48d/0x1180 iscsi_target_login_thread+0x19a9/0x3350 Allocated by task 167: __kmalloc_cache_noprof+0x158/0x370 iscsi_target_login_thread+0x971/0x3350 which belongs to the cache kmalloc-8k of size 8192 allocated 8192-byte region Falsifying the second login PDU instead reaches the other reader, on the same buffer: BUG: KASAN: slab-out-of-bounds in kmemdup_nul+0x43/0x80 Read of size 8193 at addr ffff888104d10000 by task kworker/1:1/50 Workqueue: isert_login_wq iscsi_target_do_login_rx __asan_memcpy+0x23/0x60 kmemdup_nul+0x43/0x80 iscsi_decode_text_input+0xc6/0x11c0 iscsi_target_do_login+0x261/0x1470 iscsi_target_do_login_rx+0x51d/0x7d0 iscsit over TCP is not exposed: iscsit_get_login_rx() validates the declared length with iscsi_target_check_login_request() and then reads exactly that many bytes off the socket, so the declared length governs how much arrives rather than how much is copied out of an already-filled buffer. isert does not call iscsi_target_check_login_request() at all. Reject a login PDU whose declared DataSegmentLength exceeds what was received, in both paths that reach isert_rx_login_req(): isert_get_login_rx() for the first login PDU and isert_login_recv_done() for the ones after it. dlength <= login_req_len is allowed because the received count can include up to three bytes of iSCSI padding. Once the check is in place the copy out can no longer exceed the copy in: the posted login SGE is ISER_RX_PAYLOAD_SIZE, so login_req_len cannot exceed MAX_KEY_VALUE_PAIRS and the min() in isert_rx_login_req() is login_req_len. Like the existing short-PDU check added by 29e7b925ae6d, the reject in isert_login_recv_done() returns without completing login_req_comp, so a malformed subsequent PDU leaves the login to be torn down by the login timer rather than failing immediately. The first-PDU path returns an error and fails straight away. Reproduced on 7.2.0-rc4 with soft-RoCE (rdma_rxe) under KASAN, using an initiator that sends the real key=value payload while declaring 8193 in the BHS, on the first login PDU and on the second in separate runs. The reported read size tracks the declared value exactly; 16384 and 61440 behave the same. Unpatched 3 of 3 runs report on each of the two paths, patched 0 of 3 on both, run alternately in a single session, and a normal login still completes on the patched build.	9.1	More Details
CVE-2026-20176	A vulnerability in Cisco ISE could allow an authenticated, remote attacker to execute arbitrary commands on the underlying operating system of an affected device. To exploit this vulnerability, the attacker must have valid high-privileged administrative credentials. This vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to obtain system-level access to the underlying operating system and then elevate privileges	9.1	More Details

	to root. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a DoS condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored.		
CVE-2026-73447	A privileged attacker can exploit certain operation to execute arbitrary commands with root privileges, leading to full device compromise. An authenticated user can exploit gRPC Network Security Interface (gNSI) Certz service on Arista EOS-based products to escalate privileges and execute arbitrary OS commands via a crafted Certz Rotate request. The Bootz service is also affected.	9.1	More Details
CVE-2026-86462	Apache Airflow FAB provider: changing a user's password through the Admin user-edit PATCH endpoint does not invalidate that user's existing database-backed sessions. An attacker who already holds a copy of the victim's session cookie keeps full access as that user after the password change, so the password reset does not evict them. Affects deployments using the FAB auth manager with database-backed sessions; an administrator (or the user themselves) performing a routine password change is the trigger, and no attacker interaction with the endpoint is needed. This is a second, independent route to the outcome addressed by CVE-2026-82311, which corrected an identifier comparison in the session-invalidation helper. That fix does not repair this endpoint, because the PATCH path never calls the helper at all. Deployments that applied the CVE-2026-82311 fix must also upgrade for this one. Users of apache-airflow-providers-fab are recommended to upgrade to version 3.9.0 or later, which fixes the issue.	9.1	More Details
CVE-2026-89779	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: validate ef->size covers the record's name and value When an EA record has a non-zero ef->size, ntfs_read_ea() only checks that the record fits in the remaining buffer (ea_size > bytes), not that ef->size is large enough to hold the record's own name_len + 1 + elength. A crafted image can pass validation with, e.g., ef->size = 24 but elength = 0xffff. ntfs_get_ea() then trusts elength and copies it out of the undersized record, reading past the kmalloc(info->size) allocation and leaking heap memory to userspace via getxattr(): BUG: KASAN: slab-out-of-bounds in ntfs_get_ea (fs/ntfs3/xattr.c:302) Read of size 65535 at addr ffff888100794550 by task exploit __asan_memcpy (mm/kasan/shadow.c:105) ntfs_get_ea (fs/ntfs3/xattr.c:302) ntfs_getxattr (fs/ntfs3/xattr.c:848) __vfs_getxattr (fs/xattr.c:441) vfs_getxattr (fs/xattr.c:474) do_getxattr (fs/xattr.c:800) path_getxattrat (fs/xattr.c:868) do_syscall_64 (arch/x86/entry/syscall_64.c:94) The buggy address is located 80 bytes inside of allocated 84-byte region in cache kmalloc-96 Compute the size the record needs and require ef->size to cover it.	9.1	More Details
CVE-2026-89786	In the Linux kernel, the following vulnerability has been resolved: ext4: fix out-of-bounds read in ext4_read_inline_dir() ext4_read_inline_dir() can read a dirent header past the end of its inline buffer, triggering a slab-out-of-bounds read during getdents64(): BUG: KASAN: slab-out-of-bounds in __ext4_check_dir_entry Read of size 2 at addr ffff88800f3dd23c by task exploit/148 ... __ext4_check_dir_entry ext4_read_inline_dir iterate_dir The dirent payload lives in a buffer of exactly inline_size bytes: dir_buf = kmalloc(inline_size, GFP_NOFS); but iteration runs in a position space extra_offset bytes larger (extra_size = extra_offset + inline_size) so the synthetic "." and ".." land at their block-dir offsets. A dirent is formed at "dir_buf + pos - extra_offset", yet the ext4_check_dir_entry() length argument uses the larger extra_size. A position whose dirent header would extend past extra_size is therefore accepted, and the rescan loop's rec_len probe and ext4_check_dir_entry() dereference de->rec_len before the entry is rejected. Reject a position whose minimum-size dirent header would not fit within extra_size before forming de, in both the rescan and main loops, and pass inline_size rather than extra_size to ext4_check_dir_entry() so the length check matches the physical buffer.	9.1	More Details
CVE-2026-76186	Apache Airflow Keycloak provider: from Airflow 3.3 the Keycloak auth manager takes a user's identity from the signed Airflow session token but takes the Keycloak access and refresh tokens used for every authorization decision from separate, unauthenticated cookies, and never checks that the two describe the same subject. A user who holds any valid Airflow login of their own, together with another subject's Keycloak access or refresh token obtained out of band, can pair the two: Airflow then authorizes requests with the foreign token's privileges while the session identity, audit log and cache keys continue to name the attacker's own account. The refresh path re-issues an Airflow session token for the original identity carrying the foreign tokens, so the mismatched pairing survives across sessions. Affects deployments running Airflow 3.3 or later with the Keycloak auth manager. Earlier versions carried the Keycloak tokens inside the signed session token, so the binding existed and was lost when they moved into separate cookies. Users of apache-airflow-providers-keycloak are recommended to upgrade to version 0.10.0 or later, which binds the cookie-supplied tokens to the session identity.	9.1	More Details
CVE-2026-89846	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Bound rsp_info_len to avoid OOB sense-data read In qla2x00_status_entry(), the FWI2 status path advances sense_data and shrinks par_sense_len by rsp_info_len: if (IS_FWI2_CAPABLE(ha)) { sense_data += rsp_info_len; par_sense_len -= rsp_info_len; } rsp_info_len is a 32-bit value taken directly from the target's FCP response (sf.rsp_data_len), while par_sense_len is the IOCB data area size (28 bytes for 24xx, 60 bytes for 29xx). A hostile or buggy target reporting an rsp_info_len larger than par_sense_len makes the unsigned subtraction underflow to a huge value and advances sense_data out of bounds. The underflowed par_sense_len then defeats the cap in qla2x00_handle_sense(): if (sense_len > par_sense_len) sense_len = par_sense_len; memcpy(cp->sense_buffer, sense_data, sense_len); so the memcpy reads up to SCSI_SENSE_BUFFERSIZE bytes from the out-of-bounds sense_data pointer, leaking adjacent response-ring/heap memory into the command's sense buffer. Clamp rsp_info_len to par_sense_len before the subtraction so par_sense_len can never underflow and sense_data stays within the IOCB data area. The fix sits before the comp_status switch, covering both qla2x00_handle_sense() call sites.	9.1	More Details
CVE-2026-90011	In the Linux kernel, the following vulnerability has been resolved: scsi: target: iscsi: Reserve a terminator byte for the login payload iscsi_target_check_login_request() rejects a login PDU whose DataSegmentLength exceeds MAX_KEY_VALUE_PAIRS, but the test is '>' and login->req_buf is allocated with exactly MAX_KEY_VALUE_PAIRS bytes. Since iscsit_get_login_rx() receives payload_length + padding bytes, where padding = ((-payload_length) & 3); any payload_length from 8189 to 8192 fills the whole 8192 byte buffer. The write stays in bounds, but no byte is left for a NUL terminator. The buffer is subsequently consumed as a C string. In the CHAP path chap_check_algorithm() calls kstrdup(a_str), and extract_param() calls strstr(in_buf, pattern) followed by strlen_semi(), none of which take a length. convert_null_to_semi() additionally rewrites every embedded NUL to ';', so even a payload made of well formed NUL separated key=value records is left without a terminator. These walk past the end of the object into adjacent slab memory. It is reachable by an unauthenticated initiator against a portal configured for CHAP; when authentication is not required iscsi_login_zero_tsih_s2() rewrites AuthMethod to None and the CHAP path is never entered. Allocate one	9.1	More Details

	extra byte. kzalloc() zeroes it and nothing ever writes to it, as every writer copies to offset 0 for at most MAX_KEY_VALUE_PAIRS bytes, so the buffer is always terminated.		
CVE-2026-92395	@fastify/proxy-addr is a Fastify plugin that determines a request's client address behind trusted reverse proxies, and it backs Fastify request.ip and request.ips. In versions 3.0.0 through 5.1.0, a trust subnet written in IPv4-mapped IPv6 notation with an IPv4-sized prefix, such as ::ffff:10.0.0.0/8 instead of the correct ::ffff:10.0.0.0/104, is accepted without error but trusts every IPv4 address on the internet rather than the block it names. Because the socket peer then becomes trusted at hop 0, any unauthenticated client can supply an arbitrary X-Forwarded-For header and control the address the application reads, which defeats IP-based access control, rate limiting, geolocation, and audit logging. The plugin inherited this defect from the upstream proxy-addr module (CVE-2026-90711). The issue is fixed in @fastify/proxy-addr 5.1.1, and users should upgrade to 5.1.1 or later. As a workaround, ensure any IPv4-mapped IPv6 trust subnet uses a prefix length of at least 97, or express the range in plain IPv4 notation.	9.1	More Details
CVE-2026-92397	A vulnerability has been found in Ruijie RG-EW3000GX EW_3.0(1)B11P380. Affected by this vulnerability is the function cc_set of the file unifyframe-sgi.elf of the component configChange. Such manipulation of the argument data.url leads to os command injection. The attack may be performed from remote. The exploit has been disclosed to the public and may be used.	9.1	More Details
CVE-2026-20305	A vulnerability in the diagnostic tools of Cisco ISE and ISE-PIC could allow an authenticated, remote attacker to perform command injection attacks on the underlying operating system and elevate privileges to root. To exploit this vulnerability, the attacker must have valid administrative credentials. This vulnerability is due to improper validation of user-supplied input. An attacker could exploit this vulnerability by sending crafted commands to the web-based management interface of an affected device. A successful exploit could allow the attacker to execute arbitrary code on the device and elevate privileges to root. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a denial of service (DoS) condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored.	9.1	More Details
CVE-2026-20306	A vulnerability in the REST API of Cisco ISE and ISE-PIC could allow an authenticated, remote attacker to perform command injection attacks on the underlying operating system and elevate privileges to root. To exploit this vulnerability, the attacker must have valid administrative credentials. This vulnerability is due to improper validation of user-supplied input. An attacker could exploit this vulnerability by sending crafted commands to the web-based management interface of an affected device. A successful exploit could allow the attacker to execute arbitrary code on the device and elevate privileges to root. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a DoS condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored.	9.1	More Details
CVE-2026-92398	A vulnerability was found in Ruijie RG-EW3000GX EW_3.0(1)B11P380. Affected by this issue is some unknown functionality of the file /etc/rg_config/admin of the component user_list_note Module. Performing a manipulation of the argument Name results in os command injection. It is possible to initiate the attack remotely. The exploit has been made public and could be used.	9.1	More Details
CVE-2026-92717	Covenant through 0.6 registers the CovenantHub SignalR hub without an Authorize attribute, allowing unauthenticated callers to invoke CreateHttpListener and receive a signed JWT token. Attackers can use the obtained token to authenticate against the entire operator API and access grunts, credentials, binaries, events, and the operator roster.	9.1	More Details
CVE-2026-90230	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix heap out-of-bounds read in nvmet_auth_negotiate() nvmet_execute_auth_send() allocates the DH-HMAC-CHAP message buffer with the host-supplied transfer length (tl) and hands it to nvmet_auth_negotiate() without passing tl along. nvmet_auth_negotiate() then reads the negotiate header and, for each of the halen hash identifiers and dhlen DH group identifiers, indexes into the fixed idlist[60] array (hashes at idlist[0..halen), groups at idlist[30..]). Neither the transfer length nor halen/dhlen is validated. A malicious or non-conformant host can report a tl smaller than the negotiate structure, or a halen/dhlen larger than the array (both are u8, up to 255), making the loops read past the end of the allocated buffer (heap out-of-bounds read). The sibling nvmet_auth_reply() already validates tl against the structure size; the negotiate path did not. Pass tl into nvmet_auth_negotiate(), reject a tl that does not cover the negotiate data plus one full protocol descriptor, and reject halen/dhlen larger than NVME_AUTH_DHCHAP_MAX_DH_IDS.	9.1	More Details
CVE-2026-92720	Kubero through 3.1.1 fails to apply authentication guards to the notifications API endpoints, allowing unauthenticated attackers to read webhook secrets and service URLs. Attackers can retrieve stored credentials and register malicious webhooks to intercept pipeline events or suppress alerting by deleting existing configurations.	9.1	More Details
CVE-2026-20194	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC), engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20194 are related to incorrect resource transfer between spheres that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-669.	9.1	More Details
CVE-2026-20284	A vulnerability in the SXP REST API of Cisco ISE could allow an authenticated, remote attacker to conduct SQL injection attacks. This vulnerability is due to insufficient validation of user-supplied input in REST API calls. An attacker could exploit this vulnerability by sending crafted input to an affected device. A successful exploit could allow the attacker to view or modify data on the underlying database for the affected device. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a DoS condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored. To exploit this vulnerability, the attacker must have valid administrative credentials, have the SXP service enabled, and have at least one SXP connection configured.	9.1	More Details
CVE-2026-63472	Vendure is an open-source headless commerce platform. Prior to 3.7.0, ExternalAuthenticationService.createCustomerAndUser in packages/core/src/service/helpers/external-authentication/external-authentication.service.ts selects an existing customer user by emailAddress and attaches a newly presented ExternalAuthenticationMethod without requiring verified to be true. In deployments with a custom external AuthenticationStrategy that forwards an email whose ownership the provider has not verified, an attacker can authenticate with a victim's email and bind the attacker's external identity to the victim's existing account. This can	9.1	More Details

	expose orders, addresses, and personal information and permit account changes or orders as the victim. Native-only email and password deployments and external strategies that always require provider-verified email ownership are unaffected, and new-account creation for an unused email remains permitted. This issue is fixed in version 3.7.0.		
CVE-2026-92860	A security flaw has been discovered in rcourtman Pulse up to 6.0.4/6.1.0-rc.4. Affected by this issue is the function fmt.Sprintf of the file /api/security/quick-setup of the component Quick Security Setup Handler. The manipulation of the argument Username results in improper input validation. The attack may be performed from remote. Upgrading the affected component is advised.	9.1	More Details
CVE-2026-61594	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, the live (WebSocket) transport authorizes a mount via `check_view_auth`, not Django's `View.dispatch()` chain. As a result, standard Django authorization — `LoginRequiredMixin`, `PermissionRequiredMixin`, `UserPassesTestMixin`, `@method_decorator(login_required, name="dispatch")`, and custom `dispatch()` guards — and the djust admin extension's staff gate (applied only in the HTTP `as_view` wrapper) were enforced on the initial HTTP GET but silently bypassed over WebSocket, where all events and state flow. An anonymous or under-privileged client could open a WebSocket and mount such a view — including admin list/create/change/delete — and dispatch its handlers. This is fixed in djust 1.0.7. `check_view_auth` now honors the Django `AccessMixin` family on every transport; a new system check S004 fails loud at startup on auth patterns the runtime cannot safely replay (decorator/overridden-`dispatch` forms); and the admin base mixin declares `login_required = True` + an active-staff `check_permissions` gate. As a workaround, gate views using djust's `login_required` / `permission_required` / `check_permissions` attributes (honored on all transports) rather than HTTP-only mixins/decorators.	9.1	More Details
CVE-2026-20211	A vulnerability in Cisco ISE could allow an authenticated, remote attacker to execute arbitrary commands on the underlying operating system of an affected device. To exploit this vulnerability, the attacker must have valid high-privileged administrative credentials. This vulnerability is due to insecure deserialization of Java objects by the affected software. An attacker could exploit this vulnerability by sending a crafted serialized Java object to an affected device. A successful exploit could allow the attacker to obtain user-level access to the underlying operating system and then elevate privileges to root. In single-node deployments, successful exploitation of this vulnerability could cause the affected ISE node to become unavailable, resulting in a DoS condition. In that condition, endpoints that have not already authenticated would be unable to access the network until the node is restored.	9.1	More Details
CVE-2026-75513	Marten is a .NET Transactional Document DB and Event Store on PostgreSQL. From version 7.0.0 until 9.13.0, several Marten LINQ and tenant-management paths interpolate runtime, potentially attacker-controlled strings into single-quoted SQL literals without escaping or parameterization. The primary confirmed vector is a dictionary indexer key used by Where filters in src/Marten/Linq/Members/Dictionaries/DictionaryItemMember.cs. Additional affected sinks include SelectParser.cs, DatabaseScopedTenantPartitions.cs, and DeleteAllForTenant.cs reached through IEventStore.DeleteProjectionProgressAsync, while DictionaryContainsKeyFilter.cs (Newtonsoft serializer only; System.Text.Json is not affected) handles ContainsKey calls. Events/Daemon/Internals/EventLoader.cs contains a related per-tenant partition-pruning literal that the advisory identifies as a defense-in-depth sink. A crafted single quote can escape the generated literal, enabling filter or multi-tenant authorization bypass and blind data exfiltration, and deployments that permit semicolon-batched Npgsql statements may also allow data modification. This issue is fixed in version 9.13.0.	9.1	More Details
CVE-2026-88592	kkFileView >= 4.2.0 is vulnerable to Server-Side Request Forgery (SSRF). The cross-origin file proxy endpoint /getCorsFile is protected by TrustHostFilter against the trust.host whitelist. However, the URL parameter validated by the filter is not the same parameter the controller actually fetches: the filter validates the first non-empty parameter in a fixed priority order, while the controller only reads and fetches urlPath. By supplying both urlPath=<real target> and url=<whitelisted decoy address> in the same request, the decoy passes validation while the unvalidated real target is fetched server-side — and the response body is echoed back to the attacker.	9.1	More Details
CVE-2026-20237	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC), engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20237 are related to improper input validation issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-20.	9.1	More Details
CVE-2026-84738	The AF Companion WordPress plugin before 2.2.0 does not validate the type of files uploaded through one of its import features, allowing users with a low-privileged store-management role to upload arbitrary files, including PHP ones, leading to Remote Code Execution.	9.1	More Details
CVE-2026-77903	Authentication bypass by spoofing in Microsoft Dataverse allows an unauthorized attacker to elevate privileges over a network.	9.0	More Details
CVE-2026-45143	Chamilo LMS is an open-source learning management system. From 2.0.0 through at least 2.1.0, Chamilo LMS stores private Message.content without server-side sanitization and renders it as HTML in assets/vue/views/message/MessageShow.vue and public/main/template/default/message/view_message.html.twig. An authenticated low-privilege user, including a student, can directly address crafted message content to an administrator because the message creation flow permits a sender to select another user as the recipient. The content executes in the recipient's browser when the recipient opens the routine inbox or message view, without requiring a link click, and can expose session credentials or permit actions as the administrator. This vulnerability is fixed in 2.0.1.	9.0	More Details
CVE-2026-88795	The wpShopGermany IT-RECHT KANZLEI WordPress plugin before 2.4 does not generate its API authentication token securely, deriving it from data the requester controls and creating it as a side effect of the check that is supposed to validate it, allowing unauthenticated attackers to predict the token and use the access it grants to write arbitrary files, leading to remote code execution.	9.0	More Details
CVE-2026-	A vulnerability in the internal configuration of the Apache JServ Protocol (AJP) connector for Cisco Secure FMC Software could allow an unauthenticated, remote attacker to impersonate a peer device. This vulnerability is due to incorrect initialization of encryption parameters for the AJP connector at boot time. An attacker could exploit this vulnerability by sending crafted packets to the AJP connector. A successful exploit could allow the attacker to	9.0	More Details

76420	execute commands as root and gain full control over the FMC REST APIs on the affected device. Note: This vulnerability can be exploited only if the valid sftunnel connection between Cisco Secure FMC Software and Cisco Secure FTD Software is down.		
CVE-2026-92934	vm2 before 3.11.8 contains an incomplete fix for Error.cause sanitization that allows sandbox escape when revisited host-wrapped AggregateError objects are caught within a single exception handler traversal. Attackers can exploit cycle detection bypass in handleException to access unsanitized host proxies embedded in the errors array, enabling full remote code execution and process information disclosure from the sandbox.	9.0	More Details
CVE-2026-92935	vm2 is a sandbox for running untrusted Node.js code. In versions >= 3.11.4 and <= 3.11.6, the NodeVM constructor computes `hasRealRequireConfig` with `typeof requireOpts === 'object' && requireOpts !== null`, so an array-shaped `require` value (for example `require: []`) satisfies the guard that is meant to reject nesting without an explicit require configuration. `makeResolverFromLegacyOptions()` then destructures the array into undefined option fields and returns a resolver containing only `NESTING_OVERRIDE.vm2`. As a result, an attacker who can supply JavaScript executed by a NodeVM configured with truthy `nesting` and an array-shaped `require` (e.g. `new NodeVM({nesting: true, require: []})`) can require the host `vm2` module, create an inner NodeVM with an attacker-chosen builtin allowlist (such as `child_process`), and execute arbitrary commands with the privileges of the host Node.js process, escaping the sandbox. Outer builtin restrictions do not constrain the attacker-created inner NodeVM. This issue is fixed in vm2 3.11.7.	9.0	More Details
CVE-2026-47252	Anyquery is an SQL query engine built on top of SQLite. Prior to 0.4.5, authenticated users with INSERT or UPDATE access to affected macOS virtual tables can execute operating-system commands because the Chrome plugin and equivalent Brave, Edge, and Safari variants interpolate a SQL-controlled URL into AppleScript or JXA source passed to osascript. In plugins/chrome/tabs.go, tabsTable.Insert() passes the URL through fmt.Sprintf(newTabScript, url), and tabsTable.Update() uses fmt.Sprintf(setURLScript, pk, url). A URL containing quote and newline characters can break out of the intended string or property record and append script statements, resulting in arbitrary command execution with the privileges of the anyquery process on the macOS host. This issue is fixed in version 0.4.5.	9.0	More Details

OTHER VULNERABILITIES

CVE Number	Description	Base Score	Reference
CVE-2026-86792	Apache Airflow Apache Kafka provider versions 1.15.0 before 2.0.0 resolve dotted-path strings found in a Kafka connection's `extra` field into Python callables via `import_string`, with no allowlist, and hand them to the confluent-kafka client which invokes them. Deployments that have enabled the Kafka event producer — `dag_run_events_enabled` or `task_instance_events_enabled`, both disabled by default — build that client inside the scheduler process, so a user whose only privilege is editing Airflow connections gains arbitrary code execution in the control plane; the Airflow security model limits connection-configuration users to code execution on workers, not the scheduler. Deployments using Google Managed Kafka are not affected, because that code path overwrites any user-supplied `oauth_cb`; plain brokers and Amazon MSK are exposed. Users are recommended to upgrade to apache-airflow-providers-apache-kafka 2.0.0 or later, which adds an allowlist configuration option for connection-string callbacks.	8.8	More Details
CVE-2026-89898	In the Linux kernel, the following vulnerability has been resolved: media: cec: extron-da-hd-4k-plus: add sanity check Add check to prevent overflowing msg.msg[] in case the incoming data is malformed.	8.8	More Details
CVE-2026-92580	In AVideo through 29.0, the CloneSite plugin is vulnerable to stored OS command injection. In plugin/CloneSite/cloneClient.json.php (line ~270) the stored SSH password is substituted into the command string `sshpass -p '{password}' rsync ...` with a plain str_replace and no escaping, so a single quote in the password breaks out of the quoted word and injects arbitrary shell. The password is written through the admin-only endpoint objects/pluginAddDataObject.json.php, whose only CSRF defense (isUntrustedRequest()/forbidIfIsUntrustedRequest()) is a no-op when the request source appears to be loopback — as happens behind a same-host TLS-terminating reverse proxy with $global['trustedProxies'] unset — or when an attacker-controlled application is co-hosted on the same hostname; on HTTPS the session cookie is issued with SameSite=None, so a cross-site POST carries it. An unauthenticated remote attacker can therefore lure an authenticated administrator into planting a malicious password (and an attacker-controlled cloneSiteURL), after which the plugin's documented crontab entry executes the injected command with no further administrator action, as the crontab owner (commonly root or www-data). Exploitation requires the CloneSite plugin to be enabled with the documented crontab installed and one of the above CSRF channels; default single-process Apache deployments are reported as not CSRF-exploitable. This is a residual sink of CVE-2026-41304. The issue is confirmed at master HEAD (8963b6a1); no patched version is available.	8.8	More Details
CVE-2026-93042	In the Linux kernel, the following vulnerability has been resolved: dmaengine: dw-edma: Terminate all descriptors without callbacks The DMA Engine client documentation says in the "Terminate APIs" section of Documentation/driver-api/dmaengine/client.rst: "No callback functions will be called for any incomplete transfers." dw-edma instead calls vchan_cookie_complete() when a deferred STOP reaches the interrupt handler. This schedules a callback for the active descriptor and leaves other issued or submitted descriptors queued. A late callback after dmaengine_terminate_sync() can dereference client state that has already been freed, while leftover descriptors may later restart into reused buffers or leak. Move all issued and submitted descriptors to the terminated list whenever termination completes. For a pending STOP, do this from both the DONE and ABORT paths. Complete their cookies in order without scheduling callbacks. A STOP can remain pending until the running transfer raises an interrupt. Make device_synchronize() wait for such a pending STOP to complete before releasing terminated descriptors. Reuse it from free_chan_resources(), then release the remaining virt-dma resources. Sleep instead of busy-polling while waiting, and warn if the existing timeout expires.	8.8	More Details
CVE-2026-54239	Faust.js is a headless WordPress toolkit. Prior to 1.8.11, the FaustWP WordPress plugin authenticates only the ciphertext in its token envelope and excludes the 16-byte initialization vector from the HMAC in WPE\FaustWP\Auth\encrypt() and WPE\FaustWP\Auth\decrypt() in plugins/faustwp/includes/auth/functions.php. A logged-in non-administrator who obtains an authorization code from GET /generate can modify the unauthenticated initialization vector so that CBC decryption changes the token type and user identifier while the HMAC remains valid. This can produce an access token for an Administrator and permit	8.8	More Details

	full WordPress REST API access, administrator-account creation, plugin installation, and arbitrary code execution. This issue is fixed in repository version 1.8.11.		
CVE-2026-54504	MCP Documentation Server is a local-first document management and semantic search server for AI coding agents. From 1.13.0 until 1.13.1, the automatically started Web UI in src/server.ts calls startWebServer in src/web-server.ts with START_WEB_UI enabled by default and WEB_PORT set to 3080. startWebServer uses app.listen(PORT) without a host, which binds the unauthenticated document-management API to all interfaces rather than localhost. A network-reachable client can invoke GET /api/documents, GET /api/documents/:id, POST /api/documents, POST /api/search-all, DELETE /api/documents/:id, and GET /api/config without credentials to enumerate and read documents, search the corpus, insert or delete documents, and tamper with the MCP assistant's knowledge base. The service must be reachable from the attacker's LAN, VM network, container bridge, VPN, or another routed network, and the issue does not provide remote code execution. This issue is fixed in 1.13.1.	8.8	More Details
CVE-2026-88064	Backstage is an open framework for building developer portals. Prior to 1.14.6 and from 1.15.0 until 1.15.4, the @backstage/plugin-techdocs-node package insufficiently validates mkdocs.yml supplied by an authenticated user who can register or modify a TechDocs source. Unsafe Python YAML tags, markdown_extensions names and configuration, theme options, and extra_templates values can reach the documentation generator and cause unintended code execution. The resulting impact is limited to the files, credentials, network access, and other resources available to the TechDocs backend or build container. This issue is fixed in versions 1.14.6 and 1.15.4.	8.8	More Details
CVE-2026-90240	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Flush context cache with correct SID when tearing down aliases domain_context_clear_one() and device_pasid_table_teardown() are both invoked once per DMA alias of a device. Each function locates the context entry using the bus/devfn pair provided by the pci_for_each_dma_alias() callback, then calls intel_context_flush_no_pasid(), which constructs a device-selective context-cache invalidation from info->bus and info->devfn (that is, always the requester ID of the device itself). As a result, for every alias other than the device’s own RID, the context entry that was just cleared in memory is never invalidated in the context cache. Hardware may continue using that stale cached entry. In the scalable-mode teardown path, intel_pasid_free_table() can then free the PASID directory still referenced by that stale entry, allowing the IOMMU to walk freed memory. Fix this by passing the source ID of the entry being torn down to intel_context_flush_no_pasid(), instead of deriving it from @info.	8.8	More Details
CVE-2026-90018	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: fix OOB read / stack overflow in rtw_get_wps_attr() rtw_get_wps_attr() walks WPS attributes inside a WPS IE taken from a wireless management frame. For each candidate attribute it only checks that the fixed 4-byte attribute header (2-byte ID + 2-byte length) fits inside the IE: if (attr_ptr + 4 > wps_ie + wps_ielen) break; u16 attr_id = get_unaligned_be16(attr_ptr); u16 attr_data_len = get_unaligned_be16(attr_ptr + 2); u16 attr_len = attr_data_len + 4; attr_data_len (and therefore attr_len) is read directly from the wire and is never checked against the remaining bytes in the IE before being used as the size of: memcpy(buf_attr, attr_ptr, attr_len); Since attr_len is fully attacker controlled (0 to 65535+4), this is both a heap OOB read of wps_ie, and, more seriously, a stack buffer overflow at several call sites where buf_attr is a single-byte stack variable, e.g. rtw_get_wps_attr_content()'s callers passing WPS_ATTR_SELECTED_REGISTRAR into a stack "u8 sr"/"u8 selected_registrar" (drivers/staging/rtl8723bs/os_dep/ioctl_cfg80211.c, drivers/staging/rtl8723bs/core/rtw_mlme_ext.c). A crafted WPS IE in a beacon or probe response processed during scanning can therefore smash the stack of the parsing thread. rtw_get_wps_attr_content() itself has no independent length check and simply trusts the attr_len it gets back from rtw_get_wps_attr(), so fixing the bound here also fixes that caller. The "attr_ptr + 4 > wps_ie + wps_ielen" header check above was added by commit 1463ca3ec6601 ("staging: rtl8723bs: fix OOB reads in rtw_get_sec_ie(), rtw_get_wapi_ie(), and rtw_get_wps_attr()"), which bounded the fixed header but never extended the check to cover the variable-length attribute data that follows it. Add that missing check before attr_len is used as a memcpy() length or accepted as a match.	8.8	More Details
CVE-2026-73464	On affected platforms running Arista EOS with gRPC Network Management Interface (gNMI) enabled, a specially crafted request could allow a malicious authenticated client with gRPC Network Management Interface (gNMI) access to execute arbitrary code with root privileges on the switch.	8.8	More Details
CVE-2026-92801	cc-connect through 1.5.0 fails to enforce per-user allowlist filtering in the onCardAction handler for Feishu interactive card callbacks. Attackers can dispatch agent commands by triggering card actions in admitted chats, bypassing the per-user access controls that protect the text message handler.	8.8	More Details
CVE-2026-92796	Manticore Search versions 27.0.0 before 28.4.4 fail to validate permissions for all statements in multi-statement SQL requests, allowing read-only users to execute unauthorized queries. Attackers can append additional SELECT statements after the first statement to read credential tables and obtain password hashes that authenticate as administrators without plaintext recovery.	8.8	More Details
CVE-2026-90329	In the Linux kernel, the following vulnerability has been resolved: HID: synchronize input before cleaning up a failed probe hid_device_io_start() allows reports to run concurrently with probe. If the probe subsequently fails, __hid_device_probe() releases driver resources and clears hdev->driver without first excluding those report callbacks. For example, a report may enter hidraw_report_event() while the failure path frees the associated hidraw object, leading to a use-after-free when the report takes the object's list lock. Stop input before performing failed-probe cleanup. This reacquires driver_input_lock and waits for any report callback already in progress.	8.8	More Details
CVE-2026-92788	Coze Studio through 0.5.1 fails to validate that table names in workflow SQL customization nodes belong to the caller's workspace. Authenticated attackers can enumerate predictable table identifiers and execute SQL statements against other workspaces' memory databases to read, insert, or delete data.	8.8	More Details
CVE-2026-89774	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: hold sk properly in sco_conn_ready sk deref in sco_conn_ready must be done either under conn->lock, or holding a refcount, to avoid concurrent close. conn->sk and parent sk is currently accessed without either, and without checking parent->sk_state: [Task 1] [Task 2] sco_sock_release sco_conn_ready sk = conn->sk lock_sock(sk) conn->sk = NULL lock_sock(sk) release_sock(sk) sco_sock_kill(sk) UAF on sk deref and similarly for access to sco_get_sock_listen() return value. Fix possible UAF by holding sk refcount in sco_conn_ready() and making sco_get_sock_listen() increase refcount. Also recheck after lock_sock that the socket is still valid. Adjust conn->sk locking so it's protected also by lock_sock() of the associated socket if any.	8.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: clear vdev->msi_perm after freeing it on init failure vfio_msi_cap_len() lazily allocates the per-device MSI permission table: vdev->msi_perm = kmalloc_obj(struct perm_bits, GFP_KERNEL_ACCOUNT); if (!vdev->msi_perm) return -ENOMEM; ret = init_pci_cap_msi_perm(vdev->msi_perm, len, flags); if (ret) { kfree(vdev->msi_perm); return ret; /* vdev->msi_perm left dangling */ } When init_pci_cap_msi_perm() ->		

CVE-2026-89777	alloc_perm_bits() fails with -ENOMEM, the error path frees vdev->msi_perm but leaves the freed pointer stored in it. vdev->msi_perm is not re-zeroed later because struct vfio_pci_core_device is per-device and persists across open/close cycles, and the vfio_config_init() error path returns without calling vfio_config_free(). So the dangling pointer outlives the failed open. That leads to two use-after-frees on the same device: 1. Reuse. The next vfio_config_init() sees the stale pointer at "if (vdev->msi_perm) return len;" and reuses the freed object. MSI config accesses in vfio_pci_config_rw_single() then dereference and call the freed perm->readfn / perm->writefn function pointers. 2. Double free. A later vfio_config_free() runs free_perm_bits() and kfree() on the already-freed object. Fix it by NULLing vdev->msi_perm after the kfree(), matching the NULL-after-free discipline already used in free_perm_bits() and vfio_config_free(). BUG: KASAN: slab-use-after-free in vfio_pci_config_rw_single (drivers/vfio/pci/vfio_pci_config.c:1961) Read of size 8 at addr ffff88800fcc88d0 by task exploit/143 Call Trace: ... kasan_report (mm/kasan/report.c:595) vfio_pci_config_rw_single (drivers/vfio/pci/vfio_pci_config.c:1961) vfio_pci_config_rw (drivers/vfio/pci/vfio_pci_config.c:1986) vfio_pci_rw (drivers/vfio/pci/vfio_pci_core.c:1599) vfs_read (fs/read_write.c:572) __x64_sys_pread64 (fs/read_write.c:764) do_syscall_64 (arch/x86/entry/syscall_64.c:94) ... Followed on device close by a double free of the same object: Oops: general protection fault, probably for non-canonical address 0x1f63e0e8000008: 0000 [#1] SMP KASAN NOPTI RIP: 0010:kfree (mm/slub.c:6711) Call Trace: vfio_config_free (drivers/vfio/pci/vfio_pci_config.c:1861) vfio_pci_core_disable (drivers/vfio/pci/vfio_pci_core.c:685) vfio_pci_core_close_device (drivers/vfio/pci/vfio_pci_core.c:777) vfio_df_close (drivers/vfio/vfio_main.c:602) vfio_device_fops_release (drivers/vfio/vfio_main.c:648) __fput (fs/file_table.c:512) __x64_sys_close (fs/open.c:1496) do_syscall_64 (arch/x86/entry/syscall_64.c:94) ... Kernel panic - not syncing: Fatal exception	8.8	More Details
CVE-2026-89860	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Initialize NVMe abort_work once at submission qla_nvme_fcp_abort() and qla_nvme_ls_abort() ran INIT_WORK() on priv->abort_work immediately before schedule_work(). INIT_WORK() reinitializes the work_struct, resetting its list head and clearing the pending bit. If an abort is issued more than once for the same command (for example, concurrent transport teardown and a timeout-driven abort), the second INIT_WORK() reinitializes a work item that is already queued, which can corrupt the workqueue list and lead to crashes or a looping worker. Initialize priv->abort_work once at command submission, next to the existing per-command spin_lock_init(&priv->cmd_lock), and leave only schedule_work() in the abort paths. schedule_work() already does nothing when the work item is still pending, so a repeated abort no longer disturbs an in-flight work item. The command is not returned to the transport until the final kref_put()/release callback runs after abort_work has completed, so the work item is idle before priv is reused and the single submission-time INIT_WORK() is safe.	8.8	More Details
CVE-2026-87105	Tanium addressed a SQL injection vulnerability in Threat Response.	8.8	More Details
CVE-2026-27559	A low-privileged remote attacker can exploit a command injection vulnerability in the /api/status/data endpoint by sending a crafted GET request with user credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-90000	In the Linux kernel, the following vulnerability has been resolved: HID: rmi: fix OOB access with undersized RMI reports The hid-rmi driver sizes its writeReport/readReport buffer purely from the report descriptor supplied by the device, with no minimum bound: data->input_report_size = hid_report_len(input_report); data->output_report_size = hid_report_len(output_report); alloc_size = data->output_report_size + data->input_report_size; data->writeReport = devm_kzalloc(&hdev->dev, alloc_size, GFP_KERNEL); data->readReport = data->writeReport + data->output_report_size; but then reads and writes fixed offsets into it. A device declaring a 1-byte output and a 1-byte input report makes hid_report_len() return 2 for each, so alloc_size is 4, while rmi_set_page() -- reached unconditionally at probe time through rmi_input_configured() -- stores writeReport[4] and rmi_hid_read_block() stores writeReport[0..5]. Since readReport lives at writeReport + output_report_size, those stores also corrupt the window the next reply is parsed out of. The read path is worse: the copy length comes from readReport[1], which the device fills in and can be up to 255, and the copy starts at &readReport[2] with no regard for input_report_size, so it runs past the end of the allocation into adjacent slab objects. This does not even need a lying device -- rmi_f01_probe() issues a fixed 21-byte register read, so any device declaring an input report smaller than 23 bytes reads out of bounds even when it answers truthfully. Those bytes become the register values the RMI core acts on: rmi_f01_probe() prints them to the kernel log as the product id and exports them through the mode 0444 sysfs attribute of the same name, and rmi_driver_set_irq_bits() sends them back to the device as the interrupt mask, so an undersized report descriptor leaks heap contents both to unprivileged userspace and to the device itself. The write path has no bound either: rmi_hid_write_block() copies an unbounded len to &writeReport[4], and the largest caller a device can drive at probe time is rmi_driver_set_irq_bits(), whose length is derived from the interrupt source counts the device declares in its Page Description Table. Finally, the read loop cannot terminate on a zero-length reply: such a reply copies nothing and advances neither bytes_read nor bytes_needed, and because a reply did arrive the one second wait_event_timeout() does not fire either, so a device answering 0 forever keeps the loop running inside the probe worker with page_mutex held. khungtaskd does not notice, because every reply wakes the task. Reject reports too small for what the driver builds -- 6 output bytes for the write reports and 3 input bytes for the read handshake -- at probe time, clamp the write and the read copy to the report sizes the device declared, and treat a zero-length reply as an error. A device refused this way is started as an ordinary HID device, like one that does not carry the RMI report ids at all. RMI_DEVICE must not be left set in device_flags on that path, because rmi_input_configured() would then run the RMI setup and reach rmi_set_page(), which writes the writeReport buffer the refusal just skipped allocating. The bit can arrive set: rmi_probe() copies id->driver_data into device_flags before the report checks, and a bind through the new_id sysfs attribute can supply driver_data with RMI_DEVICE (BIT(0)) set. Strip the bit where driver_data is copied, so RMI_DEVICE keeps meaning exactly "this probe validated the reports"; the three jumps to start that predate this patch are covered as well. The error path also clears RMI_READ_DATA_PENDING on its way out, because that flag is what the wait at the top of the loop tests: leaving it set would make every later wait_event_timeout() return immediately on the stale reply and kill the read path for the rest of the device's life. Clamping does not regress working hardware: the read loop already handles ---truncated---	8.8	More Details
CVE-2026-27558	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/attached_devices_tab/ajax_remove_uploaded_iodd_files endpoint using operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-92125	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not reject the @GroovyASTTransformationClass annotation, allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to run an arbitrary AST transformation at compile time, bypassing the sandbox protection and executing arbitrary code in the context of the Jenkins controller JVM.	8.8	More Details
CVE-	SiYuan versions before 3.8.4 fail to escape bookmark labels imported from notebook files when rendering them in the dock tree.		More

2026-92985	Attackers can craft malicious .sy notebook files with unescaped HTML in bookmark attributes that execute scripts in the Electron renderer with access to child_process for command execution.	8.8	Details
CVE-2026-92124	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier checks the operations Groovy will perform with the elements it reads from a collection that a sandboxed script casts to another type but performs the cast on the collection itself, allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM.	8.8	More Details
CVE-2026-92123	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not intercept operations performed on a null receiver (method calls, property and attribute accesses, and array accesses), allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM.	8.8	More Details
CVE-2026-92122	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not check the method called through the proxy created when a sandboxed script coerces a value to an interface, if the value inherits a method of the same name as an interface method, allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM.	8.8	More Details
CVE-2026-27547	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/ajax/get_iodd_menu_info endpoint using valid user or operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-27548	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/ajax/get_iodd_port_info endpoint using user or operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-27549	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/attached_devices_tab/do_upload endpoint using operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-27550	A low-privileged remote attacker can exploit a command injection vulnerability in the Field_Shadow_Password class using operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-27551	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/ajax/parameterManage endpoint using user credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-92593	Craft CMS versions 5.10.0 through 5.10.12 contain an incomplete fix for CVE-2026-55794: the Controller::getPostedRedirectUrl() -> View::renderObjectTemplate() sink remained unsandboxed, and the same fix commit added a self-signing oracle in Cp::elementLabelHtml(). Because Craft/Yii HMAC tokens are not bound to a parameter name, an authenticated low-privilege control panel user with edit rights on a single element type can mint a token over attacker-controlled Twig for the returnUrl parameter and replay it as the redirect POST parameter, reaching the unsandboxed sink and achieving server-side template injection that executes arbitrary PHP code (full server compromise). The issue is fixed in 5.10.13.	8.8	More Details
CVE-2026-27554	A low-privileged remote attacker can exploit a command injection vulnerability in the /index.php/ajax/save_iodd_parameters endpoint using operator credentials allowing execution of commands with root privileges on the device.	8.8	More Details
CVE-2026-27555	A low-privileged remote attacker can exploit a local file inclusion vulnerability in the /index.php/ajax/get_iodd_port_info endpoint using a valid user cookie allowing execution of arbitrary PHP code on the device.	8.8	More Details
CVE-2026-27556	A low-privileged remote attacker can exploit a local file inclusion vulnerability in the /index.php/ajax/save_iodd_parameters endpoint using a valid operator cookie allowing execution of arbitrary PHP code on the device.	8.8	More Details
CVE-2026-92592	Craft CMS 4.8.0 through 4.18.5 and 5.0.0 through 5.10.12 sign an authenticated user's attacker-controlled license-shun cookie with the same key and format used to validate signed redirect parameters, because the HMAC signature is not bound to its purpose (Yii's cookieValidationKey is derived from the same Craft securityKey used for signed request parameters). An authenticated, non-administrator user (Control Panel access is not required) can set the cookie via the license-shun endpoint and transplant the signed envelope into the redirect parameter; on a successful login, Craft validates the signature and renders the authenticated bytes as an unsandboxed Twig template, where Twig's map filter accepts a string callback and allows PHP system() to execute arbitrary operating-system commands as the web-server user. Exploitation requires an account using password authentication without active 2FA, the default request configuration, and availability of PHP system(). The issue is fixed in 4.18.6 and 5.10.13.	8.8	More Details
CVE-2026-15815	Grafana OSS and Grafana Enterprise did not safely resolve symbolic links when extracting plugin archives. A crafted plugin archive can chain relative symbolic link entries to escape the plugin installation directory, writing arbitrary files and an executable backend binary outside that directory. The dropped executable runs with the privileges of the Grafana server process, resulting in remote code execution. Plugin archives are extracted before their signature is verified, so a valid plugin signature does not prevent the write. An operator can therefore be affected by installing a plugin that appears legitimate, as well as by installing a plugin from an arbitrary archive using grafana-cli, the GF_INSTALL_PLUGINS environment variable, or preinstall configuration. Grafana Enterprise is affected because it includes the same plugin extraction code as Grafana OSS.	8.8	More Details
CVE-2026-92780	KnowStreaming through 3.4.1 fails to enforce role-based access control on REST API endpoints, allowing any authenticated user to access protected functionality. Attackers can call identity-management endpoints to create administrator accounts or grant themselves administrative privileges without proper authorization.	8.8	More Details
CVE-	WeGIA is a web manager for charitable institutions. Prior to 3.8.5, WeGIA maps InternoControle to an empty resource array in web/controle/control.php, and verificarPermissao in web/dao/MiddlewareDAO.php treats that empty array as unconditional access for every authenticated user. The methods in web/controle/InternoControle.php, including listarUm, alterar, and excluir,		More

2026-54671	accept user-controlled id or idInterno values without verifying ownership, allowing a low-privileged user to read, modify, or delete another person's records and expose personal, identity, address, medical, and family information. The advisory notes that a self-referencing load bug can crash this controller in the reported revision, but the empty-resource authorization pattern and affected methods remain the vulnerability under review. This issue is fixed in version 3.8.5.	8.8	Details
CVE-2026-90357	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: unlink TWT flow if the MCU rejects the agreement The flow is added to dev->twt_list before sending the agreement to the firmware, but the error path leaves it linked while flowid_mask is never set. The flow slot can then be reused and memset while still on the list, corrupting twt_list, and station removal leaves a dangling entry behind that mt7915_mac_twt_sched_list_add() later walks.	8.8	More Details
CVE-2026-89804	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/dmem: fix mismatched DMA unmap size for large folios Device-private THP migration maps migration buffers with page_size() and records that length in dma_info->size. For a compound folio page_size() is PAGE_SIZE << order, but two teardown sites still pass a literal PAGE_SIZE to dma_unmap_page(): - nouveau_dmem_migrate_to_ram() on the success path, and - nouveau_dmem_migrate_copy_one() on the copy-error path. For an order > 0 folio this unmaps less than was mapped, leaking the remainder of the IOMMU/IOVA mapping. The other unmap sites, in nouveau_dmem_migrate_chunk() and nouveau_dmem_evict_chunk(), already use the saved size; use it here too.	8.8	More Details
CVE-2026-54916	NetBox Device Type Library is a collection of community-sourced device type definitions for import into NetBox. The absence of tests/init.py and the lack of --import-mode=importlib cause pytest prepend import mode to place the tests directory at the front of sys.path during collection. An unauthenticated contributor can add a module such as tests/git.py that shadows GitPython when tests/definitions_test.py executes from git import Git, Repo, or add tests/conftest.py for automatic collection-time execution. Python imports and runs the pull-request module before any test function, allowing arbitrary code execution on the GitHub Actions runner, test-result tampering, and access to tokens or network resources exposed to the workflow. This module-shadowing path is independent of the earlier pickle deserialization flaw and the separately tracked NETBOX_DT_LIBRARY_URL issue. This vulnerability is fixed by commit b0d9a3dadd0a0a9d3c93b0b2777559fd4bad1037.	8.8	More Details
CVE-2026-20336	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20336 are related to issues concerning improper control of a resource through its lifetime that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-664.	8.8	More Details
CVE-2026-85731	oras-go is a Go library for managing OCI artifacts. Prior to 2.6.2, content/file.Store extraction of OCI layers marked with io.deis.oras.content.unpack=true can write outside the store working directory. The pushDir path through extractTarDirectory and ensureLinkPath validates symlink targets lexically, resolveRelToBase skips its parent-symlink walk for root-level entries, and writeFile follows a terminal symlink when opening a regular file. A malicious archive can therefore create a symlink chain whose lexical target remains inside the extraction root but whose resolved target is an attacker-selected absolute path, then overwrite that target with a same-named regular-file entry even when AllowPathTraversalOnWrite is false. Pulling an attacker-controlled artifact can create or overwrite any file writable by the process and may lead to code execution. This issue is fixed in version 2.6.2.	8.8	More Details
CVE-2026-89811	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Add TLB flush after MES queue eviction/suspension MES (Micro Engine Scheduler) does not perform heavy-weight TLB invalidation after unmapping queues, unlike HWS which does this automatically. This causes a race condition where in-flight DMA descriptors can access memory that has been unmapped, leading to page faults and GPU queue hangs during SVM page migration. The issue manifests as KFDSVMRangeTest.MultiThreadMigrationTest failures on gfx1151 (Strix Point) with XNACK mode 1 enabled - the GPU compute queue hangs with packets submitted but never consumed. Add kfd_flush_tlb() calls after MES queue removal in two locations: - evict_process_queues_cpsch(): after all queues removed during eviction - suspend_queues(): after debug/criu queue suspension (with mem_fence barrier) This ensures all in-flight memory accesses from unmapped queues are flushed before memory is freed or migrated. (cherry picked from commit f5c4f88e0f9c45a8fb9dfac0c1df726c95e41b77)	8.8	More Details
CVE-2026-76409	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-76409 are related to improper limitation of a pathname issues that are grouped under the Common Weakness Enumeration (CWE) CWE-22.	8.8	More Details
CVE-2026-93382	Use after free in PDFium in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to execute arbitrary code inside the sandbox via a crafted HTML page. (Chromium security severity: High)	8.8	More Details
CVE-2026-20340	A vulnerability in Cisco Secure FMC Software could allow an authenticated, remote attacker to execute arbitrary commands at the root privilege level. This vulnerability is due to unsecured deserialization of web-management user-controlled data. An attacker could exploit this vulnerability by authenticating to the device and sending a crafted HTTP payload. A successful exploit could allow the attacker to save the crafted payload and then execute it on the underlying operating system as root. To exploit this vulnerability, the attacker must have valid credentials for a user account with at least the role of Security Analyst (read-only).	8.8	More Details
CVE-2026-90041	In the Linux kernel, the following vulnerability has been resolved: HID: sony: clean up device list on probe failure sony_input_configured() adds some controllers to sony_device_list before HID core registers their input devices. input_register_device() can fail after the callback returns successfully. sony_probe() then observes that HID_CLAIMED_INPUT is clear and unwinds, but only stops the HID hardware. The devres-managed sony_sc is freed while its list node remains linked, so the next matching controller traverses freed memory. Initialize the list node and device ID to inactive states. Make list removal idempotent and run the driver-private cleanup on every probe failure path. This also makes a second cleanup safe when sony_input_configured() already unwound a partial initialization before sony_probe() handles the missing input claim. Found by 0sec (https://0sec.ai) using automated source analysis; verified against the HID input registration and probe unwind paths.	8.8	More Details
CVE-2026-93381	Buffer overflow in PDFium in Google Chrome on on Windows prior to 153.0.8010.52 allowed a remote attacker leveraging social engineering to potentially execute arbitrary code inside the sandbox via a crafted PDF file. (Chromium security severity: High)	8.8	More Details

CVE-2026-93377	Type confusion in V8 in Google Chrome prior to 153.0.8010.52 allowed a remote attacker leveraging social engineering to execute arbitrary code inside the sandbox via a crafted HTML page. (Chromium security severity: High)	8.8	More Details
CVE-2026-63506	Tina is a headless content management system. Prior to @tinacms/auth 1.1.4 and next-tinacms-azure 15.0.1, isAuthorized accepts a request-controlled clientID and asks isUserAuthorized to validate the bearer token against that selected TinaCloud app instead of the self-hosted site's configured app. An attacker with any TinaCloud account can submit the attacker's own app ID and valid token to a victim endpoint, causing TinaCloudBackendAuthProvider or an affected media authorized callback to accept the attacker's verified status across the tenant boundary. The vulnerable logic is present in packages/@tinacms/auth/src/index.ts and packages/next-tinacms-azure/src/auth.ts. Successful exploitation permits media listing, reading, upload, or deletion and, when TinaCloudBackendAuthProvider is used, GraphQL read, create, update, and delete operations on the victim's content without a victim account or victim interaction. This vulnerability is fixed in @tinacms/auth 1.1.4 and next-tinacms-azure 15.0.1.	8.8	More Details
CVE-2026-90425	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Require exactly one Stream ID for a vSID tegra241_vintf_init_vsid() maps a guest vSID to a single physical Stream ID taken from master->streams[0], and only warns when the device does not have exactly one stream. A device with several streams gets only its first one mapped, so a guest vSID invalidation cannot reach the others' ATC and IOTLB entries; a device with none makes master->streams a ZERO_SIZE_PTR, read out of bounds. Reject the mapping with -EOPNOTSUPP if master->num_streams is not one.	8.8	More Details
CVE-2026-20344	A vulnerability in the web-based management interface of Cisco Secure FMC Software could allow an authenticated, remote attacker to perform a SQL injection attack against an affected device. To exploit this vulnerability, the attacker must have a valid account on the device with the role of Security Approver, Access Admin, or Network Admin. This vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted HTTP request to the web-based management interface of an affected device. A successful exploit could allow the attacker to obtain any data from the database, obtain the session credentials of an authenticated Administrator, and take actions with administrative privileges on the affected device.	8.8	More Details
CVE-2026-20360	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20360 are related to information exposure and insecure handling issues that are grouped under the Common Weakness Enumeration (CWE) CWE-200.	8.8	More Details
CVE-2026-20333	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20333 are related to incorrect comparison conditions that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-697.	8.8	More Details
CVE-2026-90381	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: fix handling channel context with different bands in mt76_switch_vif_chanctx() When performing channel switches on different radios within a short timeframe, channel contexts with different bands can be carried for each struct ieee80211_vif_chanctx_switch. Rework mt76_switch_vif_chanctx() to properly handle this scenario.	8.8	More Details
CVE-2026-90380	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt792x: fix use-after-free in mt76_rx_poll_complete A use-after-free issue occurs in mt76_rx_poll_complete due to a race condition. The STA has already been removed, but the rx_status still had a pointer to the wcid in the STA. Set the links' wcid pointers to be NULL for a MLD in mt7925_sta_pre_rcu_remove() BUG: KASAN: invalid-access in mt76_rx_poll_complete+0x280/0x470 Call trace: dump_backtrace+0xec/0x128 show_stack+0x18/0x28 dump_stack_lvl+0x40/0xc8 print_report+0x1b8/0x710 kasan_report+0xe0/0x144 do_bad_area+0x120/0x260 do_tag_check_fault+0x20/0x34 do_mem_abort+0x54/0xa8 el1_abort+0x3c/0x5c el1h_64_sync_handler+0x40/0xcc el1h_64_sync+0x7c/0x80 mt76_rx_poll_complete+0x280/0x470 mt76_dma_rx_poll+0x114/0x51c mt792x_poll_rx+0x60/0xf8 napi_threaded_poll_loop+0xe0/0x450 napi_threaded_poll+0x80/0x9c kthread+0x11c/0x158 ret_from_fork+0x10/0x20	8.8	More Details
CVE-2026-90367	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: hold dev->mt76.mutex while disabling tx worker in SER mt7996_mac_reset_work() parked the tx worker and disabled the RX/TX NAPIs before taking dev->mt76.mutex. mt76_worker_disable()/_enable() are plain kthread park/unpark, not refcounted, and __mt76_set_channel() toggles the same worker and the MT76_RESET bit under the mutex. An L1 SER racing a channel switch could therefore have the worker unparked and MT76_RESET cleared while the reset path resets the DMA rings, corrupting descriptors or tokens. Take the mutex before disabling the worker, as mt7915 does.	8.8	More Details
CVE-2026-47094	SIMAC MyPHR 1.1 contains an insecure direct object reference (IDOR) vulnerability that allows authenticated attackers to access and modify arbitrary employee records due to missing server-side ownership validation. Attackers can send a PUT request to the employee update endpoint with an arbitrary employee identifier and a controlled password value to take over target accounts, enumerate employee records, and retrieve sensitive personally identifiable information including private pay bulletins.	8.8	More Details
CVE-2026-92762	Pelican Panel versions before 1.0.0-beta35 enforce startup write permissions only through disabled form controls rather than server-side authorization checks. Attackers with startup.read permission can craft Livewire state updates to invoke afterStateUpdated callbacks and modify startup commands, docker images, and variables to execute arbitrary commands in the container.	8.8	More Details
CVE-2026-92761	WebVirtCloud fails to properly validate permission flags in UserInstance grants, allowing view-only users to perform privileged actions. Attackers with read-only grants can power off virtual machines, reset root passwords, install SSH keys, and manage ISO images by exploiting the get_instance gate that only checks grant existence.	8.8	More Details
CVE-2026-52483	The ping diagnostics and other similar functions of the MitraStar GPT-2741GNAC-N2-SV router with firmware BR_g8.10_1.11(WVK.0)b46 allow authenticated users execute arbitrary OS command via concatenated params on a crafted POST request for the endpoint /cgi-bin/device-management-utilities-internet.cgi.	8.8	More Details
CVE-	BC Security Empire before 6.7.1 fails to validate the multipart filename parameter in upload endpoints, allowing authenticated		

2026-92748	operators to write files to arbitrary paths on the C2 server. Attackers can use path traversal sequences in the filename to bypass directory containment and write malicious files to sensitive locations for code execution.	8.8	More Details
CVE-2026-89849	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Reject non-SCSI SRB on status IOCB fast path qla2x00_status_entry() filters out non-TYPE_SRB entries and the SRB_NVME_CMD, SRB_BIDI_CMD and SRB_TM_CMD types, then falls through to a SCSI fast path that assumes the command is an SRB_SCSI_CMD. The first thing on that path, qla_chk_edif_rx_sa_delete_pending(), and the subsequent handling both evaluate GET_CMD_SP(sp), i.e. sp->u.scmd.cmd. The srb u union overlays the SCSI command pointer with other command layouts (bsg_job, iocb_cmd). If firmware delivers an unexpected STATUS_TYPE IOCB for a non-SCSI handle, sp->u.scmd.cmd can read as a non-NULL garbage pointer, bypassing the NULL checks in qla_chk_edif_rx_sa_delete_pending() and at the cp == NULL test, and leading to a wild pointer dereference. Reject any SRB whose type is not SRB_SCSI_CMD before entering the fast path. The outstanding_cmds slot is left untouched so a genuinely non-SCSI command still completes through its proper handler.	8.8	More Details
CVE-2026-86864	pgAdmin 4's Backup tool appended the client-supplied 'database' field from the /backup/job/<sid>/object request to the pg_dump argument vector as a bare trailing positional argument, without validation. Because pg_dump parses its options with getopt_long, which permutes arguments, a value beginning with a dash was interpreted as an option rather than as a database name. A value such as --file=/absolute/path therefore overrode the storage-confined --file that pgAdmin had constructed earlier, causing pg_dump to write its output anywhere the pgAdmin process could write, outside the user's File Manager storage directory. This yields arbitrary file creation and overwrite as the operating-system account running pgAdmin, which can destroy pgAdmin's own configuration database and, depending on the target chosen, be escalated further. The same field additionally permitted connection-string injection. libpq expands a database name containing an equals sign into a full connection string, and keywords embedded there override the --host and --port that pgAdmin passes, so a value such as 'host=attacker.example port=5432 dbname=x' redirected pg_dump to a server of the attacker's choosing. Because pgAdmin exports the decrypted stored database password in the PGPASSWORD environment variable before executing the utility, the redirected connection carries that credential to the attacker-nominated endpoint. Both behaviours are reachable by any authenticated user holding the tools_backup permission, which is granted to the default User role. The fix stops passing the database name through the argument vector altogether and supplies it in the PGDATABASE environment variable, which libpq treats as a literal database name and never expands as a connection string. This matches the approach already used by the Import/Export tool. Regression tests assert that the database name is absent from the constructed argument vector and that PGDATABASE carries the exact requested value. This issue affects pgAdmin 4: from the introduction of the trailing positional database argument in the Backup tool before 9.18.	8.8	More Details
CVE-2026-54519	AI Agent Automation is a modular AI agent workflow automation platform with schedulers, tools, and observability. Prior to 0.9.1, backend/src/controllers/memory.controller.js authenticates requests but listMemories, deleteMemory, and clearAgentMemory use a caller-supplied agentId or memory _id without verifying through the related Agent that the record belongs to req.user. An authenticated attacker who knows or obtains another user's identifiers can read victim AgentMemory content, including conversation history, agent context, task data, embeddings, and metadata, delete an individual victim memory, or clear all memory belonging to a victim agent. This breaks tenant isolation and causes unauthorized disclosure and data loss. This issue is fixed in version 0.9.1.	8.8	More Details
CVE-2026-89036	Appwrite before 2.0.0 contains an argument injection vulnerability that allows authenticated users with functions.write or sites.write permissions to execute arbitrary commands by injecting TAB characters into the providerRootDirectory parameter used to construct GNU tar commands. The application uses escapeshellcmd instead of escapeshellarg and fails to quote the parameter, allowing TAB characters to survive sanitization and be interpreted as argument separators, enabling injection of arbitrary GNU tar arguments such as --checkpoint-action=exec to achieve remote code execution as the builds worker process user.	8.8	More Details
CVE-2026-28326	SolarWinds Access Rights Manager was reported to be affected by an unauthenticated remote code execution vulnerability. The issue stems from a hardcoded static key.	8.8	More Details
CVE-2026-90371	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: fix RXDMAD_C buffer recycling race The RXDMAD_C buffers come from the RRO data queues' page pools, which are bound to a different NAPI, so the direct page-pool recycle used here could race the owning NAPI; take the non-direct path as is already done for WED RX queues.	8.8	More Details
CVE-2026-54612	Vvveb is a powerful and easy to use CMS with page builder to build websites, blogs or ecommerce stores. From 1.0.0 until 1.0.8.5, saveGlobalElements() in admin/controller/editor/global-trait.php concatenates the attacker-controlled file portion of data-v-save-global to the active theme directory before loadHTMLFile() and file_put_contents() operate on it. An authenticated user with the default Editor role and editor/* permission can submit crafted HTML to module=editor/editor&action=save and traverse to an existing writable PHP file outside the theme directory. If the target is web-accessible, editor-controlled PHP content executes in the web server context; a shipped public/vadmin/index.php entrypoint can be used as an execution trampoline rather than requiring a test-only file. This can permit persistent webshell placement and compromise application confidentiality, integrity, and availability. This issue is fixed in version 1.0.8.5.	8.8	More Details
CVE-2026-89844	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Hold vport_slock for host map update in report ID acquisition qla24xx_report_id_acquisition() format-1 handling drops vport_slock after taking the vport reference and then calls qla_update_host_map() without the lock. That reaches qla_update_vp_map(), which mutates the ha->host_map btree via btree_insert32()/btree_update32()/btree_remove32() and is documented to require vport_slock to be held by the caller. Running it unlocked can race concurrent host_map updates and corrupt the btree. The format-2 path in the same function already wraps its host_map update (SET_AL_PA) in vport_slock; the format-1 path is the lone outlier. Hold vport_slock across the format-1 qla_update_host_map() call to honor the documented locking contract. The vref_count taken in the loop keeps the vport valid, so this only adds the missing host_map serialization.	8.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921: Add PCIe AER handler support to prevent system crash When an AER error occurs and the bus is hung, the register reads return 0xFFFFFFFF, causing the DMA queue state to be corrupted and resulting in an invalid memory access when accessing q->desc[] or q->entry[]. Unable to handle kernel paging request at virtual address ffffffc01099eac0 pc : mt76_dma_add_buf+0x124/0x188 [mt76] lr : mt76_dma_rx_fill+0x11c/0x1d8 [mt76] sp : ffffffc016d9bbf0 x29: ffffffc016d9bc10 x28: 0000000000000000 x27: 0000000000000000 x26: ffffffb7855e50b8 x25: ffffffb80d04f000 x24: 0000000000000000 x23: 0000000000000ec0 x22: ffffffb796803648 x21: ffffffb796801f80 x20: ffffffb7968035f8 x19: 0000000000000ec0 x18: 0000000000000000 x17:		

CVE-2026-90379	000000004ec00000 x16: 000000000ec00000 x15: ffffffc01099eac0 x14: 000000004ec00000 x13: 00000000ffc5a000 x12: ffffffc016d9bc32 x11: 00000000ffffffff x10: 0000000000000002 x9 : 0000000000000000 x8 : 000000000000b4ac x7 : 0000000000000a20 x6 : ffffffb6c1806400 x5 : 0000000000000000 x4 : ffffffb80d04f000 x3 : 0000000000000000 x2 : 0000000000000001 x1 : 000000000ec04000 x0 : ffffffb7968035f8 Call trace: mt76_dma_add_buf+0x124/0x188 [mt76 (HASH:1029 4)] mt76_dma_rx_reset+0xe8/0xfc [mt76 (HASH:1029 4)] mt7921_wpdma_reset+0x188/0x1b0 [mt7921e (HASH:ee48 5)] mt7921e_mac_reset+0x128/0x418 [mt7921e (HASH:ee48 5)] mt7921_mac_reset_work+0xac/0x1a8 [mt7921_common (HASH:f721 6)] process_one_work+0x188/0x514 worker_thread+0x12c/0x300 kthread+0x140/0x1fc ret_from_fork+0x10/0x30 Fix the invalid memory access by validating the DMA index read from the hardware before it is used as a queue index. An out-of-range value, such as the 0xFFFFFFFF returned while the bus is hung, is now clamped so it can no longer corrupt q->head or q->tail. In addition, check the bus_hung flag in mt7921_mac_reset_work() before attempting the reset sequence, reject MCU messages while the bus is hung, and install no-op bus operations when an unrecoverable AER error is detected, preventing further invalid hardware accesses. Due to hardware limitations - such as the lack of a connected hardware reset pin or the absence of host re-probe functionality - affected Wi-Fi devices may not fully recover to a normal operational state after certain errors, even with AER enabled.	8.8	More Details
CVE-2026-17086	The ShortPixel Image Optimizer – Optimize Images, Convert WebP & AVIF plugin for WordPress is vulnerable to PHP Object Injection in all versions up to, and including, 6.5.5 via deserialization of untrusted input . This makes it possible for authenticated attackers, with author-level access and above, to inject a PHP Object. No known POP chain is present in the vulnerable software, which means this vulnerability has no impact unless another plugin or theme containing a POP chain is installed on the site. If a POP chain is present via an additional plugin or theme installed on the target system, it may allow the attacker to perform actions like delete arbitrary files, retrieve sensitive data, or execute code depending on the POP chain present.	8.8	More Details
CVE-2026-92986	SiYuan before 3.8.4 renders document titles as HTML in the backlink dock tree without escaping markup characters. Attackers can set malicious titles through the rename API or crafted notebooks to execute scripts in the Electron renderer with access to child_process for command execution.	8.8	More Details
CVE-2026-86801	The To Do List Member WordPress plugin from 1.4 through 1.6 ships a file upload endpoint that does not load WordPress and therefore applies no authentication, capability or nonce check of any kind, and validates only the name of an uploaded file rather than its content, allowing unauthenticated users to store active content served from the site's own origin, and to list and delete the files already staged there.	8.8	More Details
CVE-2026-25283	Memory Corruption when copying unverified data from an external source exceeds the allocated buffer size.	8.8	More Details
CVE-2026-89951	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix stale receive device on merged fragments Fragment reassembly reuses the skb from the highest-numbered buffered fragment as the merged packet. When that fragment was received on a hard interface which is deleted before the chain completes, the merged skb can re-enter the receive path with a stale skb->dev and skb_iif. batadv_batman_skb_recv() passes such merged packets through the normal receive handlers again. DAT and bridge loop avoidance both derive the ARP header length from skb->dev, so they can dereference the freed net_device before the packet reaches the local mesh interface. Refresh the receive device metadata from the current receive device before running the packet handlers. This keeps internally reinjected merged fragments consistent with the normal receive path after hard interface teardown.	8.8	More Details
CVE-2026-89928	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Consume the locked rmap value in the lockless rmap walk __kvm_rmap_lock() deliberately elides the rmap lock when it observes an empty rmap. In that case kvm_rmap_lock_readonly() also re-enables preemption and returns zero, so the caller holds neither the rmap lock nor a preemption reference. The elision documents the invariant it relies on: * Elide the lock if the rmap is empty, as lockless walkers (read-only * mode) don't need to (and can't) walk an empty rmap, nor can they add * entries to the rmap. I.e. the only paths that process empty rmaps * do so while holding mmu_lock for write, and are mutually exclusive. kvm_rmap_age_gfn_range() ignores the returned value and unconditionally enters for_each_rmap_spte_lockless(). The iterator started with rmap_get_first(), which re-reads rmap_head->val rather than using the value returned by the lock. If a writer populates the rmap between the lock's read and the iterator's re-read, the aging path walks the newly installed rmap without holding its lock. For a KVM_RMAP_MANY rmap this leaves the walker following a pte_list_desc chain that it never locked. A writer holding mmu_lock for write may free that chain (e.g. kvm_zap_all_rmap_sptes() on the recycle path, or any rmap zap) via kmem_cache_free() while the walk is in progress, giving a slab use-after-free. Nothing serialises the two: the aging path runs without mmu_lock when CONFIG_KVM_MMU_LOCKLESS_AGING=y, and the rmap lock that would otherwise exclude the writer was elided. Because the empty path re-enables preemption, the interval between the two reads can span an arbitrary scheduling delay. Fix the class of bug by having the lockless walk consume the value returned by the lock instead of re-reading the rmap. Split rmap_get_first() into __rmap_get_first(), which starts an iterator from an already-read rmap value, and make for_each_rmap_spte_lockless() take that value and call __rmap_get_first() directly. kvm_rmap_age_gfn_range() passes the value returned by kvm_rmap_lock_readonly(): when the lock was elided the value is zero, __rmap_get_first() returns NULL, and the walk is skipped. No lockless walker re-reads the rmap, so the lock-elision invariant cannot be violated, and no lock()-without-paired-unlock() path is added to the aging code.	8.8	More Details
CVE-2026-90162	In the Linux kernel, the following vulnerability has been resolved: ksmbd: defer publishing granted locks to prevent UAF/double-free race In smb2_lock(), mid-batch granted locks are published to connection-wide (conn->lock_list) and file-wide (fp->lock_list) lists immediately upon vfs_lock_file() success, while also remaining tracked on the stack-local rollback_list. If a subsequent element in the same SMB2_LOCK request array fails validation or execution, the thread jumps to out: and walks rollback_list to undo previously granted locks. However, because the granted lock was already published to conn->lock_list, a concurrent UNLOCK request on the same connection can find the lock object and kfree() it before the rollback loop executes. When the granting thread subsequently walks rollback_list, it dereferences and frees the already-freed ksmbd_lock structure, resulting in a Use-After-Free and Double-Free (on both ksmbd_lock and struct file_lock). Fix this by deferring the publication of granted locks to conn->lock_list and fp->lock_list until after the entire array of lock elements has been processed without error. Mid-batch grants remain tracked exclusively on the request-local rollback_list until the whole batch succeeds, eliminating the race window.	8.8	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: use proto_lock for l2cap_data to fix l2cap_disconn_ind hci_conn::l2cap_data is accessed without locks in l2cap_disconn_ind via hci_conn_timeout (disc_work) ->		

2026-90256	hci_proto_disconn_ind -> l2cap_disconn_ind. This is UAF if the l2cap_conn is deleted concurrently. disc_work is disabled sync in hci_conn_del(), so we cannot take hci_dev_lock in disc_work. Fix by using proto_lock to guard l2cap_data, in addition to hdev->lock which is held in other access paths.	8.8	More Details
CVE-2026-89929	In the Linux kernel, the following vulnerability has been resolved: KVM: nVM: Ensure INVVPID is emulated on the correct physical CPU When emulating INVVPID, KVM executes INVVPID on the physical CPU using vpid02 (instead of the L1 assigned VPID), after doing some validations on the operands. However, it is possible that the physical CPU KVM executes INVVPID on is different from the CPU L2 is running on. For example, in the following scenario: - L2 runs on CPU #1 and exits to L1 (vmx->nested.vmcs02.cpu=1) - L1 migrates to CPU #2 and executes INVVPID - KVM executes INVVPID on CPU #2 - L1 migrates back to CPU #1 and runs L2 (vmx->nested.vmcs02.cpu=1) The TLB entries on CPU #1 are never invalidated, because INVVPID was executed on CPU #2, and vmcs02 never ran on a different pCPU (i.e. vmx_vcpu_load_vmcs() will *not* request KVM_REQ_TLB_FLUSH). Ensure that INVVPID is being executed on the same pCPU that L2 last ran on, and if not, fallback to clearing last_vpid=0 to trigger a full VPID flush on the next nested VM-Enter (as KVM will detect L1 using a different VPID for L2). If L2 ends up running on a different pCPU, KVM will flush the TLB anyway through vmx_vcpu_load_vmcs().	8.8	More Details
CVE-2026-93189	In the Linux kernel, the following vulnerability has been resolved: HID: core: quiesce input in hid_hw_stop() to prevent use-after-free A driver's probe calls hid_device_io_start() to enable input delivery, then fails at a later initialization step and unwinds via hid_hw_stop(). The unwind frees struct hidraw via hidraw_disconnect() while in-flight HID reports may still be running on another CPU, dereferencing the freed object through hidraw_report_event(). syzbot reports the resulting use-after-free for the corsair-psu HID driver. Edward Adam Davis posted a per-driver fix for corsair-psu that adds an explicit hid_device_io_stop() before hid_hw_stop() in the probe error path ("hwmon: prevent packets from going to driver for probe", 2026-04-28). Auditing the tree shows 15 drivers call hid_device_io_start(); 7 also call hid_device_io_stop() and 8 do not: drivers calling hid_device_io_start() without a matching hid_device_io_stop() before hid_hw_stop(): drivers/hwmon/corsair-psu.c (fix posted by Edward) drivers/hwmon/corsair-cpro.c drivers/hwmon/nzxt-kraken3.c drivers/hwmon/nzxt-smart2.c drivers/hwmon/gigabyte_waterforce.c drivers/hid/hid-logitech-dj.c drivers/hid/hid-nintendo.c drivers/hid/hid-mcp2221.c Roughly half of all callers of the API are exposed. Centralize the quiesce in hid_hw_stop() so callers do not have to remember the matching stop: if a driver has left hdev->io_started true on entry, call hid_device_io_stop() before hid_disconnect(). For the 7 drivers that already call hid_device_io_stop() correctly, hdev->io_started is false on entry, the guard short-circuits, and behavior is unchanged. No Fixes: tag because the affected drivers gained their hid_device_io_start() calls independently over years; the bug is a class-wide API misuse rather than a regression from one commit.	8.8	More Details
CVE-2026-78295	Unauthenticated Cross Site Request Forgery (CSRF) in Xagio SEO <= 7.1.0.43 versions.	8.8	More Details
CVE-2026-88825	The iGMS Direct Booking WordPress plugin before 2.0 does not authorise or escape its widget appearance settings, allowing unauthenticated users to store arbitrary web scripts that execute in the context of an administrator viewing the iGMS Direct Booking WordPress plugin before 2.0 settings, and in the browser of any visitor to a page displaying the booking widget.	8.8	More Details
CVE-2026-85130	The WPLP Cookie Consent WordPress plugin before 4.4.4 does not escape a value submitted through a public endpoint for the JavaScript context it is later output in on an administrative screen, allowing unauthenticated users to run arbitrary JavaScript in the session of an administrator who interacts with the logged entry. Only multisite installations are affected.	8.8	More Details
CVE-2026-92970	HUBzero CMS through 2.2.32 contains a path traversal vulnerability in project file upload handlers that allows authenticated project members to write arbitrary files outside the project repository. Attackers can supply traversal sequences in upload parameters to write files to attacker-chosen paths with web server privileges, potentially enabling code execution.	8.8	More Details
CVE-2026-78088	The Contest Gallery – Upload & Vote Photos, Media, Sell with PayPal & Stripe plugin for WordPress is vulnerable to Unauthenticated Arbitrary File Overwrite in all versions up to, and including, 32.0.1 due to insufficient file path validation in the 'baseUrlForFacebook' parameter. This makes it possible for authenticated attackers, with subscriber-level access and above, to overwrite known files which may lead to remote code execution when certain preconditions are met.	8.8	More Details
CVE-2026-92466	zlt2000 microservices-platform through 6.0.0 contains a missing authorization vulnerability where the zlt.security.auth.urlPermission.enable flag defaults to false, disabling all permission checks after authentication. Authenticated users with no roles can access administrative APIs including user management, role assignment, and Elasticsearch index operations by bypassing the disabled authorization enforcement.	8.8	More Details
CVE-2026-90255	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: fix the SCO setup context lifetime hci_setup_sync() queues a conn_handle_t with a NULL destroy callback, so the context is only freed if hci_enhanced_setup_sync() actually runs. An entry that is cancelled instead is leaked, as _hci_cmd_sync_cancel_entry() does not release entry->data when there is no destroy callback, and hci_cmd_sync_clear() cancels every pending entry when the controller is unregistered. The context also stores a bare hci_conn pointer, so the connection can be freed while the work is queued. The dequeue in hci_conn_del() does not cover it either, as it matches on entry->data == conn and entry->data is the wrapper here. Same problem as commit 2f5d635ad590 ("Bluetooth: hci_sync: hold conn in hci_connect_acl/le_sync() callbacks"). Hold the connection and release both from a destroy callback. The submission failure path drops both, since hci_cmd_sync_submit() does not call the destroy callback when it fails to queue.	8.8	More Details
CVE-2026-87786	The Dewa Kirim WordPress plugin through 1.0.0 does not escape delivery coordinates submitted at checkout before outputting them inside an inline script, allowing unauthenticated users to store JavaScript that runs in the session of an administrator who later opens the order.	8.8	More Details
CVE-2026-76552	The WP Import Export Lite WordPress plugin before 3.9.33 does not validate the type, extension or content of files it retrieves from a user-supplied URL during import, allowing users granted its import permission to store arbitrary files, including executable ones, on the server and achieve remote code execution.	8.8	More Details
CVE-2026-84829	The Optimole WordPress plugin before 4.2.12 does not properly escape a user supplied value before using it to build an image tag attribute, allowing unauthenticated users to inject arbitrary attributes into pages served to every visitor, which leads to Stored Cross-Site Scripting.	8.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: KVM: nVMX: Always flush vpid02 on first use Make sure		

CVE-2026-89932	vpid02 is always flushed on first use by setting last_vpid=0 when allocating vpid02. nested_vmx_transition_tlb_flush() will always detect a VPID change on first VM-Enter after VMXON, because VPID=0 in vmcs12 is not allowed if L1 enables VPID. This avoids using stale TLB entries from a previous lifetime of the VPID, that might have been associated with a different vCPU (or a completely different VM). Note that last_vpid is already being initialized as 0 when the vCPU is created, but it is not reset when vpid02 is freed on VMXOFF. Hence, the problem can only occur if L1 does VMXOFF -> VMXON, runs an L2, and KVM happens to reuse a VPID that has TLB entries on the physical CPU.	8.8	More Details
CVE-2026-88792	The Dictionary WordPress plugin through 1.0 does not have authorisation, sanitisation or escaping in place when adding or updating dictionary entries, allowing unauthenticated users to store arbitrary web scripts which will execute when a user views an affected entry.	8.8	More Details
CVE-2026-88904	The PuppyFW WordPress plugin through 0.4.4 does not have proper authorisation on one of its REST routes, which tests the caller against a capability taken from the request itself, allowing any authenticated user, including subscribers, to add, modify and delete arbitrary blog options and thereby escalate their privileges.	8.8	More Details
CVE-2026-92918	admin3 through 3.0.0 persists user session tokens in the audit log event body when publishing UserLoggedIn domain events. Attackers with log:view permission can read the JSON response from the GET /logs endpoint to harvest session tokens and replay them as bearer credentials for full user access.	8.8	More Details
CVE-2026-77614	Opencast is a free, open-source platform to support the management of educational audio and video content. Prior to versions 19.7 and 20.2, the default security configuration in etc/security/mh_default_org.xml accepts a client-selected JSESSIONID from the ;jsessionid= URL path parameter and does not replace it when the victim logs in. An unauthenticated attacker can send a crafted link to a victim whose browser has no active Opencast session cookie, wait for the victim to authenticate, and then reuse the known identifier as the victim's authenticated session. This can expose the victim's data and actions and can produce full administrative account takeover when the victim is an administrator. This issue is fixed in versions 19.7 and 20.2.	8.8	More Details
CVE-2026-89995	In the Linux kernel, the following vulnerability has been resolved: dma-direct: return struct page from dma_direct_alloc_from_pool() Commit 5b138c534fda ("dma-direct: factor out a dma_direct_alloc_from_pool helper") changed dma_direct_alloc_from_pool() to return the CPU address from dma_alloc_from_pool(). That fits dma_direct_alloc(), but dma_direct_alloc_pages() also uses the helper and expects a struct page *. Fix this by making dma_direct_alloc_from_pool() return the struct page * again, and pass the CPU address back through an out-parameter for the dma_direct_alloc() caller.	8.8	More Details
CVE-2026-89908	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Preserve memslot arch flags on KVM_MR_FLAGS_ONLY kvm_arch_prepare_memory_region() computes new->arch.flags, i.e. whether a memslot is KVM_MEM_HUGEPAGE_CAPABLE or KVM_MEM_HUGEPAGE_INCAPABLE, only for KVM_MR_CREATE and KVM_MR_MOVE, and returns early for every other change. But the generic code allocates a zeroed memslot for every change and never copies old->arch, so after a KVM_MR_FLAGS_ONLY update, e.g. toggling KVM_MEM_LOG_DIRTY_PAGES for live migration, the active memslot has arch.flags == 0. With both flags clear, fault_supports_huge_mapping() falls through to the alignment check on the HVA range alone, which no longer verifies that the GPA and HVA have the same offset within a PMD. A memslot that was marked KVM_MEM_HUGEPAGE_INCAPABLE because of a GPA/HVA offset mismatch can then be mapped with PMD entries on read faults, and since kvm_map_page() aligns the gfn and the pfn independently, the guest ends up accessing the wrong host pages, exactly the "d -> f, e -> g" case described in the comment above the check. Carry the arch flags over from the old memslot for KVM_MR_FLAGS_ONLY, as the GPA, HVA and size are guaranteed to be unchanged for that case.	8.8	More Details
CVE-2026-84860	ScadaLTS 2.8.1-release-candidate build 0 is affected by an Authorization Bypass Spring Security gates DWR endpoints by URL path pattern, but DWR itself dispatches method calls based on the POST body parameters c0-scriptName and c0-methodName. The crossDomainSessionSecurity setting in web.xml is set to false, which disables DWR's built-in origin validation. This means any authenticated user can invoke any DWR method (regardless of the URL-based access control) by sending their request to a URL they are permitted to access (e.g. MiscDwr.initializeLongPoll.dwr) while targeting a restricted class in the POST body. This is the systemic root cause that enables multiple other findings to be exploited as a low privilege user.	8.8	More Details
CVE-2026-84858	ScadaLTS 2.8.1-release-candidate build 0 is affected by an Authenticated Remote Code Execution via Scripting Sandbox Bypass The DWR "DataSourceEditDwr" class exposes the "validateScript" method that compiles and executes attacker-supplied JavaScript via the Rhino scripting engine. There are no authorization checks on this method and so it is possible for an attacker with access to a low privilege user to abuse this flaw by leveraging the DWR routing bypass.	8.8	More Details
CVE-2026-89960	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: fix stale pqap_hook pointer on error in vfio_ap_mdev_set_kvm() In vfio_ap_mdev_set_kvm(), kvm->arch.crypto.pqap_hook is set to &matrix_mdev->pqap_hook before the update locks are acquired and the mdev list is checked for a conflicting assignment. If another mdev is already attached to the same KVM instance, the function returns -EPERM without restoring the hook pointer, leaving kvm->arch.crypto.pqap_hook pointing at the failing matrix_mdev instead of the mdev that legitimately owns the KVM. Since matrix_mdev->kvm is never set on this error path, vfio_ap_mdev_unset_kvm() will not clean up the hook when matrix_mdev is later closed. If matrix_mdev is subsequently freed, any PQAP instruction executed by the guest will dereference the stale pointer through pqap_hook_rwsem, resulting in a use-after-free. Since kvm->arch.crypto.pqap_hook is only set in the vfio_ap_mdev_set_kvm() function and is cleared in the vfio_ap_mdev_unset_kvm() function, a check for 'kvm->arch.crypto.pqap_hook != NULL' is all that is needed to determine whether it belongs to another mdev. This will alleviate the need to iterate the matrix_dev->mdev_list list to see if the kvm object is assigned to another mdev.This was introduced in v3 to alleviate the need to take the mdevs_lock while iterating the list; however, this did not prevent a potential race condition. The pqap_hook_rwsem(write) is now performed inside get_update_locks_for_kvm(), which is updated to acquire pqap_hook_rwsem(write) between kvm->lock and mdevs_lock. This ordering is consistent with the PQAP intercept path, which acquires pqap_hook_rwsem in read mode while srcu is held under vcpu->mutex, establishing the dependency: kvm->lock -> vcpu->mutex -> srcu -> pqap_hook_rwsem(read). The pqap_hook_rwsem is now released inside the release_update_locks_for_kvm(), which is updated to release pqap_hook_rwsem(write) between mdevs_lock and kvm->lock. Additionally, kvm_put_kvm() in vfio_ap_mdev_unset_kvm() is moved after release_update_locks_for_kvm(). Previously it was called while kvm->lock was held; if it were ever the last reference, kvm_destroy_vm() would run under kvm->lock, which would deadlock.	8.8	More Details
CVE-2026-89959	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: Fix control domain removal in vfio_ap_mdev_cfg_remove The vfio_ap_config_remove function uses the bitmap_andnot function to clear bits from the matrix_mdev->matrix.adm bitmap (specifies the control domains assigned to the mdev). This prevents the explicitly unplugged control domains from being removed the KVM guest. The bitmap_and function is used instead.	8.8	More Details

CVE-2026-89957	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: Fix hot-unplug skipped when last AP adapter or domain removed The vfio_ap_mdev_hot_unplug_cfg() function uses the return value of bitmap_andnot() to determine whether the guest APCB needs to be updated. However, bitmap_andnot() returns false when the resulting destination bitmap is empty. This means that if the only adapter, domain or control domain assigned to an mdev is removed from the host's AP configuration, the bit is correctly cleared from the shadow APCB, but bitmap_andnot() returns false because the result is an empty bitmap. Consequently, do_hotplug remains 0 and vfio_ap_mdev_update_guest_apcb() is never called, leaving the KVM guest with stale hardware access to the unplugged AP devices. Fix this by replacing the bitmap_andnot() return value check with bitmap_intersects() to determine whether the shadow APCB actually overlaps with the removal mask. If there is an intersection, call bitmap_andnot() solely for its side effect of clearing the bits, then unconditionally set do_hotplug to trigger the guest APCB update.	8.8	More Details
CVE-2026-92137	Jenkins Robot Framework Plugin 6.2.2 and earlier does not check that the archive directory configured for Robot Framework report files is contained within the build directory on the Jenkins controller, allowing attackers with Item/Configure permission to create or replace arbitrary files on the Jenkins controller file system with attacker-specified content, which can lead to remote code execution.	8.8	More Details
CVE-2026-89907	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Validate MSI data before routing it to EIOINTC pch_msi_set_irq() passes e->msi.data straight into eiointc_set_irq() as the irq number. The MSI data comes from userspace, that either via a KVM_IRQ_ROUTING_MSI entry set with KVM_SET_GSI_ROUTING (used by irqfd and KVM_IRQ_LINE) or directly via KVM_SIGNAL_MSI, and is never checked against EIOINTC_IRQS. eiointc_set_irq() uses the value with __set_bit()/__clear_bit() on the 256-bit isr bitmap, eiointc_update_irq() then indexes sw_coremap[] and the per-cpu coreisr/sw_coreisr bitmaps with it. Therefore a data value >= 256 reads and writes memory past the end of those arrays, i.e. any process holding a VM fd can corrupt kernel memory beyond the allocation of loongarch_eiointc. Reject MSI data that doesn't fit in the EIOINTC irq space. The DMSINTC path is unaffected as it decodes the vector from the address and masks it.	8.8	More Details
CVE-2026-89913	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: vgic-v3: take an LPI reference in vgic_v3_save_pending_tables vgic_v3_save_pending_tables() iterates dist->lpi_xa using xa_for_each() and dereferences the returned struct vgic_irq in the loop body without holding a reference on the LPI. The xarray iterator only provides temporary RCU coverage while looking up the current entry. That is not sufficient for this loop body, which reads fields from struct vgic_irq and performs guest memory accesses before the iteration completes. A concurrent path can trigger this race: the irqfd cached injection path (vgic_its_inject_cached_translation) obtains a transient LPI reference via vgic_its_check_cache() without holding kvm->lock, vcpu->mutex, config_lock, or its_lock. If guest ITS DISCARD then drops the cache and ITE references under its_lock, the transient inject reference may become the final one. When vgic_put_irq() drops it, the LPI is erased from lpi_xa and freed via kfree_rcu(). Meanwhile, vgic_v3_save_pending_tables() may still hold a stale pointer obtained from the xarray iterator and dereference it after the RCU grace period completes. Fix this by re-fetching each iterated LPI via vgic_get_irq(), which takes a stable reference, and dropping it with vgic_put_irq() on all paths. This matches the pattern already used by other lpi_xa iterators in the vgic ITS code.	8.8	More Details
CVE-2026-85127	The VikBooking Hotel Booking Engine & PMS WordPress plugin before 1.8.15 does not restrict the type of files unauthenticated visitors may attach to its live chat, nor sanitize their contents, allowing them to store active content which is executed in the context of an administrator viewing the conversation.	8.8	More Details
CVE-2026-20361	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Nexus Dashboard engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20361 are related to SQL injection issues that are grouped under the Common Weakness Enumeration (CWE) CWE-89.	8.8	More Details
CVE-2026-85122	The Easy Form Builder by WhiteStudio WordPress plugin before 4.2.0 does not validate a submitted value against the stored configuration for some of its form types, allowing unauthenticated users to store arbitrary content which is then rendered unescaped in an admin page, leading to Stored XSS.	8.8	More Details
CVE-2026-82964	Improper preservation of permissions in the Avast sandbox minifilter driver (aswSnx.sys) on Windows allows a local, low-privileged attacker executing inside the sandbox to escape file isolation and escalate to SYSTEM. When the sandbox virtualizes a file it copies the original security descriptor, but the driver opened the virtualization target object with GENERIC_WRITE and FILE_WRITE_ATTRIBUTES only, omitting WRITE_DAC. Every attempt to apply the original DACL therefore failed, and the failure was discarded silently, leaving virtualized copies of sensitive files with permissive permissions. Because the IRP_MJ_CREATE callback additionally did not strip WRITE_DAC for sensitive directories, a sandboxed process could rewrite the security descriptor of a virtualized object, read the virtualized copy of the SAM database, extract local NTLM password hashes and execute code as SYSTEM. The absence of an IRP_MJ_SET_SECURITY callback in the driver's operation registration table is a related defense-in-depth gap, but it is not the control that prevents this attack.	8.8	More Details
CVE-2026-90286	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/gfx6: Use PFP on the compute queues too On GFX6, the compute rings use the same CP path as the graphics ring. The only difference is that they don't support draw commands. (As opposed to GFX7 and newer which have a separate command parser that is called MEC for compute queues.) This means that we have to take into consideration that the PFP also exists on compute queues on GFX6: Use PFP for register writes on both graphics and compute queues. In the pipeline sync, use the PFP to wait for the previous fence (and not the ME) to prevent the PFP from starting to execute the next submission while the ME is still in the previous submission. After a VM flush, emit PFP_SYNC_ME on compute queues as well.	8.8	More Details
CVE-2026-63459	Vendure is an open-source headless commerce platform. Prior to 3.6.5, RichTextDescriptionCell in packages/dashboard/src/lib/components/shared/table-cell/order-table-cell-components.tsx attempts to strip markup by assigning an administrator-controlled description to a live element's innerHTML and then reading textContent. Active resource markup can execute an event handler during the innerHTML assignment before textContent is read. A lower-privilege administrator can store such markup in descriptions rendered by the Products list, Collections list, Promotions list, Payment Methods list, or Shipping Methods list, and script executes when another administrator views the affected row. This stored cross-site scripting can compromise the viewing administrator's session and enable cross-privilege or cross-channel administrative actions. This issue is fixed in version 3.6.5.	8.7	More Details
CVE-	Improper validation of pod identifier uniqueness in aws-network-policy-agent in Amazon EKS Network Policy Agent before v1.4.0 might allow an authenticated remote user to bypass NetworkPolicy enforcement on co-located pods in other namespaces via		

2026-86831	crafted pod and namespace names that produce pod identifier collisions. To remediate this issue, users should upgrade to Amazon EKS Network Policy Agent 1.4.0 or later and Amazon VPC CNI Managed Add-on v1.22.4 or later (which includes Network Policy Agent v1.4.0).	8.7	More Details
CVE-2026-77615	Paella Player is a set of libraries to create a multi stream video player. Prior to Paella Player 2.12.11 (as used in Opencast prior to 19.7 and 20.2), there is a potential XSS attack though closed captions cue text. This vulnerability is fixed in 2.12.11.	8.7	More Details
CVE-2026-20135	A vulnerability in the TLS 1.3 implementation in Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause an affected device to reload unexpectedly, resulting in a denial of service (DoS) condition. This vulnerability is due to improper buffer management during the TLS 1.3 connection. An attacker could exploit this vulnerability by sending a crafted TLS 1.3 packet to an affected system through a TLS 1.3-enabled listening socket. A successful exploit could allow the attacker to cause the LINA process to crash, which would cause the device to reload. The reload can happen before or after authentication of the connection.Note: TLS 1.3 connections include both data traffic and user-management traffic.	8.6	More Details
CVE-2026-20250	A vulnerability in Datagram TLS (DTLS) message handling of Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software for Cisco Secure Firewall 3100 Series and 4200 Series devices could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to improper resource management when processing certain DTLS messages. An attacker could exploit this vulnerability by sending a crafted stream of DTLS traffic to an affected device. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition.	8.6	More Details
CVE-2026-87963	The Yo WordPress plugin from 1.1 through 1.3.1 does not sanitize or parameterize the username request parameter before using it in a SQL query, and reads it before WordPress applies its request escaping, allowing unauthenticated attackers to perform SQL injection and read arbitrary database contents including administrator password hashes.	8.6	More Details
CVE-2026-68791	Incorrect authorization in Azure Machine Learning allows an unauthorized attacker to disclose information over a network.	8.6	More Details
CVE-2026-20154	A vulnerability in the system rate-limiting process for syslog message 419002 of Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause high CPU utilization on an affected device, resulting in a denial of service (DoS) condition. This vulnerability is due to improper rate limiting for syslog message 419002. An attacker could exploit this vulnerability by sending a flood of TCP synchronization (SYN) packets to an affected device. A successful exploit could allow the attacker to cause high CPU utilization, resulting in performance degradation.	8.6	More Details
CVE-2026-92972	SGLang through 0.5.19 in prefill/decode disaggregation mode contains an unauthenticated PUT /route endpoint on the prefill bootstrap service that allows attackers to poison the KV transfer routing table. Attackers can supply arbitrary rank_ip and rank_port values to redirect decode workers to attacker-controlled endpoints, causing denial of service or disclosure of KV transfer metadata including session identifiers and tensor-parallel topology parameters.	8.6	More Details
CVE-2026-78472	The Ni WooCommerce Sales Report WordPress plugin before 4.2.0 does not sanitise and escape a parameter before using it in a SQL statement, allowing unauthenticated users to perform SQL injection attacks.	8.6	More Details
CVE-2026-20249	A vulnerability in the certification authentication feature of Internet Key Exchange version 2 (IKEv2) for Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause an affected device to reload unexpectedly. This vulnerability is due to a logic error during the certificate authentication phase of the IKEv2 connection setup. An attacker could exploit this vulnerability by attempting to establish an IKEv2 VPN connection with a crafted certificate. A successful exploit could allow the attacker to cause the IKEv2 process to crash, causing a denial of service (DoS) condition.	8.6	More Details
CVE-2026-20352	A vulnerability in the RADIUS feature of Cisco Identity Services Engine (ISE) could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to improper handling of certain RADIUS requests. An attacker could exploit this vulnerability by sending a crafted RADIUS request directly to an affected device. A successful exploit could allow the attacker to cause the ISE node to become unavailable. For single node deployments in that condition, endpoints that have not already authenticated would be unable to access the network until the node comes back up on its own.	8.6	More Details
CVE-2026-92954	vm2 is a sandbox library for running untrusted JavaScript in Node.js. In versions >= 3.10.0 and <= 3.11.7, Promises returned from the host realm into the sandbox are not marked as handled at the bridge boundary; only Promises created inside the sandbox are wrapped with a rejection-swallowing handler (lib/setup-sandbox.js), and the bridge only installs host-side rejection sanitizers when sandbox code calls .then/.catch/.finally. As a result, code running in the sandbox can invoke a host function that returns a rejected Promise (for example events.once() exposed via the NodeVM events builtin, or any embedder-provided Promise-returning API) and simply ignore the return value, leaving the host Promise unhandled so that Node.js's default unhandled-rejection behavior terminates the host process. This is an incomplete fix of GHSA-hw58-p9xv-2mjh. The issue is fixed in version 3.11.8.	8.6	More Details
CVE-2026-92950	vm2 before 3.11.7 contains a sandbox escape vulnerability in the CLI tool that allows attackers to execute arbitrary code in the host Node.js process. Attackers can supply a malicious script file to the vm2 CLI that uses require(__filename) to re-execute itself in the host realm, bypassing sandbox isolation and accessing host modules like fs and child_process.	8.6	More Details
CVE-2026-20295	A vulnerability in the sftunnel inter-device communication protocol of Cisco Secure FMC Software and Cisco Secure FTD Software could allow an unauthenticated, remote attacker to exhaust the available memory of an affected device. This vulnerability is due to improper management of memory resources during sftunnel TLS connection setup. An attacker could exploit this vulnerability by sending crafted sftunnel TLS frames to an affected device during the connection setup. A successful exploit could allow the attacker to exhaust the available memory on the affected device, which could result in a DoS condition.	8.6	More Details

CVE-2026-87775	The Tz Weekly Radio Schedule WordPress plugin through 1.8.1 does not sanitize and escape a parameter before using it to build a SQL query on an AJAX action available to unauthenticated users, allowing unauthenticated attackers to perform SQL injection attacks and extract sensitive data from the database.	8.6	More Details
CVE-2026-87774	The Tz Weekly Radio Schedule WordPress plugin through 1.8.1 does not sanitize and escape a parameter before using it to build a SQL query on an AJAX action available to unauthenticated users, allowing unauthenticated attackers to perform SQL injection attacks and extract sensitive data from the database.	8.6	More Details
CVE-2026-87771	The Product Question and Answer WordPress plugin through 1.1.0 does not sanitize and escape parameters before using them in SQL queries on AJAX actions available to unauthenticated users, allowing unauthenticated attackers to perform SQL injection attacks and extract sensitive data from the database.	8.6	More Details
CVE-2026-87770	The Price Drop Alert for Woo Commerce WordPress plugin through 1.1 does not sanitize and escape parameters before using them in a SQL query on an AJAX action available to unauthenticated users, allowing unauthenticated attackers to perform SQL injection attacks and extract sensitive data from the database.	8.6	More Details
CVE-2026-87767	The wp shortcut link and advertisement baner WordPress plugin through 1.2.0 does not sanitize and escape a parameter before using it in a SQL query on an AJAX action available to unauthenticated users, allowing unauthenticated attackers to perform SQL injection attacks and extract sensitive data from the database.	8.6	More Details
CVE-2026-92576	HKUDS nanobot before 0.3.0 contains a server-side request forgery vulnerability in the WebFetchTool component where the _validate_url() function fails to block internal IP ranges and private addresses. Attackers can send messages instructing the bot to fetch cloud metadata endpoints, localhost services, and RFC 1918 addresses to extract IAM credentials and internal service data.	8.6	More Details
CVE-2026-92958	vm2 through 3.11.6 contains a builtin-module denylist bypass in NodeVM. When the embedder uses the builtin wildcard together with negative entries (e.g. require: { builtin: ['*', '-fs', '-child_process'] }), negative entries are matched by exact module name in lib/builtin.js, so -fs removes only the builtin named fs and does not remove builtin subpaths such as fs/promises. Sandboxed code can therefore call require('fs/promises') or require('node:fs/promises') and reach the promise-based filesystem API despite fs being denied; node: prefix handling is likewise inconsistent (a -node:fs/promises entry does not block require('fs/promises')). Host file creation and writing were confirmed via fsp.writeFile(), and other fs/promises operations (cp, mkdir, rename, rm, rmdir, truncate, read operations, etc.) are also reachable. This issue is fixed in vm2 3.11.7.	8.5	More Details
CVE-2026-76412	A vulnerability in the remote diagnostics debugger of Cisco Secure FMC Software could allow an authenticated, remote attacker to enable the remote diagnostics debugger service. This vulnerability is due to an error when checking the privilege level of a user who is invoking remote diagnostics. An attacker could exploit this vulnerability by authenticating to the device, either through the web-based management interface or the REST API, and using the remote diagnostics debugger to grant a user elevated privileges. A successful exploit could allow the attacker to elevate privileges to root. Notes: To exploit this vulnerability, the attacker must have valid user credentials on the affected device. The CVSSv3.1 Attack Complexity is High due to the multistage process required to fully exploit this vulnerability.	8.5	More Details
CVE-2026-93292	SigNoz versions from 0.88.0 before 0.142.1 contain a SQL injection vulnerability in trace-funnel analytics endpoints that interpolate service_name and span_name fields into ClickHouse string literals without escaping. Authenticated attackers can inject SQL through funnel step definitions to execute arbitrary queries and read results in HTTP responses.	8.5	More Details
CVE-2026-66580	Contributor SQL Injection in Product Feed Manager <= 7.12.0 versions.	8.5	More Details
CVE-2026-86359	Dell Repository Manager, versions prior to 3.5.2, contains an Incorrect Default Permissions vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Elevation of privileges.	8.5	More Details
CVE-2026-92126	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not reject @Builder annotations whose builderStrategy member names an arbitrary class, allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to execute code outside the sandbox if a suitable class is present on the classpath of the component that evaluates the script.	8.5	More Details
CVE-2026-93426	SigNoz versions 0.87.0 before 0.142.0 fail to escape user-supplied telemetry field-key names in the v5 query_range API, allowing authenticated users to inject SQL. Attackers with Viewer role or higher can embed backticks and quotes in field names to break out of identifiers and string literals, executing arbitrary ClickHouse SQL to read system tables and exfiltrate data.	8.5	More Details
CVE-2026-79708	GitLab has remediated an issue in GitLab EE affecting all versions from 19.0 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that, under certain conditions could have allowed an authenticated user with developer permissions to execute a policy test pipeline on projects within their group and access protected CI/CD variables restricted to higher-privileged roles, due to insufficient scope validation.	8.5	More Details
CVE-2026-20334	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20334 are related to issues concerning improper adherence to coding standards that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-710.	8.4	More Details
CVE-2026-89885	In the Linux kernel, the following vulnerability has been resolved: media: platform: mtk-mdp3: Fix SCP device refcounting mdp_probe() first tries to get the SCP handle with scp_get(). When that fails, it falls back to looking up the SCP platform device with __get_pdev_by_id() and then reads its driver data. The fallback lookup returns the platform device with a reference, just like scp_get() does. However, the fallback path currently drops that reference immediately after platform_get_drvdata(). The driver later still calls scp_put(mdp->scp) unconditionally from the probe error path and from mdp_video_device_release(), which drops the SCP device reference again. Keep the fallback reference until the existing scp_put() call, so that the fallback path follows the same ownership rules as the scp_get() path.	8.4	More Details
	In the Linux kernel, the following vulnerability has been resolved: platform/chrome: cros_ec_typec: Reject out-of-bounds PD cap		

CVE-2026-93190	count cros_typec_register_partner_pdos() copies the partner PDOs from the EC TYPEC_STATUS response into the fixed caps_desc.pdo[PDO_MAX_OBJECTS] array. memcpy(caps_desc.pdo, resp->source_cap_pdos, sizeof(u32) * resp->source_cap_count); ... memcpy(caps_desc.pdo, resp->sink_cap_pdos, sizeof(u32) * resp->sink_cap_count); PDO_MAX_OBJECTS is 7. source_cap_count and sink_cap_count are u8 fields from the EC. The only check is that they are not both zero. If either is larger than 7, the memcpy writes past the end of the array on the stack. A count of 255 overflows it by about 1 KB. The EC source arrays are only seven entries wide. A larger count reads past them too. The ChromeOS EC firmware caps these counts today, so a compliant setup does not hit this. The kernel should still validate these values rather than trust them. Validate the counts in cros_typec_register_partner_pdos() next to the memcpy. Skip the PDO registration if either count is above PDO_MAX_OBJECTS. The rest of cros_typec_handle_status() still runs so events are handled and cleared.	8.4	More Details
CVE-2026-89904	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Fix acpi_package_ids[] array overflow With LoongArch virt machine, a typical setting is one core per socket, there will max 256 sockets (packages) on one VM. With PPTT acpi table, array acpi_package_ids[] will be overflowed. Here change the array size of acpi_package_ids[] with the max value of MAX_PACKAGES and KVM_MAX_VCPUS.	8.4	More Details
CVE-2026-89795	In the Linux kernel, the following vulnerability has been resolved: PCI: Allow per function PCI slots to fix slot reset on s390 On s390 systems, which use a machine level hypervisor, PCI devices are always accessed through a form of PCI pass-through which fundamentally operates on a per PCI function granularity. This is also reflected in the s390 PCI hotplug driver which creates hotplug slots for individual PCI functions. Its reset_slot() function, which is a wrapper for zpci_hot_reset_device(), thus also resets individual functions. Currently, the pci_create_slot() assigns the same pci_slot object to multifunction devices. This approach worked fine on s390 systems that only exposed virtual functions as individual PCI domains to the operating system. Since commit 44510d6fa0c0 ("s390/pci: Handling multifunctions") s390 supports exposing the topology of multifunction PCI devices by grouping them in a shared PCI domain. This creates a problem when resetting a function through the hotplug driver's slot_reset() interface. When attempting to reset a function through the hotplug driver, the shared slot assignment causes the wrong function to be reset instead of the intended one. It also leaks memory as we do create a pci_slot object for the function, but don't correctly free it in pci_slot_release(). Add a flag for struct pci_slot to allow per function PCI slots for functions managed through a hypervisor, which exposes individual PCI functions while retaining the topology. Since we can use all 8 bits for slot 'number' (for ARI devices), change slot 'number' u16 to account for special values PCI_SLOT_PLACEHOLDER and PCI_SLOT_ALL_DEVICES.	8.4	More Details
CVE-2026-89943	In the Linux kernel, the following vulnerability has been resolved: ASoC: loongson: Fix error handling in ACPI property parsing In loongson_card_parse_acpi(), the return value of device_property_read_string() for the `codec-dai-name` property was ignored. If the property is missing or invalid, an uninitialized pointer would be used later, potentially leading to undefined behavior. Fix this by checking the return value and propagating the error appropriately.	8.4	More Details
CVE-2026-90398	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix stride mismatch in mac_phy_caps_parse() Currently, in ath11k_wmi_tlv_mac_phy_caps_parse(), kcalloc() sizes the mac_phy_caps buffer as tot_phy_id * len, where len is clamped to min(firmware_len, sizeof(struct wmi_mac_phy_capabilities)). The subsequent memcpy() destination advances by sizeof(full struct) per slot via C pointer arithmetic, not by the clamped len. When firmware sends short TLVs, the second and later slots are written past the end of the allocation. The reader in ath11k_pull_mac_phy_cap_svc_ready_ext() also indexes the buffer with full-struct pointer arithmetic, so the allocation must match that stride. Fix by using kzalloc_objs(), which derives the element size from the pointer type, making allocation size and pointer stride provably consistent regardless of what len the firmware provides. Compile tested only.	8.4	More Details
CVE-2026-90399	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix stride mismatch in mac_phy_caps_parse() Currently, in ath12k_wmi_mac_phy_caps_parse(), kzalloc() sizes the mac_phy_caps buffer as tot_phy_id * len, where len is clamped to min(firmware_len, sizeof(struct ath12k_wmi_mac_phy_caps_params)). The subsequent memcpy() destination advances by sizeof(full struct) per slot via C pointer arithmetic, not by the clamped len. When firmware sends short TLVs, the second and later slots are written past the end of the allocation. The reader in ath12k_pull_mac_phy_cap_svc_ready_ext() also indexes the buffer with full-struct pointer arithmetic, so the allocation must match that stride. Fix by using kzalloc_objs(), which derives the element size from the pointer type, making allocation size and pointer stride provably consistent regardless of what len the firmware provides. Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.1.c7-00108-QCAHMTSWPL_V1.0_V2.0_SILICONZ_UPSTREAM-3	8.4	More Details
CVE-2026-90120	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v5: Check get_logical_index() return value in MADT IAFFID parsing In gic_acpi_parse_iaffid() a given MADT GICC entry might not correspond to a logical cpu recognized by the kernel, resulting in the cpu variable initialization to an error value. Currently, the get_logical_index() return value is not checked for failure, which might result in out-of-bounds memory corruption while trying to index a per_cpu variable array. Add a check to evaluate get_logical_index() return value.	8.4	More Details
CVE-2026-89806	In the Linux kernel, the following vulnerability has been resolved: drm/sysfb: ofdrm: Fix integer overflow in fb_size calculation The framebuffer size calculation `fb_size = linebytes * height` can overflow when both values are large (e.g., 46341 * 46341 > INT_MAX). Since linebytes and height are both int types, the multiplication is performed as int * int, which results in undefined behavior on overflow. Use check_mul_overflow() to detect and prevent this overflow, consistent with the approach used in simpledrm.c and corebootdrm.c.	8.4	More Details
CVE-2026-90191	In the Linux kernel, the following vulnerability has been resolved: mailbox: riscv-sbi-mpxy: validate RPMI notification lengths The SBI return value controls how many bytes are copied from shared memory into the RPMI notification buffer. It is not validated against the negotiated shared-memory size before that copy. The event walker also uses a reversed loop condition and can inspect a short event record. Validate the complete notification length before copying it, iterate only while a full event header remains, and stop when a declared event payload extends beyond the copied notification data.	8.4	More Details
CVE-2026-89992	In the Linux kernel, the following vulnerability has been resolved: cpuidle: dt_idle_genpd: kfree() the original name allocation dt_idle_pd_alloc() kasprintf()s the full node path, then points pd->name at kbasename() of that string. dt_idle_pd_free() kfree()s pd->name, which is no longer the start of the allocation. Copy the basename instead.	8.4	More Details
CVE-2026-93063	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mei: check SAP message length before reading it Verify the SAP message size is not larger than the local buffer before reading the message to avoid buffer overflow.	8.4	More Details
CVE-			

2026-79298	An issue in Howyar Technologies Inc SysReturn Versions prior to 11.3.034 and fixed in v.11.3.0.34 allows a local attcker to execute arbitrary code via the BOOTia32.efi and a crafted cloak32.dat file on the ESP.	8.4	More Details
CVE-2026-76825	RestrictedPython is a tool that helps define a subset of the Python language for accepting program input in a trusted environment. Prior to 8.4, RestrictedPython could allow a sandbox escape when a custom import policy or globals exposed the standard library string module, the string.Formatter class, a Formatter instance, or a Formatter subclass to restricted code. The string.Formatter methods format, get_field, get_value, and vformat performed attribute and item traversal internally without passing through RestrictedPython's safer_getattr protections. Restricted code could use those live object references to reach function globals, builtins, file access, or code execution primitives, affecting confidentiality, integrity, and availability in the host environment. This issue is fixed in version 8.4.	8.4	More Details
CVE-2026-93196	In the Linux kernel, the following vulnerability has been resolved: nvdimm: virtio_pmem: refcount requests for token lifetime KASAN reports slab-use-after-free in __wake_up_common(): BUG: KASAN: slab-use-after-free in __wake_up_common+0x114/0x160 Read of size 8 at addr ffff88810fdcb710 by task swapper/0/0 CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.19.0-next-20260220-00006-g1eae5f204ec3 #4 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Arch Linux 1.17.0-2-2 04/01/2014 Call Trace: <IRQ> dump_stack_lvl+0x6d/0xb0 print_report+0x170/0x4e2 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 ? __virt_addr_valid+0x1dc/0x380 kasan_report+0xbc/0xf0 ? __wake_up_common+0x114/0x160 ? __wake_up_common+0x114/0x160 __wake_up_common+0x114/0x160 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 __wake_up+0x36/0x60 virtio_pmem_host_ack+0x11d/0x3b0 ? sched_balance_domains+0x29f/0xb00 ? __pfx_virtio_pmem_host_ack+0x10/0x10 ? _raw_spin_lock_irqsave+0x98/0x100 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 vring_interrupt+0x1c9/0x5e0 ? __pfx_vp_interrupt+0x10/0x10 vp_vring_interrupt+0x87/0x100 ? __pfx_vp_interrupt+0x10/0x10 __handle_irq_event_percpu+0x17f/0x550 ? __pfx__raw_spin_lock+0x10/0x10 handle_irq_event+0xab/0x1c0 handle_fasteoi_irq+0x276/0xae0 __common_interrupt+0x65/0x130 common_interrupt+0x78/0xa0 </IRQ> virtio_pmem_host_ack() wakes a request that has already been freed by the submitter. This happens when the request token is still reachable via the virtqueue, but virtio_pmem_flush() returns and frees it. Fix the token lifetime by refcounting struct virtio_pmem_request. virtio_pmem_flush() holds a submitter reference, and the virtqueue holds an extra reference once the request is queued. The completion path drops the virtqueue reference, and the submitter drops its reference before returning.	8.4	More Details
CVE-2026-89980	In the Linux kernel, the following vulnerability has been resolved: ALSA: harmony: initialize locks before requesting IRQ snd_harmony_create() registers the IRQ before initializing h->lock and h->mixer_lock. A pending interrupt can invoke the handler while these locks are uninitialized. Initialize both locks before requesting the IRQ so the handler always sees valid lock state.	8.4	More Details
CVE-2026-89856	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Clamp MSI-X derived queue counts to avoid truncation ha->msix_count is u16, but ha->max_req_queues, ha->max_rsp_queues and ha->max_qpairs are u8. Deriving the queue count as "ha->max_req_queues = ha->msix_count - 1" therefore truncates: a board (or a misconfigured/malicious hot-plugged device) advertising 257 MSI-X vectors yields msix_count - 1 == 256, which truncates to 0. An MSI-X count of 1 zeroes it as well, and in target mode the subsequent "ha->max_req_queues--" then underflows 0 to 255. When the count is 0, qla2x00_alloc_queues() calls kzalloc_objs(struct req_que *, 0), which returns ZERO_SIZE_PTR. That is not NULL, so the allocation check passes and the following "ha->req_q_map[0] = req" dereferences ZERO_SIZE_PTR, corrupting memory or crashing the kernel. Add qla_calc_queue_count() to clamp the derived value into [1, QLA_MAX_QUEUES - 1] so it always fits in u8 and is never zero, and use it at all three derivation sites (qla25xx_iospace_config(), qla83xx_iospace_config() and qla24xx_enable_msix()). Also guard the target-mode decrement so it cannot reintroduce a zero (which would in turn underflow max_qpairs).	8.4	More Details
CVE-2026-89782	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: reject restart table growth beyond U16_MAX entries During $LogFile replay, log_replay() indexes the transaction table by the transact_id taken from the log record header. check_log_rec() only verifies that transact_id is non-zero and properly aligned, not its magnitude, so a crafted image can request an arbitrarily large index. alloc_rsttbl_from_idx() grows the table to cover that index via extend_rsttbl(), which passes the new entry count to init_rsttbl(): rt = init_rsttbl(esize, used + add); used + add is computed as u32 but init_rsttbl() takes a u16, and the count is stored in struct RESTART_TABLE as a __le16. When used + add exceeds U16_MAX it is truncated, init_rsttbl() allocates a table far smaller than the index requires, and alloc_rsttbl_from_idx() then dereferences and writes at the original, untruncated offset -- an out-of-bounds access past the allocation, reachable by mounting a crafted NTFS image. BUG: KASAN: use-after-free in alloc_rsttbl_from_idx (fs/ntfs3/fslog.c:950) Read of size 4 at addr ffff8880327ffff8 by task exploit alloc_rsttbl_from_idx (fs/ntfs3/fslog.c:950) log_replay (fs/ntfs3/fslog.c:4562) ntfs_loadlog_and_replay (fs/ntfs3/fsntfs.c:324) ntfs_fill_super (fs/ntfs3/super.c:1393) get_tree_bdev_flags vfs_get_tree path_mount __x64_sys_mount A restart table is limited to U16_MAX entries by its __le16 count, so a larger growth request is invalid input. Reject it in extend_rsttbl(); all callers already handle a NULL return.	8.4	More Details
CVE-2026-89781	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: fix out-of-bounds read in read_log_rec_buf() read_log_rec_buf() copies a log record into a caller buffer starting at u32 off = lsn_to_page_off(log, lsn) + log->record_header_len; log->record_header_len (and log->data_off, used for the following pages) comes verbatim from the on-disk restart area and is only checked for 8-byte alignment in is_rst_area_valid(), so off can exceed log->page_size. "tail = log->page_size - off" then underflows and memcpy() reads past the page_size-sized buffer returned by read_log_page(), spilling adjacent slab memory into the replay buffer. This is reachable by mounting a crafted NTFS image: BUG: KASAN: slab-out-of-bounds in read_log_rec_buf+0x216/0x580 Read of size 64 at addr ffff88800a877ff8 by task exploit/127 read_log_rec_buf fs/ntfs3/fslog.c:2299 log_replay fs/ntfs3/fslog.c:4216 ntfs_loadlog_and_replay fs/ntfs3/fsntfs.c:324 ntfs_fill_super fs/ntfs3/super.c:1392 get_tree_bdev_flags fs/super.c:1694 __x64_sys_mount fs/namespace.c:4360 The buggy address is located 4088 bytes to the right of the 4096-byte region [ffff88800a876000, ffff88800a877000) Reject an in-page offset outside the current page before the copy. [almaz.alexandrovich@paragon-software.com: replaced the >= sign with >]	8.4	More Details
CVE-2026-90231	In the Linux kernel, the following vulnerability has been resolved: apparmor: fix unconfined user namespace restriction forced stack If a task is already confined by a stack the unprivileged transition restriction on unconfined is not correctly, applied. This results in an escape if two transitions through an unconfined profile can be executed. Fix this by pushing the check into the per profile label build. The check will always be done against unconfined and result in a stack of just the unconfined component when necessary.	8.4	More Details
	In the Linux kernel, the following vulnerability has been resolved: media: saa7164: fix cleanup on resource allocation failure saa7164_dev_setup() adds the device to the global saa7164_devlist before requesting the PCI BAR memory regions. If		

CVE-2026-89877	get_resources() fails, saa7164_dev_setup() decrements the device count and returns an error, but leaves the device on saa7164_devlist. The probe error path then frees the device, leaving a dangling entry on the global list. Reuse the existing MMIO mapping error path to remove the device from saa7164_devlist and decrement the device count before returning. Also release BAR0 if it was successfully requested but the BAR2 request fails.	8.4	More Details
CVE-2026-90347	In the Linux kernel, the following vulnerability has been resolved: arm64: ptrace: Keep 'orig_x0' in-sync with x0 on syscall entry Commit e057b9477232 ("arm64: syscall: Ensure saved x0 is kept in-sync with tracer updates") attempted to resolve a long-standing issue with syscall entry tracing, where a tracer is able to manipulate the first syscall argument without being subjected to seccomp or audit checking. Unfortunately, that fix was incomplete [1], as it failed to update 'orig_x0' between a tracer updating x0 during a seccomp ptrace exit (SECCOMP_RET_TRACE) and the seccomp filter being re-evaluated. Rather than add hooks to the core seccomp code, instead move the synchronisation code into the ptrace GPR and syscall setting code so that 'orig_x0' is kept up to date with x0 whenever we're stopped on the syscall entry path.	8.4	More Details
CVE-2026-92752	metasfresh DocumentAttachmentsRestController and CommentsRestController endpoints check only that callers are logged in without enforcing record-level permissions. Attackers can enumerate sequential document identifiers to read, replace, and delete attachments and comments on records their role cannot access.	8.3	More Details
CVE-2026-92467	zlt2000 microservices-platform through 6.0.0 contains an unverified password change vulnerability in the PUT /users/password endpoint that allows authenticated users to change any account password by omitting the current password check. Attackers can supply an arbitrary user id in the request body and a new password to overwrite credentials of any non-administrator account without verification.	8.3	More Details
CVE-2026-93371	A security vulnerability has been detected in marcopiovanello yt-dlp-web-ui up to v4. This issue affects the function NewGenericDownload of the file server/internal/downloaders/generic.go. Such manipulation of the argument params leads to command injection. It is possible to launch the attack remotely. The exploit has been disclosed publicly and may be used. The name of the patch is c7ad3bd79c7c520a7d17e7f2ba19d962be8e7897. A patch should be applied to remediate this issue.	8.3	More Details
CVE-2026-93453	SOGo before 5.12.11 constructs password-reset links using the client-supplied Origin header as the authority, allowing unauthenticated attackers to redirect recovery tokens to attacker-controlled domains. Attackers can submit password recovery requests with a malicious Origin header to have valid password-reset tokens mailed to victim recovery addresses within links pointing to attacker infrastructure, enabling account takeover.	8.3	More Details
CVE-2026-54597	ITFlow provides an IT documentation, ticketing and accounting system for small managed service providers. Prior to version 26.07, an authenticated user with module_support write permission and access to a credential record can perform time-based blind SQL injection through the expires parameter of the share_generate_link handler in agent/ajax.php. sanitizeInput applies string-context escaping, but expires is inserted unquoted into the item_expire_at MySQL INTERVAL expression, allowing a crafted expression and interval unit to execute conditional database queries whose results are inferred from response delays. This can expose password hashes, SMTP credentials, API keys, encrypted vault data, and database metadata and support administrative takeover after credential cracking. This issue is fixed in version 26.07.	8.3	More Details
CVE-2026-20323	A vulnerability in the sftunnel inter-device communication protocol of Cisco Secure FMC Software and Cisco Secure FTD Software could allow an unauthenticated, adjacent attacker to impersonate the peer device and obtain access at the level of the manager role, which is equivalent to root. This vulnerability is due to improper management of the TLS certificate for the sftunnel management connection. An attacker could exploit this vulnerability by connecting to the sftunnel port using a crafted TLS certificate. A successful exploit could allow the attacker to become a registered sftunnel peer with root access. Note: The attack is successful only if the sftunnel connection is down or the attack can disrupt the sftunnel connection long enough to execute the attack.	8.3	More Details
CVE-2026-85077	Sanic is an opensource python web server/framework. Prior to version 24.12.1, and in version 25.12.0, the HTTP/1.1 response pipeline in sanic/response/types.py serializes response header names and values without rejecting carriage-return or line-feed characters. Applications that place attacker-controlled data in response.headers, file(..., filename=...), or cookie path and domain attributes can therefore emit injected headers and may split responses. Depending on application and proxy behavior, this can enable session fixation through injected cookies, cache poisoning, or security-header corruption. This issue is fixed in versions 24.12.1 and 25.12.1.	8.2	More Details
CVE-2026-90332	In the Linux kernel, the following vulnerability has been resolved: PCI: dwc: ep: Flush cached MSI write before unmapping the iATU The MSI-X path already flushes any posted MSI-X write before tearing down its iATU mapping. That was added by commit c22533c66cca ("PCI: dwc: ep: Flush MSI-X write before unmapping its ATU entry") to make sure the write reaches the Root Complex before the outbound window that translates it disappears. The MSI path has the same problem but no equivalent flush. When the Endpoint driver caches an MSI target address and later observes that the Root Complex has changed it, dw_pcie_ep_raise_msi_irq() unmaps the existing iATU entry and reprograms it for the new address. Between the last MSI writel() and the unmap there may still be a posted write sitting in the fabric, and unmapping the iATU entry can drop or misroute that write. Fix this by reading back from the mapped MSI window before the unmap. The readback drains any posted MSI writes through the same iATU entry that mapped them, which is the same logic the MSI-X path uses. [mani: commit log]	8.2	More Details
CVE-2026-56795	Dell Server Update Utility, versions prior to 26.07.01, contains an Uncontrolled Search Path Element vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Code execution.	8.2	More Details
CVE-2026-92729	SigNoz versions 0.88.0 through 0.141.0 fail to apply authorization wrappers to trace-funnel analytics endpoints in the HTTP handler. Unauthenticated attackers can submit arbitrary funnel definitions to retrieve trace analytics including identifiers, durations, span counts, service topology, and error activity without credentials.	8.2	More Details
	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: check the data direction of a C2HData PDU nvme_tcp_handle_c2h_data() finds the request by command id and checks that it has a payload, but it does not check that the command asked for data to be read. A controller that answers a write command with C2HData therefore reaches nvme_tcp_recv_data(), where _copy_to_iter() hits WARN_ON_ONCE(i->data_source) and returns 0. The receive path turns that into -EFAULT and resets the controller. No data is copied, so this is not memory corruption. What a controller gets is a kernel warning it can raise at will, which is fatal on a host booted with panic_on_warn. The send path already knows the direction - it consults rq_data_dir() when it builds a command - and nvme_tcp_handle_r2t() checks the length and the offset of the request it names. The C2HData path does not check the direction at all. Reject a C2HData PDU whose command is not a read. Rejecting it		

CVE-2026-89973	fails the command and resets the controller, as the neighbouring check in this function does; what goes away is the warning. [6.885580] ------------[cut here]------------ [6.886457] WARNING: lib/iov_iter.c:193 at _copy_to_iter+0x289/0x1330, CPU#0: kworker/0:1H/71 [6.888137] CPU: 0 UID: 0 PID: 71 Comm: kworker/0:1H Not tainted 7.2.0-rc5-NVMETCP-gf5098b6bae76 #1 PREEMPT(lazy) [6.891165] Workqueue: nvme_tcp_wq nvme_tcp_io_work [6.891875] RIP: 0010:_copy_to_iter+0x289/0x1330 [6.903739] Call Trace: [6.904085] <TASK> [6.909254] __skb_datagram_iter+0x433/0x820 [6.911026] skb_copy_datagram_iter+0x37/0x120 [6.911622] nvme_tcp_recv_skb+0xa07/0x4320 [6.913378] __tcp_read_sock+0x1ab/0x810 [6.915788] nvme_tcp_try_recv+0x152/0x1e0 [6.918222] nvme_tcp_io_work+0x1e4/0x6c0 [6.926906] </TASK> [6.927226] ---[end trace 0000000000000000]--- [6.927878] nvme nvme0: queue 1 failed to copy request 0x71 data [6.928709] nvme nvme0: receive failed: -14	8.2	More Details
CVE-2026-93456	django-page-cms through 2.0.13 exempts five admin mutation views from CSRF protection in pages/admin/views.py, allowing attackers to forge requests that modify page content. Signed-in editors visiting a malicious page can be tricked into storing unescaped content that renders to all visitors, enabling stored cross-site scripting attacks.	8.2	More Details
CVE-2026-78252	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 15.3 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that, under certain conditions, an authenticated user could have induced a targeted user to perform unintended state-changing HTTP requests due to improper sanitization of user-controlled data in the Markdown JSON table renderer.	8.2	More Details
CVE-2026-63127	RMCP is an official Rust SDK for the Model Context Protocol. Prior to 2.0.0, the rmcp crate's OAuth implementation in crates/rmcp/src/transport/auth.rs omits the RFC 9728 resource field from ResourceServerMetadata and allows discover_oauth_server_via_resource_metadata to use protected-resource metadata without confirming that the returned resource identifier exactly matches the configured MCP server. A malicious MCP server can publish metadata for a different legitimate MCP resource and its authorization server, causing a victim who connects and completes the authorization flow to obtain a legitimate access token that the client subsequently sends to the malicious server. The attacker can capture the token and impersonate the victim against the legitimate MCP resource within the token's granted scopes. This issue is fixed in version 2.0.0.	8.2	More Details
CVE-2026-71180	Dell Update Package Framework, versions prior to 26.07.03, contains an Unchecked Return Value vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of privileges.	8.2	More Details
CVE-2026-73435	On affected platforms running Arista EOS with Open Shortest Path First version 2 (OSPFv2) configured, a specially crafted OSPFv2 packet from an unauthenticated attacker on the same broadcast segment, with OSPFv2 authentication configured can cause adjacency flapping and packet loss. The disruption can affect routing across the broader OSPF domain.	8.2	More Details
CVE-2026-93107	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Avoid reprocessing the current packet after the QP enters the error state When do_complete() finds the QP in the error state it returns RESPST_CHK_RESOURCE. Before commit 49dc9c1f0c7e ("RDMA/rxe: Cleanup reset state handling in rxe_resp.c") this was the flush loop: check_resource() had an error-state branch that fetched each remaining recv WQE and completed it with IB_WC_WR_FLUSH_ERR, without touching the current packet. That commit removed the error-state branch from check_resource() (draining is now done at rxe_receiver() entry) but kept the do_complete() error-state return. As a result, when a QP moves to the error state while a packet is being completed - e.g. an rdma_cm disconnect racing with receive processing - the responder state machine loops back into the request processing chain with the already-completed packet still in hand: check_resource() fetches a fresh recv WQE, execute()/send_data_in() copies the same packet payload again, do_complete() posts another IB_WC_SUCCESS CQE (qp->resp.status is still 0), and control returns to the error-state check. The loop re-executes the same packet once per posted recv WQE (observed: ~1000 duplicate IB_WC_SUCCESS completions of one SEND, one per ~8us, matching the RQ occupancy) until the RQ is exhausted, after which qp->resp.wqe is NULL and send_data_in() dereferences it: BUG: kernel NULL pointer dereference, address: 0000000000000014 Workqueue: rxe_wq do_work RIP: copy_data+0x29/0x1f0 Call Trace: send_data_in+0x25/0x50 rxe_receiver+0xf36/0x1dd0 The duplicate completions are indistinguishable from real receives to the ULP. During an rds stress test, the message was accepted as new and delivered the same datagram to user space hundreds of times, corrupting the stream; any ULP that relies on RC exactly-once delivery is affected. A live packet reaching the error-state check in do_complete() has been executed and completed exactly once and must be consumed, not re-processed. Return RESPST_CLEANUP for it (dequeue and free); keep returning RESPST_CHK_RESOURCE for the pkt == NULL case.	8.2	More Details
CVE-2026-54253	TS3 Manager is modern web interface for maintaining Teamspeak3 servers. Prior to 2.2.6, the /api/download handler in packages/server/routes/api.js passes the attacker-controlled port query parameter to socket.connect(port, host) and returns the resulting error.message through res.status(400).send(error.message) as text/html without a Content Security Policy. When a logged-in operator follows a crafted top-level link, the reflected value executes in the manager origin. The token cookie set in packages/ui/src/store/modules/query.js lacks HttpOnly, Secure, and an explicit SameSite attribute, allowing the script to read the token and call the autofillform event in packages/server/socket.js. autofillform returns the decoded JWT, including the cleartext ServerQuery password, enabling operator-session hijacking and control of the managed TeamSpeak server when the operator uses administrative ServerQuery credentials. A valid operator session and user interaction are required. This issue is fixed in 2.2.6.	8.2	More Details
CVE-2026-86039	libp2p is a JavaScript implementation of the libp2p networking stack. From 8.0.0 until 12.0.24, @libp2p/peer-store in packages/peer-store/src/index.ts uses consumePeerRecord to verify a RecordEnvelope signature but does not require PeerRecord.peerId in the signed payload to equal the signer peer ID derived by RecordEnvelope.openAndCertify. The expectedPeer option checks only the envelope signer, and the gossipsub Peer Exchange path can provide the attacker's own peer ID as expectedPeer. An attacker can therefore sign a record with the attacker's key, place a victim peer ID and attacker-controlled multiaddrs in the payload, and have certified addresses stored for the victim. The poisoned addresses can cause address-book corruption, dial redirection or failure, routing manipulation, and reachability disruption, although the connection upgrade still verifies remote peer identity and prevents a complete identity takeover. The issue is fixed in version 12.0.24.	8.2	More Details
CVE-2026-76413	A vulnerability in Cisco Adaptive Security Device Manager (ASDM) single sign-on (SSO) handler for Cisco Secure FMC Software could allow an unauthenticated, remote attacker to log in as the Cisco ASDM administrator user. This vulnerability is due to improper management of the Cisco ASDM SSO token. An attacker could exploit this vulnerability by performing session token forgery techniques. A successful exploit could allow the attacker to log in as the administrator user and, by repeating this action, keep legitimate administrators locked out of the ASDM indefinitely.	8.2	More Details

CVE-2026-90241	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Tear down scalable-mode context on probe failure intel_pasid_setup_sm_context() walks a PCI device's DMA aliases via pci_for_each_dma_alias() and programs a scalable-mode context entry for each RID. For a device with a dma_alias_mask, the callback is invoked once for the device's own RID and once for each alias bit, all with the same pci_dev, so device_pasid_table_setup() runs for multiple RIDs. pci_for_each_dma_alias() stops at the first callback error. Therefore, a failure partway through the walk can leave context entries for already processed RIDs present and still pointing to the device's PASID table. On this error path, intel_iommu_probe_device() currently jumps directly to intel_pasid_free_table(), which frees the PASID table without first tearing down those context entries. The IOMMU may then walk a present context entry whose PASID table pointer references freed memory. intel_iommu_release_device() already performs teardown before freeing the table. Apply the same ordering on the probe failure path. device_pasid_table_teardown() safely handles RIDs that were never programmed: iommu_context_addr() returns NULL when no context table has been allocated, and clearing the Present bit of an already non-present entry is a no-op. So unwind is safe for both the alias that failed and any aliases not yet reached.	8.2	More Details
CVE-2026-92566	DataGear through 6.0.0 contains a server-side request forgery vulnerability in the /dataSet/preview/Http endpoint that allows unauthenticated attackers to execute arbitrary HTTP requests by supplying a caller-controlled URI. Attackers can issue GET, POST, PUT, PATCH, or DELETE requests to internal endpoints and cloud metadata services, receiving full response bodies without authentication or validation.	8.2	More Details
CVE-2026-83946	Improper neutralization of input during web page generation ('cross-site scripting') in Azure Portal allows an unauthorized attacker to perform spoofing over a network.	8.2	More Details
CVE-2026-92903	Improper input validation in Snowflake CLI versions prior to 3.27.0 allowed unsanitized user-controlled values to be interpolated into SQL strings that are executed as multi-statement queries. An attacker who is able to supply a malicious project configuration file or craft command-line input can cause Snowflake CLI to execute attacker-controlled SQL statements in the context of the victim's Snowflake session and active role. Successful exploitation requires either write or pull-request access to a project repository whose CI/CD pipeline runs Snowflake CLI under an elevated service account role, or the ability to supply untrusted input to CLI-wrapping automation. Impact is limited by the privileges held by the configured Snowflake role at execution time. The fix is available in Snowflake CLI version 3.27.0, which also addresses several additional security findings. Users must manually upgrade.	8.2	More Details
CVE-2026-54354	MapServer is a system for developing web-based GIS applications. Prior to 8.6.4, MapServer's PostGIS runtime filter translation in src/mappostgis.cpp and msPostGISLayerTranslateFilter() treats a filteritem as numeric when CONNECTIONTYPE POSTGIS and metadata such as gml_<item>_type=Integer are configured, but it does not verify that attacker-controlled CGI qstring or OGC API Features featureId input is a numeric literal. The unquoted input is concatenated into the generated PostgreSQL/PostGIS predicate, allowing an unauthenticated remote attacker with access to an affected query endpoint to bypass predicates, enumerate unintended records, perform boolean-based or time-based SQL injection, and increase database load. The issue does not by itself establish database modification capabilities. This issue is fixed in version 8.6.4.	8.2	More Details
CVE-2026-89999	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: validate report length in wacom_intuos_pro2_bt_irq wacom_intuos_pro2_bt_irq() receives the wire report length in `len` but never consults it before parsing. After the report-id gate it unconditionally calls wacom_intuos_pro2_bt_pen() and then, selected by features.type, a fixed chain of sub-parsers, none of which receive `len`: wacom_intuos_pro2_bt_pen(wacom); if (type == INTUOSP2_BT || type == INTUOSP2S_BT) { wacom_intuos_pro2_bt_touch(wacom); wacom_intuos_pro2_bt_pad(wacom); wacom_intuos_pro2_bt_battery(wacom); } else { wacom_intuos_gen3_bt_pad(wacom); wacom_intuos_gen3_bt_battery(wacom); } Each sub-parser dereferences wacom->data at fixed offsets. The furthest byte touched on each branch is: INTUOSP2_BT / INTUOSP2S_BT: wacom_intuos_pro2_bt_pad() reads data[285] (the touchring byte), so the report must be at least 286 bytes; INTUOSHT3_BT ("gen3"): wacom_intuos_gen3_bt_battery() reads data[45], so the report must be at least 46 bytes. features.type is selected from the VID/PID id_table entry and wacom_setup_device_quirks() force-registers the pen/pad/touch inputs for that type independent of the report descriptor, so a malicious or malfunctioning paired/spoofed Bluetooth peripheral can advertise that VID/PID and send an undersized report that still satisfies the data[0] == 0x80/0x81 gate. The driver then reads past the received report and forwards the bytes to userspace via evdev (MSC_SERIAL / ABS_MISC / ABS_WHEEL on the pen and pad input nodes), an out-of-bounds read with a concrete userspace read-back channel, and a true out-of-bounds read on transports whose backing buffer is sized to the (small) report descriptor rather than a fixed-size staging buffer. This is the same class of bug commit 2f1763f62909 ("HID: wacom: fix out-of-bounds read in wacom_intuos_bt_irq") already hardened in the sibling wacom_intuos_bt_irq(), which guards each report id against its minimum length before parsing. Guard wacom_intuos_pro2_bt_irq() the same way: before parsing, reject reports shorter than the furthest offset the selected branch actually dereferences, warn, and bail out. Because the whole pen/touch/pad/ battery chain runs unconditionally per branch, a single up-front check against the maximum offset (286 bytes for INTUOSP2_BT/INTUOSP2S_BT, 46 bytes for the gen3 branch) bounds every sub-parser. Returning 0 on a short report also skips those calls for the same malformed report, which is the safe, conservative behavior.	8.1	More Details
CVE-2026-89848	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Quiesce response IRQ before freeing request queue qla2xxx_delete_qpair() deletes the request queue before the response queue. qla25xx_delete_req_que() frees the request queue memory (kfree(req) in qla25xx_free_req_que()), but the response-queue MSI-X is only released later, in qla25xx_free_rsp_que(). In that window the response interrupt can still fire, qla2xxx_msix_rsp_q() queues qpair->q_work, and qla_do_work() -> qla24xx_process_response_queue() dereferences the now-freed rsp->req (LOGINOUT/CT/ELS entries and the status path), a use-after-free. The cancel_work_sync() added for the qpair teardown lives in the response free path, which runs after the request queue is already freed, so it does not protect rsp->req. Release the response-queue interrupt and flush qpair->q_work before deleting the request queue, so no late completion can reach the freed request queue. Clearing have_irq makes the subsequent qla25xx_free_rsp_que() skip its free_irq(), and the firmware queue-delete order (request then response) is preserved; the request-delete mailbox completes on the default vector and is unaffected by dropping the qpair response interrupt early.	8.1	More Details
CVE-2026-92604	Scirius through 3.8.0 contains an arbitrary file write vulnerability in the PCAP filestore upload endpoint that allows default User role users to write attacker-controlled JSON content to filesystem paths. Attackers can supply path traversal sequences in the uploaded document's _id field to escape the intended directory and write files with .json extension to arbitrary locations as root.	8.1	More Details
	NetLicensing MCP Server is a natural-language interface that enables agentic applications to manage the software-licensing lifecycle in Labs64 NetLicensing. Prior to 0.1.6, network-reachable HTTP transport requests to /mcp that omit x-netlicensing-api-		

CVE-2026-54446	key, Authorization: Bearer, and the apikey query parameter pass through ApiKeyMiddleware in src/netlicensing_mcp/server.py without authentication. The downstream api_key_ctx in src/netlicensing_mcp/client.py then falls back to the operator's NETLICENSING_API_KEY and authenticates upstream NetLicensing REST API calls under the operator account. An unauthenticated attacker can invoke MCP tools to enumerate products, licenses, licensees, and transactions, create or modify licensing objects, perform validations, and execute destructive delete operations. The issue affects HTTP deployments configured with a server-side key and does not require user interaction. This issue is fixed in version 0.1.6.	8.1	More Details
CVE-2026-90153	In the Linux kernel, the following vulnerability has been resolved: ksmbd: bound smb_check_perm_dacl() ACE walks by DACL size smb_check_perm_dacl() validates that the DACL fits inside the NT security descriptor, but then bounds its two ACE walks by the remaining NTSD length (acl_size) rather than the DACL's declared size (pdacl_size). When pdacl->size is smaller than the trailing NTSD buffer, bytes after the declared DACL boundary - still inside the stored security descriptor - are parsed as ACEs during access checks. A crafted DACL can place an access-granting ACE beyond pdacl->size, and the current code accepts it during SMB2_CREATE access validation, while parse_dacl() and smb_inherit_dacl() stop at pdacl_size. Bound both ACE walks by pdacl_size to match the DACL boundary semantics used elsewhere in the server. Validation: - semantic KUnit harness shows the post-boundary ACE is selected before the fix and rejected (EACCES) after it - linux master (7.2-rc6), x86_64	8.1	More Details
CVE-2026-90176	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Do not skip lock checks for single-byte ranges check_lock_range() uses inclusive ranges. Its callers pass the end offset as start + length - 1, so start == end represents a valid single-byte range rather than an empty range. The start == end shortcut therefore skips mandatory byte-range lock checks for one-byte reads, writes, copychunk operations and one-byte truncate ranges. A conflicting lock covering that byte is not checked and the operation is allowed to proceed. Remove the shortcut. The truncate size == inode->i_size case is already handled by only calling check_lock_range() when the new size differs from the current file size.	8.1	More Details
CVE-2026-87976	Apache NiFi Registry 0.4.0 through 2.11.0 are subject to path manipulation when storing extension bundle content using group, artifact, and version coordinates from uploaded NAR manifests. The default file persistence provider used coordinates as filesystem path components without rejected parent-directory names, and the path-containment check compared an unnormalized resolved path. An authenticated user authorized to write and delete bundles in a bucket can upload a NAR with a crafted manifest resulting in file system operations outside of the file persistence directory. Upgrading to Apache NiFi Registry 2.12.0 is the recommended mitigation, which rejects parent-directory coordinates and requires a normalized path to remain a strict child of the storage root location.	8.1	More Details
CVE-2026-54596	ITFlow provides an IT documentation, ticketing and accounting system for small managed service providers. Prior to version 26.07, an authenticated Technician or higher with access to at least one client invoice can inject SQL through the frequency parameter handled by agent/post/recurring_invoice.php. The handler passes recurring_invoice_frequency through sanitizeInput but interpolates it unquoted into DATE_ADD, allowing SQL syntax to escape the interval expression, assign additional INSERT columns, store subquery results in recurring_invoice_note, and expose those results through agent/recurring_invoice.php. The persisted recurring_invoice_frequency can execute again when Force Recurring uses it in a later UPDATE, allowing another legitimate user to trigger the second-order injection. This can expose password hashes, SMTP credentials, user records, and database metadata, modify database fields, and enable administrative takeover after credential cracking. This issue is fixed in version 26.07.	8.1	More Details
CVE-2026-20335	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Secure Adaptive Security Appliance Software, Cisco Secure Firewall Threat Defense Software and Cisco Secure Firewall Management Center Software engineering team has conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20335 are related to incorrect calculation issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-682.	8.1	More Details
CVE-2026-92943	Improper validation of certificate with host mismatch in the MQTT client TLS connection layer in AWS IoT Device SDK for Python 1.5.3 through 1.6.0 on Python 3.7 and later might allow an adversary-in-the-middle actor to impersonate the AWS IoT Core endpoint, read device telemetry, and inject arbitrary MQTT messages that the device processes as authentic, via a certificate issued for an unrelated hostname by a certificate authority present in the device trust store. To remediate this issue, users should upgrade to version 1.6.1.	8.1	More Details
CVE-2026-89861	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Hold vport reference in qla24xx_report_id_acquisition() In the format 1 path, the virtual port is located on ha->vp_list while holding vport_slock, but the lock is dropped before vp is used: qla_update_host_map() is called and VP_IDX_ACQUIRED/REGISTER_FC4_NEEDED/ REGISTER_FDMI_NEEDED are set on vp. No reference is taken across that window, so a concurrent qla24xx_deallocate_vp_id() can tear the vport down and free it, leading to a use-after-free. Take a vport reference (vref_count) under vport_slock when the matching vp is found, and drop it after the last use of vp. qla24xx_deallocate_vp_id() waits for vref_count to reach zero before unlinking and freeing the vport, so the pointer stays valid. This matches the reference idiom already used by the other ha->vp_list traversals.	8.1	More Details
CVE-2026-74909	Keycloak provides a policy enforcer to protect applications by matching incoming web requests against defined security policies. A flaw was found where the enforcer fails to correctly normalize web addresses that contain special encoded characters, such as those representing semicolons or directory traversal segments. An authenticated user can use these encoded characters to trick the enforcer into applying a less restrictive security policy than intended, potentially gaining unauthorized access to sensitive administrative or private application endpoints.	8.1	More Details
CVE-2026-92793	GoAdmin through 1.2.26 fails to properly anchor the logout pattern when checking permissions, allowing authenticated users to bypass permission checks by appending a query parameter. Attackers can append a query string containing the admin prefix followed by /logout to reach administrative endpoints and perform unauthorized actions including reading sensitive data and modifying application state.	8.1	More Details
CVE-2026-81442	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Privilege Management vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Information tampering and Unauthorized access.	8.1	More Details
CVE-2026-92776	Wiki.js through 2.5.314 fails to require path separators when matching START and END page rules, allowing attackers to access pages sharing a prefix with authorized folders. Users granted access to a folder can read and modify unrelated pages with matching prefixes, bypassing intended access controls.	8.1	More Details

CVE-2026-92914	AVideo LoginControl contains an authentication bypass vulnerability in the PGP second factor verification that compares challenge responses using loose equality against an uninitialized session variable. Attackers with a victim's password can bypass the second factor by sending a parameter-less GET request to verifyChallenge.json.php, which evaluates null == null and marks authentication complete.	8.1	More Details
CVE-2026-92782	Chroma through 1.5.9 fails to validate tenant and database segments when resolving collections, allowing authenticated attackers to access collections from other tenants by knowing the collection identifier. Attackers can read, modify, and update records in foreign collections by issuing requests under their own tenant path, bypassing authorization checks.	8.1	More Details
CVE-2026-61593	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, the SSE client→server POST endpoints are `@csrf_exempt` and the SSE GET stream endpoint had no Origin check, so a cross-origin page could drive a victim-cookie-authenticated SSE session: force the victim's browser to GET the stream URL (which creates and mounts a LiveView as the victim) and POST to the message endpoint with `credentials: include` to fire state-changing event handlers as the victim. The URL `session_id` is client-chosen (validated only for UUID *format*), so it is not a CSRF token, and a JSON body sent as `text/plain` is a CORS simple request with no preflight. The issue is fixed in 1.0.7. All three SSE endpoints validate the request `Origin` against `ALLOWED_HOSTS` (mirroring the WebSocket CSWSH defense) and reject cross-origin requests with 403; the POST endpoints additionally require `Content-Type: application/json` (415 otherwise), closing the `text/plain` simple-request bypass. As a workaround, disable the SSE transport, or front it with a proxy that enforces an Origin allowlist.	8.1	More Details
CVE-2026-92783	Yeti through 2.11.0 fails to validate caller permissions in the DELETE /api/v2/rbac/{id} endpoint, allowing users with read access to delete access control relationships. Attackers can revoke the owner's grant and permanently lock legitimate owners out of objects.	8.1	More Details
CVE-2026-92785	Angel through 3.3.0 deserializes untrusted setAlgoMetrics payload using Kryo without class registration or allowlist validation. Unauthenticated network attackers can instantiate arbitrary classes or exhaust coordinator memory by sending crafted serialized objects to the master RPC endpoint.	8.1	More Details
CVE-2026-81478	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Use of Hard-coded Cryptographic Key vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Unauthorized access.	8.1	More Details
CVE-2026-81476	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Remote execution.	8.1	More Details
CVE-2025-43936	Dell ObjectScale, versions prior to ObjectScale 4.4.0.0, contains an Improper Authentication vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Unauthorized access.	8.1	More Details
CVE-2026-81475	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Missing Authentication for Critical Function vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Remote execution.	8.1	More Details
CVE-2026-26950	Dell SmartFabric Manager, versions prior to 2.2.1, contains an Insufficient Verification of Data Authenticity vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Elevation of privileges.	8.1	More Details
CVE-2026-92751	CMAK through 3.0.0.6 fails to install a cross-site request forgery filter, allowing attackers to perform state-changing actions on behalf of authenticated operators. Attackers can craft hidden forms that submit to destructive endpoints like topic deletion and cluster configuration changes, leveraging the operator's HTTP Basic authentication credentials or play-basic-authentication cookie without SameSite protection.	8.1	More Details
CVE-2026-92806	phpList versions before 3.6.17 fail to validate cross-site request forgery tokens in the mass subscriber removal form handler. Attackers can induce logged-in administrators to visit crafted pages that silently delete and blacklist arbitrary subscriber addresses without authentication verification.	8.1	More Details
CVE-2026-73454	On affected platforms running Arista EOS with gRPC Network Security Interface (gNSI) Credentialz configured, a specially crafted request can cause unintended modifications to the target account's properties. This may result in the account being assigned elevated privileges or access beyond what an administrator intended.	8.1	More Details
CVE-2026-85530	The GiveWP WordPress plugin before 4.16.8.1 does not consistently normalise a donor's e-mail address between the value it stores and the value it later uses to look that donor up, allowing unauthenticated users to be resolved as an arbitrary donor and to set the WordPress password of any user account linked to one, including an administrator's.	8.1	More Details
CVE-2026-92087	@fastify/auth is a Fastify plugin that composes multiple authentication and authorization strategies into a single route guard. In versions 5.0.0 through 5.1.0, when strategies are composed with the relation "or" option together with the run "all" option and one entry is a nested array acting as an AND group, the group is evaluated in an order-dependent way: an earlier failing check is silently dropped and the group's result becomes the outcome of its last check. As a result, a request that satisfies only the last member of an AND group, for example an attacker who holds a valid API key but is not an administrator, is authorized instead of rejected, and a related order-dependent bypass affects the mirror configuration where the top-level relation is "and" and a nested group uses "or". The issue is fixed in @fastify/auth 5.1.1, and users should upgrade to 5.1.1 or later. As a workaround, omit the run "all" option where it is not required, order each AND group so its stricter check is evaluated last, or replace nested AND groups with an explicit top-level "and" composition.	8.1	More Details
CVE-2026-61591	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, for views that opt into state snapshots, the snapshot `state_json` embedded in the client page was restored on reconnect as trusted view state with no integrity check. A client could edit the unsigned `state_json` in their page and return it in the reconnect mount frame to inject arbitrary view attributes — e.g. flip `is_admin` to `True`, or change `account_id` / `balance` — escalating privilege or tampering with business state held in public view attributes (the normal djust pattern). This issue is fixed in djust 1.0.7. State snapshots are signed; unsigned or forged snapshots are rejected on the back-navigation restore path. As a workaround, do not enable state snapshots; do not hold authorization/ownership state in public view	8.1	More Details

	attributes.		
CVE-2026-87935	The Paid Downloads plugin for WordPress is vulnerable to Arbitrary File Upload in all versions up to, and including, 3.15 via the admin_request_handler function. This is due to missing authorization and file type validation in the admin_request_handler function, which is reachable unauthenticated via is_admin() returning true for /wp-admin/admin-post.php. This makes it possible for unauthenticated attackers to upload files that may be executable, which makes remote code execution possible. On Apache servers where AllowOverride is enabled, an .htaccess file placed in the upload directory may block direct HTTP retrieval of uploaded files, limiting exploitability to stacks that do not honor .htaccess directives such as nginx, LiteSpeed, and Apache with AllowOverride None.	8.1	More Details
CVE-2026-92578	WWBN AVideo through 29.0 contains an authentication bypass vulnerability where the stored password hash is accepted as a valid login credential through two independent code paths in loginFromRequest() and encryptPasswordVerify(). Attackers who obtain the stored users.password hash value can authenticate as any user by submitting the hash directly to login endpoints, completely bypassing password verification.	8.1	More Details
CVE-2026-76834	b2evolution CMS versions 6.7.8 through 7.2.5 contain an incomplete fix for CVE-2016-8901 where the serialized-array object check in param_check_serialized_array() fails to reject payloads with negative integer array keys. Unauthenticated attackers can submit crafted serialized PHP objects via POST requests to htsrv/call_plugin.php that bypass validation and reach unserialize(), instantiating arbitrary PHP objects with attacker-chosen properties that may enable code execution if suitable POP gadget chains exist.	8.1	More Details
CVE-2026-27552	A low-privileged remote attacker can exploit improper authorization in the /index.php/attached_devices_tab/do_upload endpoint to upload IODD files to the device, potentially altering device behavior or causing system crashes.	8.1	More Details
CVE-2026-92984	HUBzero CMS through 2.2.32 accepts session identifiers from query strings and request variables instead of cookies alone, allowing unauthenticated attackers to fixate victim sessions. Attackers can obtain a valid session identifier, send victims a crafted link containing it, and replay the identifier after the victim authenticates to hijack their account and access.	8.1	More Details
CVE-2026-92763	Rundeck through 6.2.1 fails to properly authorize the importConfig and importNodesSources parameters in the project archive import endpoint. Attackers with only the import action can replace project configuration files including security-relevant settings like node executors and SSH key paths that affect job execution.	8.1	More Details
CVE-2026-90243	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Clear Present bit before tearing down copied context entry copied_context_tear_down() zeroes the 128-bit context entry with context_clear_entry() while the Present bit is still set, and only then issues the context-cache and IOTLB invalidations. This leaves a window in which hardware can fetch a torn entry, with some fields already zeroed while Present is still set, leading to unpredictable behaviour or spurious faults. While x86 provides strong write ordering, the compiler may reorder the writes to the two 64-bit halves of the entry, and the hardware fetch is not guaranteed to be atomic with respect to multiple CPU writes. There is no cacheline flush before the invalidation either, so on an IOMMU without coherent access to the context table the zeroed entry may not be visible to hardware at the point the invalidation is submitted. Apply the same ownership handshake described in the VT-d spec, Section 6.5.3.3 ("Guidance to Software for Invalidations"): clear only the Present bit, flush it out to the IOMMU, perform the invalidations, and only then zero the remainder of the entry.	8.1	More Details
CVE-2026-90268	In the Linux kernel, the following vulnerability has been resolved: scsi: sd: Fix error handling in sd_probe() after large pool creation failure After device_add(&sdkp->disk_dev) succeeds, sd_large_pool_create() failure must unregister disk_dev and let scsi_disk_release() free sdkp. Going through out_free_index kfree()s an already registered device and leaks the sysfs entry.	8.1	More Details
CVE-2026-89413	The Filter Gallery plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 1.1.4. This is due to the plugin not properly verifying that a user is authorized to perform an action. This makes it possible for authenticated attackers, with subscriber-level access and above, to delete any arbitrary Filter Gallery records — including all associated filters, image mappings, settings, and details options — by supplying attacker-controlled gallery IDs. The nonce bypass requires omitting the nonce POST field entirely rather than submitting an invalid value, as a present-but-invalid nonce is correctly rejected.	8.1	More Details
CVE-2026-92749	SafeLine through 9.4.1 derives the management console session-signing secret from a time-seeded math/rand generator, allowing attackers to reconstruct the key offline. Unauthenticated remote attackers who can bound the install timestamp can regenerate the secret and forge valid administrator session cookies to gain control of protected sites.	8.1	More Details
CVE-2026-93393	A heap-based buffer overflow exists in the TLS transport layer of the MongoDB C Driver when built with the Windows platform TLS backend. A remote endpoint that the client connects to can cause the driver to write uncontrolled data outside the bounds of a heap allocation while processing incoming encrypted traffic after the TLS handshake completes. No authentication or user interaction is required, because the affected processing occurs before any application-level authentication completes. Triggering this issue may lead to memory corruption in the client process, disclosure of adjacent heap memory, or termination of the process.	8.1	More Details
CVE-2026-86466	Apache Airflow FAB provider: the Authentik OAuth path in the FAB auth manager does not validate the issuer or audience claims of the id_token it accepts. An attacker holding a token that the same Authentik identity provider minted for a different client application can present it to Airflow and be authenticated as the user it names, because the audience claim is never checked. Affects deployments using the FAB auth manager with Authentik OAuth where the same Authentik instance also serves other applications; the attacker needs a valid token for any of those other applications, not for Airflow. CVE-2026-75156 corrected the same missing validation on the Azure AD path in this file; the Authentik path was left unchanged and is fixed here. Deployments that applied the CVE-2026-75156 fix and use Authentik must also upgrade for this one. Users of apache-airflow-providers-fab are recommended to upgrade to version 3.9.0 or later, which fixes the issue.	8.1	More Details
CVE-2026-54520	AI Agent Automation is a modular AI agent workflow automation platform with schedulers, tools, and observability. Prior to 0.9.1, the executeStep file-step implementation in backend/src/agents/executor.js passes the user-controlled step.path value through path.resolve with process.cwd() and then uses the resulting path for read or write operations without checking that it remains in an approved workflow directory. An authenticated user who can create or modify workflow file steps can supply traversal segments to escape the intended workspace and read sensitive files or write and overwrite files accessible to the backend process, including application-adjacent files when process permissions allow. This issue is fixed in version 0.9.1.	8.1	More Details

CVE-2026-93375	Incorrect reference resolution in Tracing in Google Chrome on on Windows prior to 153.0.8010.52 allowed a local attacker to potentially execute arbitrary code outside the sandbox via a local program. (Chromium security severity: High)	8.1	More Details
CVE-2026-90301	In the Linux kernel, the following vulnerability has been resolved: ocfs2: o2hb: quiesce negotiate handlers and timeout work Heartbeat regions publish struct o2hb_region as the private data for the NEGO_TIMEOUT and NEGO_APPROVE o2net handlers as soon as make_item() creates the configfs region. The approve handler can call o2hb_arm_timeout(), so a peer can touch the region timeout work before dev_store() has finished building the heartbeat runtime, or after teardown has started to shut that runtime back down. The final configfs put also has to keep reg alive until the last in-flight o2net callback drops its handler reference. o2net_unregister_handler_list() blocks future handler lookups, but it does not wait for sc_rx_work that already passed o2net_handler_get(). That drain needs to cover local listener teardown as well, where the o2net ordered workqueue may already be inside destroy_workqueue(). Fix the lifetime rule in both directions. Initialize the region delayed works before publishing reg through the o2net handler table, keep new or stopping regions non-armable with hr_stopping, and quiesce both delayed works on failed-start and teardown paths even when no heartbeat thread is left to call o2hb_disarm_timeout(). Then unregister handlers before tearing down handler-visible region state and make the drain wait for the active or destroying o2net ordered workqueue before release frees reg. The buggy scenario involves two paths, with each column showing the order within that path: region lifecycle: late negotiate callback: 1. make_item() registers the 1. o2net_process_message() gets a region handlers before heartbeat handler for reg. dev_store() has built a 2. The callback runs after the lookup runnable heartbeat context. lock is dropped and dereferences reg. 2. A failed start or rmdir 3. An approve or timeout path tries to stops the heartbeat thread, queue reg's delayed work, or release quiesces existing work, and races the callback body after handler drops the final configfs ref. unregister. 3. region_release() must drain 4. The callback or delayed work can handler-visible o2net rx work outlive reg unless lifecycle code before freeing reg. keeps the region non-armable and drains the active-or-destroying o2net workqueue. Validation reproduced this kernel report: KASAN slab-use-after-free in __run_timers+0x22c/0x5b0 Write of size 8 Call trace: dump_stack_lvl+0x66/0xa0 print_report+0xce/0x630 __run_timers+0x22c/0x5b0 kasan_report+0xe0/0x110 _raw_spin_unlock_irqrestore+0x27/0x60 try_to_wake_up+0x191/0xf70 timer_expire_remote+0xae/0xf0 run_timer_softirq+0x19b/0x1a0 handle_softirqs+0x156/0x660 __irq_exit_rcu+0xc4/0x160 irq_exit_rcu+0xe/0x20 sysvec_apic_timer_interrupt+0x6c/0x80 asm_sysvec_apic_timer_interrupt+0x1a/0x20 Allocated by task stack: kasan_save_stack+0x33/0x60 kasan_save_track+0x14/0x30 __kasan_kmalloc+0xaa/0xb0 o2hb_heartbeat_group_make_item+0x3c/0x600	8.1	More Details
CVE-2026-92919	admin3 through 3.0.0 fails to sanitize client-supplied filenames in the upload handler, allowing authenticated users to write files outside the storage root on Windows deployments. Attackers can use dot-dot path segments in filenames to escape the configured storage directory and overwrite arbitrary files accessible to the server process.	8.1	More Details
CVE-2026-63671	MDC is a tool to take regular Markdown and write documents interacting deeply with a Vue component. Prior to 0.22.1, @nuxtjs/mdc uses parseMarkdown with allowDangerousHtml enabled by default and relies on validateProps, validateProp, and unsafeLinkPrefix to remove executable URLs from untrusted Markdown. validateProp checks only attributes named href or src, allowing an SVG xlink:href value represented as xLinkHref to retain a javascript: URL that executes in the page origin when selected. The data:text/html denylist entries are also compared against url.protocol, which is only data:, so an iframe src containing data:text/html survives sanitization and executes in an opaque origin when loaded. Plain href javascript: URLs, srcdoc, object, script, and base elements are already blocked, making these two paths specific sibling gaps in the sanitizer. This issue is fixed in version 0.22.1.	8.1	More Details
CVE-2026-92469	zlt2000 microservices-platform through 6.0.0 contains an authorization bypass vulnerability in the file-center module DELETE /files/{id} endpoint that performs no ownership validation. Authenticated attackers can enumerate file identifiers via GET /files and delete arbitrary users' files and metadata by supplying their identifiers to the delete endpoint.	8.1	More Details
CVE-2026-90091	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: fix race l2cap_sock_cleanup_listen() vs. put_chan For L2CAP sockets without owning sk->sk_socket, reading l2cap_pi(sk)->chan may race against concurrent l2cap_sock_kill() -> l2cap_sock_put_chan(). This excludes simultaneous proto_ops callbacks, but access in l2cap_sock_cleanup_listen() has unsafe lockless read. [Task 1] [Task 2 (hdev->workqueue)] l2cap_sock_release(parent) l2cap_disconn_cfm l2cap_sock_cleanup_listen l2cap_conn_del bt_accept_dequeue l2cap_chan_del lock_sock(sk) l2cap_sock_teardown_cb bt_accept_unlink bt_sk(sk)->parent = NULL release_sock(sk) ----------> lock_sock(sk) parent = /* NULL */ lock_sock(sk) <---------- release_sock(sk) sock_set_flag(sk, SOCK_ZAPPED) l2cap_sock_close_cb l2cap_sock_kill(sk) l2cap_sock_put_chan chan = READ l2cap_pi(sk)->chan l2cap_pi(sk)->chan = NULL l2cap_chan_hold_unless_zero l2cap_put_chan(chan) kref_get_unless_zero(&chan->ref) Task 1 may observe NULL which causes null-ptr-deref. Fix the race by taking lock_sock() in l2cap_sock_kill() to synchronize with l2cap_sock_cleanup_listen(). hold_unless_zero() is not needed here, l2cap_pi(sk)->chan owns reference if it is non-NULL. Clarify code comments vs. locking.	8.0	More Details
CVE-2026-92136	Jenkins OWASP Dependency-Check Plugin 5.6.4 and earlier does not escape CWE values from Dependency-Check reports on the Jenkins UI, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers with Item/Configure permission.	8.0	More Details
CVE-2026-85469	A flaw was found in quay-builder-qemu. A remote attacker could exploit this by compromising the upstream `Noelware/docker-manifest-action` used in the release workflow, which is pinned to a mutable branch. This allows the attacker to inject arbitrary code, leading to the exfiltration of sensitive registry credentials or the publication of malicious images. The workflow also exposes the default GitHub token, increasing the severity of the compromise.	8.0	More Details
CVE-2026-92135	Jenkins Coverage Plugin 3.3358.v9487dde48783 and earlier does not validate the coverage results ID when a job configuration is submitted through the REST API, allowing attackers with Item/Configure permission to use a javascript: scheme URL as identifier, resulting in a stored cross-site scripting (XSS) vulnerability.	8.0	More Details
CVE-2026-92134	Jenkins Warnings Plugin 13.10258.va_17d49a_78c3b_ and earlier does not validate the analysis results ID when a job configuration is submitted through the REST API, allowing attackers with Item/Configure permission to use a javascript: scheme URL as identifier, resulting in a stored cross-site scripting (XSS) vulnerability.	8.0	More Details
CVE-2026-92127	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier automatically approves the classpath entries in an item configuration when a user with Overall/Administer permission copies the item, or updates that configuration through the REST API or CLI, allowing attackers able to define classpath entries to execute arbitrary code in the context of the Jenkins controller JVM.	8.0	More Details

CVE-2026-89954	In the Linux kernel, the following vulnerability has been resolved: mtd: afs: validate v2 image info bounds The AFS v2 parser uses footer[8] to locate the image information block inside the current erase block, then uses the image information region_count to walk entries from a fixed local array. The footer offset and region count come from flash contents and are not checked against the erase block or the local image-info array before use. Reject v2 entries whose image information offset would underflow the erase block calculation, and reject region counts that cannot fit in the local image-info array before walking region entries.	8.0	More Details
CVE-2026-73461	On affected EOS platforms with AAA-based gRPC authorization enabled for OpenConfig, gRPC requests of an authenticated user to OpenConfig may use the wrong privilege level, resulting in an authorization using the wrong AAA method list. This does not impact non-gRPC OpenConfig requests such as NETCONF.	8.0	More Details
CVE-2026-90092	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: reject accept queue add unless BT_LISTEN New sk should not be added to parent socket accept queue after last l2cap_sock_cleanup_listen() has run in l2cap_sock_teardown_cb() and state set to BT_CLOSED, as that can result to UAF on dereferencing the dangling parent reference. l2cap_sock_new_connection_cb() may race with parent l2cap_chan teardown, due to chan->state accessed without consistent locking: [Task 1] [Task 2] l2cap_sock_release(parent) l2cap_connect l2cap_sock_shutdown pchan = l2cap_global_chan_by_psm l2cap_chan_lock(pchan) l2cap_chan_close l2cap_sock_teardown_cb pchan->state = BT_CLOSED l2cap_chan_unlock(pchan) ------> l2cap_chan_lock(pchan) l2cap_new_connection l2cap_sock_new_connection_cb l2cap_chan_lock(pchan) <-------- l2cap_chan_unlock(pchan) l2cap_sock_kill(parent) /* bt_sk(sk)->parent dangling */ Fix by adding check for sk_state == BT_LISTEN after acquiring sk lock in l2cap_sock_new_connection_cb(). Add lock_sock() around sk_state writes where missing, to avoid data races. Although the data races on pchan->state should be fixed too, this defensive sk_state check probably makes sense in any case.	8.0	More Details
CVE-2026-78428	For users authenticated through SAML or OpenID Connect (OIDC), this vulnerability can result in one user receiving another user's authenticated session when multiple SSO login attempts occur concurrently	8.0	More Details
CVE-2026-86108	Insufficient validation of inputs supplied through affected VeloCloud Edge management and configuration workflows may allow an authorized management request or configuration value to be interpreted as an operating-system command. Successful exploitation may allow command execution with elevated privileges on the affected VeloCloud Edge.	8.0	More Details
CVE-2026-89947	In the Linux kernel, the following vulnerability has been resolved: clk: meson: align gxbb_32k_clk_sel number of parents with actual count The following out-of-bounds read has been observed by Christian on a GXBB WeTek Hub: == BUG: KASAN: global-out-of-bounds in __clk_register+0x1b70/0x2418 Read of size 8 at addr ffffd66320cf88e0 by task swapper/0/1 CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 7.0.0-rc5 #1 PREEMPT Hardware name: WeTek Hub (DT) Call trace: show_stack+0x14/0x20 (C) dump_stack_lvl+0x74/0x94 print_report+0x164/0x4b0 kasan_report+0x98/0xd8 __asan_report_load8_noabort+0x1c/0x24 __clk_register+0x1b70/0x2418 devm_clk_hw_register+0x74/0x15c meson_clkc_init+0xd4/0x20c meson_clkc_syscon_probe+0x5c/0x94 platform_probe+0xbc/0x17c really_probe+0x184/0x844 __driver_probe_device+0x154/0x35c driver_probe_device+0x60/0x188 __driver_attach+0x168/0x4a0 bus_for_each_dev+0xec/0x180 driver_attach+0x38/0x58 bus_add_driver+0x238/0x4c0 driver_register+0x150/0x388 __platform_driver_register+0x54/0x7c gxbb_clkc_driver_init+0x18/0x20 do_one_initcall+0xb8/0x340 kernel_init_freeable+0x49c/0x52c kernel_init+0x24/0x148 ret_from_fork+0x10/0x20 The buggy address belongs to the variable: gxbb_32k_clk_parents+0x60/0x400 The buggy address belongs to a vmalloc virtual mapping The buggy address belongs to the physical page: Memory state around the buggy address: ffffd66320cf8780: 00 00 00 00 f9 f9 f9 f9 00 f9 f9 f9 f9 f9 f9 f9 ffffd66320cf8800: 00 04 f9 f9 f9 f9 f9 00 04 f9 f9 f9 f9 f9 f9 >ffffd66320cf8880: 00 00 00 00 00 00 00 00 00 00 00 00 f9 f9 f9 f9 ^ ffffd66320cf8900: 00 01 f9 f9 f9 f9 f9 00 06 f9 f9 f9 f9 f9 f9 ffffd66320cf8980: 00 00 02 f9 f9 f9 f9 f9 00 00 02 f9 f9 f9 f9 f9 == Commit 7915d7d5407c ("clk: amlogic: gxbb: drop non existing 32k clock parent") dropped a non-existing clock parent from the gxbb_32k_clk_sel mux but didn't adjust the hard-coded num_parents field. Fix the actual number of parents of that mux by using ARRAY_SIZE instead (avoiding similar problems in future).	8.0	More Details
CVE-2026-89911	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Correctly cap TLBI Range to the architural limit TLB Invalidation by Range has a fairly powerful way of encoding pretty large ranges in a small number of bits. This range can be based on an arbitrary VA, which means it is pretty easy for a guest to generate an overflow should the hypervisor be naive enough to add the range to the base... Make sure the range is capped to the limit dictated by the address bit that determines the VA range. For an IPA invalidation, this is further corrected down the line to ignore the upper range.	7.9	More Details
CVE-2026-25282	Transient DOS when processing unverified data from a neighboring system causes out of bound memory access.	7.9	More Details
CVE-2026-93111	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark tracing_multi trampolines as ftrace managed Since tracing_multi link does not set ftrace_managed, it would fail to release the tracing_multi link when attaching tracing_multi link and then attaching fentry link. [3.714215] WARNING: kernel/bpf/trampoline.c:1727 at bpf_trampoline_multi_detach+0x20b/0x240, CPU#1: test_progs/97 ... [3.733170] bpf_tracing_multi_link_release+0x14/0x30 [3.733890] bpf_link_free+0x58/0x130 [3.734414] bpf_link_release+0x23/0x30 Fix it by setting 'ftrace_managed = true' in register_fentry_multi().	7.8	More Details
CVE-2026-93122	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: uac: validate rate list length before storing UAC1 and UAC2 configfs rate-list attributes parse a comma-separated list of sampling rates and store each parsed value in fixed-size arrays. The arrays have UAC_MAX_RATES entries, but the store paths do not check that the input contains at most that many tokens before writing through opts->name##s[i++]. Writing more than ten rates therefore writes past the end of the p_srates[] or c_srates[] array in struct f_uac1_opts or struct f_uac2_opts. With CONFIG_UBSAN_BOUNDS enabled, writing an 11-entry rate list to the UAC1 p_srate attribute reports: UBSAN: array-index-out-of-bounds drivers/usb/gadget/function/f_uac1.c:1669:1 index 10 is out of range for type 'int [10]' __ubsan_handle_out_of_bounds.cold f_uac1_opts_p_srate_store configfs_write_iter vfs_write ksys_write do_syscall_64 The same reproducer against the UAC2 p_srate attribute reports: UBSAN: array-index-out-of-bounds drivers/usb/gadget/function/f_uac2.c:2087:1 index 10 is out of range for type 'int [10]' __ubsan_handle_out_of_bounds.cold f_uac2_opts_p_srate_store configfs_write_iter vfs_write ksys_write do_syscall_64 Reject additional tokens once UAC_MAX_RATES entries have been parsed. Also keep the original kstrdup() pointer for kfree(), because strsep() advances the parsing cursor. Freeing the advanced cursor leaks the original buffer on successful parses and can free an interior pointer on some error paths.	7.8	More Details

CVE-2026-89994	In the Linux kernel, the following vulnerability has been resolved: dmaengine: fsl-edma: tracing: no ptr dereference during log output The fsl edma events store a pointer to a struct fsl_edma_engine in the ringbuffer and dereference it when a log entry is printed. At this time, the pointer may no longer be valid. Event injection can be used to trigger a crash: $ cd /sys/kernel/tracing $ echo 'value = 0' > events/fsl_edma/edma_writeb/inject $ cat trace The log output needs only edma->membase. Add a membase field at the end of the event and use the new field for log output. Keep the existing fields for backward compatibility.	7.8	More Details
CVE-2026-90291	In the Linux kernel, the following vulnerability has been resolved: module/dups: Fix use-after-free in kmod_dup_req lifetime handling The kmod dups code uses RCU to ensure that a kmod_dup_req instance is freed only after it is no longer referenced. When releasing an instance, the kmod_dup_request_delete() function removes the kmod_dup_req from the dup_kmod_reqs list, waits via synchronize_rcu() and finally frees it. However, this doesn't work correctly because parallel users referencing the instance in kmod_dup_request_exists_wait() don't enter an RCU read-side critical section. This can result in a use-after-free. The kmod_dup_request_exists_wait() function may need to hold a valid reference to a kmod_dup_req instance across a blocking wait until the corresponding modprobe command completes. This makes it unsuitable for RCU. Fix the issue by changing the lifecycle management of kmod_dup_req to use reference counting.	7.8	More Details
CVE-2026-89985	In the Linux kernel, the following vulnerability has been resolved: memcg: keep folio's objcg same as its node memcg_reparent_objcgs() has an inherent assumption that a folio's objcg is the objcg of the folio's node. Folio migration across nodes breaks that assumption: the new folio simply inherits the old folio's objcg while living on a different node. Once the assumption is broken, the reparenting of the folio's objcg and the reparenting of the folio's LRU list are no longer atomic. memcg_reparent_objcgs() handles one node per iteration and drops all the locks in between, so the objcg gets reparented in the iteration for the objcg's node while the LRU list gets spliced in the iteration for the folio's node. Any LRU operation on that folio in between resolves its lruvec through the objcg, and thus takes the lru_lock of the wrong memcg, not the lru_lock of the list the folio is actually on. Fix this by selecting the objcg by folio_nid() at charge time, and by re-deriving it for the destination node in mem_cgroup_migrate() and mem_cgroup_replace_folio().	7.8	More Details
CVE-2026-90199	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: reject out-of-range evcn in mi_enum_attr() In mi_enum_attr(), the start/end VCN validation for non-resident attributes is: if (svcn > evcn + 1) goto out; When evcn is U64_MAX the "evcn + 1" expression wraps to 0 and any svcn passes the check. For evcn values close to U64_MAX (but not equal to it) the right-hand side is still a meaningless near-wrap upper bound, so a malformed on-disk attribute with svcn == 0 and evcn near U64_MAX can pass mi_enum_attr() unrejected. VCN (virtual cluster number) is a cluster index, so any valid evcn is bounded by the volume's total cluster count, which ntfs3 holds in sbi->used.bitmap.nbits (set up in ntfs_init_from_boot() before any caller of mi_enum_attr() runs). Reject evcn values that fall outside this range. However, an empty non-resident attribute (no allocated clusters) is legitimately encoded with svcn == 0 and evcn == -1 (U64_MAX), e.g. via attr->nres.evcn = cpu_to_le64((u64)vcn - 1) with vcn == 0. That sentinel must keep passing, so exclude evcn == U64_MAX from the range check. The existing "svcn > evcn + 1" test still tolerates the sentinel ("0 > 0" is false) and continues to require svcn == 0 for it, while the range check rejects every other out-of-range evcn and thereby also defuses the "evcn + 1" wraparound. svcn does not need its own bound: once evcn < nbits, "svcn > evcn + 1" implies svcn <= nbits. [almaz.alexandrovich@paragon-software.com: fixed evcn check]	7.8	More Details
CVE-2026-89967	In the Linux kernel, the following vulnerability has been resolved: mm/migrate_device: avoid out-of-bounds writes for compound folios migrate_device_range() and migrate_device_pfns() clear the entries following a compound folio so that the PFN arrays retain their page-granular representation. If a compound folio extends beyond the end of the caller-provided range, the loops clear all following folio entries without limiting them to the number of slots remaining in the npages-sized array, causing an out-of-bounds write. Do not proceed with a compound folio if its page-granular representation does not fit entirely in the remaining PFN array. If this happens, drop any reference and lock acquired for the folio, clear the remaining entries, and stop collecting. Observed with a KASAN x86 QEMU kernel using the HMM migrate_anon_huge_zero selftest. Closing /dev/hmm_dmirror0 after migrating an anonymous huge page to device memory exercises: dmirror_fops_release() -> dmirror_device_evict_chunk() -> migrate_device_range()	7.8	More Details
CVE-2026-63325	Redocly CLI makes OpenAPI validation, linting, and documentation workflows easier. Prior to version 2.33.0 of @redocly/respect-core and @redocly/cli, the respect command dynamically evaluates $faker runtime expressions in Arazzo descriptions. A crafted expression can traverse constructor, prototype, or __proto__ properties in packages/respect-core/src/modules/context-parser/get-value-from-context.ts, reach the JavaScript Function constructor, and execute arbitrary code when a user processes an untrusted description. The executed code runs with the privileges of the CLI process and can execute shell commands or read CI secrets. Users processing only trusted, self-authored workflows are not affected. This issue is fixed in @redocly/respect-core and @redocly/cli version 2.33.0.	7.8	More Details
CVE-2026-93125	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject rdonly/rdwr_buf_size kfunc arguments that exceed u32 max check_kfunc_args() detects a kfunc argument named rdonly_buf_size or rdwr_buf_size and stores reg->var_off.value into meta->r0_size, a u64, and does not bound it. check_kfunc_call() later copies that value into the returned register's mem_size field: meta->r0_size = reg->var_off.value; ... regs[BPF_REG_0].mem_size = meta.r0_size; regs[BPF_REG_0].mem_size is u32. A constant whose upper 32 bits are set gets truncated instead of causing a load-time rejection, so the verifier records a PTR_TO_MEM register with an approximately 4 GiB mem_size for whatever allocation the kfunc returned. A later access check against that register uses the truncated, wrong bound. Reject rdonly_buf_size/rdwr_buf_size values that exceed U32_MAX at the point meta->r0_size is set.	7.8	More Details
CVE-2026-89997	In the Linux kernel, the following vulnerability has been resolved: dm: fix resume-vs-remove race If the user issues the resume ioctl and the remove ioctl at the same time, it may be possible that the device is resumed after it is suspended in __dm_destroy. The result is that the table is destroyed without calling the postsuspend method. Dm targets expect that they may be removed only after the postsuspend method method was called. If we break this expectation, it can cause misbehavior in various targets. For example - in the dm-integrity target, the reboot notifier is not unregistered, leading to use-after-free. Fix this bug by refusing to resume if the device is being destroyed.	7.8	More Details
CVE-2026-90316	In the Linux kernel, the following vulnerability has been resolved: drm/omap: dsi: Do not copy isr table To be able to unregister stuff from isrs, the corresponding table was copied. Nobody seems to unregister stuff that way, so it does not help. But there are stack-allocated objects passed to these isrs giving chances of UAF of these objects if irqs are unregistered while they are handled, so better do not copy that table.	7.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Resize MST HDCP per-connector arrays to		

CVE-2026-90289	32 AMDGPU_DM_MAX_DISPLAY_INDEX is 31. It suggest a maximum number of 32 connectors. But the way it's used is like MAX_DISPLAY_COUNT. Hence we're off by one with DRM core, which supports a max of 32 connectors. Rename AMDGPU_DM_MAX_DISPLAY_INDEX to AMDGPU_DM_MAX_DISPLAY_COUNT to match its actual use, and increase the size to 32 to match the originally intended size.	7.8	More Details
CVE-2026-89986	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix sleeping allocation in alloc_pages_bulk_weighted_interleave() syzbot reported a sleeping function called from invalid context splat in bucket_table_alloc(). When rhashtable_insert_slow() rehashes the table under rcu_read_lock(), it calls bucket_table_alloc(..., GFP_ATOMIC | __GFP_NOWARN). If the bucket table allocation uses vmalloc, __vmalloc_node_range_noprof() invokes vm_area_alloc_pages() -> alloc_pages_bulk_mempolicy_noprof() with the passed GFP_ATOMIC flags. If the current task has an MPOL_WEIGHTED_INTERLEAVE mempolicy, alloc_pages_bulk_weighted_interleave() is called and currently hardcodes GFP_KERNEL when allocating the temporary weights array, triggering a might_alloc() splat in atomic/RCU contexts. Pass the gfp flags (masked with GFP_RECLAIM_MASK to strip page-allocator zone modifiers like __GFP_HIGHMEM) received by alloc_pages_bulk_weighted_interleave() to kmalloc() instead of hardcoding GFP_KERNEL. Since the weights buffer is immediately initialized in full, kmalloc() is sufficient.	7.8	More Details
CVE-2026-93070	In the Linux kernel, the following vulnerability has been resolved: media: ipu6: Do not free aux device pdata after init ipu6_bus_initialize_device() stores the isys/psys pdata pointer in struct ipu6_bus_device and initializes the auxiliary device. After that point, error unwinding must drop the auxiliary device reference and let ipu6_bus_release() free both the bus device and adev->pdata. The isys and psys init paths already call put_device() when MMU initialization fails, and ipu6_bus_add_device() calls auxiliary_device_uninit() on auxiliary_device_add() failure. Both paths therefore run the bus release callback. The extra kfree(pdata) in the callers can release the same object a second time. Remove the manual pdata frees after the auxiliary device has been initialized. This issue was found by a static analysis checker and confirmed by manual source review.	7.8	More Details
CVE-2026-93074	In the Linux kernel, the following vulnerability has been resolved: dax/fsdev: use __va(phys) for kaddr in direct_access Use __va(phys) instead of virt_addr + linear_offset for the kaddr return in __fsdev_dax_direct_access(). The previous code added a device-linear byte offset to virt_addr (which is __va of ranges[0]), but for multi-range devices with physical gaps between ranges, this linear arithmetic crosses the gap and produces a wrong kernel virtual address. Using __va(phys) where phys comes from dax_pgoff_to_phys() is correct for any range layout because the direct map translates each physical address independently. This leaves dev_dax->virt_addr write-only, so remove the field (suggested by Dave Jiang).	7.8	More Details
CVE-2026-90312	In the Linux kernel, the following vulnerability has been resolved: bpf: Check load-acquire src ptr type before the load check_atomic_load() calls check_load_mem() before atomic_ptr_type_ok(). For a load-acquire that fetches into its own source register (dst_reg == src_reg), check_load_mem() overwrites src_reg's type with the type of the loaded value, so the subsequent atomic_ptr_type_ok() no longer sees the source pointer and fails to reject the disallowed types (ctx, pkt, flow_keys, sock). Since bpf_convert_ctx_accesses() does not rewrite atomic loads, the raw access to the underlying kernel object is left in place. The destination type is taken from the ctx access itself, so a load-acquire of the sk field of struct __sk_buff for example leaves the register typed as PTR_TO_SOCK_COMMON_OR_NULL, which type_is_sk_pointer() does not match either, while it actually holds unconverted struct sk_buff bytes. Once the NULL check has passed this is a type confusion, not just a leak of kernel data. Validate src_reg with check_reg_arg() and check the source pointer type with atomic_ptr_type_ok() before the load again, mirroring check_atomic_rmw(). Out-of-range register numbers are already rejected earlier by check_and_resolve_insns() (commit 503d21ef8eac ("bpf: Do register range validation early")), and the only exemption there, is_stack_arg_ldx(), requires BPF_LDX | BPF_MEM | BPF_DW and thus never matches a BPF_ATOMIC insn. atomic_ptr_type_ok() can therefore not dereference register state out of bounds, that is, the out-of-bounds read addressed by the Fixes commit below does not reappear (as proven also via selftest).	7.8	More Details
CVE-2026-90317	In the Linux kernel, the following vulnerability has been resolved: bpf: Invalidate RCU pointers after final spin unlock In a sleepable BPF program, a spin lock can provide the only RCU protection for a kptr. The final bpf_spin_unlock() ends that protection, but the verifier leaves the pointer valid. Another CPU can then free the object before the pointer is used. A capability-limited runtime PoC triggered a task_struct use-after-free in __bpf_get_task_stack(). Record whether the program is in an RCU-protected context before releasing the lock. Invalidate RCU-protected pointers only when the unlock leaves the final such context. This preserves valid pointers in non-sleepable programs and inside an explicit RCU read-side section.	7.8	More Details
CVE-2026-93105	In the Linux kernel, the following vulnerability has been resolved: esp: do not unref managed frag pages in esp_ssg_unref() esp_ssg_unref() releases the page references held on the source scatterlist after the AEAD operation completes. It calls skb_page_unref() on every frag page for an out-of-place transform (req->src != req->dst), and in the error path of esp_output_tail() (already_unref == true) on the request's own scatterlist. This is wrong when the skb carries managed frags (SKBFL_MANAGED_FRAG_REFS). Managed frags are owned by a zerocopy ubuf and the skb does not hold a per-frag page reference; io_uring SEND_ZC with a registered buffer attaches the bvec pages this way via io_sg_from_iter(). The rest of the stack honours this invariant: skb_release_data() skips the per-frag unref when SKBFL_MANAGED_FRAG_REFS is set, and skb_zcopy_managed() is the guard used at the other unref sites. esp_ssg_unref() is missing that guard, so for a managed-frag skb it drops a page reference the skb never acquired. This can underflow the page reference count and free a page that is still in use. Guard the function with skb_zcopy_managed() so both unref paths are skipped for managed-frag skbs, matching skb_release_data().	7.8	More Details
CVE-2026-89988	In the Linux kernel, the following vulnerability has been resolved: kprobes: Protect kprobe_blacklist with RCU __within_kprobe_blacklist() traverses kprobe_blacklist without holding kprobe_mutex. When a module is unloaded, kprobe_remove_area_blacklist() removes blacklist entries and immediately frees them with kfree(). A concurrent call to within_kprobe_blacklist() can therefore dereference freed memory. Furthermore, within_kprobe_blacklist() can be called in atomic or non-preemptible contexts where the sleeping kprobe_mutex cannot be taken. Protect kprobe_blacklist with RCU. Use guard(rcu)() and list_for_each_entry_rcu() for traversal, list_add_tail_rcu() for insertions, list_del_rcu() for deletions, and kfree_rcu() to reclaim entries safely after a grace period.	7.8	More Details
CVE-2026-93079	In the Linux kernel, the following vulnerability has been resolved: cxl/features: Reject Get Feature count larger than the output buffer cxlctl_get_feature() sizes its output buffer from the user's fwctl_rpc.out_len, but the device is told to write cxl_mbox_get_feat_in.count bytes into rpc_out->payload, which is a separate user-controlled value. Nothing bounds count against out_len, so a small out_len with a large count overflows the kvzalloc()'d buffer. A heap OOB write reachable from FWCTL_RPC. Reject requests where count exceeds the available payload room, before allocating.	7.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: bpf: Check pointer type for all atomic RMW paths Atomic RMW		

CVE-2026-90177	verification records an instruction pointer type only when the current destination is PTR_TO_ARENA. A second path can therefore reach the same instruction with an ordinary pointer without comparing it against the saved arena type. The post-verification fixup uses the saved type to rewrite the instruction to BPF_PROBE_ATOMIC for every path. Record the actual destination type for all atomic RMW paths so the existing mismatch check rejects incompatible uses of one instruction.	7.8	More Details
CVE-2026-89979	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix race between non-atomic ops and trigger-start We protect the races of the concurrent state transitions between atomic PCM ops, but the checks between the non-atomic ops (hw_params, hw_free and prepare) and the atomic ops aren't perfect; there is a check of the conflicting PCM state at the beginning of hw_params & co, but the atomic PCM ops can be still issued during the non-atomic PCM operations. An example such scenario is that a thread A re-issues the PREPARE or HW_PARAMS for the already prepared stream, while another thread B triggers the PCM start in the middle of the prepare operation. Although this usually doesn't lead to much serious issues, it can give some inconsistency as reported by syzkaller (such as ODEBUG warning). There are various atomic PCM ops, and basically the only problem is the PCM start as it operates from the PREPARED state. Other trigger commands (stop, etc) are for the running or the other special state, hence they are filtered as pre-condition. This patch is for preventing the PCM trigger-start during the non- atomic operations in order to address the problems above. Fortunately, the hw_params, hw_free and prepare operations call snd_pcm_buffer_access_lock(), and this can be used for checking the concurrent operations at the PCM trigger -- which sets the runtime->buffer_accessing to a negative (if possible), so the PCM trigger just needs to check the runtime->buffer_accessing value; if it's negative, it means the concurrent non-atomic PCM ops is running.	7.8	More Details
CVE-2026-93095	In the Linux kernel, the following vulnerability has been resolved: hfsplus: validate thread record before delete key rebuild hfsplus_delete_cat() is called with str == NULL when the last open reference to an unlinked HFS+ hardlink backing inode is closed. In that case, the function finds the catalog thread by CNID and rebuilds the catalog key from thread.nodeName. That reconstruction path reads thread.nodeName.length directly from the catalog B-tree into fd.search_key and then copies length * 2 bytes into fd.search_key->cat.name.unicode. It does not first check that the found record is a thread record or that its size matches the thread name. A corrupted image can therefore provide an oversized thread name length and make hfs_bnode_read() write past the catalog search-key allocation. Read the CNID record through hfsplus_brec_read_cat(), which bounds the record read to sizeof(hfsplus_cat_entry) and verifies that a thread record's size exactly matches nodeName.length. Together, these checks ensure an accepted thread name fits HFSPLUS_MAX_STRLEN. Reject non-thread records before building the delete key from the validated thread name. Share the thread-record-type helper between hfsplus_find_cat() and hfsplus_delete_cat().	7.8	More Details
CVE-2026-90244	In the Linux kernel, the following vulnerability has been resolved: iommu/dma: Restore locking around msi_page_list Unlike a group's default domain, which is always freshly allocated and privately owned (iommu_group_alloc_default_domain()), VFIO type1's legacy container merges any newly attached group into an existing domain whenever their iommu_ops and cache-coherency enforcement match. iommu_dma_get_msi_page() only asserts the caller's own group mutex is held (iommu_group_mutex_assert()). On an IOMMU that publishes IOMMU_RESV_SW_MSI, e.g. ARM SMMU, a VM with two such devices assigned through the legacy container can have their guest drivers probe and allocate MSIs in parallel; each host-side VFIO_DEVICE_SET_IRQS lands on a different device fd and group mutex, but both devices' domains are the same merged domain, so both can enter iommu_dma_get_msi_page() concurrently and corrupt msi_page_list. commit 288683c92b1a ("iommu: Make iommu_dma_prepare_msi() into a generic operation") dropped the prior msi_prepare_lock on the reasoning that "each iommu_domain is unique to a group," which holds for default domains but not this VFIO type1 case. Restore the static lock, since it's only guarding a corner case and will likely never be contended. iommufd avoids the equivalent problem by having its own callers (iommufd_sw_map_msi()) take a ctx-wide sw_msi_lock before ever reaching the shared list. VFIO type1 can't mirror that since it dispatches to iommu_dma_sw_msi() which is outside VFIO's jurisdiction.	7.8	More Details
CVE-2026-90309	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: Hold CQ references when processing EQ events EQ handlers look up CQs from dev->cq_xa and invoke CQ completion or error callbacks outside the xarray lock. erdma_destroy_cq() can erase the CQ from the xarray and free its queue buffer and doorbell record while a previously scheduled EQ handler is still using the CQ. Add a CQ refcount and take a reference under the xarray lock with refcount_inc_not_zero(). Remove the CQ from the xarray before dropping the destroy-path reference, then wait for in-flight EQ users before releasing CQ resources.	7.8	More Details
CVE-2026-90308	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: Hold QP references for AE and CM processing AE QP fatal events and iWARP CM paths load QPs from dev->qp_xa and then use or reference them outside the xarray lock. erdma_destroy_qp() can drop the destroy-path reference and free QP resources while such a lookup is in flight. Add erdma_qp_get_by_qpn() to acquire a kref under the xarray lock with kref_get_unless_zero(). Remove the QP from the xarray before dropping the destroy-path reference so no new lookup can acquire it while destruction waits for existing users.	7.8	More Details
CVE-2026-90046	In the Linux kernel, the following vulnerability has been resolved: mm/page_alloc: don't spin_trylock() in NMI on UP Patch series "mm/page_alloc: fixes for free_pages_nolock() on RT/UP". Pre-existing bugs found by Sashiko during review of this other series: https://lore.kernel.org/all/20260703-alloc-trylock-v5-0-c87b714e19d3@google.com/ I have not reproduced these bugs, and I suspect there is no real-world user that is affected by them. This patch (of 2): As noted in can_spin_trylock(), using this is unsafe in this context. commit 620b46ed6ae17 ("mm/page_alloc: return NULL early from alloc_frozen_pages_nolock() in NMI on UP") fixed this on the alloc side but missed the free side. Impact: If BPF programs using these features in NMI (probably tracing) are present on non-SMP builds this might crash the kernel and is probably exploitable by local attackers for privilege escalation.	7.8	More Details
CVE-2026-90324	In the Linux kernel, the following vulnerability has been resolved: ublk: check import_ubuf() return value import_ubuf() can fail if the address range (provided by the userspace ublk server) is outside the allowed user address space. Return that 0 bytes were copied if import_ubuf() fails rather than passing an uninitialized struct iov_iter to ublk_copy_user_pages().	7.8	More Details
CVE-2026-89998	In the Linux kernel, the following vulnerability has been resolved: dm: fix race when loading and unloading a table If the userspace calls two concurrent table load ioctls and one of them succeeds and the other fails, there is a race condition because dm_setup_md_queue walks &md->table_devices without any lock. If the walk races with dm_table_destroy -> free_devices -> dm_put_table_device, there is access to invalid memory. Fix this race by extending the lock over the list walk.	7.8	More Details
CVE-2026-90388	In the Linux kernel, the following vulnerability has been resolved: iommu/dma: Check atomic pool allocation result directly The non-blocking, non-coherent allocation path uses dma_alloc_from_pool(), which returns the allocated page and fills cpu_addr only on success. Do not rely on cpu_addr to detect allocation failure in this path. Check the returned page directly before using it for the IOMMU mapping.	7.8	More Details

CVE-2026-90030	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: clear forceRM when issuing EndTransfer The forceRM bit of the DEPCMD register controls the behavior of the EndTransfer command used to stop an active transfer. Older DWC3 programming guide revisions recommended setting forceRM=1 when issuing EndTransfer. Newer programming guide revisions recommend issuing EndTransfer with forceRM cleared. With forceRM=1 on DWC_usb31 v2.00a and v2.10a controllers, a transfer aborted through the ep_dequeue path was observed to remain active after EndTransfer completion. A subsequent StartTransfer issued on the same endpoint triggered writes associated with the aborted transfer. This resulted in an SMMU fault because the transfer buffer had already been unmapped during EndTransfer command-completion cleanup. Using forceRM=0 eliminates the issue. Although older DWC3 programming guide revisions recommended setting forceRM=1, no issues are known from using forceRM=0. Clear forceRM when issuing EndTransfer to provide consistent EndTransfer behavior and align with newer programming guide recommendations.	7.8	More Details
CVE-2026-92507	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix potential use after free in ib_dealloc_pd_user() When accessing a PD via the netlink path the only synchronization mechanism for the said PD is rdma_restrack_get(). Currently, rdma_restrack_del() is invoked at the end of ib_dealloc_pd_user(), which is too late, since by that point vendor-specific resources associated with the PD might already be freed. This can leave a short window where the PD remains accessible through restrack, leading to a potential use-after-free. Fix this by moving the rdma_restrack_begin_del() call to the start of ib_dealloc_pd_user(), ensuring that the PD is removed from restrack before its internal resources are released. This guarantees that no new users hold references to a PD that is in the process of destruction. In addition, this change preserves the intended inverted order between create and destroy routines: resources are added to restrack at the end of successful creation, and hence shall be removed from the restrack first thing during the destruction flow, which keeps the lifecycle management consistent and predictable.	7.8	More Details
CVE-2026-90032	In the Linux kernel, the following vulnerability has been resolved: media: usbtv: keep device alive while ALSA card exists The ALSA PCM callbacks store the driver state in pcm->private_data. An open PCM file can outlive USB disconnect because usbtv_audio_free() uses snd_card_free_when_closed(). The disconnect path can then drop the V4L2 device reference and free struct usbtv before ALSA releases the substream, so a later close dereferences freed memory in snd_usbtv_pcm_close(). Take a V4L2 device reference for the ALSA card and drop it from the card private_free callback. This keeps struct usbtv valid until ALSA has closed the remaining files and freed the card.	7.8	More Details
CVE-2026-90343	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: stop PMSR before P2P and NAN teardown PMSR request teardown must abort active measurements while the wireless_dev is still present in the driver. cfg80211_leave_locked() and cfg80211_stop_pd() already do this before invoking the driver's stop callback, but cfg80211_stop_p2p_device() and cfg80211_stop_nan() do not. Those helpers are also called directly by nl80211, rfkill shutdown, and wireless_dev unregister paths. If one of these paths stops a P2P device or NAN interface with a pending request, it removes the mac80211 subinterface from the driver first. Subsequent request cleanup cannot reach the lower driver's abort callback, but cfg80211 frees the request regardless. Driver state can then retain a stale request and use it when it later reports a result. Call cfg80211_pmsr_wdev_down() before stopping the P2P device or NAN interface. This keeps lower-driver request state and cfg80211 request ownership in sync for all of the helpers' callers.	7.8	More Details
CVE-2026-90210	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix UAF in bpf_trampoline_multi_attach_free on update failure When bpf_trampoline_update() fails before modify_fentry_multi()/ unregister_fentry_multi() is called, cur_image is unchanged (cur_image == old_image) and ftrace still calls into it. Freeing old_image in that case causes a UAF. Only free old_image when it differs from cur_image.	7.8	More Details
CVE-2026-90358	In the Linux kernel, the following vulnerability has been resolved: bpf, x86: Fix trampoline stack size for 128-bit arguments btf_distill_func_proto() accepts a function argument up to 16 bytes, so a 128-bit scalar such as __int128 reaches the x86 trampoline with arg_size == 16. But the current implementation assumes an __int128 argument only needs one register, so the register save area is under-allocated and save_args() overwrites adjacent stack slots. Compute the register count from arg_size for all arguments to fix it.	7.8	More Details
CVE-2026-93127	In the Linux kernel, the following vulnerability has been resolved: bpf: Drop scalar id on sign-extending narrowing stack fills When a spilled scalar is filled back with a sign-extending narrowing load (BPF_MEMSX), check_stack_read_fixed_off() copies the spilled register including its scalar id, but coerce_reg_to_size_sx() then sign-extends the filled register's value. If the same slot is also filled with a plain zero-extending load (BPF_MEM), both destination registers share the id yet hold different values. A later 'if <zext-reg> == const' then refines the sign-extended register through sync_linked_regs() to a value it does not have at runtime (e.g. the verifier believes 0x80000000 while the register is 0xffffffff80000000), which can be turned into an out-of-bounds access. Drop the shared scalar id at the sign-extension site in check_mem_access() when sign extension actually changes the value, mirroring the BPF_MOVSX handling in check_alu_op() (no_sext = reg_umax < 2^(size*8-1)).	7.8	More Details
CVE-2026-90383	In the Linux kernel, the following vulnerability has been resolved: misc: sgi-gru: remove interrupt-context page-table walks The GRU TLB miss handler walks a process's page tables without holding page-table locks or a reference to the mapped page. It also uses a kernel page-table accessor on user page tables and supports only PMD-level large mappings on x86-64. Remove the direct walker. Send interrupt faults directly to user polling mode so the existing call-OS fallback retries them in process context. Remove the mmap-lock failure statistic that can no longer be incremented.	7.8	More Details
CVE-2026-90387	In the Linux kernel, the following vulnerability has been resolved: swiotlb: Preserve allocation virtual address for dynamic pools swiotlb_alloc_tlb() can allocate from the DMA atomic pool when a decrypted pool is needed from atomic context. With CONFIG_DMA_DIRECT_REMAP, the atomic pool is backed by remapped virtual addresses, which are not the same as the direct-map addresses returned by phys_to_virt(). swiotlb_init_io_tlb_pool() currently reconstructs the pool virtual address from the physical start address. For atomic-pool backed allocations this stores the wrong address in pool->vaddr. Later, swiotlb_free_tlb() passes that address to dma_free_from_pool(), which will fail to recognize the chunk Pass the virtual address returned by the allocation path into swiotlb_init_io_tlb_pool(), and store that address in pool->vaddr. This keeps the pool free path using the same virtual address as the allocator.	7.8	More Details
CVE-2026-92485	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix WARNING in bpf_tracing_link_release The trampoline could be corrupted by the blindly 'tr->flags = BPF_TRAMP_F_TAIL_CALL_CTX' in verifier. 1. A fexit attached to a tail_call_reachable prog. 'tr->flags' became 'BPF_TRAMP_F_CALL_ORIG | BPF_TRAMP_F_TAIL_CALL_CTX'. And, the trampoline would poke the target prog's nop insn using jmp insn instead of call insn. 2. Another fexit loaded with the same tail_call_reachable prog target. 'tr->flags' became 'BPF_TRAMP_F_TAIL_CALL_CTX'. 3. Close the first fexit link. Due to no BPF_TRAMP_F_CALL_ORIG in 'tr->flags', the trampoline will fail to restore the prog's nop insn using call insn. [3.410719]	7.8	More Details

	WARNING: kernel/bpf/syscall.c:3551 at bpf_tracing_link_release+0x53/0x60, CPU#1: test_progs/98 ... [3.428793] bpf_link_free+0x58/0x130 [3.429293] bpf_link_release+0x23/0x30 Fix the warning by updating 'tr->flags' with '|=' and lock.		
CVE-2026-90392	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix potential UAF when reading bpf link info In bpf_link_show_fdinfo and bpf_link_get_info_by_fd, link->prog is accessed without holding any locks. If the prog is concurrently replaced via bpf_link_update, the old prog can be freed, leading to a potential UAF issue. Fix this by accessing link->prog under RCU protection to safely fetch the pointer and guarantee its lifetime while reading its fields.	7.8	More Details
CVE-2026-92508	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix potential use after free in ib_free_cq() When accessing a CQ via the netlink path the only synchronization mechanism for the said CQ is rdma_restrack_get(). Currently, rdma_restrack_del() is invoked at the end of ib_free_cq(), which is too late, since by that point vendor-specific resources associated with the CQ might already be freed. This can leave a short window where the CQ remains accessible through restrack, leading to a potential use-after-free. Fix this by moving the rdma_restrack_del() call to be before the freeing of the vendor-specific resources ensuring that the CQ is removed from restrack before its internal resources are released. This guarantees that no new users hold references to a CQ that is in the process of destruction.	7.8	More Details
CVE-2026-90227	In the Linux kernel, the following vulnerability has been resolved: nvme/ioctl: check SUBMIT_IO with nvme_cmd_allowed() Unlike IO_CMD / IO64_CMD, NVME_IOCTL_SUBMIT_IO never calls nvme_cmd_allowed(). Unprivileged callers can thus issue I/O on a partition device or write through a read-only file descriptor. Pass flags and open_for_write through and reject disallowed commands with -EACCES.	7.8	More Details
CVE-2026-59974	Stanza is a Stanford NLP Python library for tokenization, sentence segmentation, NER, and parsing of many human languages. Prior to 1.14.0, stanza.resources.common.unzip in stanza/resources/common.py passes downloaded model and resource archives to zipfile.ZipFile.extractall without validating member paths, and the vulnerable extraction path is reachable through stanza.download and stanza.install_corenlp. A malicious archive containing parent-directory traversal entries can write outside the intended model directory, allowing files writable by the Stanza process to be overwritten and potentially enabling code execution through modified shell configuration, SSH authorization data, Python packages, or executable scripts. This issue is fixed in version 1.14.0.	7.8	More Details
CVE-2026-90435	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix integer overflow of user QP buffer size set_user_buf_size() computes the QP buffer size by left-shifting the user-supplied rq.wqe_cnt and rq.wqe_shift values as signed integers. A sufficiently large rq.wqe_cnt causes signed integer overflow, which is undefined behavior, and yields a small or negative buf_size, causing ib_umem_get() to map a buffer smaller than the hardware will actually write into. Replace the shifts and addition with check_shl_overflow() and check_add_overflow(), rejecting invalid user inputs. Moreover, guard the identical shift computing qp->sq.offset in _create_user_qp() before set_user_buf_size() is reached.	7.8	More Details
CVE-2026-90225	In the Linux kernel, the following vulnerability has been resolved: nfc: llcp: read llcp_sock->local under the socket lock in getsockopt nfc_llcp_getsockopt() read llcp_sock->local before lock_sock(sk) and then dereferenced the cached pointer inside the locked region. llcp_sock_bind() assigns and clears llcp_sock->local under the same socket lock, dropping the last reference on its error path. A getsockopt() racing an in-flight bind() can observe the pointer, block on lock_sock(), and then dereference a freed nfc_llcp_local once bind() has unwound. Move the llcp_sock->local read and the NULL check inside the lock_sock(sk) region so bind() cannot mutate or free the pointer between the load and the use.	7.8	More Details
CVE-2026-90217	In the Linux kernel, the following vulnerability has been resolved: bpf: Compare iterator types during state pruning An iterator stack slot can be MEM_RCU or PTR_UNTRUSTED. These states must not be equal, or the verifier can prune an unsafe path. Compare the pointer type for STACK_ITER slots.	7.8	More Details
CVE-2026-90429	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Synchronize the error ISR against VINTF (de)init A user VINTF is torn down by tegra241_cmdqv_deinit_vintf(), which runs from the destroy callback and from the init-failure unwind in the alloc handler. It clears the cmdqv->vintfs[] slot and lets the iommufd core free it, but nothing serializes that against the error interrupt: tegra241_cmdqv_isr() reads cmdqv->vintfs[idx] and dereferences the vintf. A concurrent error can make the ISR read a slot mid-clear (a NULL deref) or use a vintf which is about to be freed (a use-after-free). deinit_vintf() also returns idx to the IDA before clearing the slot, so a concurrent create that reuses idx can publish its new vintf into the slot, only for this teardown to erase it again with the stale NULL store. On the other end, tegra241_cmdqv_init_vintf() publishes a new vintf with a plain store to the cmdqv->vintfs[] slot, and the ISR dereferences fields of a published vintf such as vintf->base. A plain store gives no ordering on a weakly-ordered CPU, and a stale VINTF_ERR_MAP bit on a reused idx can make the ISR pick a vintf the moment it is published, before its fields are set or tegra241_vintf_hw_init() runs. The cmdqv->vintfs[0] slot stays NULL until tegra241_cmdqv_init_structures() first creates VINTF0, so the slot 0 read needs the same NULL check. Publish every slot with an smp_store_release(), and read each slot in the ISR with an smp_load_acquire() under a NULL check, so the ISR always sees a fully built vintf or NULL. Also make deinit_vintf() clear the slot, and synchronize_irq() prior to returning idx to the IDA, so no vintf is freed under a running handler and no reused idx is clobbered.	7.8	More Details
CVE-2026-90423	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix UAF in ODP init error-handling path rxe_odp_mr_init_user() stores &umem_odp->umem in mr->umem before calling rxe_odp_init_pages(). If rxe_odp_init_pages() fails, rxe_odp_mr_init_user() releases umem_odp and returns an error. rxe_reg_user_mr() then unwinds the error through rxe_cleanup(), rxe_mr_cleanup(), ib_umem_release(mr->umem). There is an IS_ERR_OR_NULL(umem) check at the start of ib_umem_release(). But since mr->umem is NOT reset to NULL in the error handling path of rxe_odp_mr_init_user(), the check passes and it reads already-freed fields like umem->is_dmabuf, causing UAF. Fix the UAF by clearing mr->umem after releasing the failed ODP umem so the MR cleanup path does not release it again.	7.8	More Details
CVE-2026-90043	In the Linux kernel, the following vulnerability has been resolved: zram: fix slot lock bit position on big-endian 64-bit The slot lock is a bit operation on the whole __lock word, which flags and ac_time alias as two u32s. On little-endian the lock bit lands in the position ZRAM_ENTRY_LOCK reserves in flags, so the aliasing works out. On 64-bit big-endian it lands in ac_time instead: with ZRAM_TRACK_ENTRY_ACTIME enabled, storing the access time from mark_slot_accessed() or slot_free() wipes out the held lock bit, letting another CPU take the same slot lock; an access time value with that bit set makes the slot look locked forever. Shift the lock bit into the flags half of the word on big-endian 64-bit.	7.8	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_fs: Fix Use-After-Free in AIO error path In ffs_epfile_write_iter() and ffs_epfile_read_iter(), when ffs_epfile_io() fails with an error other than -EIOCBQUEUED, the io_data structure (`p`) is freed. However, for AIO operations, the kiocb cancel function was already armed and kiocb->private was set to `p`. If a concurrent cancel operation (such as sys_io_cancel()) executes after ffs_epfile_io() fails but before the function frees		

2026-90044	`p`, a Use-After-Free can occur when the cancellation handler accesses the freed pointer. To securely fix this race condition, we must properly un-arm the cancellation. Invoking `kiocb->ki_complete()` does exactly this by acquiring `ctx->ctx_lock` and safely removing the kiocb from the active sequence. In doing so, it ensures that a parallel io_cancel can no longer discover the kiocb, effectively closing the race window. We then return -EIOCBQUEUED to notify the VFS layer that the kiocb has been consumed and it should avoid attempting to complete the request again or triggering subsequent completion handlers.	7.8	More Details
CVE-2026-90045	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: ffs: fix mm lifetime handling io_data stores a pointer to the submitting task's mm_struct, but does not currently hold a reference to it while async requests are pending. This can result in a use-after-free if the task exits before completion handling finishes. Take a reference with mmgrab() when queuing the read request and release it with mmdrop() on request completion.	7.8	More Details
CVE-2026-90237	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_ct: move custom expectation support to helper Originally, the ct expectation support called nf_ct_helper_ext_add() for confirmed conntracks, which is invalid, triggering a splat. This was fixed by commit 1710eb913bdc ("netfilter: nft_ct: skip expectations for confirmed conntrack") which restricted it to unconfirmed conntracks. However, early insertion of expectations into the expectations list when the conntrack is unconfirmed leads to stale entries pointing to the wrong hlist_head through .pprev due to ct extension reallocation. Commit 7c9664351980 ("netfilter: move nat hlist_head to nf_conn") moved the nat hlist_head to nf_conn for this reason: 1. ... 2. When reallocation of extension area occurs we need to fixup the bysource hash head via hlist_replace_rcu. I'd rather not increase the size of the struct nf_conn for this feature has very limited scope: only one expectation can be created at a time given expect_clash() will make nf_ct_expect_related() reports EBUSY. For this reason, relax nf_ct_expect_related() not to drop packets in case expectation creation fails, therefore, expectation creation becomes best effort. To address this issue, add an internal ct helper and attach it to the conntrack entry to streamline the custom ct expectation support with existing ct helpers. Expose a new nf_conntrack_helper_release() function to release the internal helper that is allocated and attached to the conntrack entry to create the custom expectations. The nft_ct module removal always waits for rcu grace period, then the NULL helper callback is observed after this. This patch also restricts the creation of expectations to different helpers other than this custom helper that is created for this type of expectations.	7.8	More Details
CVE-2026-92510	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix potential use after free in ib_destroy_srq_user() When accessing a SRQ via the netlink path the only synchronization mechanism for the said SRQ is rdma_restrack_get(). Currently, rdma_restrack_del() is invoked at the end of ib_destroy_srq_user(), which is too late, since by that point vendor-specific resources associated with the SRQ might already be freed. This can leave a short window where the SRQ remains accessible through restrack, leading to a potential use-after-free. Fix this by moving the rdma_restrack_begin_del() call to the start of ib_destroy_srq_user(), ensuring that the SRQ is removed from restrack before its internal resources are released. This guarantees that no new users hold references to a SRQ that is in the process of destruction. In addition, this change preserves the intended inverted order between create and destroy routines: resources are added to restrack at the end of successful creation, and hence shall be removed from the restrack first thing during the destruction flow, which keeps the lifecycle management consistent and predictable.	7.8	More Details
CVE-2026-90320	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate external xattr entries when reading metadata ocfs2_validate_xattr_block() checks the xattr block header before the block reaches higher-level xattr users, but it does not verify that a non-indexed block's xh_count and entry offsets fit inside the block. Indexed buckets likewise reach list/get consumers after ECC without an entry-bounds check. Use the flat xattr entry validator for non-indexed external xattr blocks, and use a bucket-specific validator for indexed buckets at metadata read time. The bucket validator keeps the entry array bounded by the first bucket block while checking name/value offsets against the bucket block they target. Reject corrupted external xattr metadata before listxattr() or getxattr() can walk out-of-range entry arrays or name/value offsets. Validation reproduced this kernel report: BUG: KASAN: use-after-free in ocfs2_xattr_list_entries+0xd7/0x190 Read of size 1 at addr ffff88810a654007 by task ocfs2_xattr_lis/630 Call Trace: dump_stack_lvl+0x66/0xa0 print_report+0xce/0x630 kasan_report+0xe0/0x110 ocfs2_xattr_list_entries+0xd7/0x190 ocfs2_listxattr+0x3f6/0x610 listxattr+0x90/0xe0 path_listxattrat+0xed/0x220 do_syscall_64+0x115/0x6a0 entry_SYSCALL_64_after_hwframe+0x77/0x7f	7.8	More Details
CVE-2026-90009	In the Linux kernel, the following vulnerability has been resolved: scsi: bsg: Fix TOCTOU in io_uring passthrough command setup scsi_bsg_uring_cmd() reads bsg_uring_cmd from the shared mmap'd SQE. Userspace can change a field after we check it and before we use it. request_len is the sharp case: it can grow past sizeof(scmd->cmnd) after the bound check and overflow scmd->cmnd in copy_from_user(). READ_ONCE() the SQE fields we check or use into locals before use.	7.8	More Details
CVE-2026-90204	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate DIO orphan slot during inode read [BUG] A corrupted append-DIO dinode (high byte at offset 0xa1 corrupted from 0 to 1) can carry an i_dio_orphaned_slot outside the mounted filesystem slot range and trigger a use-after-free error: BUG: KASAN: slab-use-after-free in ocfs2_get_system_file_inode+0x780/0x820 fs/ocfs2/sysfile.c:102 Read of size 8 at addr ffff88800b767c00 by task kworker/u8:3/85 Call Trace: ... ocfs2_get_system_file_inode+0x780/0x820 fs/ocfs2/sysfile.c:102 ocfs2_wipe_inode+0x292/0xf70 fs/ocfs2/inode.c:840 ocfs2_delete_inode fs/ocfs2/inode.c:1155 [inline] ocfs2_evict_inode+0x6c9/0x1170 fs/ocfs2/inode.c:1295 evict+0x38e/0x8f0 fs/inode.c:810 iput_final fs/inode.c:1914 [inline] iput fs/inode.c:1966 [inline] iput+0x55b/0x8b0 fs/inode.c:1926 ocfs2_recover_orphans+0x610/0xe40 fs/ocfs2/journal.c:2374 ocfs2_complete_recovery+0x5af/0xd00 fs/ocfs2/journal.c:1373 ... [CAUSE] ocfs2_del_inode_from_orphan() uses i_dio_orphaned_slot to index the slot-local system inode cache. The dinode validator does not check this active slot, so an out-of-range value produces an invalid cache entry pointer that is dereferenced as an inode pointer. [FIX] Reject an active i_dio_orphaned_slot outside the slot range during dinode validation, before DIO orphan recovery can consume it.	7.8	More Details
CVE-2026-90001	In the Linux kernel, the following vulnerability has been resolved: HID: bpf: serialize device reference release in struct_ops destroy path __hid_bpf_ops_destroy_device() and hid_bpf_unreg() can race on the same registration reference, double-putting struct hid_device and freeing it while hid_destroy_device() still uses it. Serialize the remove/NULL decision under hdev->bpf.prog_list_lock so exactly one path releases each registration reference: unreg re-checks ops->hdev under the lock and returns without putting when the destroy path already cleared it; all put_device() calls happen after the lock is dropped, which is safe because a concurrent unreg then observes ops->hdev == NULL under the lock. Background: each successful attach (hid_bpf_ops_reg) acquires one device reference (hid_get_device()). Two paths can release it: - device destruction: hid_destroy_device() -> hid_bpf_destroy_device() -> __hid_bpf_ops_destroy_device(), which walks hdev->bpf.prog_list under rcu_read_lock() and drops one reference per attached program; - BPF link release: bpf map delete (no BPF_F_LINK) synchronously calls st_ops->unreg() -> hid_bpf_unreg(), which drops the reference for its own registration. The coordination handshake (e->hdev = NULL on the destroy side vs "if (!hdev) return" on the unreg side) is a TOCTOU check: the two paths run under different lock domains (rcu_read_lock vs prog_list_lock), so a concurrent unreg can read ops->hdev as non-NULL, block	7.8	More Details

	on prog_list_lock, and then proceed while the destroy traversal executes - both paths then drop the same reference. The refcount reaches zero legitimately (each decrement is individually valid), so no refcount_t saturation fires: the device is simply freed while the transport is still inside hid_destroy_device(), and subsequent teardown touches freed memory. The fix serializes the remove/NULL decision under prog_list_lock on both sides and moves the destroy-side puts outside the lock. With the lock held, plain reads/writes of ops->hdev are sufficient; no READ_ONCE/WRITE_ONCE are added, keeping the patch minimal. Unlocked-read safety: the unlocked read of ops->hdev at the top of hid_bpf_unreg() cannot touch a freed device, because the unreg path itself still holds this registration's reference (released only by its own hid_put_device() after the lock is dropped), and a destroy traversal that already cleared ops->hdev makes the lock-internal re-check return early without any put. At most one of the two paths releases each registration reference.		
CVE-2026-90002	In the Linux kernel, the following vulnerability has been resolved: ftrace: Take trace_array reference before accessing its ftrace_ops The trace instance files set_ftrace_filter and set_ftrace_notrace was updated to work with specific trace instances (trace_arrays). The issue is that when these files are opened, there is a small race window where it will use the ftrace_ops from the inode->private pointer to get a reference to the trace_array and then take its reference. The problem is that the ftrace_ops itself could be freed. If the rmdir on the instance happens at the same time the set_ftrace_filter file is opened, the rmdir could have also freed the ftrace_ops and referencing it will cause a use-after-free bug and crash the kernel. Instead, pass in the trace_array as the file private data (NULL for the top level instance), and then pass both the trace_array and the ftrace_ops to the ftrace_regex_open() function. If the trace_array is NULL, then it just uses the ftrace_ops without the need to take its reference (like normal). If the ftrace_ops is NULL, that is only the case for the top level instance and the global_ops can be used. This allows the trace_array to have its reference incremented before touching the ftrace_ops that could also be freed when the instance is.	7.8	More Details
CVE-2026-90321	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate inline xattrs during inode block validation Patch series "ocfs2: validate xattr entry bounds", v7. This series validates OCFS2 xattr entry name/value bounds when xattr metadata is read and validated, before getxattr() or listxattr() can walk out-of-range entry arrays or offsets from corrupted metadata. This patch (of 2): ocfs2_validate_inode_block() verifies a dinode before OCFS2 users walk metadata from it, but inline xattr metadata is still checked only in operation-specific consumers. The existing ibody lookup helper validates inline header placement and entry count, but inode block validation does not reject entry name/value bounds. Add a flat xattr entry validator and call it from inode block validation for inline xattrs. Keep the operation paths on their existing header/count lookup checks; the full entry bounds check now runs when the inode block is validated at read time. Reject corrupted inline xattr metadata before ocfs2_xattr_ibody_get() or listxattr() can walk past the inline storage. Validation reproduced this kernel report: BUG: KASAN: use-after-free in ocfs2_xattr_find_entry+0x5a/0x170 Read of size 2 at addr ffff8881242a2000 by task python3/529 Call Trace: dump_stack_lvl+0x66/0xa0 print_report+0xce/0x630 kasan_report+0xe0/0x110 ocfs2_xattr_find_entry+0x5a/0x170 ocfs2_xattr_get_nolock+0x20a/0x820 ocfs2_xattr_get+0x10c/0x1e0 __vfs_getxattr+0xe2/0x130 vfs_getxattr+0x185/0x1b0	7.8	More Details
CVE-2026-90003	In the Linux kernel, the following vulnerability has been resolved: futex: Prevent rcuwait use-after-free during requeue PI On PREEMPT_RT, FUTEX_CMP_REQUEUE_PI can trigger a KASAN report (slab-out-of-bounds) in futex_requeue_pi_complete() invocation of rcuwait_wake_up(). The futex_q used by futex_wait_requeue_pi() is allocated on the waiter's stack. An early wakeup can race with a PI requeue as follows: waiter requeue task ------ ------------ futex_wait_requeue_pi() futex_do_wait() schedule() futex_requeue futex_proxy_trylock_atomic() futex_requeue_pi_prepare() Q_REQUEUE_PI_NONE -> Q_REQUEUE_PI_IN_PROGRESS * timeout/ signal wakes waiter * futex_requeue_pi_wakeup_sync() Q_REQUEUE_PI_IN_PROGRESS -> Q_REQUEUE_PI_WAIT requeue_pi_wake_futex futex_requeue_pi_complete() cmpxchg Q_REQUEUE_PI_WAIT -> Q_REQUEUE_PI_LOCKED rcuwait_wait_event() if (atomic_read(&q->requeue_state) != Q_REQUEUE_PI_WAIT) break /* no schedule() */ /* q.pi_state->owner == current */ futex_private_hash_put() /* return from syscall */ rcuwait_wake_up(&q->requeue_wait) /* q is gone */ futex_requeue_pi_complete() publishes Q_REQUEUE_PI_LOCKED before calling rcuwait_wake_up(). The waiter observes this state in rcuwait_wait_event() before invoking schedule() in rcuwait_wait_event(). Here, the waiter is free leave the syscall before requeue task can complete the wake. To address this race skip rcuwait_wake_up() in the Q_REQUEUE_PI_LOCKED case. This state is only published by requeue_pi_wake_futex(), which saves q->task before futex_requeue_pi_complete() and wakes the waiter via wake_up_state(). This wake is intended to wake the waiter from its futex_do_wait() sleep. If the waiter is still sleeping there, it can not get into the Q_REQUEUE_PI_WAIT state (and require this removed wake). Should the waiter be woken up from futex_do_wait() by other means (as in this example) and sleep in futex_requeue_pi_wakeup_sync() then the wake_up_state() from requeue_pi_wake_futex() will wake it, too. Should the waiter task terminate before wake_up_state() had a chance to wake the task then the task pointer does not become invalid because the futex_hash_bucket::lock is held and the task pointer is RCU protected. [bigeasy: Updated comment and commit message]	7.8	More Details
CVE-2026-93045	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject arena frees below the arena base bpf_arena_free_pages() accepts scalar arena addresses. The runtime masks the address to the low 32 bits and reconstructs a full user address from the arena base before returning the range to the arena free tree. When the scalar value is below the low 32 bits of the arena base, full_uaddr falls below user_vm_start. The existing upper-end clipping then turns this into an out-of-range free-tree offset. A later allocation can reuse that offset and return an address below the arena mapping. Reject such frees before computing the clipped range.	7.8	More Details
CVE-2026-90047	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Don't hand out the flat CCS storage as usable VRAM get_flat_ccs_offset() reads the base of the flat CCS storage from the hardware, scales it by the number of enabled L3 nodes, and rounds the result up to 128K. Everything below that offset is then handed to the VRAM allocator as usable memory. Rounding a limit that means "usable memory ends here" upwards publishes whatever lies between the real base and the rounded one as free memory, and that memory belongs to the compression hardware. The scaled value has no reason to be 128K aligned, and on a Battlemage G21 with 16 GiB it is not: flat CCS base: raw 0x3fafff800, rounded 0x3fb000000 so the last 2 KiB of page 0x3fafff000 is CCS storage, in the allocator's pool. Whatever is allocated there gets that tail overwritten by the compression hardware, which needs no page-table entry, no buffer object and no GPU submission to do it, and does it before userspace exists. On this machine a Mesa VM's level-3 page table landed on that page on every cold boot. It lost the entry covering the compositor's batch-buffer heap, so the compositor's first submission faulted fetching its batch and gdm restarted it forever: a black screen on an otherwise working machine. Restarting gdm cleared it because the next VM's page tables were allocated somewhere else. Round down instead, to the page size the allocator works in. On this machine that excludes exactly one page. Reading the reserved page afterwards shows what had been writing it: [369] 0xcccc000000000000 [371] 0xcc77000000000000 [373] 0xcccc000000000000 [375] 0xcc77000000000000 compression metadata, two bytes per sixteen, sitting where the driver used to hand out memory. The assertion that should have caught this compares the offset against GSMBASE - ccs_size for equality. That value is 128K aligned, so it agrees with the rounded-up offset precisely when the base is not aligned - the check cannot fail in the case it exists to catch, and is compiled out unless CONFIG_DRM_XE_DEBUG is set. Replace it with one that can fail: CCS storage must not run into GSM. [And this was a debug session from hell, enormously	7.8	More Details

	helped by an AI doing much of the grunt-work. I'd like to call it my tireless helper, but the AI several times stated flat out that this was impossible and unsolvable and that we should just write a report about it. I suspect those things have been trained by people who may not be quite as stubborn as I am. But while the AI was ready to give up several times, it did keep adding debug code and analyzing it faithfully when I pushed. So credit where credit is due and I let the AI write the commit message above. This is basically a one-liner fixing a bogus "round_up()" to a "round_down()", but there were 24 patches adding more and more debug information to this, and 18 kernel boot to finally narrow it down to this. - Linus]		
CVE-2026-90325	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: skip dying blkg in blkcg_activate_policy() When switching IO schedulers on a block device, blkcg_activate_policy() can race with concurrent blkcg deletion, leading to a use-after-free in rcu_accelerate_cbs. T1: T2: blkg_destroy kill(&blkg->refcnt) // blkg->refcnt=1->0 blkg_release // call_rcu(__blkg_release) ... blkg_free_workfn ->pd_free_fn(pd) elv_iosched_store elevator_switch ... iterate blkg list blkg_get(blkg) // blkg->refcnt=0->1 list_del_init(&blkg->q_node) blkg_put(pinned_blkg) // blkg->refcnt=1->0 blkg_release // call_rcu again rcu_accelerate_cbs // uaf Fix this by checking hlist_unhashed(&blkg->blkcg_node) before getting a reference to the blkg. This is the same check used in blkg_destroy() to detect if a blkg has already been destroyed. If the blkg is already unhashed, skip processing it since it's being destroyed.	7.8	More Details
CVE-2026-90007	In the Linux kernel, the following vulnerability has been resolved: scsi: pm8001: Use rollback index when freeing MSI-X vectors pm8001_request_msix() unwinds previously registered handlers with free_irq() when request_irq() fails. The rollback loop uses the failing index i for every iteration instead of the already registered vector index j. That passes the wrong IRQ/dev_id pair to free_irq() and leaves the earlier handlers installed. Use j for both pci_irq_vector() and the matching irq_vector entry in the rollback loop.	7.8	More Details
CVE-2026-90008	In the Linux kernel, the following vulnerability has been resolved: scsi: megaraid_sas: Limit NVMe request size to the PRP chain frame megasas_make_prp_nvme() builds a command's PRP list in cmd->sg_frame, a DMA pool buffer of instance->max_chain_frame_sz bytes, spending one entry per NVMe page of the transfer plus one per page of the buffer for the chain pointer. The loop runs until the transfer is described and never checks the buffer bound. max_hw_sectors comes straight from the MDTS the firmware reports for the drive. On drives with a large MDTS the only thing keeping the list inside the buffer was the block layer default of 1280 KiB, which needs 320 entries, which fit into a 4 KiB frame as that holds 512. But since commit 9b8b84879d4a ("block: Increase BLK_DEF_MAX_SECTORS_CAP") that default is 4 MiB, and such a transfer needs 1025 entries, so the list runs a full page past the end of the frame: sd 1:0:1:0: [sdb] tag#630 page boundary ptr_sgl: 0x00000000ba62d13f BUG: unable to handle page fault for address: ff663bcb81e7c000 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page RIP: 0010:megasas_build_and_issue_cmd_fusion+0xeaa/0x1870 [megaraid_sas] If the page after the frame happens to be mapped, the overrun does not fault but silently corrupts the neighbouring pool entry, which is another in-flight command's PRP list. Cap max_hw_sectors at what the chain frame can describe, less one page for transfers that do not start on a page boundary and so need one entry more. This is the megaraid_sas counterpart of commit 04631f55afc5 ("scsi: mpt3sas: Limit NVMe request size to 2 MiB"), but derives the limit from max_chain_frame_sz rather than hardcoding it.	7.8	More Details
CVE-2026-90010	In the Linux kernel, the following vulnerability has been resolved: scsi: bsg: Cap io_uring sense copy to max_response_len Completion copied scmd->sense_len to the user response buffer without honoring max_response_len. After a valid sense, the midlayer sets sense_len to the real length (up to SCSI_SENSE_BUFFERSIZE), so a smaller user buffer was overrun.	7.8	More Details
CVE-2026-90027	In the Linux kernel, the following vulnerability has been resolved: usb: typec: qcom-pmic-typec: disable cc_debounce_dwork on stop cc_debounce_dwork is queued from the set_cc() and start_toggling() callbacks, which run from TCPM's kthread worker. port_stop() returns before tcpm_unregister_port() destroys that worker. Flushing the worker during unregister may therefore run a callback which queues the delayed work after port_stop() has returned. The delayed work can then run after devres has freed pmic_typec_port. Use disable_delayed_work_sync() in port_stop() to cancel a pending instance and prevent the TCPM callbacks from queueing another one. This issue was found by an in-house static analysis tool.	7.8	More Details
CVE-2026-90013	In the Linux kernel, the following vulnerability has been resolved: tracing: Take trace_array reference when opening options file The options files do not take the trace_array reference for the options they represent. This could cause a use-after-free kernel crash if one of these files is opened by one task and another task removes the instance that the option is for. Because it doesn't take a reference upon opening, it will not stop the removal which will free the options descriptor that is being used. As the options are somewhat dynamic in their creation at boot up, each file represents a flag in the trace_array. The trace_array has an array of indexes to represent each of these flags that is stored in the trace_flags_index array. The address of the index array element is used to pass to the inode->i_private pointer. Then that element is read which holds the index (which represents the flag) and then the index is used to calculate the trace_array descriptor from its trace_flags_index array. One issue is that the index element can not be referenced until the trace_array's reference is taken. To handle this, create a new helper function called: trace_array_options_get() that will iterate all the existing trace_arrays in the ftrace_trace_arrays list (under the trace_types_lock), and compare the passed in address of the index element with the entire array of the trace_array's trace_flags_index array. If it matches, then up the corresponding trace_array's reference and return.	7.8	More Details
CVE-2026-90014	In the Linux kernel, the following vulnerability has been resolved: tracing: Have show_event_filters/triggers files take trace array ref The newly added files show_event_filters and show_event_triggers that show all filters or triggers that are set within the trace array do not take a reference for the trace array it is showing. Without taking a reference, the trace_array may be freed via "rmdir" while a task is reading one of theses files. Those files iterate all the events within an instance (trace_array) and nothing prevents that instance from being freed while its data is being read. This causes a use-after-free crash. Have the open of both those files take the trace_array reference via the trace_array_get() that prevents the trace_array from being freed while the files are opened.	7.8	More Details
CVE-2026-93037	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Propagate sdma_txinit_ahg() errors set_txreq_header_ahg() ignores the return value of sdma_txinit_ahg(). If sdma_txinit_ahg() fails, it returns before initializing tx->txreq. However, set_txreq_header_ahg() ignores the error and returns the AHG change count, causing the caller to continue processing the request as though initialization had succeeded. Propagate sdma_txinit_ahg() failures to the caller and abort request processing when initialization fails. Found by Linux Verification Center (linuxtesting.org) with SVACE.	7.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate orphan slot during inode read Patch series "ocfs2: validate active orphan slots during inode read". OCFS2 trusts active ordinary and append-DIO orphan slots read from dinodes. A corrupted slot can therefore index osb_orphan_wipes or the slot-local system-inode cache outside their allocations before the corruption is reported. Patch 1 validates the ordinary orphan slot used by inode wipe processing. Patch 2 validates		

CVE-2026-90205	the append-DIO orphan slot used by DIO completion and orphan recovery. Both checks reject corrupt metadata at the existing inode validation boundary. This patch (of 2): [BUG] A corrupted dinode with OCFS2_ORPHANED_FL can carry an i_orphaned_slot outside the mounted filesystem slot range. ocfs2_wipe_inode() uses it to index osb_orphan_wipes before looking up the orphan directory, causing an out-of-bounds memory access. BUG: KASAN: slab-use-after-free in ocfs2_get_system_file_inode+0x780/0x820 fs/ocfs2/sysfile.c:102 Read of size 8 at addr ffff88800b767c00 by task kworker/u8:3/85 Call Trace: ... ocfs2_get_system_file_inode+0x780/0x820 fs/ocfs2/sysfile.c:102 ocfs2_wipe_inode+0x292/0xf70 fs/ocfs2/inode.c:840 ocfs2_delete_inode fs/ocfs2/inode.c:1155 [inline] ocfs2_evict_inode+0x6c9/0x1170 fs/ocfs2/inode.c:1295 evict+0x38e/0x8f0 fs/inode.c:810 iput_final fs/inode.c:1914 [inline] iput fs/inode.c:1966 [inline] iput+0x55b/0x8b0 fs/inode.c:1926 ocfs2_recover_orphans+0x610/0xe40 fs/ocfs2/journal.c:2374 ocfs2_complete_recovery+0x5af/0xd00 fs/ocfs2/journal.c:1373 ... [CAUSE] ocfs2_validate_inode_block() validates i_suballoc_slot but leaves the active ordinary orphan slot unchecked. Downstream consumers assume that the value is smaller than osb->max_slots. [FIX] Reject an active i_orphaned_slot outside the slot range during dinode validation, before the inode reaches orphan wipe processing.	7.8	More Details
CVE-2026-90326	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: fix race between policy activation and blkg destruction When switching an IO scheduler on a block device, blkcg_activate_policy() allocates blkg_policy_data (pd) for all blkgs attached to the queue. However, blkcg_activate_policy() may race with concurrent blkcg deletion, leading to use-after-free and memory leak issues. The use-after-free occurs in the following race: T1 (blkcg_activate_policy): - Successfully allocates pd for blkg1 (loop0->queue, blkcgA) - Fails to allocate pd for blkg2 (loop0->queue, blkcgB) - Enters the enomem rollback path to release blkg1 resources T2 (blkcg deletion): - blkcgA is deleted concurrently - blkg1 is freed via blkg_free_workfn() - blkg1->pd is freed T1 (continued): - Rollback path accesses blkg1->pd->online after pd is freed - Triggers use-after-free In addition, blkg_free_workfn() frees pd before removing the blkg from q->blkg_list. This allows blkcg_activate_policy() to allocate a new pd for a blkg that is being destroyed, leaving the newly allocated pd unreachable when the blkg is finally freed. Fix these races by extending blkcg_mutex coverage to serialize blkcg_activate_policy() rollback and blkg destruction, ensuring pd lifecycle is synchronized with blkg list visibility.	7.8	More Details
CVE-2026-90207	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: midi: Serialize input teardown with event_input snd_midi_input_event() must not be running while a rawmidi substream is closing, since this can lead to the trigger state becoming out-of-step through this sequence in snd_rawmidi_input_trigger(): snd_rawmidi_input_trigger(up=0) snd_midi_input_event() -> snd_rawmidi_kernel_read() -> snd_rawmidi_input_trigger(up=1) -> cancel_work_sync() which ends with the underlying device being active unexpectedly. When this is called from close_substream(), further input can re-trigger the input event leaving it running after rawmidi_release_priv() has set rfile->rmidi to NULL which leads to: Unable to handle kernel NULL pointer dereference at virtual address 00000000000000b0 Call trace: snd_midi_input_event+0x3c/0x134 [snd_seq_midi] (P) snd_rawmidi_input_event_work+0x1c/0x2c process_one_work+0x150/0x3a4 worker_thread+0x190/0x318 Apply a similar approach to commit ef7607ab1c8ad ("ALSA: seq: midi: Serialize output teardown with event_input") which fixed the same issue in the output direction, but updated to use RCU following Takashi Iwai's proposed follow-on patch [1]. With this change in place, midisynth_unsubscribe() clears the input file so snd_midi_input_event() will not re-trigger the stream and will be quiesced by the cancel_work_sync() in snd_rawmidi_input_trigger(). [1] https://lore.kernel.org/linux-sound/20260813144224.753399-1-tiwai@suse.de/	7.8	More Details
CVE-2026-90022	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_midi2: fix use-after-free in string attribute show path f_midi2_opts_str_show() takes the string lock internally, but its callers dereference the opts->info.<field> pointer before calling it, outside the lock. This races with f_midi2_opts_str_store(), which frees the old string under opts->lock when the attribute is written concurrently, the show path can read a pointer that gets freed before the lock inside str_show() is even taken. Change f_midi2_opts_str_show() to take a pointer to the string field, matching the existing pattern in f_midi2_opts_str_store(), and dereference it only after the lock is held. Update all three callers (iface_name, block name, and the EP string option macro) accordingly.	7.8	More Details
CVE-2026-92518	In the Linux kernel, the following vulnerability has been resolved: riscv, bpf: Fix kernel stack corruption in tailcall with CFI When CONFIG_CFI_CLANG is enabled, prog->bpf_func already skips the kcfi instruction during setup. Including it again in the tailcall jump offset causes it to jump over an extra 4 bytes, skipping the stack pointer adjustment, which will result in kernel stack corruption.	7.8	More Details
CVE-2026-92511	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix potential use after free in ib_destroy_cq_user() When accessing a CQ via the netlink path the only synchronization mechanism for the said CQ is rdma_restrack_get(). Currently, rdma_restrack_del() is invoked at the end of ib_destroy_cq_user(), which is too late, since by that point vendor-specific resources associated with the CQ might already be freed. This can leave a short window where the CQ remains accessible through restrack, leading to a potential use-after-free. Fix this by moving the rdma_restrack_begin_del() call to the start of ib_destroy_cq_user(), ensuring that the CQ is removed from restrack before its internal resources are released. This guarantees that no new users hold references to a CQ that is in the process of destruction. In addition, this change preserves the intended inverted order between create and destroy routines: resources are added to restrack at the end of successful creation, and hence shall be removed from the restrack first thing during the destruction flow, which keeps the lifecycle management consistent and predictable.	7.8	More Details
CVE-2026-90026	In the Linux kernel, the following vulnerability has been resolved: usb: typec: qcom-pmic: cancel reset_work on stop pdphy_stop() disables IRQs but leaves reset_work pending. If the IRQ handler schedules it just before disable_irq(), the work runs after remove() frees the struct via devm. Call cancel_work_sync() after disabling IRQs to close the window. This issue was found by an in-house static analysis tool.	7.8	More Details
CVE-2026-89965	In the Linux kernel, the following vulnerability has been resolved: nvdimm/btt: reject an arena whose nfree is below the lane count The BTT info block's nfree field, the number of reserve free blocks, is read from the medium without validation. btt_freelist_init() and btt_rtt_init() size the per-lane freelist[] and rtt[] arrays by nfree, but the I/O path indexes them by the lane from nd_region_acquire_lane(), which is bounded by nd_region->num_lanes (ND_MAX_LANES), not by nfree. A crafted or foreign arena whose nfree is below the lane count makes freelist[lane]/rtt[lane] run past the allocation: an out-of-bounds write. btt.rst documents the nlanes = min(nfree, num_cpus) invariant, which the code does not currently honor: num_lanes is ND_MAX_LANES regardless of nfree. Reject an arena whose nfree is below num_lanes at discovery, before the per-lane arrays are allocated, enforcing that invariant.	7.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: ip6mr: do not clone dst in ip6mr_cache_report() IPv6 input attaches a non-refcounted (NOREF) dst to skbs under RCU. When an ingress multicast packet misses MFC lookup, ip6mr_cache_unresolved() places the skb onto the unresolved queue, escaping the receive-side RCU grace period. If the		

CVE-2026-90111	underlying route is deleted and freed, and the MFC queue is later resolved with a wrong parent interface, ip6_mr_forward() invokes ip6mr_cache_report(..., MRT6MSG_WRONGMIF), which executes dst_clone(skb_dst(pkt)) on the freed dst entry, triggering a slab use-after-free. Report packets queued to mroute6_sk (a raw socket) and netlink notifications do not require an attached dst entry. Fix this by: 1. Removing dst_clone() in ip6mr_cache_report() and ensuring report skbs do not hold a dst. 2. Dropping skb_dst before queuing unresolved skbs in ip6mr_cache_unresolved(), matching the fact that multicast forwarding resolves outgoing routes anew via ip6_route_output().	7.8	More Details
CVE-2026-93137	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix use-after-free on mm_struct in bpf_find_vma() bpf_find_vma() reads task->mm and calls mmap_read_trylock(mm) without holding a reference on the mm. On a foreign task, a concurrent exit_mm() can free the mm_struct between the lockless read and the trylock, resulting in a use-after-free. mm_struct is not SLAB_TYPESAFE_BY_RCU. For the current task, task->mm is stable. For a foreign task, pin the mm under task->alloc_lock and release it with mmput_async(), mirroring commit d8e27d2d22b6 ("bpf: fix mm lifecycle in open-coded task_vma iterator"). Use spin_trylock() instead of get_task_mm() so BPF context does not block on alloc_lock. Reject irqs-disabled contexts and !CONFIG_MMU on the foreign-task path because dropping the mm reference is not safe there. Race: CPU0 (BPF program) CPU1 (exiting task) ========== ========== bpf_find_vma(foreign_task): mm = task->mm exit_mm(): task->mm = NULL mmput(mm) -> frees mm_struct mmap_read_trylock(mm) // UAF on mm	7.8	More Details
CVE-2026-89832	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to clear dirty flag on folio in error path If node block is corrupted due to chksum mismatch or inconsistent footer info, it needs to drop clear flag of node folio, in order to persist inconsistent node data to storage.	7.8	More Details
CVE-2026-93138	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix vmlinux BTF prep race in bpf_get_btf_vmlinux bpf_get_btf_vmlinux() lazily parses the vmlinux BTF under the bpf_verifier_lock, but publishes the result through a plain store and re-checks it through a plain lockless load. Nothing orders the stores initializing the struct btf inside btf_parse_vmlinux() against the store publishing the pointer: On a weakly ordered arch, a concurrent first-time caller taking the lockless fast path could in principle observe the pointer before the parsed contents are visible. The mutex_unlock() does not help such a reader given it only synchronizes with a later acquisition of the same lock. Thus, publish the pointer with smp_store_release() and read it on the fast path with smp_load_acquire(). Acquire semantics are needed rather than a dependency-ordered READ_ONCE(): btf_parse_vmlinux() also populates globals outside the returned object (e.g. bpf_ctx_convert.t). An address dependency would only order accesses performed through the pointer and not cover other globals.	7.8	More Details
CVE-2026-89814	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: clamp the isolation index for rings outside a partition adev->isolation[] has one slot per partition, but a ring that is not assigned to one keeps AMDGPU_XCP_NO_PARTITION, which is ~0, so indexing the array with it is out of bounds. SDMA submissions hit this on both the isolation enforcement and the VM flush path and trip UBSAN. Fall back to the first slot the way the cleaner shader path already does, and stop taking the address before the ring type check that makes it relevant.	7.8	More Details
CVE-2026-89815	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: Drop tt->restore after successful restore ttm_pool_restore_and_alloc() can successfully complete the restore process via ttm_pool_restore_commit(), but tt->restore is not dropped afterward. As a result, subsequent backup/restore flows observe what appears to be a completed restore, while in reality shmem handles are still installed in tt->pages, leading to the stack trace below. Fix this by freeing and dropping tt->restore in ttm_pool_restore_and_alloc() upon successful completion of the restore. 20545 [309.784531] RIP: 0010:sg_alloc_append_table_from_pages+0x38c/0x490 20547 [309.809570] RSP: 0018:ffffc9000623b838 EFLAGS: 00010206 20548 [309.814827] RAX: 0000000000001000 RBX: ffff88816e42a160 RCX: 0000000000000000 20549 [309.821986] RDX: 0000000000002000 RSI: 0000000000000003 RDI: 0000000000001000 20550 [309.829147] RBP: ffff88816e42a168 R08: 0000000000000002 R09: 000000007ffff000 20551 [309.836310] R10: ffffc9000623b928 R11: 0000000000000000 R12: 000000007ffff000 20552 [309.843471] R13: ffff88815ba5a100 R14: 0000000000000000 R15: 0000000000000001 20553 [309.850634] FS: 00007f9ff305e700(0000) GS:ffff888276c94000(0000) knlGS:0000000000000000 20554 [309.858749] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 20555 [309.864519] CR2: 00007f9fca701000 CR3: 00000001565e2005 CR4: 0000000008f70ef0 20556 [309.871678] PKRU: 55555558 20557 [309.874403] Call Trace: 20558 [309.876866] <TASK> 20559 [309.878988] sg_alloc_table_from_pages_segment+0x60/0x100 20560 [309.884415] ? ttm_resource_manager_usage+0x36/0x60 [ttm] 20561 [309.889845] ? xe_tt_map_sg+0x7d/0xd0 [xe] 20562 [309.894045] xe_tt_map_sg+0x7d/0xd0 [xe] 20563 [309.898037] xe_bo_move+0x927/0xaa0 [xe] 20564 [309.902029] ttm_bo_handle_move_mem+0xba/0x170 [ttm] 20565 [309.907022] ttm_bo_validate+0xbe/0x190 [ttm] 20566 [309.911405] xe_bo_validate+0x9a/0x120 [xe] 20567 [309.915663] xe_gpuvm_validate+0xd9/0x140 [xe] 20568 [309.920206] drm_gpuvm_validate+0x2f0/0x5b0 [drm_gpuvm] 20569 [309.925459] ? drm_exec_lock_obj+0x63/0x210 [drm_exec] 20570 [309.930627] xe_vm_validate_rebind+0x46/0xb0 [xe] 20571 [309.935428] xe_exec_fn+0x20/0x40 [xe] 20572 [309.939249] drm_gpuvm_exec_lock+0x78/0xc0 [drm_gpuvm] 20573 [309.944410] xe_validation_exec_lock+0x5a/0xa0 [xe] 20574 [309.949385] xe_exec_ioctl+0x806/0xc30 [xe] 20575 [309.953639] ? ttwu_queue_wakelist+0xd9/0xf0 20576 [309.957935] ? __pfx_xe_exec_fn+0x10/0x10 [xe] 20577 [309.962449] ? __wake_up_common+0x73/0xa0 20578 [309.966482] ? __pfx_xe_exec_ioctl+0x10/0x10 [xe] 20579 [309.971263] drm_ioctl_kernel+0xa3/0x100 20580 [309.975209] drm_ioctl+0x213/0x440 20581 [309.978637] ? __pfx_xe_exec_ioctl+0x10/0x10 [xe] 20582 [309.983415] xe_drm_ioctl+0x67/0xd0 [xe] 20583 [309.987408] __x64_sys_ioctl+0x7f/0xd0	7.8	More Details
CVE-2026-89819	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: validate plane degamma LUT size for private color prop Unlike the CRTC degamma path, which is guarded by amdgpu_dm_verify_lut_sizes(), the per-plane degamma LUT size was never validated before use. __set_dm_plane_degamma() passed the user-supplied size straight into __is_lut_linear() and, for a non-linear LUT, into __set_input_tf() -> __drm_lut_to_dc_gamma(), the latter always iterating MAX_COLOR_LUT_ENTRIES entries regardless of the actual LUT size. A malformed AMD_PLANE_DEGAMMA_LUT blob (e.g. a single entry) could thus trigger a divide-by-zero in __is_lut_linear() or an out-of-bounds read in __drm_lut_to_dc_gamma(). Reject any plane degamma LUT whose size does not match MAX_COLOR_LUT_ENTRIES, mirroring the invariant the code already asserts a few lines below (and which the CRTC path enforces). The AMD_PLANE_DEGAMMA_LUT property is only exposed on builds with AMD_PRIVATE_COLOR defined.	7.8	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: crypto: acomp - allocate async request context when cloning ACOMP_REQUEST_ON_STACK() reserves only enough storage for the synchronous fallback. When an async implementation is selected, callers clone that stack request before retrying, but acomp_request_clone() currently copies only the stack-sized object. The clone therefore has no storage for the async provider request context, and providers such as QAT write past the	7.8	More Details

90069	allocation through acomp_request_ctx(). KASAN does report a slab OOB write. Allocate a zeroed clone large enough for the runtime acomp request size, copy only the bytes present in the source object, and preserve the existing fallback-on-allocation-failure behavior. Use the runtime reqsize because an implementation may adjust it during tfm initialization.		
CVE-2026-89823	In the Linux kernel, the following vulnerability has been resolved: drm: fix race between partial drm_dev_register() failure and ioctl If drm_dev_register() fails after registering a minor (e.g. render minor registered, primary minor fails), userspace could have opened the first minor and entered a drm_dev_enter() critical section. Since the unplugged flag was never set, the ioctl proceeds while the error path tears down device resources. Fix this by introducing drm_dev_synchronize_unplug(), which sets the unplugged flag and waits for the SRCU barrier, ensuring all in-flight drm_dev_enter() critical sections complete before cleanup proceeds; call it on the error path of drm_dev_register().	7.8	More Details
CVE-2026-89825	In the Linux kernel, the following vulnerability has been resolved: drm/panthor: fix firmware control interface bounds checks panthor_init_cs_iface() and panthor_init_csg_iface() validate firmware control interface offsets with 32-bit arithmetic and the size of the host wrapper structures. The offsets are derived from firmware-provided strides, so the arithmetic can wrap before the bounds check, and the host wrapper size is not the size of the firmware control interface being mapped. Use 64-bit arithmetic for the computed offsets and validate against the actual firmware control interface structure sizes with subtraction-based bounds checks. Also validate that the shared section is large enough for the global control interface before using it.	7.8	More Details
CVE-2026-90071	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_teql: restore skb->dev on the slave failure path teql_master_xmit() sets skb->dev = slave before calling the slave's ndo_start_xmit(), but never restores it when that transmit fails. The skb then walks on to the next slave still pointing at the previous one. If a later slave has no resolved neighbour, teql_resolve() hands the skb to neigh_event_send(), which queues it on that neighbour's arp_queue with the stale skb->dev. skb->dev holds no reference, so deleting the previous slave frees the net_device while the skb is still queued. Whatever runs next on that skb - arp_error_report() on timeout, or neigh_direct_output() -> dev_queue_xmit() once the neighbour resolves - causes a UAF like the one below: BUG: KASAN: slab-use-after-free in __icmp_send (net/ipv4/icmp.c:914 (discriminator 2)) Read of size 4 at addr ffff888106e100b0 by task flood_packet/527 CPU: 0 UID: 0 PID: 527 Comm: flood_packet Not tainted 7.2.0-rc6-g594d90519502 #1 PREEMPT(lazy) Hardware name: QEMU Ubuntu 24.04 PC v2 (i440FX + PIIX, arch_caps fix, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Call Trace: <IRQ> dump_stack_lvl (lib/dump_stack.c:94 lib/dump_stack.c:120) print_report (mm/kasan/report.c:378 mm/kasan/report.c:482) ? __pfx__raw_spin_lock_irqsave (./include/asm-generic/qrwlock.h:122 (discriminator 4)) ? __icmp_send (net/ipv4/icmp.c:914 (discriminator 2)) kasan_report (mm/kasan/report.c:595) ? __icmp_send (net/ipv4/icmp.c:914 (discriminator 2)) __icmp_send (net/ipv4/icmp.c:914 (discriminator 2)) [...] ipv4_link_failure (net/ipv4/route.c:1251 net/ipv4/route.c:1258) ? __pfx_ipv4_link_failure (./include/linux/skbuff.h:4327) ? _raw_write_lock (./include/linux/instrumented.h:55 ./include/linux/atomic/atomic-instrumented.h:1301 ./include/asm-generic/qrwlock.h:98 ./include/linux/rwlock_api_smp.h:230 kernel/locking/spinlock.c:304) ? __pfx__raw_write_lock (kernel/locking/spinlock.c:175) arp_error_report (./include/net/dst.h:438 net/ipv4/arp.c:296) neigh_invalidate (net/core/neighbour.c:1077) neigh_timer_handler (net/core/neighbour.c:1169) [...] Allocated by task 505: kasan_save_stack (mm/kasan/common.c:57) kasan_save_track (mm/kasan/common.c:78) __kasan_kmalloc (mm/kasan/common.c:398 mm/kasan/common.c:415) __kvmalloc_node_noprof (./include/linux/kasan.h:263 mm/slub.c:5334 mm/slub.c:6905) alloc_netdev_mqs (net/core/dev.c:12055 (discriminator 2)) rtnl_create_link (net/core/rtnetlink.c:3721) rtnl_newlink (net/core/rtnetlink.c:3903 net/core/rtnetlink.c:4044 net/core/rtnetlink.c:4159) rtnetlink_rcv_msg (net/core/rtnetlink.c:7076) [...] Freed by task 536: kasan_save_stack (mm/kasan/common.c:57) kasan_save_track (mm/kasan/common.c:78) kasan_save_free_info (mm/kasan/generic.c:584) __kasan_slab_free (mm/kasan/common.c:253 mm/kasan/common.c:285) kfree (./include/linux/kasan.h:235 mm/slub.c:2677 mm/slub.c:6377 mm/slub.c:6692) device_release (drivers/base/core.c:2636) kobject_put (lib/kobject.c:689 lib/kobject.c:720 ./include/linux/kref.h:65 lib/kobject.c:737) netdev_run_todo (net/core/dev.c:11756) rtnl_dellink (net/core/rtnetlink.c:157 ./include/linux/rtnetlink.h:135 net/core/rtnetlink.c:3651) rtnetlink_rcv_msg (net/core/rtnetlink.c:7076) [...] Fix this by restoring skb->dev to the master at the end of each slave's iteration.	7.8	More Details
CVE-2026-89829	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to pass folio->index to f2fs_sanity_check_node_footer() Otherwise in f2fs_sanity_check_node_footer(), it will check the same nid incorrectly.	7.8	More Details
CVE-2026-89836	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix folio_nr_pages() race after put in large folio invalidate Our v6.18 based Android system is continuely suffering livelock and bad page stat as shown in[1] which related to broken xarray slot status. By investigating big folio operations within f2fs, we find below races and fix it by get the nr_pages before drop the refcount and folio_lock. f2fs_get_read_data_folio() calls f2fs_folio_put() before folio_nr_pages() when invalidating a large folio from the page cache. That unlocks the folio and drops the caller reference, leaving a window where a concurrent truncate or folio split can shrink the compound folio or free it before the invalidate range is computed. An undersized range then leaves split sub-folios in mapping->i_pages, which can later interact badly with truncate and reclaim (stale xarray entries and bad page state when folio->mapping no longer matches the mapping being truncated). [1] PID: 2594 TASK: ffffff8169b81580 CPU: 7 COMMAND: "Thread-3" #0 [ffffffc08ef2b8a0] xas_load at ffffffe52d1f42a4 #1 [ffffffc08ef2b900] find_get_entries at ffffffe52c185798 #2 [ffffffc08ef2bb60] truncate_inode_pages_range at ffffffe52c19e83c #3 [ffffffc08ef2bbc0] truncate_inode_pages_final at ffffffe52c19ec2c #4 [ffffffc08ef2bc20] f2fs_evict_inode at ffffffe52c4c8400 #5 [ffffffc08ef2bcc0] evict at ffffffe52c2de9f4 #6 [ffffffc08ef2bd00] iput at ffffffe52c2db1b4 #7 [ffffffc08ef2bd30] dentry_unlink_inode at ffffffe52c2d7204 #8 [ffffffc08ef2bd50] __dentry_kill at ffffffe52c2d3dcc #9 [ffffffc08ef2bd80] dput at ffffffe52c2d3c3c #10 [ffffffc08ef2bda0] __fput at ffffffe52c2b0a7c #11 [ffffffc08ef2bde0] ____fput at ffffffe52c2b1034 #12 [ffffffc08ef2bdf0] task_work_run at ffffffe52beea200 #13 [ffffffc08ef2be20] exit_to_user_mode_loop at ffffffe52bfbc17c #14 [ffffffc08ef2be80] el0_svc at ffffffe52d1f8e54 #15 [ffffffc08ef2beb0] el0t_64_sync_handler at ffffffe52d1f8d10	7.8	More Details
CVE-2026-89880	In the Linux kernel, the following vulnerability has been resolved: media: rtl2832_sdr: release URBs and stream buffers on start_streaming() failure rtl2832_sdr_start_streaming() calls rtl2832_sdr_alloc_stream_bufs(), rtl2832_sdr_alloc_urbs() and rtl2832_sdr_submit_urbs() in sequence and shares a single err: label that only unlocks the mutex and returns. When alloc_urbs() succeeds but submit_urbs() fails, or when alloc_urbs() itself returns -ENOMEM after alloc_stream_bufs() has already succeeded, the URBs and/or the coherent DMA stream buffers stay allocated while streaming reports failure to vb2. Two latent defects follow on the next VIDIOC_STREAMON: 1) rtl2832_sdr_alloc_stream_bufs() unconditionally resets dev->buf_num to 0 and overwrites dev->buf_list[]/dev->dma_addr[], permanently leaking the coherent DMA memory allocated by the previous attempt. 2) rtl2832_sdr_alloc_urbs() never resets dev->urbs_initialized and only increments it. After a second successful pass urbs_initialized can exceed MAX_BULK_BUFS, so the subsequent rtl2832_sdr_free_urbs() walks from urbs_initialized - 1 down to 0 and reads past the end of dev->urb_list[], passing garbage pointers to usb_free_urb(). Mirror the teardown that	7.8	More Details

	stop_streaming() already performs: on the error path call rtl2832_sdr_free_urbs() and rtl2832_sdr_free_stream_bufs() before unlocking. Both helpers are idempotent (free_urbs kills and zeros urbs_initialized; free_stream_bufs is gated on URB_BUF and clears the buf_num counter), so partial-failure paths and the no-allocation paths remain safe. Issue identified by automated review of the INV-003 series at https://sashiko.dev/		
CVE-2026-89841	In the Linux kernel, the following vulnerability has been resolved: f2fs: only redirty pinned folios in redirty_blocks redirty_blocks() pins folios with read_cache_folio() and then walks the same range again with filemap_lock_folio() to redirty them and drop the references it took. Commit 5951fee46bef ("f2fs: Use a folio in redirty_blocks()") changed the second pass to a do/while loop. If read_cache_folio() fails before anything is pinned, page_idx does not advance but the cleanup loop still runs once. If readahead has already populated the failed folio in page cache, that extra iteration finds it and folio_put_refs(folio, 2) drops one reference too many. Later drop_caches or reclaim can then report "BUG: Bad page state". Only redirty the range that was pinned successfully.	7.8	More Details
CVE-2026-89854	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix cs84xx use-after-free on host teardown qla84xx_put_chip() drops the last reference to ha->cs84xx and frees it via __qla84xx_chip_release() without clearing ha->cs84xx. During teardown it ran before scsi_remove_host(), which is what removes the 84xx_fw_version host sysfs attribute. A concurrent read of that attribute in the window between the two calls executes qla24xx_84xx_fw_version_show(), which dereferences the freed ha->cs84xx, resulting in a use-after-free. Move qla84xx_put_chip() to after scsi_remove_host() in both qla2x00_remove_one() and qla2x00_disable_board_on_pci_error(). Once scsi_remove_host() returns, the sysfs attribute is gone and kernfs has drained any in-flight show(), so no reader can touch cs84xx; the put still runs before the host and ha are freed.	7.8	More Details
CVE-2026-93337	NetworkManager-l2tp contains an improper input validation vulnerability that allows local users with VPN connection creation permissions to inject arbitrary pppd directives by supplying mru or mtu property values containing trailing non-numeric content after a valid integer. Attackers can exploit the verbatim write of unvalidated strings into the pppd options file via write_config_option() to inject the plugin directive, causing the privileged pppd process to load an attacker-controlled shared object and achieve arbitrary code execution as root.	7.8	More Details
CVE-2026-90093	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: access chan->conn safely in get/setsockopt Since commit b66774b48dd9 ("Bluetooth: L2CAP: Fix UAF in channel timeout by holding conn ref") l2cap_chan::conn has held reference and remains non-NULL also after the corresponding hci_conn is deleted. In this state accessing various fields eg. hci_conn::hdev is invalid, which leads to KASAN crash in l2cap_sock_setsockopt() access of conn->hcon->hdev. Check l2cap_chan::conn.hcon corresponds to an alive hci_conn before trying to use it in l2cap_sock.c. Hold l2cap_chan_lock() in getsockopt/setsockopt to ensure it stays alive, and to avoid data races in l2cap_chan fields.	7.8	More Details
CVE-2026-89870	In the Linux kernel, the following vulnerability has been resolved: media: zoran: Avoid freeing a registered video_device twice zoran_init_video_device() installs zoran_vdev_release() as the video_device release callback through zoran_template. After video_register_device() succeeds, video_unregister_device() drops the registered video_device reference and the V4L2 core eventually invokes that release callback, which kfree()s the video_device. zoran_exit_video_devices() called video_unregister_device() and then kfree(zr->video_dev), so device teardown could free the same video_device twice. Remove the direct kfree() and clear the cached pointer after unregistering. The pre-registration failure path keeps its manual free because the video_device was not registered there. This issue was found by a static analysis checker and confirmed by manual source review.	7.8	More Details
CVE-2026-54692	SAIL is a cross-platform library for loading and saving images with support for animation, metadata, and ICC profiles. Prior to 1.0.0, sail_codec_load_frame_v8_xbm() in src/sail-codecs/xbm/xbm.c allocates the decoded pixel buffer using the X11 one-byte-per-literal layout, but an X10 static short file causes the flat decode loop to write two file-controlled bytes per literal. When ceil(width/8) produces an odd row stride, the X10 literal count includes a padding byte for every row, but the destination has no space for those bytes, so loading the XBM through sail_load_from_file, sail_load_from_memory, or sail_start_loading_* produces a forward heap overwrite that scales with image height. The X11 static char path is not affected. The overwrite can corrupt process state, cause reliable crashes, and potentially enable code execution in a susceptible consuming application. This issue is fixed in version 1.0.0.	7.8	More Details
CVE-2026-89873	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-ctrls: validate HEVC EXT SPS RPS counts The HEVC SPS control carries the short-term and long-term RPS counts that decoder drivers use to walk the matching EXT SPS dynamic arrays. Reject SPS values that exceed the HEVC limits of 64 short-term sets and 32 long-term references so drivers cannot later index beyond those controls. Also reject EXT SPS ST RPS entries whose negative or positive picture counts exceed the 16-entry arrays, or whose combined delta-POC count exceeds the HEVC DPB maximum.	7.8	More Details
CVE-2026-90292	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix use-after-free in siw_accept() siw_accept() looks up the QP supplied by userspace. If that QP is already in RTS, the function jumps to error cleanup before associating the incoming CEP with it. The cleanup tests whether qp->cep is non-NULL and assumes the current call installed the association. However, qp->cep can point to the CEP of an existing connection. The cleanup then drops a reference from the incoming cep, not qp->cep. Once the incoming endpoint loses its remaining references, this can free it before the subsequent cep->qp store, causing a use-after-free. It also clears the existing QP association. Only release the association reference when qp->cep is the incoming CEP. This preserves an existing association and avoids accessing the freed endpoint.	7.8	More Details
CVE-2026-89875	In the Linux kernel, the following vulnerability has been resolved: media: ti: vpe: quiesce overflow recovery before freeing streams The VIP overflow recovery worker is armed from the hardirq handler when a FIFO overflow is detected, and the list-complete path looks the stream up through the VPDMA list private pointer. Both keep touching stream, port and device state; the recovery worker also resets the parser and VPDMA, repopulates the descriptor list, and re-enables the per-list IRQs. vip_stop_streaming() masks and clears the per-list IRQs, but it neither synchronizes the hardirq handler nor disables recovery_work. An overflow IRQ that has already queued recovery_work, or a list-complete IRQ in flight when the stream is torn down, can therefore still dereference the stream after its resources are released: the descriptor list is freed by vip_release_stream() on file release, and the stream itself by free_stream() on unbind/remove. Drain the recovery worker and the IRQ handler at both teardown points through a shared vip_quiesce_stream() helper, before any stream-owned resource is released. disable_work_sync() cancels pending recovery_work, drains a running instance, and raises its disable depth, so a subsequent schedule_work() issued by a racing IRQ handler is rejected at the workqueue scheduler: recovery_work cannot be requeued after disable_work_sync() takes effect. The worker may still re-enable the per-list IRQs before disable_work_sync() returns; disable_irqs() then masks those sources and synchronize_irq() waits for any in-flight handler that still dereferences stream state. In vip_stop_streaming() the helper runs before the parser is stopped, since a worker drained by disable_work_sync() may re-enable the parser before exiting and would otherwise undo the stop. recovery_work is created	7.8	More Details

	disabled and enabled in vip_start_streaming() before IRQs, pairing the enable with the teardown disable across the streaming lifecycle. This issue was found by an in-house static analysis tool and confirmed by manual code review.		
CVE-2026-89808	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix the case that vm range is hole at svm_migrate_copy_to_vram When migration vm range is hole at cpu side(MIGRATE_PFN_MIGRATE set + MIGRATE_PFN_VALID unset) driver still allocates device pages. There is no dma map of src pages and migration. j is 0 and svm_migrate_copy_memory_gart() will return an uninitialized r. That can trigger out_free_vram_pages to drop all VRAM just set up. Initialize r and only call the last svm_migrate_copy_memory_gart if j > 0. Current code postponed the last page to the final copy. This patch flushes on the last page when reach to the end of current drm_buddy_block; avoids another svm_migrate_copy_memory_gart.	7.8	More Details
CVE-2026-89805	In the Linux kernel, the following vulnerability has been resolved: drm/pagemap: Fix folio allocation fallback and use-after-put drm_pagemap_migrate_populate_ram_pfn() had two issues when populating RAM PFNs with higher-order folios: 1. The higher-order vma_alloc_folio()/folio_alloc() calls did not pass __GFP_NOWARN, so a THP allocation failure under memory pressure would spam the kernel log, and there was no fallback path despite a TODO comment stating one was needed. Add __GFP_NOWARN to the higher-order allocation and, on failure, fall back to order-0 allocations for the entire range originally covered by the failed higher-order allocation, leaving MIGRATE_PFN_COMPOUND unset for those PFNs. 2. In the free_pages error path, order was computed via folio_order(page_folio(page)) *after* put_page(page) had already dropped the reference, resulting in a use-after-free/put when that was the last reference on the page. Compute order before releasing the page. Introducing the fallback in 1. also requires the source page array handed to ->copy_to_ram() to be built differently. Both callers only populated the entry at the head of each source folio, relying on the copy callback to derive the rest of the folio from the order recorded in the matching drm_pagemap_addr. Once the destination has been demoted to order-0 folios the drm_pagemap_addr entries are per-page, so a source page is needed for every one of them; leaving them NULL makes the copy callback stop after the first page and the remainder of the range is never copied. The source folio is only split later, by migrate_vma_pages() / migrate_device_pages(), so its order cannot be used to detect the demotion - test the destination for MIGRATE_PFN_COMPOUND instead. Factor the array population out into drm_pagemap_migrate_populate_src_pages() and use it from both drm_pagemap_evict_to_ram() and __drm_pagemap_migrate_to_ram().	7.8	More Details
CVE-2026-89803	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: unsubscribe the channel-kill event before the fence context nouveau_channel_del() tears the fence context down first and only drops the channel-kill subscription later, in the middle of the nvif object teardown: if (chan->fence) nouveau_fence(chan->cli->drm)->context_del(chan); ... nvif_object_dtor(&chan->vram); nvif_event_dtor(&chan->kill); The subscribed handler is nouveau_channel_killed(), which calls nouveau_channel_kill() and from there nouveau_fence_context_kill() on chan->fence. A kill event delivered in that window takes fctx->lock and walks fctx->pending on a fence context that context_del() has already freed. Nothing reaches this below Fermi today, because the subscription is gated on FERMI_CHANNEL_GPFIFO and nothing kills a channel there. On Fermi and newer the window is real but narrow, since a kill has to land exactly while the channel is being destroyed. That is reason enough on its own, which is why this carries a Fixes: tag. The last patch in this series subscribes Tesla channels as well; nothing kills those today, so it does not widen the exposure now, but it is the groundwork for a recovery path that would, and the ordering is better fixed before that lands than alongside it. Drop the subscription before anything it depends on is torn down.	7.8	More Details
CVE-2026-89801	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/uvmm: fix premature region free on failed OP_UNMAP_SPARSE In nouveau_uvmm_bind_job_submit()'s OP_UNMAP_SPARSE arm, op->reg is set from nouveau_uvma_region_find(), which only looks the region up and takes no reference; a region's sole reference is its membership in uvmm->region_mt. Two failure paths leave op->reg set: the -ENOENT check when the region is busy, and the drm_gpuvm_sm_unmap_ops_create() failure. The sibling nouveau_uvmm_sm_unmap_prepare() failure just below clears op->reg; these two do not. unwind_continue steps back one op, so the failing op is skipped by the unwind loop and its op->reg stays set. nouveau_uvmm_bind_job_cleanup() then enters its if (op->reg) branch and calls nouveau_uvma_region_remove() and nouveau_uvma_region_put() on it, dropping the tree's sole reference and freeing a region this job never created. The comment above the cleanup loop documents the broken invariant: op->reg must be NULL on submit failure. This frees a live region on an unrelated failure, reachable single-job when drm_gpuvm_sm_unmap_ops_create() returns -ENOMEM; if another job owns the same region, its cleanup then removes and puts the freed region, a use-after-free. Clear op->reg on both failure paths.	7.8	More Details
CVE-2026-81474	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Heap-based Buffer Overflow vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of privileges.	7.8	More Details
CVE-2026-86320	A flaw was found in flatpak-builder where Git hooks are not disabled when applying patch sources with use-git-am: true. An attacker who can provide a malicious source containing a Git post-applypatch hook can cause the hook to execute on the host during the build process, resulting in arbitrary code execution with the privileges of the user running flatpak-builder.	7.8	More Details
CVE-2026-25290	Memory Corruption when validating large data buffers from external sources using addition to check buffer length.	7.8	More Details
CVE-2026-25280	Memory corruption when processing escape handling flow with insufficient user buffer sizes.	7.8	More Details
CVE-2026-25278	Memory Corruption when processing I2C transfer requests due to a race condition between memory allocation and data copying.	7.8	More Details
CVE-2026-24075	Memory Corruption when multiple threads issue concurrent IOCTL requests to the device control handler due to improper synchronization and race conditions.	7.8	More Details
CVE-2026-24074	Memory Corruption when processing data with large offset and length values exceeds buffer limits during data copy operations.	7.8	More Details
CVE-			More

2026-24073	Memory corruption when processing decode statistics due to insufficient validation of offset against structure size.	7.8	Details
CVE-2025-59607	Memory Corruption when copying large input data exceeds normal allocation limits.	7.8	More Details
CVE-2026-92838	A DLL hijacking vulnerability exists in the GeoVision GV-Remote E-Map desktop application. The application loads one or more dynamic-link libraries (DLLs) from an unsafe search path, allowing a local attacker to place a malicious DLL in a location searched before the legitimate library location. If successfully exploited, an attacker with local write access to the affected directory could achieve arbitrary code execution in the security context of the GV-Remote E-Map process.	7.8	More Details
CVE-2026-92816	ComfyUI before 0.30.0 fails to sanitize folder_name input in dataset save nodes, allowing attackers to write files to arbitrary paths outside the output directory. Attackers can load a crafted workflow that writes attacker-controlled content to arbitrary locations, enabling code execution through modified startup files or package initializers.	7.8	More Details
CVE-2026-92786	LightGBM through 4.7.0 fails to validate child and split array values when parsing text models, allowing attackers to write out-of-bounds memory during SHAP prediction. Attackers can craft malicious model files with invalid node references that trigger out-of-bounds writes at attacker-chosen offsets in the leaf_depth_ buffer during feature contribution computation.	7.8	More Details
CVE-2026-89789	In the Linux kernel, the following vulnerability has been resolved: gtp: add synchronize_net() in gtp_newlink() error path to prevent use-after-free gtp_newlink()'s error path frees tid_hash and addr_hash without waiting for an RCU grace period after clearing sk_user_data. A concurrent gtp_encap_recv() in softirq may still hold the gtp_dev pointer obtained via rcu_dereference_sk_user_data() and access the freed memory. BUG: KASAN: slab-use-after-free in gtp0_pdp_find+0x1f6/0x200 (gtp.c:152) Call Trace: <IRQ> gtp0_pdp_find+0x1f6/0x200 gtp_encap_recv+0x527/0x24b0 udp_queue_rcv_one_skb+0x75f/0xc10 Add synchronize_net() before the kfree calls in out_hashtable, which covers all error paths from both gtp_encap_enable() and gtp_create_sockets().	7.8	More Details
CVE-2026-89791	In the Linux kernel, the following vulnerability has been resolved: perf: Fix use-after-free when perf mmap() revival races with the last munmap() perf_mmap_close() drops rb->mmap_count *without* holding event->mmap_mutex (the refcount_dec_and_test() right before the refcount_dec_and_mutex_lock() of event->mmap_count). A concurrent perf_mmap_rb() can slot its entire "revival" path into that window (perf_mmap holds event->mmap_mutex for its whole duration, including rb_alloc): munmap side (perf_mmap_close) mmap side (perf_mmap_rb) ---------- ---------- rb->mmap_count 1 -> 0 (no lock) (holds event->mmap_mutex) inc_not_zero(rb->mmap_count) fails ring_buffer_attach(event, NULL) rb_alloc() + attach new rb refcount_set(&event->mmap_count, 1) lock; event->mmap_count 1 -> 0 ring_buffer_attach(event, NULL) ring_buffer_put() -> frees the *new* rb The revival's refcount_set(&event->mmap_count, 1) is an invisible 1 -> 1 write: the close frees the just-revived buffer although the other process still has it mapped -- a page-level use-after-free allowing local privilege escalation to root by any unprivileged user (default kernel.perf_event_paranoid=2). Swap the order of the two counter updates: event->mmap_count is dropped first via refcount_dec_and_mutex_lock(), so its 1 -> 0 transition and the ring_buffer_attach() stay serialized with perf_mmap(). rb->mmap_count == 0 then implies every event using the buffer is detached already, so the result of the rb->mmap_count drop can gate the remaining teardown directly and detach_rest is no longer needed. An earlier fix for this race from Kyle Zeng and David Lee takes event->mmap_mutex around both counter updates [0]; here the not-last close stays lockless.	7.8	More Details
CVE-2026-90051	In the Linux kernel, the following vulnerability has been resolved: tcp: reject non zerocopy devmem tx Devmem tcp tx doesn't work without zero-copy, however it's not currently enforced if NETIF_F_SG isn't present. In this case, tcp_sendmsg_locked() will try the copy path and try to copy data from an iovec which consists of offsets into the dma-buf and would normally fail. Moreover, d9c56501c72fd ("net: tcp: block mixing readable and unreadable frags") relies on that and assumes that the devmem binding is present IFF we're using the zero-copy path, which can be used to mix net-iov and pages in a single skb, and break invariants. Let's reject devmem tx without zero-copy. Note, the parameter check the patch is modifying is too loose, we can create an io_uring request with dmabuf_id and all ZC flags, but which won't have the binding. We replace it with stricter validation.	7.8	More Details
CVE-2026-89793	In the Linux kernel, the following vulnerability has been resolved: ublk: clear VM_MAYWRITE on read-only ublk char device mmap ublk_ch_mmap() rejects mmap requests with VM_WRITE set, but never clears VM_MAYWRITE on the resulting read-only mapping. This allows a userspace daemon to mmap the per-queue command buffer PROT_READ, then upgrade it to PROT_WRITE via mprotect(), since VM_MAYWRITE was never cleared. The command buffer holds struct ublksrv_io_desc entries that are kernel-written ABI; a writable mapping lets an unprivileged daemon process corrupt fields such as addr, op_flags, nr_sectors, and start_sector. Same bug class as the drm/panthor and drm/vc4 VM_MAYWRITE fixes, and the 2026-08-13 ptp/vmclock fix (a5edadbae57e). Verified via mprotect() PoC: before the fix, a PROT_READ mapping can be upgraded to PROT_READ|PROT_WRITE and a write into the command buffer corrupts io_desc fields (confirmed under KASAN). After the fix, mprotect() returns -EACCES.	7.8	More Details
CVE-2026-90052	In the Linux kernel, the following vulnerability has been resolved: dm-integrity: fix buffer overflow with keyed discard Since commit 68c5c42567bc ("dm-integrity: replace forgeable discard filler with a keyed sector marker"), integrity_metadata computes a checksum for every discarded block into the "checksums" buffer. integrity_sector_checksum always writes the whole digest. So if the tag size is smaller than the digest size, the checksum of the last block that fits into the buffer is written past the end of it. For example, with hmac(sha256) and tag size 16, a 4MiB discard writes 16 bytes past the kmalloc'ed page. Fix this by subtracting extra_space from the buffer size when computing max_blocks, like we do for writes.	7.8	More Details
CVE-2026-90057	In the Linux kernel, the following vulnerability has been resolved: slip: remove slip_hangup() to fix use-after-free in slip_receive_buf() Jaeyoung Chung and Eulgyu Kim reported a slab-use-after-free read in slip_receive_buf() when racing against tty hangup. tty_ldisc_hangup() calls ld->ops->hangup() while holding only a read lock on tty->ldisc_sem (via tty_ldisc_ref()). Because slip_hangup() simply called slip_close(), it ran concurrently with reader functions such as slip_receive_buf(). slip_close() unregisters and frees the net device and its private struct slip, causing concurrent reader threads in slip_receive_buf() to dereference freed memory. Line discipline close() is already guaranteed to be called under the write lock of tty->ldisc_sem during hangup processing (in tty_ldisc_reinit() or tty_ldisc_kill()). Remove slip_hangup() so teardown is serialized cleanly by slip_close().	7.8	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: bpf: Disable preemption in bpf_get_stackid The get_perf_callchain call needs disabled preemption plus we need it disabled as long as we access its returned trace entries	7.8	More Details

89799	buffer. Note the bpf_get_stackid_pe function is executed already with preemption disabled.		
CVE-2026-90118	In the Linux kernel, the following vulnerability has been resolved: ntfs: fix off-by-one page overflow in ntfs_decompress() The per-token range check in ntfs_decompress() uses if (cb >= cb_sb_end || dp_addr > dp_sb_end) break; so dp_addr == dp_sb_end falls through to the symbol copy `*dp_addr++ = *cb++`, writing one byte past the destination page. Since NTFS_SB_SIZE == PAGE_SIZE the destination is a single page, so the byte lands in the adjacent page, and *dest_ofs is left one past the sub-block end (the later `*dest_ofs &= ~PAGE_MASK` then yields 1, not 0, so the page is never finalized and later sub-blocks keep writing further past it). A corrupted compressed $DATA attribute thus produces a bounded run of out-of-bounds writes when the file is read. Break as soon as dp_addr reaches dp_sb_end; a full sub-block still completes, as its final copy advances dp_addr to exactly dp_sb_end.	7.8	More Details
CVE-2026-89810	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix error path at svm_migrate_copy_to_ram If page migration from device to sys ram fails for some reasons driver needs release and unlock allocated system pages. To do that driver should use page physical address, or pfn, then get struct page*. Current driver uses dma address(for adev) that is not correct with IOMMU enabled, or even in general. The patch releases and unlocks allocated system pages based on where migration failed by struct page* of sys ram pages. Also dma_unmap correspodent system ram pages at error path.	7.8	More Details
CVE-2026-90143	In the Linux kernel, the following vulnerability has been resolved: net: kcm: Hold RCU read lock while running BPF parser kcm_parse_func_strparser() calls bpf_prog_run_pin_on_cpu() which prevents CPU migration, but does not establish an RCU read-side critical section. Consequently, BPF map operations can trigger WARN_ON_ONCE(!bpf_rcu_lock_held()) when called from the KCM strparser program. Hold the RCU read lock while running the program.	7.8	More Details
CVE-2026-89920	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Fix memory corruption by not reinjecting CK machine checks Channel-subsystem damage machine checks are for the host channel subsystem. The guest channel subsystem is emulated in the userspace VMM. There is no point in forwarding such machine checks into the guest. This also simplifies the machine check reinjection and avoids kfree of a stack variable as reported by sashiko. There might be still machine checks that have the ck bit set with another bit (like instruction damage), mask out the CK bit in s390_backup_mcck_info(), like the CP and ED bits already are.	7.8	More Details
CVE-2026-89903	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Do not save/restore percpu base register in rethook trampoline The rethook trampoline saves $r21 ($u0), the percpu base, into its frame at entry and restores it at exit. Inbetween rethook_trampoline_handler() may schedule via preempt_enable_notrace(). If the task migrates to another CPU, the frame's $r21 holds the old CPU's percpu base, and restoring it poisons $r21 on the new CPU. Until the next user->kernel transition heals $r21, all this_cpu_*() accesses (runqueues, RCU per-CPU data, timer tick programming, FPU ownership) hit the wrong CPU's percpu area. Under kretprobe-heavy preemptible load this can corrupt scheduler and timer state: scheduling-while-atomic splats, wrong-CPU RCU warnings, WARN_ON_ONCE(rq != this_rq()) in nohz_balance_exit_idle(), and CPUs parking in the idle loop with the constant timer never re-armed (hard lockup). Reproduces on a Loongson-3A6000 with kretprobes on VFS paths plus heavy file churn (OS install / unsquashfs). By convention $r21 always holds the current CPU's percpu base in kernel mode: SAVE_SOME() at exception entry reloads it only when coming from user mode, and RESTORE_SOME() restores it only when returning to user mode; the context-switch path never writes it. Therefore the live $r21 at trampoline exit is already correct, and nothing inbetween can change it legitimately (kernel C code cannot write a global register variable). The same flaw existed even in the pre-rethook kretprobe trampoline since v6.3; it was carried over when rethook replaced it. Drop both the save and the restore here. Drop the restore is enough to solve the issue, and drop the save is to keep the code tidy and no need to clear it.	7.8	More Details
CVE-2026-89938	In the Linux kernel, the following vulnerability has been resolved: iio: chemical: atlas-sensor: use iio_trigger_poll_nested() to fix remove UAF The atlas driver requests its hardware data-ready IRQ with devm_request_threaded_irq(); its threaded handler queues an irq_work, atlas_work_handler(), that calls iio_trigger_poll(data->trig). The IRQ is devm-managed, so free_irq() runs from the devres unwind after atlas_remove() returns without flushing that irq_work. Once a buffer is enabled, conversion-complete IRQs keep firing and queueing it; a pending irq_work can therefore run after the unwind has freed atlas_data/indio_dev and the trigger, when atlas_work_handler() derives the atlas_data pointer via container_of() and dereferences data->trig, a use-after-free. Call iio_trigger_poll_nested() directly from the threaded handler instead of bouncing through irq_work. free_irq() then drains the threaded handler, closing the window; other iio drivers with a threaded data-ready IRQ do the same (e.g. bmi270). This issue was found by an in-house static analysis tool.	7.8	More Details
CVE-2026-89906	In the Linux kernel, the following vulnerability has been resolved: LoongArch: BPF: Refactor jump offset calculation in tail call The old macro-based jmp_offset calculation derives the jump distance from a stale prior-pass code stride, which can lead to wrong branch offsets and soft lockups under extra JIT passes. Fix this by calculating the offset directly on the absolute target: "ctx->offset[insn + 1] - ctx->idx". To avoid a false 16-bit range check abort during size estimation, add a "ctx->image == NULL" guard to inject a safe dummy offset.	7.8	More Details
CVE-2026-89961	In the Linux kernel, the following vulnerability has been resolved: powerpc/mm: fix wrong addr_pfn tracking in compound vmemmap population vmemmap_populate_compound_pages() uses addr_pfn to determine the PFN offset within a compound page and to decide whether the current vmemmap slot should be populated as a head page mapping or should reuse a tail page mapping. However, addr_pfn is advanced manually in parallel with addr. The loop itself progresses in vmemmap address space, so each PAGE_SIZE step in addr covers PAGE_SIZE / sizeof(struct page) struct page slots. Since addr_pfn is compared against nr_pages in data-PFN units, it should advance by the same number of PFNs. The existing manual increments do not match that and therefore do not reliably track the PFN corresponding to the current addr. As a result, pfn_offset can be computed from the wrong PFN and the code can make the head/tail decision for the wrong compound-page position. Fix this by deriving addr_pfn directly from the current vmemmap address instead of carrying it as loop state.	7.8	More Details
CVE-2026-93144	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject writes through untrusted BTF pointers check_ptr_to_btf_access() lets program-type btf_struct_access callbacks validate writes before the default BTF access path rejects non-read accesses. That bypasses the read-only policy for untrusted BTF pointers created by helpers such as bpf_rdonly_cast(). Reject non-read accesses through PTR_UNTRUSTED BTF pointers at the common entry point, before the callback branch to handle all cases.	7.8	More Details
CVE-2026-93147	In the Linux kernel, the following vulnerability has been resolved: s390/bpf: Replace ly instruction with llgf cpu_nr is a 32 bit value and BPF_REG_0 is a 64 bit register, when ly loads the cpu_nr into BPF_REG_0 it does not zero the upper bits, but llgf does.	7.8	More Details

CVE-2026-93148	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject MEM_ALLOC BTF accesses past object bounds BTF struct walks relax the struct-size check for accesses through a trailing flexible array. That is valid for ordinary BTF type walking, but PTR_TO_BTF_ID | MEM_ALLOC values point to objects allocated with the static BTF type size. When walking a MEM_ALLOC object, reject the access before applying the flexible-array relaxation if the access range extends past the struct size. Apply the same policy to struct ID matching so kfunc and kptr type checks do not walk past the allocated object bounds either.	7.8	More Details
CVE-2026-89919	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: keyop: use mmu_lock to read gmap->asce Every other dat_* consumer in this file (kvm_s390_get_skeys, set_skeys, get_cmma_bits, set_cmma_bits, MEM_CLR_CMMA, kvm_s390_fixup_prefix, kvm_test_age_gfn, kvm_age_gfn) reads kvm->arch.gmap->asce *inside* the mmu_lock read-side. keyop is the only outlier. gmap->asce is mutated under write_lock(mmu_lock) by gmap_set_limit() and keyop might use a stale asce value for walking as KVM_S390_KEYOP and KVM_S390_VM_MEM_LIMIT_SIZE can run concurrently. This can result in memory corruption.	7.8	More Details
CVE-2026-93154	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Add refcounting to user ring MRs Prevent userspace from deregistering the MRs that back QP/CQ/SRQ rings by bumping the MR's refcount upon association.	7.8	More Details
CVE-2026-19477	There is stack-based buffer overflow vulnerability recently discovered in MCC Universal Library for Linux (uldaq). This may result in information disclosure or arbitrary code execution. This vulnerability affects MCC Universal Library for Linux (uldaq) v1.2.1 and prior versions.	7.8	More Details
CVE-2026-90146	In the Linux kernel, the following vulnerability has been resolved: bpf, xdp: move offload check into dev_xdp_install() bpf_xdp_link_update() calls dev_xdp_install() directly and skips dev_xdp_attach(), so the checks in dev_xdp_attach() do not run. A user can make an XDP link with a normal program and then swap in an offloaded or device-bound program with BPF_LINK_UPDATE, which puts it on the software path. dev_xdp_install() is the one place all three paths go through: "ip link set xdp" and BPF_LINK_CREATE reach it via dev_xdp_attach(), and BPF_LINK_UPDATE calls it directly. So move the program checks (offloaded, bound to another device, device-bound in generic mode, native vs generic, DEVMAP and CPUMAP) there, and keep only the netlink-flag check (XDP_FLAGS_UPDATE_IF_NOEXIST) in dev_xdp_attach().	7.8	More Details
CVE-2026-89922	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Take srcu when importing watchpoint data __import_wp_info() backs up the original guest memory contents of a watchpoint with read_guest_abs(), which is kvm_read_guest() and therefore resolves the memslot via __kvm_memslots(). That requires kvm->srcu (or kvm->slots_lock) to be held, otherwise a concurrent memslot update can free the memslots array under us once its SRCU grace period has elapsed. As this is not fast path, following lock ordering (mutex first, then srcu) take the big hammer and hold the srcu for the full import.	7.8	More Details
CVE-2026-93170	In the Linux kernel, the following vulnerability has been resolved: dmaengine: xilinx_dma: Fix channel idle state management in AXIDMA and MCDMA interrupt handlers Fix a race condition in AXIDMA and MCDMA irq handlers where the channel could be incorrectly marked as idle and attempt spurious transfers when descriptors are still being processed. The issue occurs when: 1. Multiple descriptors are queued and active. 2. An interrupt fires after completing some descriptors. 3. xilinx_dma_complete_descriptor() moves completed descriptors to done_list. 4. Channel is marked idle and start_transfer() is called even though active_list still contains unprocessed descriptors. 5. This leads to premature transfer attempts and potential descriptor corruption or missed completions. Only mark the channel as idle and start new transfers when the active list is actually empty, ensuring proper channel state management and avoiding spurious transfer attempts.	7.8	More Details
CVE-2026-93175	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix dangling pointer in CRTC reset function amdgpu_dm_crtc_reset_state() frees the old state before allocating a new one. If kzalloc() fails, the function returns without updating the state pointer, leaving a dangling pointer to already freed memory. Fix this by allocating the new state first. On allocation failure, the old state remains untouched and the function safely returns. Found by Linux Verification Center (linuxtesting.org) with SVACE. [adjust for movement around current amd-staging-drm-next]	7.8	More Details
CVE-2026-89942	In the Linux kernel, the following vulnerability has been resolved: iio: buffer: Fix potential use-after-free in anonymous buffer release An anonymous buffer handle holds a reference to the underlying IIO device. The reference is dropped in the buffer handle's release function. If the device has been removed, either through unbind or hot-unplug, the buffer handle might hold the last reference. The release function takes the mutex for the buffer using a guard, which means the unlock happens after all the code in the function, including `iio_device_put()`. If the anonymous buffer holds the last reference this might free both the IIO device and the buffer, which contains the mutex, leading to use-after-free when the mutex is unlocked. Fix this by using a scoped guard just around the buffer dmabuf list access, making sure the mutex is unlocked before releasing the IIO device. Version 10 of the patch that introduced this issue used this exact scheme of first unlocking and then dropping the reference [1]. During review it was suggested to use a guard instead, and version 11 made that change [2].	7.8	More Details
CVE-2026-89941	In the Linux kernel, the following vulnerability has been resolved: iio: buffer: Make IIO DMA fence release RCU-safe The `dma_fence` documentation states that if a custom release implementation is provided, the `dma_fence` object must be freed in an RCU-safe way. The current `iio_dma_fence` implementation uses `kfree()`, which might result in a use-after-free. Remove the custom `release` implementation. This makes the DMA fence core fall back to `dma_fence_free()`, which calls `kfree_rcu()` on the fence. This requires that the fence be the first member of `struct iio_dma_fence`. Using the default release method for extended DMA fence structures is a common pattern.	7.8	More Details
CVE-2026-89940	In the Linux kernel, the following vulnerability has been resolved: iio: buffer: Tie IIO dma fence lock lifetime to the fence The `iio_dma_fence` implementation currently uses a lock embedded in the `iio_dmabuf_priv`. But the `iio_dma_fence` can outlive the `iio_dmabuf_priv`, which can cause a use-after-free. Tie the lifetime of the lock to the lifetime of the fence by embedding them in the same struct. We can't just hold a reference to the `iio_dmabuf_priv` from the `iio_dma_fence` since `iio_buffer_dmabuf_release()` might sleep and the fence release callback is not allowed to sleep. Note that the `dma_fence` framework now has an internal lock that gets used when the passing `NULL` for `lock` in `dma_fence_init()`, but in order to allow this patch to be backportable use an external lock.	7.8	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: dm-pcache: validate seg_id fields from persistent memory cache_pos_decode(), cache_key_decode() and the last-kset branches of cache_replay(), the writeback worker and the GC worker take a cache segment id from the cache device metadata and index cache->segments[] with it without checking it against cache->n_segs. That metadata is only CRC-protected with a fixed public seed, so whoever supplies the cache device on a table load (CAP_SYS_ADMIN) controls the id; an out-of-range value forms a wild pcache_cache_segment pointer that is		More

2026-93201	dereferenced and written through -- an out-of-bounds read and write driven by on-disk data. Add cache_seg_id_valid() and reject an out-of-range id at each decode site, failing the operation with -EIO instead of indexing past the array. Bound the id against the initialized-segment count (cache_info.n_segs) rather than the physical device total. A forged cache_info.n_segs below seg_num otherwise leaves segments[cache_info.n_segs..seg_num) as zeroed structs whose data pointer is NULL, so a forged id in that window would still be dereferenced. A later patch guarantees cache_info.n_segs <= seg_num, and a driver-created cache sets the two equal, so valid images are unaffected.	7.8	Details
CVE-2026-89902	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Avoid preempt count underflow without probe LoongArch uses break 11 for the breakpoint placed after an instruction that Kprobes executes out of line. Since userspace can issue the same break instruction, do_bp() can reach kprobe_singlestep_handler() when there is no current probe. The handler actually returns false in this case, but it first calls preempt_enable_no_resched(). The corresponding preempt_disable() is done by kprobe_breakpoint_handler() on a real Kprobe hit, so it has not run here. As a result, an ordinary userspace breakpoint (code 11) underflows the current task's preempt count. This also makes in_interrupt() return true until the task schedules. One visible consequence is the socket cgroup attribution: cgroup_sk_alloc() treats the allocation as interrupt context and assigns the socket to the root cgroup. A socket opened from the SIGTRAP handler can then avoid a BPF_CGROUP_INET_SOCK_CREATE policy attached to the task's own cgroup. Return as soon as kprobe_running() reports no active probe. The same check has appeared in [PATCH v10 2/4] of the original LoongArch Kprobes series, but was dropped before the feature reached mainline.	7.8	More Details
CVE-2026-93192	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Clear queue->active_job when v3d_fence_create() fails The run_job() callbacks for BIN, RENDER, TFU and CSD assign the incoming job to queue->active_job before calling v3d_fence_create(). If v3d_fence_create() fails, the callback returns NULL without clearing active_job, leaving a dangling pointer. Create a failure path in all run_job() callbacks that clears the active job before returning NULL. The BIN path takes queue->queue_lock around the clear as it races against v3d_overflow_mem_work(); RENDER, TFU and CSD paths have no concurrent reader, so the clear is lock-free.	7.8	More Details
CVE-2026-89887	In the Linux kernel, the following vulnerability has been resolved: media: i2c: ov7740: fix use-after-destroy in remove The ov7740_remove() function had a severe teardown order bug where it destroyed the driver's mutex before freeing the V4L2 control handler which relies on that mutex, leading to a use-after-destroy kernel panic. Furthermore, the driver explicitly called v4l2_ctrl_handler_free() and mutex_destroy() sequentially, but then called ov7740_free_controls() which invokes both of them a second time, resulting in a double-free. This patch fixes the issue by unregistering the subdevice first, and relying exclusively on ov7740_free_controls() to safely tear down the mutex and control handler in the correct order.	7.8	More Details
CVE-2026-89890	In the Linux kernel, the following vulnerability has been resolved: media: go7007: defer the ALSA v4l2 put until card release go7007_snd_init() already takes a v4l2_device reference for the ALSA side, but go7007_snd_remove() drops it immediately after calling snd_card_free_when_closed(). That is too early when a userspace process still has the capture PCM open. The ALSA card and its PCM callbacks remain alive until the last file is closed, so the release path can still reach struct go7007 through pcm->private_data and call go7007_snd_hw_free() after the V4L2 release path has freed the object. Move the matching v4l2_device_put() to the ALSA card private_free callback so the existing ALSA reference covers the whole deferred card lifetime.	7.8	More Details
CVE-2026-89899	In the Linux kernel, the following vulnerability has been resolved: media: cec: disable delayed work before freeing an interrupted transmit cec_transmit_msg_fh() drops adap->lock to wait for a blocking transmit in wait_for_completion_killable(). If that wait is interrupted by a signal, cancel_delayed_work_sync() can run before the CEC kthread arms the reply timeout via schedule_delayed_work(&data->work) in cec_transmit_done_ts(). The work is then armed after the cancel, and the data is freed with its delayed_work still pending: ODEBUG: free active (active state 0) object: ... hint: cec_wait_timeout Use disable_delayed_work_sync(): it cancels the work and disables it, so the later schedule_delayed_work() becomes a no-op and the work cannot be re-armed. The data is freed right after, so it need not be re-enabled.	7.8	More Details
CVE-2026-90142	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Fix resize of the RX ring When a AF_XDP socket is attached, the virtnet_rx_resize should resize the rq->xsk_buffs XSK buffer array. Otherwise, when the size grows, the virtnet_rx_resume() causes a write past the end of the array. This is easily reproducable with ethtool -G ens3 rx 32 ./xdpsock -i eth0 -q 0 -r -z & ethtool -G eth0 rx 256	7.8	More Details
CVE-2026-89882	In the Linux kernel, the following vulnerability has been resolved: media: rkvdec: hevc: guard INTER_REF_PIC_SET_PRED index underflow st_ref_pic_set_prediction() computes the reference RPS index as st_rps_idx - (delta_idx_minus1 + 1) per HEVC spec equation 7-59. Both operands are u8, so when delta_idx_minus1 + 1 exceeds the current index the subtraction wraps and the subsequent array access at calculated_rps_st_sets[ref_rps_idx] reads far out of bounds. A userspace V4L2 client that can open the RKVDEC m2m decoder can submit an EXT_SPS_ST_RPS control with INTER_REF_PIC_SET_PRED set and delta_idx_minus1 crafted to trigger the underflow. Reject the entry early when the reference index would underflow.	7.8	More Details
CVE-2026-89894	In the Linux kernel, the following vulnerability has been resolved: media: cx231xx: reject geometry changes while the VBI queue is busy vidioc_s_fmt_vid_cap() and vidioc_s_std() change the device-wide dev->width / dev->norm but only refuse the change when the *video* queue (dev->vidq) is busy. The VBI queue (dev->vbiq) shares that same geometry: cx231xx_init_vbi_isoc() latches dma_q->lines_per_field from dev->norm, the VBI videobuf2 plane is sized from dev->width / dev->norm in vbi_queue_setup() and vbi_buf_prepare(), and cx231xx_do_vbi_copy() then recomputes the destination offset from the *live* dev->width and the latched lines_per_field on every URB completion: offset = lines_completed * (dev->width << 1) + ...; if (dma_q->current_field == 2) offset += dev->width * 2 * dma_q->lines_per_field; memcpy(plane + offset, p_buffer, lencopy); Because the VBI node shares video_ioctl_ops with the video node, an application can size a small VBI plane (REQBUFS/QBUF with a small width, or with the NTSC standard), then enlarge dev->width (or switch dev->norm to PAL) through the video node while the VBI stream is running -- the change is allowed because only dev->vidq is checked -- and let the device deliver a field-2 VBI payload. cx231xx_do_vbi_copy() now computes the offset with the larger geometry and memcpy()s past the end of the smaller plane that was already allocated, a heap out-of-bounds write whose offset is attacker-chosen and whose contents come from the device. The per-field guard in cx231xx_copy_vbi_line() does not help: it bounds the copy against the latched lines_per_field, not the plane's real capacity, and vb2 does not re-run buf_prepare() for an already prepared buffer. Refuse the format/standard change when the VBI queue is busy as well, so the geometry cannot change underneath an allocated VBI buffer.	7.8	More Details
	In the Linux kernel, the following vulnerability has been resolved: ntfs: Fix index_root heap OOB write in ntfs_ir_to_ib() ntfs_ir_to_ib copies all entries from index_root into a freshly allocated index_block_size-byte buffer without verifying that the entries fit in the available space. The entries in index_root may be larger than the usable entry space in the index block. This can cause OOB writes past the end of the allocation. The validator ntfs_index_root_inconsistent() checks that entries are self-consistent within the IR value, but never cross-checks them against index_block_size. There is no bounds check in ntfs_ir_to_ib()		

CVE-2026-90133	before the memcpy. Fixing this at the sink in ntfs_ir_to_ib() since ntfs_index_root_inconsistent() validates the logical consistency of index_root as a structure and a root with large entries is a structurally valid root. The bug is a size conflict of ntfs_ir_to_ib(). Also, the validator is called once per inode load in ntfs_read_locked_inode() while ntfs_ir_to_ib() is only called during a reparent, a check there adds no overhead to the common path. Moreover, even a future call path that bypasses the validator would still be protected. With NULL as first parameter of ntfs_error(), the volume error flag is never set by this call, so the device name will be absent from the error message. In any case, that the caller, ntfs_ir_reparent(), prints an error message that includes the device name on NULL returns. I think this is the best solution available without adding 'struct super_block *sb' as a parameter to ntfs_ir_to_ib(). This heap out-of-bounds write is triggered by a crafted filesystem image, which is not in the kernel threat model, anyway, fixing memory errors would be nice to keep things secure.	7.8	More Details
CVE-2026-89893	In the Linux kernel, the following vulnerability has been resolved: media: cx23885: cancel NetUP CI work before teardown netup_ci_exit() frees a netup_ci_state while its work item, netup_read_ci_status(), may still be pending or running on the system workqueue. The worker obtains the state with container_of() and dereferences it, so it must not outlive the state. netup_ci_init() queues the initial status read, and CI GPIO interrupts subsequently queue the same work from netup_ci_slot_status(). During remove, cx23885_finidev() calls free_irq() before the CI device is unregistered. free_irq() prevents further IRQ handlers from running, but does not drain work queued previously, so the worker can run after netup_ci_exit() frees the state. Call cancel_work_sync() before dvb_ca_en50221_release() and kfree(). This issue was found by an in-house static analysis tool.	7.8	More Details
CVE-2026-89888	In the Linux kernel, the following vulnerability has been resolved: media: i2c: ov02a10: fix endpoint parsing use-after-free The ov02a10_check_hwcfg() function calls fwnode_handle_put(ep) immediately after allocating and parsing the endpoint. However, it subsequently calls fwnode_property_read_u32() using the same 'ep' handle, leading to a potential use-after-free. Additionally, reading the optional 'ovti,mipi-clock-voltage' property used to overwrite the 'ret' variable. If the property was missing, 'ret' would become negative, and this failure code would be incorrectly returned at the end of the function, causing probe to fail entirely. Fix the use-after-free by moving fwnode_property_read_u32() before the endpoint is parsed and freed. Avoid the error leak by not assigning the result of fwnode_property_read_u32() to 'ret'.	7.8	More Details
CVE-2026-89883	In the Linux kernel, the following vulnerability has been resolved: media: rc: sunxi-cir: Unregister rc device on probe failure After rc_register_device() succeeds, later probe failures must undo the registration with rc_unregister_device(). The current error path jumps to the allocation cleanup label and only calls rc_free_device(), leaving the rc device registration and resources created by rc_register_device() behind. Add a registered-device unwind label for the IRQ lookup, IRQ request, and hardware initialization failure paths. Keep rc_free_device() for failures before rc_register_device() succeeds.	7.8	More Details
CVE-2026-81546	The Affinity by Canva application before 3.3.0 (September 2026 release) did not perform adequate bounds checking when parsing Affinity document files leading to a stack-based buffer overflow. A threat actor could craft a Affinity document that when opened by a user in Affinity could result in arbitrary code execution.	7.7	More Details
CVE-2026-20342	A vulnerability in a specific file download API of Cisco Secure FMC Software could allow an authenticated, remote attacker to download arbitrary files from an affected system. This vulnerability exists because user input is not being sanitized. An attacker could exploit this vulnerability by sending a crafted HTTPS request. A successful exploit could allow the attacker to download arbitrary files from the affected system. To exploit this vulnerability, the attacker must have valid credentials for a user account with at least the role of Security Analyst (read-only).	7.7	More Details
CVE-2026-18912	ManageEngine DataSecurity Plus versions before 6310 are vulnerable to an authenticated SQL injection vulnerability, allowing an authenticated technician to execute arbitrary SQL queries through the Reports module.	7.7	More Details
CVE-2026-85887	Incorrect permission assignment for critical resource in M365 Copilot allows an authorized attacker to disclose information over a network.	7.7	More Details
CVE-2026-90137	In the Linux kernel, the following vulnerability has been resolved: platform/x86: hp-bioscfg: fix password encoding bounds check The password PSWD_ENCODINGS parser reads password_obj[elem + pos_values] while copying the supported password encodings from the ACPI package. The outer loop only guarantees that elem is within password_obj_count. The encoding count is bounded by MAX_ENCODINGS_SIZE, but that does not guarantee that the ACPI package contains enough entries for all elem + pos_values accesses. A malformed package can therefore declare a non-zero encoding count without providing enough string objects, causing the parser to read past the ACPI package array and pass an out-of-bounds string pointer and length to hp_convert_hexstr_to_str(). Add the same computed-index bounds check used by the other offset-based package parsing loops before reading password_obj[elem + pos_values].	7.7	More Details
CVE-2026-50158	yutu is an AI-powered toolkit for managing and growing YouTube channels. Prior to 0.10.9, the caption-download MCP tool accepts a caller-controlled file parameter through cmd/caption/download.go and passes it to Caption.Download() in pkg/caption/caption.go, where os.Create() creates or truncates that path without using the pkg.Root confinement boundary backed by YUTU_ROOT. A principal able to invoke caption-download, including a local HTTP client when the MCP server runs with its default authentication-disabled configuration, can write downloaded caption bytes to any path writable by the yutu process outside YUTU_ROOT. This can overwrite application files, configuration, shell startup files, logs, or data and can cause persistent code execution or denial of service depending on the selected writable target. Live caption retrieval also requires usable service credentials and an accessible caption identifier. This issue is fixed in version 0.10.9.	7.7	More Details
CVE-2026-90341	In the Linux kernel, the following vulnerability has been resolved: firmware: coreboot: Validate table bounds The existing coreboot_table_populate() bounds checks limit individual entries to the mapped length. However, coreboot_table_probe() replaces the platform resource length with header and table sizes supplied by firmware before mapping the full table. A malformed table can overflow the 32-bit size addition or advertise an extent beyond the resource, causing the driver to map and parse memory outside the resource. A resource shorter than the fixed header is also mapped as though it contained a complete header. Reject resources shorter than the fixed header. After validating the signature, require a complete header, calculate the advertised extent with overflow checking, and reject extents beyond the resource before remapping the table.	7.7	More Details
CVE-2026-90025	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: displayport: Fix OOB altmode array index The UCSI displayport driver indexes the connector's port altmode array with the GET_CURRENT_CAM response after checking it is not 0xff. The port altmode array is UCSI_MAX_ALTMODES elements long. If the PPM returns an invalid GET_CURRENT_CAM response above UCSI_MAX_ALTMODES and not equal to 0xff, the kernel may crash with an array index OOB error. Update the	7.7	More Details

	UCSI displayport driver to verify the current cam is less than UCSI_MAX_ALTMODES before accessing the port altmode array.		
CVE-2026-93165	In the Linux kernel, the following vulnerability has been resolved: platform/chrome: sensorhub: Fix memory overread in ring handler `max_response` and `sensor_num` are read from different EC commands: - `max_response` is from cros_ec_get_proto_info(). ec_dev->max_response = info->max_response_packet_size - sizeof(struct ec_host_response); - `sensor_num` is from cros_ec_get_sensor_count(). sensor_num = cros_ec_get_sensor_count(ec); With a malfunctioning EC firmware, it is possible that the `msg->insize` (i.e., `fifo_info_length` in the context) could be clamped in cros_ec_cmd_xfer() because `msg->insize` is greater than `max_response`. int fifo_info_length = sizeof(struct ec_response_motion_sense_fifo_info) + sizeof(u16) * sensorhub->sensor_num; This means the number of read bytes could be less than expected. As a result, the subsequent memcpy() in cros_ec_sensorhub_ring_handler() overreads the `resp->fifo_info` buffer. Check the return value of cros_ec_cmd_xfer_status() and abort if the number of bytes read does not match the expected length.	7.7	More Details
CVE-2026-90407	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix overreads in ath11k_wmi_process_csa_switch_count_event() There is no policy entry for WMI_TAG_PDEV_CSA_SWITCH_COUNT_STATUS_EVENT, so the parse infrastructure does not enforce a minimum length for the event struct. Additionally, the num_vdevs field is taken directly from firmware and used as a loop bound over the vdev_ids array without checking that it fits within the TLV payload. Either condition can cause an out-of-bounds read. Add a TLV policy entry for WMI_TAG_PDEV_CSA_SWITCH_COUNT_STATUS_EVENT so the parse infrastructure enforces a minimum length for the fixed-size event struct. Add a helper ath11k_wmi_tlv_data_len() to recover the payload length of a parsed TLV from the header preceding its data pointer. Use it in ath11k_wmi_process_csa_switch_count_event() to bound num_vdevs before the loop. Compile tested only.	7.7	More Details
CVE-2026-90408	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix overreads in ath12k_wmi_process_csa_switch_count_event() There is no policy entry for WMI_TAG_PDEV_CSA_SWITCH_COUNT_STATUS_EVENT, so the parse infrastructure does not enforce a minimum length for the event struct. Additionally, the num_vdevs field is taken directly from firmware and used as a loop bound over the vdev_ids array without checking that it fits within the TLV payload. Either condition can cause an out-of-bounds read. Add a TLV policy entry for WMI_TAG_PDEV_CSA_SWITCH_COUNT_STATUS_EVENT so the parse infrastructure enforces a minimum length for the fixed-size event struct. Add a helper ath12k_wmi_tlv_data_len() to recover the payload length of a parsed TLV from the header preceding its data pointer. Use it in ath12k_wmi_process_csa_switch_count_event() to bound num_vdevs before the loop. Compile tested only.	7.7	More Details
CVE-2026-54339	Glean is a self-hosted RSS reader and personal knowledge management tool. Prior to 0.2.6, POST /api/feeds/discover passes an attacker-supplied feed_url to discover_feed(feed_url), creates a subscription through FeedService.create_subscription(), and enqueues fetch_feed_task. The background path calls fetch_feed(feed.url) and parse_feed(), which assigns each RSS item link to ParsedEntry.url. The task then passes ParsedEntry.url to fetch_and_extract_fulltext(parsed_entry.url) without network-level validation in backend/packages/rss/glean_rss/extractor.py and backend/apps/worker/glean_worker/tasks/feed_fetcher.py. A malicious feed can therefore make the server request private, loopback, link-local, or cloud-metadata resources. The fetched response is stored in Entry.content and can be retrieved through GET /api/entries/{id}, producing non-blind server-side request forgery with full response disclosure. This can bypass network perimeters, probe internal services and ports, expose internal configuration or web content, and potentially disclose cloud metadata access tokens. This issue is fixed in version 0.2.6.	7.7	More Details
CVE-2026-61595	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, `djust.tenants` isolation was enforced only on the HTTP path. The current tenant was stored in `threading.local()` and set exclusively by the HTTP-only `TenantMiddleware`, so on the live (WebSocket/SSE) path `get_current_tenant()` was always `None` during mount and every event handler — and the tenant-aware `QuerySet` manager failed OPEN (returned the unfiltered queryset, ignoring `STRICT_MODE`), disclosing every tenant's rows to whoever held the socket. `threading.local` was additionally shared across connections on the `sync_to_async` executor thread. This issue is fixed in djust 1.0.7. Tenant storage moved to a `contextvars.ContextVar` (per async task); the resolved tenant is bound around WS/SSE mount and every dispatch; both managers scope the base queryset once and fail CLOSED (`.none()` under the default `STRICT_MODE`); and system check S006 warns when `STRICT_MODE=False`. No known workarounds are available on the live path.	7.7	More Details
CVE-2026-66618	Administrator SQL Injection in WP Maps <= 4.9.9 versions.	7.6	More Details
CVE-2026-66626	Editor SQL Injection in SKT Addons for Elementor <= 4.0 versions.	7.6	More Details
CVE-2026-66625	Administrator SQL Injection in WC Vendors Marketplace <= 2.7.2.1 versions.	7.6	More Details
CVE-2026-66628	Shop manager SQL Injection in WP-Lister Lite for eBay <= 3.8.11 versions.	7.6	More Details
CVE-2026-66624	Administrator SQL Injection in WPMasterToolKit <= 2.22.0 versions.	7.6	More Details
CVE-2026-66619	Administrator SQL Injection in Newsletters <= 4.18 versions.	7.6	More Details
CVE-2026-66630	Administrator SQL Injection in PublishPress Series <= 3.1.3 versions.	7.6	More Details
CVE-	Builder.io Gen2 SDKs through versions 5.2.11 and 0.25.13 contain a prototype pollution vulnerability in the deep-set helper function that processes content block bindings without validation. Attackers can craft content blocks with binding keys		More

2026-92779	containing __proto__, prototype, or constructor paths to pollute Object.prototype during rendering, affecting all subsequent objects created in the process including other tenants' renders.	7.6	Details
CVE-2026-66631	Administrator SQL Injection in MC Woocommerce Wishlist <= 1.9.21 versions.	7.6	More Details
CVE-2026-76425	A vulnerability in the APIs of Cisco ISE could allow an authenticated, remote attacker to conduct SQL injection attacks against the backend database. This vulnerability is due to insufficient validation of certain parameters that are concatenated directly into an SQL query. An attacker could exploit this vulnerability by sending a crafted request that contains SQL statements to an affected endpoint. A successful exploit could allow the attacker to read arbitrary content from the SQL database and conduct server-side request forgery (SSRF) attacks. To exploit this vulnerability, the attacker must have valid administrative credentials.	7.6	More Details
CVE-2025-56565	DD-WRT firmware, as deployed on TP-Link TL-WR740N v1 through v4 hardware, stores sensitive authentication credentials in cleartext within non-volatile memory. The exposed material includes SSH private keys, dynamic DNS passwords, email notification credentials and administrative passwords. An attacker with physical access to the device can extract these credentials from an SPI flash dump, leading to device compromise, infiltration of the connected network and unauthorised access to dependent third-party services.	7.6	More Details
CVE-2026-45726	Omni manages Kubernetes on bare metal, virtual machines, or in a cloud. From 1.3.0 until 1.6.6 and 1.7.3, importing a standalone Talos cluster creates an ImportedClusterSecrets resource containing the cluster's complete CA secrets bundle. The access rules in internal/backend/runtime/omni/state_access.go allow an authenticated user with the Reader role to retrieve the resource through ResourceService if the importing actor has not rotated those secrets, exposing Kubernetes, Talos, and etcd CA private keys plus the service-account key. The Kubernetes CA private key permits certificate signing for privileged identities such as system:masters and provides control of the imported cluster outside Omni's authorization boundary, including its workloads, credentials, and secrets. This issue is fixed in versions 1.6.6 and 1.7.3.	7.6	More Details
CVE-2026-54506	Vvveb is a powerful and easy to use CMS with page builder to build websites, blogs or ecommerce stores. Prior to 1.0.8.5, app/controller/user/profile.php accepts the user[bio] field and passes stored content through sanitizeHTML() in system/functions.php, whose on* event-handler regular expression omits the forward-slash delimiter and whose do-while condition compares the string to itself, so forbidden nested tags are removed only once. An Author-role or higher user can submit solidus-prefixed event-handler markup or nested forbidden tags that survive sanitization. The stored bio is rendered without sufficient output encoding on /author/{username}, in the admin user-management view, and potentially in comment displays, causing attacker-controlled JavaScript to execute when unauthenticated visitors, administrators, or other users view the content. This can expose browser-session data and permit victim-context account actions, defacement, or phishing. This issue is fixed in version 1.0.8.5.	7.6	More Details
CVE-2026-92465	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability in Themeum WP Mega Menu allows Blind SQL Injection. This issue affects WP Mega Menu: from n/a through 1.4.2.	7.6	More Details
CVE-2026-70469	Apache NiFi 2.11.0 disabled support for gzip-encoded HTTP requests for the application REST API and rejected requests that included the standard Content-Encoding header indicating gzip encoding. The framework enforcement filter did not check multiple instances of the Content-Encoding header and did not reject non-standard identifiers for gzip encoding, allowing a malicious client to send crafted requests that could consume excessive amounts of memory. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which disables decompression of gzip-encoded HTTP requests regardless of header number or encoding identifiers.	7.5	More Details
CVE-2026-63460	Vendure is an open-source headless commerce platform. Prior to 3.6.5, the public Shop GraphQL API allows an unauthenticated caller to supply a catastrophically backtracking pattern through StringOperators.regex. packages/core/src/service/helpers/list-query-builder/parse-filter-params.ts passes the raw pattern to the REGEXP implementation registered by packages/core/src/service/helpers/list-query-builder/list-query-builder.ts, and better-sqlite3 and sqljs evaluate it synchronously in the Node.js event loop. ShopProductsResolver.products is publicly reachable, so one nested-quantifier pattern can block request processing and make the storefront and admin API unavailable, while repeated requests can sustain denial of service. PostgreSQL and MySQL or MariaDB deployments do not execute this regular expression in the Node.js event loop. This issue is fixed in version 3.6.5.	7.5	More Details
CVE-2026-85128	The Choose User Role at Registration WordPress plugin before 1.3.3 does not validate the role requested at registration against the roles an administrator chose to offer, allowing unauthenticated users to request any role, including administrator, and to be granted it once the request is approved. Exploitation requires the Choose User Role at Registration WordPress plugin before 1.3.3's role selection feature and public account registration to both be enabled.	7.5	More Details
CVE-2026-79651	A flaw was found in the theme localization endpoints of the keycloak-services component, which is the core service responsible for authentication flows and theme management in Keycloak. The issue occurs because the system accepts arbitrary locale tags from unauthenticated requests and stores them in a permanent in-memory cache without limits. An attacker can exploit this by sending a large number of unique locale tags, eventually causing the server to run out of memory and crash.	7.5	More Details
CVE-2026-80274	If a BIND resolver sends a query for a DNSSEC-signed authoritative zone, and the authoritative server replies with a valid wildcard answer and signed NSEC3 proof, followed by an unsigned NSEC at the same owner name, it will trigger an unexpected program exit. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
	In the Linux kernel, the following vulnerability has been resolved: smb: smbdirect: destroy QP before mem pools on accept failure On the rdma_accept_failed error path of smbdirect_accept_connect_request(), the receive io posted just above is owned by the QP (recv_io is set to NULL after a successful post). The error path fell through to smbdirect_connection_destroy_mem_pools() before smbdirect_connection_destroy_qp(), so the mem pools and the recv_io slab cache were destroyed while that recv_io was still outstanding on the QP. The drain in smbdirect_connection_destroy_qp() (ib_drain_qp()) is what runs the recv completion that returns the recv_io to the free list, so destroying the pools first leaves the object outstanding at kmem_cache_destroy() time ("Slab cache still has objects") and later frees it into an already-destroyed mempool (mempool_free_bulk NULL-pointer dereference). Give rdma_accept_failed its own teardown that drains the QP first, then destroys the mem pools, and returns. The remaining labels (post_recv_io_failed onward) run before the recv_io was ever		

CVE-2026-90172	posted, so they keep the mem-pools-then-qp order. The outstanding recv_io at kmem_cache_destroy() time: [3487.344647] ==========[3487.349942] BUG smbdirect_recv_io_cache_ffff88811ba99000 (Not tainted): Objects remaining on __kmem_cache_shutdown() [3487.356078] ---------- [3487.356078] [3487.356738] Object 0xffff8881511c3440 @offset=13376 [3487.358464] Allocated in mempool_alloc_noprof+0x18c/0x290 age=1194 cpu=6 pid=22254 [3487.361197] mempool_alloc_noprof+0x18c/0x290 [3487.361542] smbdirect_connection_create_mem_pools+0x405/0x780 [3487.361972] smbdirect_accept_connect_request+0x5a8/0x1b80 [3487.362359] smbdirect_listen_rdma_event_handler+0x1579/0x1b90 [3487.362779] cma_cm_event_handler+0x9c/0x230 [3487.363096] cma_ib_req_handler+0x2682/0x45d0 [3487.363414] cm_process_work+0x56/0x3d0 [3487.363676] cm_work_handler+0x8a0e/0xd000 [3487.367496] process_scheduled_works+0xa07/0x13a0 [3487.367859] worker_thread+0x7c9/0xc80 [3487.368148] kthread+0x341/0x430 [3487.368407] ret_from_fork+0x3a8/0x7a0 [3487.368704] ret_from_fork_asm+0x1a/0x30 [3487.370307] Slab 0xffffea0005447000 objects=19 used=1 fp=0xffff8881511c0040 flags=0x100000000000240(workingset|head|node=0|zone=2) [3487.372840] ------------[cut here]------------ [3487.373195] WARNING: mm/slub.c:1244 at __slab_err+0x1a/0x30, CPU#6: kworker/6:84/22254 [3487.373759] Modules linked in: [3487.373993] CPU: 6 UID: 0 PID: 22254 Comm: kworker/6:84 Tainted: G B 7.1.0-next-20260623+ #88 PREEMPT(lazy) [3487.374778] Tainted: [B]=BAD_PAGE [3487.377830] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 [3487.378515] Workqueue: ib_cm cm_work_handler [3487.378820] RIP: 0010:__slab_err+0x1a/0x30 [3487.379129] Code: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 0f 1f 44 00 00 e8 36 00 00 00 bf 05 00 00 00 be 01 00 00 00 e8 f7 75 45 00 90 <0f> 0b 90 c3 cc cc cc cc cc 66 66 66 66 2e 0f 1f 84 00 00 00 00 00 [3487.383255] RSP: 0018:ffff888220fc7050 EFLAGS: 00010093 [3487.383643] RAX: ffffffff8168e60a RBX: ffff88810955e640 RCX: ffff88821c381d80 [3487.384158] RDX: 0000000000000000 RSI: 0000000000000008 RDI: ffffffff870fa080 [3487.384662] RBP: ffff888220fc7068 R08: ffffffff870fa087 R09: 1ffffffff0e1f410 [3487.385192] R10: dfffc0000000000 R11: fffffbfff0e1f411 R12: ffffea0005447210 [3487.385674] R13: ffffea0005447000 R14: ffff888220fc7068 R15: ffff88812a8ab300 [3487.388932] FS: 0000000000000000(0000) GS:ffff888427e76000(0000) knlGS:0000000000000000 [3487.389529] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [3487.389934] CR2: 00007ffcf2d84fd8 CR3: 0000000111d64006 CR4: 0000000000f72ef0 [3487.390440] PKRU: 55555554 [3487.390641] Call Trace: [3487.390826] <TASK> [3 ---truncated---	7.5	More Details
CVE-2026-85756	SSH.NET is a Secure Shell (SSH) library for .NET. Prior to 2026.0.0, ScpClient places caller-supplied remote paths into the command used to run scp on the server, and the default RemotePathTransformation.DoubleQuote transformation cannot safely quote every remote command interpreter. When an application passes an attacker-controlled path to a shell-based server, shell metacharacters not neutralized by the active IRemotePathTransformation can execute commands as the authenticated SSH user. Exploitation requires a shell-based server and a path crafted for that shell's parsing rules; non-shell servers and paths fully neutralized by the selected transformation are not affected. RemotePathTransformation.ShellQuote is available for POSIX shells, while SftpClient avoids a remote shell entirely. This issue is fixed in version 2026.0.0.	7.5	More Details
CVE-2026-63126	Wire provides gRPC and protocol buffers for Android, Kotlin, Swift, and Java. Prior to 6.4.5 and 7.0.0-alpha04, Wire protobuf readers do not consistently validate attacker-controlled lengths against the current logical message boundary before advancing cursors, pointers, limits, slices, or allocations. In Kotlin, ProtoAdapter.decode(ByteArray) and ProtoAdapter.decode(ByteString) use ByteArrayProtoReader32.internalNextLengthDelimited(), where a positive oversized length can wrap pos + length to a negative limit and escape the existing negative-length check. Related ProtoReader, ReadBuffer.readVarint(), ReadBuffer.verifyAdditional(count:), packed-repeated, nested-message, and ProtoDecoder.decodeSizeDelimited(_:from:) paths can cross logical boundaries, perform pointer arithmetic, reserve capacity, or convert an unrepresentable size before proving the requested bytes exist. An attacker who supplies malformed protobuf bytes can cause unchecked exceptions, traps, out-of-bounds behavior, or excessive allocation, resulting in denial of service without known confidentiality, integrity, or code-execution impact. This issue is fixed in versions 6.4.5 and 7.0.0-alpha04.	7.5	More Details
CVE-2026-25275	Transient DOS when processing authentication frames with invalid FILS information element header lengths.	7.5	More Details
CVE-2026-90224	In the Linux kernel, the following vulnerability has been resolved: nfc: nci: fix double completion race in nci_data_exchange_complete nci_close_device() and nci_rx_work can both call nci_data_exchange_complete() concurrently. After commit 4527025d440ce8 ("nfc: nci: fix circular locking dependency in nci_close_device") moved flush_workqueue(ndev->rx_wq) after mutex_unlock(&ndev->req_lock), rx_work is no longer serialized with the explicit completion call in the close path. Both callers read the non-NULL callback pointer and invoke rawsock_data_exchange_complete(), which calls sock_put() -- but only one sock_hold() was taken, so the second sock_put() underflows the refcount and frees the socket while it is still in use. Replace the bare clear_bit(NCI_DATA_EXCHANGE) with test_and_clear_bit() so that only the first caller proceeds to invoke the callback.	7.5	More Details
CVE-2026-90149	In the Linux kernel, the following vulnerability has been resolved: NFSv4/flexfiles: fix NULL dereference for NFSv4.0 data servers flexfiles accepts NFSv4.0 data servers, but two NFSv4 code paths assume the data server client has a session. Unlike NFSv4.1+, an NFSv4.0 client has no session (clp->cl_session is NULL; it uses clp->cl_slot_tbl), so I/O to a v4.0 flexfiles DS oopses: - nfs4_init_ds_session() dereferences clp->cl_session->session_state while seeding the DS lease. It also only seeds cl_lease_time when NFS4_SESSION_INITING is set; without a session that never happens, so cl_lease_time stays 0 and nfs4_renew_state() busy-loops, requeuing every 5 seconds. Seed the lease whenever there is no session and return before touching session state. - ff_layout_async_handle_error_v4() dereferences clp->cl_session->fc_slot_table on every DS I/O error. Fall back to the v4.0 transport slot table (clp->cl_slot_tbl) when there is no session.	7.5	More Details
CVE-2026-86043	Skipper is an HTTP router and reverse proxy for service composition. Prior to version 0.27.37, the opaAuthorizeRequestWithBody filter can authorize an oversized request after Skipper truncates the body presented to Open Policy Agent because the input.truncated_body signal is derived from Content-Length rather than the actual read result. In filters/openpolicyagent/openpolicyagent.go, ExtractHttpBodyOptionally truncates bodies at maxBodyBytes, while filters/openpolicyagent/internal/envoy/skipperadapter.go copies the request headers without adding a Content-Length value that reflects the truncation. For an HTTP/1.1 request using Transfer-Encoding: chunked or an HTTP/2 request without Content-Length, a body-inspecting policy that follows the prior mitigation and permits input.truncated_body equal to false can evaluate only the truncated prefix, allow the request, and then forward the full oversized body to the protected upstream. This residual issue is distinct from CVE-2026-50197. This issue is fixed in version 0.27.37.	7.5	More Details
CVE-2026-	InternLM LMDeploy through 0.17.0 contains a reachable assertion vulnerability in the DistServe decode migration loop that allows unauthenticated attackers to terminate the inference engine. Attackers can submit a migration_request with an empty	7.5	More

92971	remote_block_ids list to trigger an AssertionError that crashes the engine loop and causes subsequent inference requests to fail.		Details
CVE-2026-90228	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix NULL pointer dereference in nvmet_execute_identify_ns_zns() When a host issues an Identify command with CNS 05h (I/O Command Set specific Identify Namespace) and CSI 02h (ZNS) targeting a file-backed namespace, nvmet_execute_identify_ns_zns() calls bdev_is_zoned() on req->ns->bdev. A file-backed namespace has no block device, so req->ns->bdev is NULL and bdev_is_zoned() dereferences it, ooppsing. The I/O command set is selected by the host-supplied CSI field and the command is routed here whenever CONFIG_BLK_DEV_ZONED is enabled, independent of the namespace backing type, so any file-backed namespace is exposed. Reject the command with Invalid Field when the namespace is not backed by a block device.	7.5	More Details
CVE-2026-92961	vm2 before 3.11.6 fails to enforce bufferAllocLimit on ArrayBuffer, SharedArrayBuffer, and TypedArray constructors, allowing attackers to allocate arbitrary host memory. Attackers can bypass the buffer allocation cap by using these V8 intrinsics to exhaust host process memory and trigger out-of-memory conditions.	7.5	More Details
CVE-2026-87742	A flaw was found in quarkus-websockets-next. This vulnerability allows a remote attacker to cause a Denial of Service (DoS) by streaming messages over a single connection faster than the application can process them. Due to unbounded message buffering and a lack of read backpressure, this rapidly exhausts heap space, leading to a java.lang.OutOfMemoryError that crashes the Java Virtual Machine (JVM).	7.5	More Details
CVE-2026-76646	A remote attacker could cause excessive resource consumption by supplying specially crafted request parameters, potentially resulting in a denial of service condition. Older unsupported versions may also be affected. Users are recommended to upgrade to versions 2.3.12, 2.3-next-M9, 3.0.4, 4.0.4, or 4.1.4, which fix this issue.	7.5	More Details
CVE-2026-92916	Grav is a flat-file CMS. In Grav 1.7.0 through 1.7.53.2 and 2.0.0 through 2.0.21, when the debugger is enabled (system.debugger.enabled: true, which is not the default), the Clockwork profiler endpoint is exposed without authentication: InitializeProcessor::handleDebuggerRequest() intercepts any path containing /__clockwork/ during bootstrap and passes it to Debugger::debuggerRequest(), which performs no user lookup, IP restriction, or Clockwork authenticator check, and also supports anonymous pagination over the entire stored history. With the shipped censored: false default, each stored record contains raw request cookies (including Grav's session cookie, whose value is the PHP session id, allowing an attacker to resume another user's session, including an authenticated admin's), the full parsed request body (Grav's login form posts data[username]/data[password], so passwords are stored in plaintext because Clockwork's password filter only inspects top-level keys), and the site's entire system and plugin configuration, including operator-saved secrets such as SMTP credentials, third-party API keys, and licence keys. Authorization and X-API-Token headers are stored even when censored: true. On Grav 2.0, setting provider: debugbar does not avoid the issue because Grav forces the Clockwork provider for requests preferring a JSON response. The issue is fixed in 1.7.53.4 and 2.0.22, which restrict /__clockwork/ to server-local requests or requests presenting the new system.debugger.token secret and strip cookies and credential headers from stored records. Workarounds include setting debugger.enabled: false or blocking /__clockwork/ at the web server or CDN.	7.5	More Details
CVE-2026-92917	Grav is a flat-file CMS. In versions 2.0.0-rc.1 through 2.0.21, the Twig content sandbox fails to restrict the dump and serialize filters (print_r, vardump, json_encode, yaml_encode, string): GravExtension::assertSandboxDumpSafe() determines sandbox state by calling SandboxExtension::isSandboxed() without a Source argument, which reports only the global sandbox flag that Grav never enables, so the guard added in GHSA-mc5q-6hpj-rp7j never executes. As a result, an authenticated user with page-edit rights can render {{ config|print_r }} in page content with Twig processing enabled and dump Grav's entire merged configuration — print_r reflects the real Config object held in a private property of the SandboxConfig facade, bypassing its path redaction — exposing plugin secrets such as SMTP credentials, API tokens, webhook secrets and cache backend passwords. Grav 1.7 is not affected because it ships no Twig content sandbox. The issue is fixed in 2.0.22, where the affected filters are registered with Twig's needs_is_sandboxed flag.	7.5	More Details
CVE-2026-81481	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Filesystem access for attacker.	7.5	More Details
CVE-2026-20343	A vulnerability in a critical API for Cisco Secure FMC Software could allow an unauthenticated, remote attacker to download sensitive files and use unbounded disk space. This vulnerability exists because a critical API lacks authentication. An attacker could exploit this vulnerability by repeatedly invoking the API. A successful exploit could allow the attacker to download sensitive files that should be restricted and consume disk space so the device could become unresponsive, causing a DoS condition.	7.5	More Details
CVE-2026-20247	A vulnerability in Cisco ISE could allow an unauthenticated, remote attacker to conduct SQL injection attacks on an affected device. This vulnerability is due to improper validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to modify data in the underlying database.	7.5	More Details
CVE-2026-46352	Suricata is a network Intrusion Detection System, Intrusion Prevention System and Network Security Monitoring engine. Starting in version 8.0.0 and prior to version 8.0.5, Suricata's IP defragmentation code could deadlock when processing fragmented traffic containing an encapsulated tunnel protocol whose payload is itself fragmented. Version 8.0.5 contains a fix. No known workarounds are available.	7.5	More Details
CVE-2026-90102	In the Linux kernel, the following vulnerability has been resolved: NFSv4/pnfs: key the data server cache on the NFS version nfs4_pnfs_ds_add() keys the per-net data server cache on the multipath address set alone, and struct nfs4_pnfs_ds records no version. That suffices for the files layout driver, which always connects with version 4, but flexfiles takes its version tuple from GETDEVICEINFO per device, and one address can legitimately serve both NFSv3 and NFSv4. Two deviceids on one address with different ds_versions[0].version therefore share a single nfs4_pnfs_ds, and whichever mirror connects first pins ds_clp to its own version. The other one is handed that client anyway, so it selects rpc_call_ops for a version the connection does not speak, and the mismatched sequence-slot handling dereferences NULL. Add the version to the cache key so the two cannot alias, giving each version its own nfs4_pnfs_ds and connection while both mirrors stay usable. Only the major version is compared, since that is what selects rpc_call_ops and rpc_ops; v4.0 and v4.1 keep sharing a client. The files layout driver passes the 4 it already hardcodes at connect time.	7.5	More Details
	In the Linux kernel, the following vulnerability has been resolved: NFSv4.2: fix LAYOUTSTATS send buffer exhaustion encode_layoutstats_maxsz budgets XDR_QUADLEN(PNFS_LAYOUTSTATS_MAXSIZE), i.e. 256 bytes, for the layoutupdate4 body		

CVE-2026-90103	written by the layout driver. The flexfiles record can exceed that. ff_layout_encode_ff_layoutupdate() emits, per data server, a netaddr4, an nfs_fh4, two ff_io_latency4, an nfstime4 and a bool. A data server whose filehandle is NFS_MAXFHSIZE bytes long already accounts for 132 of those bytes, and the two ff_io_latency4 at 64 bytes each, the nfstime4 and the bool add a further 144, so the body passes 256 bytes before the netaddr4 is encoded at all. encode_layoutstats() additionally writes the deviceid4 and the layoutupdate4 lou_type word, neither of which the macro accounts for. The filehandle and the address are both chosen by the server, through LAYOUTGET and GETDEVICEINFO, so it can drive the encoder past the end of the send buffer. xdr_reserve_space() returns NULL once that happens, and the two ff_layout_encode_io_latency() calls run with dss_info->mirror->lock held, so a NULL return there leaves the lock permanently held. Raise PNFS_LAYOUTSTATS_MAXSIZE to 384 so that the record fits inside the reservation.	7.5	More Details
CVE-2026-76163	If BIND is loaded with a "`named.conf`" file that contains no global "`options`" block, an attacker can send a query of QTYPE TKEY which may cause an assertion failure and subsequent unexpected program exit. This issue affects BIND 9 versions 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-92942	vm2 before 3.11.7 (affected versions <= 3.11.6) does not enforce the VM({ timeout }) option on code executed outside the synchronous VM#run() call. The timeout only wraps the single call to _runScript() via doWithTimeout() in lib/vm.js, and FinalizationRegistry and WeakRef are exposed to sandboxed code unmodified (they are not among the hardened globals in lib/setup-sandbox.js). Sandboxed code can register a FinalizationRegistry cleanup callback against an object and then drop the only strong reference to it; vm.run() returns within the configured timeout, but when the V8 garbage collector later reclaims the object it invokes the sandboxed cleanup callback outside any vm2 timeout accounting. A busy loop in that callback blocks the host event loop for an unbounded period, resulting in denial of service. The time of invocation depends on the garbage collector (e.g. under memory pressure or with --expose-gc).	7.5	More Details
CVE-2026-81875	HAPI FHIR is a complete implementation of the HL7 FHIR standard for healthcare interoperability in Java. Prior to version 6.9.12, SHCParser in org.hl7.fhir.r5/src/main/java/org/hl7/fhir/r5/elementmodel/SHCParser.java can consume attacker-controlled Smart Health Card JWT content whose header contains zip: "DEF" and whose small raw-DEFLATE payload expands to a very large value. SHCParser.decodeJWT() passes the decoded payload to SHCParser.inflate(), which accumulates all decompressed bytes in a ByteArrayOutputStream without an output-size limit before JSON parsing, and SHCParser.decompress() contains the same unbounded pattern. An application or validator service that accepts attacker-supplied SHC content can therefore suffer excessive heap allocation, severe garbage-collection pressure, request failure, process instability, or process termination. This issue is fixed in version 6.9.12.	7.5	More Details
CVE-2026-92983	InternLM LMDeploy through 0.17.0 in DistServe prefill/decode disaggregation mode fails to release scheduler sessions because the proxy uses user-facing session IDs instead of internal scheduler keys. Unauthenticated attackers can send completion requests to the proxy endpoint that accumulate unreleased scheduler metadata and memory until the prefill worker is out-of-memory killed.	7.5	More Details
CVE-2026-86040	libp2p is a JavaScript implementation of the libp2p networking stack. Prior to 11.0.26, @libp2p/floodsub accepts unauthenticated RPC frames on /floodsub/1.0.0 through PeerStreams.attachInboundStream in packages/floodsub/src/peer-streams.ts without protobuf element limits, then processRpc and processRpcSubOpt in packages/floodsub/src/floodsub.ts synchronously process the subscriptions array without a per-frame cap. A single bounded-size frame can decode into millions of empty subscription entries that block the event loop, while hundreds of thousands of unique-topic SUBSCRIBE entries allocate PeerSet objects in this.topics that are not removed after peer removal or stop. Empty entries cause CPU exhaustion but do not grow this.topics; persistent memory growth requires unique topics. The subscription path bypasses message signature validation and the message-only processing queue, allowing a remote peer to cause sustained CPU denial of service, memory exhaustion, out-of-memory termination, and node unavailability. The issue is fixed in version 11.0.26.	7.5	More Details
CVE-2026-86038	libp2p is a JavaScript implementation of the libp2p networking stack. From 15.0.0 until 16.0.5, @libp2p/gossipsub uses the default StrictSign policy in packages/gossipsub/src/utils/buildRawMessage.ts, where validateToRawMessage verifies a signature with attacker-controlled msg.key but skips binding that key to msg.from when the claimed author is an RSA peer ID that does not inline a public key. An unauthenticated attacker can place a victim RSA peer ID in msg.from, sign the message with the attacker's private key, and supply the attacker's public key in msg.key, causing the message to be accepted and propagated as authored by the victim. Applications that trust message.from for validators, authorization, accounting, moderation, reputation, or audit logging can process attacker-controlled data under false origin attribution. The issue is fixed in version 16.0.5.	7.5	More Details
CVE-2026-85721	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 2.0.0 until 2.16.1 and 3.0.12, automatic response decompression on the HTTP/1.1 path uses ChannelManager.newHttpContentDecompressor() to install Http1ContentDecompressor without a cumulative output-size limit. A hostile or compromised server, or an attacker who can alter a response in transit, can send a small gzip, deflate, or snappy response that expands across chunks until the client exhausts its heap and raises OutOfMemoryError; brotli and zstd are also affected when their optional codecs are present. In versions 3.0.8 through 3.0.10, the HTTP/2 decompressor is also unbounded, so switching protocols does not mitigate the issue on those releases. A limit applied to each decode call is insufficient because the response can be delivered as many small chunks, so the fixed implementation tracks total decompressed bytes for the whole response. This issue is fixed in versions 2.16.1 and 3.0.12.	7.5	More Details
CVE-2026-92719	Quickwit through 0.9.0 fails to validate the host and scheme of the queue_url parameter in SQS file sources, allowing attackers to make the node issue requests to arbitrary internal addresses. Attackers can supply a malicious queue_url to the create-source API to scan internal networks and fingerprint services based on connection response differences.	7.5	More Details
CVE-2026-82399	CoreDNS is a DNS server written in Go. Prior to 1.14.7, the DNS-over-HTTPS, DNS-over-HTTP/3, DNS-over-QUIC, and DNS-over-gRPC request paths in plugin/pkg/doh/doh.go, core/dnsserver/server_quic.go, and core/dnsserver/server_grpc.go call dns.Msg.Unpack on attacker-controlled DNS section counts before dns.DefaultMsgAcceptFunc validates the fixed header. An unauthenticated client can use DNS name compression and excessive section counts to amplify allocation before the plugin chain, so plugin-level rate limiting cannot prevent concurrent requests from exhausting memory and terminating CoreDNS. The ordinary UDP and TCP listeners are not affected because they validate the header first. This issue is fixed in version 1.14.7.	7.5	More Details
	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: restore NET_IP_ALIGN in the RX DMA offset Since the RX path was converted to zero-copy, the page pool page is handed to the stack directly as the skb head, and the offset the DMA engine writes at is what determines the alignment of the packet headers. Before the conversion the payload was copied into an skb obtained from napi_alloc_skb(), which reserves NET_SKB_PAD + NET_IP_ALIGN. The conversion moved the headroom into stmmac_rx_offset() but did not carry over NET_IP_ALIGN, so on architectures where NET_IP_ALIGN is 2 the IP		

CVE-2026-90059	header now lands misaligned: 64 (NET_SKB_PAD) + 14 (ethernet) + 20 (IP) = 98 Same for the XDP branch: 256 (XDP_PACKET_HEADROOM) + 14 (ethernet) + 20 (IP) = 290 On ARM32 this is fatal, because ldm and ldrd trap on unaligned addresses even when CONFIG_HAVE_EFFICIENT_UNALIGNED_ACCESS is set. Any received echo request panics the machine, e.g: Unhandled fault: alignment exception (0x001) at 0x81873062 Internal error: : 1 [#1] SMP ARM Hardware name: Altera SOCFPGA Arria10 PC is at icmp_echo+0x38/0xa8 LR is at icmp_rcv+0x22c/0x370 Call trace: icmp_echo from icmp_rcv+0x22c/0x370 icmp_rcv from ip_protocol_deliver_rcu+0x2c/0x224 ip_protocol_deliver_rcu from ip_local_deliver+0xc8/0x1a0 ip_local_deliver from ip_sublist_rcv_finish+0x3c/0x50 ip_sublist_rcv_finish from ip_list_rcv_finish+0x110/0x118 ip_list_rcv_finish from ip_list_rcv+0xc8/0xdc ip_list_rcv from __netif_receive_skb_list_core+0x170/0x1c0 ... napi_complete_done from stmmac_napi_poll_rx+0xcb0/0x1030 Code: e24dd068 e59020a0 e28dc010 e0822001 (e8920003) Kernel panic - not syncing: Fatal exception in interrupt The faulting instruction is the ldm of *icmp_hdr(skb) in icmp_echo(). Fix by adding NET_IP_ALIGN back to the RX offset, which restores the alignment the stack used to get. Note that commit a955318fe67e ("stmmac: align RX buffers") made a similar change in 2021 and was reverted by commit 12d125b4574b ("stmmac: Revert "stmmac: align RX buffers""") because it caused packet corruption. That patch raised the offset from 0 without adjusting the buffer size accounting, so the DMA engine could arguably write past the end of the RX buffers, though this was never root caused. Commit df542f669307 ("net: stmmac: Switch to zero-copy in non-XDP RX path") since derives the page pool allocation from stmmac_rx_offset(), so the extra bytes are accounted for.	7.5	More Details
CVE-2026-90067	In the Linux kernel, the following vulnerability has been resolved: libceph: validate banner payload length When parsing the Ceph messenger v2 protocol banner, the `payload_len` field is decoded from the banner prefix. If a client sends a banner with a `payload_len` of 0, the kernel sets up a 0-length socket read. This violates an invariant in the state machine, triggering a warning in `populate_in_iter()`: ------------[cut here]------------ !iov_iter_count(&con->v2.in_iter) WARNING: net/ceph/messenger_v2.c:3129 at populate_in_iter net/ceph/messenger_v2.c:3129 [inline], CPU#1: kworker/1:3/5070 WARNING: net/ceph/messenger_v2.c:3129 at ceph_con_v2_try_read+0x6634/0x6810 net/ceph/messenger_v2.c:3159, CPU#1: kworker/1:3/5070 ... Call Trace: <TASK> ceph_con_workfn+0x1f5/0x14a0 net/ceph/messenger.c:1575 process_one_work kernel/workqueue.c:3322 [inline] process_scheduled_works+0xa8e/0x14e0 kernel/workqueue.c:3405 worker_thread+0xa47/0xfb0 kernel/workqueue.c:3486 kthread+0x388/0x470 kernel/kthread.c:436 ret_from_fork+0x514/0xb70 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK> According to the msgr2 protocol specification, the banner payload is expected to contain at least two 64-bit integers (`server_feat` and `server_req_feat`). Therefore, `payload_len` must be at least 16 bytes. Fix this by adding a check in `process_banner_prefix()` to reject a `payload_len` smaller than 16 bytes. This prevents the 0-length read and correctly aborts the connection with a protocol error.	7.5	More Details
CVE-2026-92784	@refinedev/inferencer through 7.0.0 fails to escape API field names when interpolating them into generated JSX source code. Attackers controlling the data provider can inject malicious JavaScript through crafted JSON property names that execute in the developer's browser when the Inferencer page renders.	7.5	More Details
CVE-2026-92791	Uber Kraken through 0.1.29 fails to validate the tag parameter in the /tags/{tag} endpoint, allowing unauthenticated attackers to traverse outside the configured storage root. Attackers can use percent-encoded parent-directory segments in the tag parameter to read arbitrary files accessible to the testfs backend process.	7.5	More Details
CVE-2026-90234	In the Linux kernel, the following vulnerability has been resolved: NFS: Return a delegation the client fails to record When an NFS server grants a delegation in an OPEN reply, nfs_inode_set_delegation() records it on the client. However, three of its error flows return without sending DELEGRETURN. A delegation can be relinquished only by DELEGRETURN (RFC 8881 Section 20.2.4), so dropping one silently leaves the server believing the client still holds it. If the server happens to recall that delegation, the client answers CB_RECALL with NFS4ERR_BADHANDLE because it has no record of the stateid. The server revokes the delegation and moves it onto its cl_revoked list, because the client never sends the FREE_STATEID that would drain it. Every subsequent SEQUENCE reply then carries SEQ4_STATUS_RECALLABLE_STATE_REVOKED, and the client's state manager loops issuing TEST_STATEID across its delegations without ever clearing the condition. The window is easy to reach now that a server offers a write delegation on any write OPEN: a delegation recalled for one opener races a re-open that the server answers with a fresh write delegation. Instead of dropping it, hand the delegation back during these error flows.	7.5	More Details
CVE-2026-92792	OpenNHP through 1.0.2 selects its trusted-execution attestation verifier based on attacker-supplied evidence containing a test_purpose key, causing the FallbackVerifier to execute unconditionally. Attackers can bypass attestation verification by including the test_purpose key in evidence and providing enrolled measure and serial number pairs from the allowlist to gain unauthorized access.	7.5	More Details
CVE-2026-92625	Control iD iDSecure versions prior to 4.8.3.0 are affected by an unauthenticated Denial of Service. The /api/license/restartService endpoint is reachable without authentication and invokes an internal routine that terminates the iDSecure service process and relaunches it by way of a generated batch script. An unauthenticated remote attacker can call this endpoint repeatedly to hold the service in a continuous restart cycle, rendering it unavailable.	7.5	More Details
CVE-2026-92794	OpenSign through 2.41.3 fails to validate caller identity in the getDocument cloud function when one-time-password verification is disabled. Attackers can supply a document identifier from guest signing links to retrieve complete document details including all signers' information, sender identity, and valid download tokens without authentication.	7.5	More Details
CVE-2026-85719	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 2.1.0 until 2.16.1 and 3.0.12, requests using an authenticated SOCKS proxy can expose the proxy's credentials to the origin because NettyRequestFactory and NettyRequestSender attach Proxy-Authorization without confirming that the request is being sent to an HTTP proxy. With preemptive proxy authentication, the header is attached to a plaintext HTTP request, exposing credentials such as directly reversible Basic credentials to the origin. With the default non-preemptive flow, a hostile origin can return a 407 response and ProxyUnauthorized407Interceptor sends the proxy credentials through the existing SOCKS tunnel, including NTLM, Kerberos, and SPNEGO credentials. Releases before 2.1.0 lack SOCKS proxy support. This issue is fixed in versions 2.16.1 and 3.0.12.	7.5	More Details
CVE-2026-85715	ExifReader is a JavaScript Exif information parser. Prior to 4.41.1, ExifReader parses attacker-controlled HEIC or AVIF ISO-BMFF files in getItems() within src/image-header-iso-bmff-iloc.js and trusts iloc itemCount and extentCount values while allocating an extent object for every nested-loop iteration. When offsetSize, lengthSize, baseOffsetSize, and indexSize are zero, the extent fields consume no input bytes and the buffer offset does not advance, but the parser can still allocate up to itemCount multiplied by extentCount objects without an allocation budget. A small malicious iloc box can therefore cause hundreds of megabytes of heap growth or exhaust system memory, terminating a Node.js process and denying service to web, desktop, or mobile applications that parse untrusted images. The zero field widths are valid ISO-BMFF values indicating absent fields, so the	7.5	More Details

	vulnerable parser must bound work rather than relying on offset advancement. The issue is fixed in version 4.41.1.		
CVE-2026-81516	Steeltoe is an open source project that provides a collection of libraries that helps users build cloud-native applications. From 4.0.0 until 4.3.0, ConsulDiscoveryClient constructs ConsulServiceInstance objects by parsing each registration's secure metadata with a strict Boolean conversion. A principal that can register a Consul service can supply a secure value other than true or false, causing the exception from one instance to abort construction of the entire instance list and make the targeted service undiscoverable. When GetAllInstancesAsync enumerates all services, one malformed instance can abort enumeration across every service. The outage persists until the offending registration is removed. This issue is fixed in version 4.3.0.	7.5	More Details
CVE-2026-92815	changedetection.io through 0.60.6 fails to validate the Goto URL action in browser steps, allowing unauthenticated attackers to access internal addresses. Attackers can supply arbitrary internal URLs in the optional_value parameter to retrieve responses from restricted network locations.	7.5	More Details
CVE-2026-81876	HAPI FHIR is a complete implementation of the HL7 FHIR standard for healthcare interoperability in Java. Prior to version 6.9.12, SHCParser in org.hl7.fhir.r5/src/main/java/org/hl7/fhir/r5/elementmodel/SHCParser.java can enter an infinite loop while processing attacker-controlled Smart Health Card JWT content whose header contains zip: "DEF" and whose raw-DEFLATE payload is empty or truncated. SHCParser.decodeJWT() reaches SHCParser.inflate(), where Inflater.inflate() can return zero while Inflater.finished() remains false and Inflater.needsInput() is true. The loop also lacks an Inflater.needsDictionary() termination check, SHCParser.decompress() contains the same zero-progress pattern, and ResourceChecker.java can reach SHC parsing during file-format detection. A malformed validation request can pin a JVM worker thread indefinitely, and concurrent requests can exhaust all validation workers. This issue is fixed in version 6.9.12.	7.5	More Details
CVE-2026-81515	Steeltoe is an open source project that provides a collection of libraries that helps users build cloud-native applications. From 4.0.0 until 4.3.0, EurekaDiscoveryClient deserializes the registry response as one unit, and an unrecognized actionType or status, a non-Boolean isCoordinatingDiscoveryServer, or a nonnumeric timestamp can abort the entire response. A principal that can register or update an instance can cause all connected Steeltoe clients to receive an empty or stale instance list until the malformed registration is removed. The JsonInstanceInfoConverter, BoolStringJsonConverter, and LongStringJsonConverter parsing paths are affected. This issue is distinct from the earlier DataCenterInfo.name parsing vulnerability. This issue is fixed in version 4.3.0.	7.5	More Details
CVE-2026-84997	react/http is an event-driven, streaming HTTP client and server implementation for ReactPHP. From 0.6.0 until 1.11.1, React\Http\Io\ChunkedDecoder could enter an infinite loop while processing a malformed Transfer-Encoding: chunked body because handleData required its buffer to shrink on every iteration. An incomplete terminal-chunk trailer without CRLF left the buffer unchanged after strpos returned false, and exactly two non-CRLF bytes after a completed non-terminal chunk bypassed both the error and wait guards. The affected decoder processes request bodies for React\Http\HttpServer and response bodies for React\Http\Browser, allowing a malicious client to freeze a server or a malicious or compromised server to freeze a client. A reverse proxy that normalizes inbound requests may protect the server direction but does not protect outbound Browser requests. This issue is fixed in version 1.11.1.	7.5	More Details
CVE-2026-92577	In AVideo through 29.0, the API get_api_video endpoint contains a broken access control vulnerability in the clean_title branch that returns user-group-restricted videos with owner PII to anonymous callers. Attackers can query videos by their public slug to bypass group restrictions and retrieve sensitive user fields including email, phone, address, birth date, and administrator status.	7.5	More Details
CVE-2026-92987	roxmltree through 0.21.1 performs quadratic-time attribute and namespace validation during XML parsing without limits on attribute count. Attackers can craft XML documents with tens of thousands of attributes on a single element to consume excessive CPU time and cause denial of service.	7.5	More Details
CVE-2026-92594	Craft CMS 5.0.0-RC1 through versions before 5.11.0 incorrectly authorize the GraphQL draftCreator and revisionCreator fields: instead of requiring the user-data scope enforced by Gql::canQueryUsers() (usergroups.*:read), these fields are gated only on the elements.drafts:read / elements.revisions:read scopes, and their resolver returns a raw User element whose email, username, fullName, and addresses fields have no per-field authorization. A client holding only the drafts or revisions scope — including an unauthenticated client when the operator has enabled the public GraphQL schema with those scopes — can therefore harvest the email addresses, usernames, full names, and postal addresses of all draft/revision creators (typically site editors and administrators). The issue is fixed in 5.11.0.	7.5	More Details
CVE-2026-92596	Nodemailer before 9.1.0 contains a quadratic time complexity vulnerability in the addressparser component that allows remote attackers to cause denial of service by supplying a crafted comma-separated address list. Attackers can send a single email with a large number of addresses to block the Node.js event loop for extended periods, consuming 100% CPU and freezing the process.	7.5	More Details
CVE-2026-92599	joi (npm package `joi`, hapi.js) versions >=17.2.0 <17.13.7 and >=18.0.0 <18.2.6 are vulnerable to regular expression denial of service in the `Joi.string().isoDate()` validation rule. One of the regular expressions the rule applies to the input is unanchored, so a valid ISO date followed by a long run of fractional-second digits causes the regex engine to restart its search from every position in the string, yielding time proportional to the square of the input length (about 1.4 s for 64 KB of digits and about 22 s for 256 KB). A remote attacker who can supply a string to an isoDate validation can stall the application with a single request. Fixed in 17.13.7 and 18.2.6; as a workaround, cap the length of the string before it reaches joi.	7.5	More Details
CVE-2026-65388	A remote attacker who controls a container registry may be able to direct a client's token request to a host of the attacker's choice, and disclose the victim's registry credentials to that host. This vulnerability is addressed in containerization version 0.41.0.	7.5	More Details
CVE-2026-92626	Control iD iDSecure versions prior to 4.8.3.0 are affected by an unauthenticated Denial of Service. The /api/dguardintegration/dguardVersion endpoint dereferences DGuard integration login state that may be unset, raising an unhandled null reference exception. The exception is thrown from an asynchronous method that returns void, so it is not observed by a caller and can terminate the iDSecure process.	7.5	More Details
CVE-2026-	CoreDNS is a DNS server written in Go. Prior to 1.14.7, the DNS-over-HTTPS, DNS-over-HTTP/3, DNS-over-QUIC, and DNS-over-gRPC listeners in plugin/pkg/doh/doh.go, core/dnsserver/server_quic.go, and core/dnsserver/server_grpc.go call dns.Msg.Unpack without the dns.DefaultMsgAcceptFunc request policy used by UDP, TCP, and DNS-over-TLS. An unauthenticated client can send an RFC 2136 UPDATE that the proxy or forward plugin passes unchanged to an update-capable upstream. If that upstream	7.5	More Details

86003	trusts CoreDNS's source address or connection and does not require an attacker-unknown end-to-end TSIG, the request appears to originate from CoreDNS and can add, replace, or delete DNS records, redirect traffic, take over names, alter mail routing, or disrupt the writable zone. This issue is fixed in version 1.14.7.		
CVE-2026-38999	A Null Pointer Dereference in the mk_sched_event_close function (mk_server/mk_scheduler.c) of Monkey through commit 4fb0c16 allows attackers to cause a Denial of Service (DoS) via sending a crafted HTTP request to the server.	7.5	More Details
CVE-2026-81563	A BIND resolver encountering an SVCB/HTTPS AliasMode record referencing 14 or more SVCB/HTTPS ServiceMode records may fail to properly deallocate internal resources. If this happens repeatedly, resource exhaustion will eventually prevent the resolver from performing new recursive lookups. This issue affects BIND 9 versions 9.18.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.18.11-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-63128	RMCP is an official Rust SDK for the Model Context Protocol. Prior to 2.0.0, the rmcp crate's stateful Streamable HTTP server in crates/rmcp/src/transport/streamable_http_server/tower.rs allows an unauthenticated client to send a well-formed JSON-RPC POST that is not an initialization request, or an initialization request with a mismatched protocol header, causing StreamableHttpService::handle_post to call LocalSessionManager.create_session before validating the message. An early validation failure returns without removing the inserted LocalSessionHandle from LocalSessionManager.sessions, permanently retaining session and channel state for the server process lifetime. Repeated requests can grow the shared session table without bound, degrade legitimate-client latency through lock contention, exhaust memory, and terminate the server. This issue is fixed in version 2.0.0.	7.5	More Details
CVE-2026-68537	`fulgur` converts untrusted HTML/CSS into PDF, commonly on a server that processes input supplied by many tenants. In versions prior to 0.19.0, a body-direct child whose CSS-resolved height greatly exceeds the page height was sliced into one fragment per page with no upper bound. This is fixed in version 0.19.0. A `MAX_PAGES` cap bounds the slice loop — halting it even for a `+inf` height — and non-finite layout heights are sanitized so they can no longer drive the loop. As a workaround, validate or constrain untrusted CSS (in particular `height` / `vh` on body-level elements) before passing HTML to fulgur.	7.5	More Details
CVE-2026-89974	In the Linux kernel, the following vulnerability has been resolved: nvme-fc: fix double free of fabrics options when nvme_add_ctrl() fails nvmf_create_ctrl() owns the fabrics options and frees them whenever ->create_ctrl() returns an error, so a transport must not free them on its own error paths. nvme-fc tracks this by testing ctrl->ctrl.opts in nvme_fc_ctrl_free(), which requires nvme_fc_init_ctrl() to clear that pointer on every error exit. The coupling is implicit, and commit 1a9e218195a5 ("nvme: split device add from initialization") broke it by adding a second error exit. When nvme_add_ctrl() fails, nvme_fc_init_ctrl() jumps to out_put_ctrl:, past the "ctrl->ctrl.opts = NULL" that only sits on the fail_ctrl: path, so nvme_fc_ctrl_free() frees the options and nvmf_create_ctrl() frees them a second time: BUG: KASAN: slab-use-after-free in nvmf_free_options+0x30/0x190 nvmf_free_options+0x30/0x190 drivers/nvme/host/fabrics.c:1284 nvmf_create_ctrl drivers/nvme/host/fabrics.c:1374 [inline] Freed by task 5534: nvme_fc_ctrl_free drivers/nvme/host/fc.c:2374 [inline] nvme_fc_init_ctrl+0xe17/0x1450 drivers/nvme/host/fc.c:3605 nvme_add_ctrl() fails when dev_set_name() cannot allocate, so this is reachable under memory pressure or fault injection. Without KASAN the options are freed twice. Rather than clear the pointer on the second exit as well, derive ownership the way nvme-tcp, nvme-rdma and nvme-loop do, from list membership: their free_ctrl leaves the options alone unless the controller made it onto the transport list. The list cannot simply be populated on the success path as it is there. nvme-fc runs the initial connect synchronously via flush_delayed_work(), and the controller has to be reachable on rport->ctrl_list for the whole of it: nvme_fc_unregister_remoteport() needs to find it to signal connectivity loss, nvme_fc_match_disconn_ls() matches an incoming Disconnect Association LS against ctrl->association_id, which is only assigned during that window, nvme_fc_resume_controller() needs it on remoteport re-registration, and nvme_fc_existing_controller() uses it to reject a duplicate connect racing the one in flight. Keep the insertion where it is and add a fail_unlist: label, falling into fail_ctrl:, for the error paths that run after it. The earlier error paths never reach the insertion and keep using fail_ctrl: directly, so the list is only touched where the controller is actually on it. nvme_fc_ctrl_free() cannot use the plain "goto free_ctrl" the other transports use, because it still has to put_device(), release the rport reference and free the ida entry for resources taken before the insertion. Sample list_empty() under rport->lock instead. ctrl->ctrl.opts also stays valid for the whole teardown now. That is not the bug being fixed, but it removes some fragility around the old idiom: nvme_free_ctrl() calls nvme_auth_free() before ->free_ctrl(), and ctrl_max_dhchaps() dereferences ctrl->opts without a NULL check when ctrl->dhchap_ctxs is set, which nvme-fc permits since NVMF_ALLOWED_OPTS allows the dhchap options. The nvme sysfs attributes that dereference ctrl->opts, such as hostnqn and address, evaluate their is_visible() test once at device_add() time and stay readable until cdev_device_del().	7.5	More Details
CVE-2026-89971	In the Linux kernel, the following vulnerability has been resolved: nvme: skip the zoned limits update if the zone info query failed nvme_query_zone_info() returns either a negative errno or a positive NVMe status code, but nvme_update_ns_info_block() only tests for the negative case: ret = nvme_query_zone_info(ns, lbaf, &zi); if (ret < 0) goto out; If the device fails the Identify Namespace (I/O Command Set specific) command, or the Identify Controller command issued by nvme_set_max_append(), the positive status falls through and setup continues with the zero-initialized zone info. nvme_update_zone_info() then marks the queue zoned with chunk_sectors and ns->head->zsze set to zero. blk_validate_zoned_limits() does not check chunk_sectors, so the limits commit succeeds. blk_revalidate_disk_zones() does reject the zero zone size, but by then the limits are live and nothing rolls them back, so I/O keeps being submitted to a zoned queue with a zero zone size and disk_zone_no() shifts by ilog2(0): nvme0n1: Invalid non power of two zone size (0) UBSAN: shift-out-of-bounds in include/linux/blkdev.h:747:16 shift exponent -1 is negative disk_zone_no include/linux/blkdev.h:747 [inline] bio_straddles_zones include/linux/blkdev.h:1058 [inline] blk_zone_wplug_handle_write block/blk-zoned.c:1423 [inline] blk_zone_plug_bio.cold+0x25/0x1c8 block/blk-zoned.c:1605 blk_mq_submit_bio+0x18fb/0x2870 block/blk-mq.c:3196 submit_bh_wbc+0x575/0x740 fs/buffer.c:2824 __block_write_full_folio+0x728/0xdd0 fs/buffer.c:1933 Any device, firmware or NVMe-oF target that fails this one command reaches this. Skip the zoned limits update in that case, and log which of the two things happened: during a revalidation the queue keeps the zone geometry it was last validated with, and on a first scan the namespace is registered without zoned limits, so that it is still available as a handle for admin commands. Neither of the paths in nvme_query_zone_info() that return a positive status logs anything, so the failure would otherwise be silent. zi.zone_size is an exact indicator: every path that returns a positive status returns before it is assigned, and after that the only failure left is -ENODEV, which the caller already handles. Found by FuzzNvme.	7.5	More Details
	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: reject unsolicited H2CData PDUs nvmet_tcp_handle_h2c_data_pdu() accepts an H2CData PDU after only checking that its TTAG is a valid in-range command index and that the command's data buffers are mapped. It never checks that the target has actually solicited that data by sending an R2T for the command. A remote host can abuse this. It submits a write command that takes the R2T path and,		

CVE-2026-89968	before the target transmits the R2T, sends an H2CData PDU for that command's tag. The data completes the command early, and when the command then fails synchronously (e.g. a length mismatch caught by nvmet_check_transfer_len()), it is completed a second time. Each completion calls nvmet_tcp_queue_response(), so the same command is added to queue->resp_list twice while it is still linked; the second llist_add() makes the node point to itself (lentry->next == lentry). nvmet_tcp_process_resp_list() then walks that self-referential node and adds the command to resp_send_list twice. With CONFIG_DEBUG_LIST this trips the "list_add double add" check (kernel BUG); without it the loop never terminates and the nvmet_tcp workqueue wedges (soft-lockup). It is remotely triggerable and needs no authentication on an allow_any_host subsystem. Track whether an R2T has been transmitted for a command and reject an H2CData PDU that arrives before it. The flag is cleared on command reuse (nvmet_tcp_get_cmd() zeroes cmd->flags) and stays set across the multiple H2CData PDUs of a single solicited transfer.	7.5	More Details
CVE-2026-93151	In the Linux kernel, the following vulnerability has been resolved: nvmet-rdma: fix response resource leak on queue teardown When an nvme target with rdma transport is removed while I/Os are in flight, a response can be posted but its send completion is never delivered before the connection is torn down. As a result nvmet_rdma_send_done() and nvmet_rdma_release_rsp() are never called for the response, and this leaks the allocated RDMA read/write context and request SGLs. These leaks are recreated by running blktests nvme/061 with the rdma transport and the siw driver. Kernel kmemleak feature reports them as follows: unreferenced object 0xffff88812bc490c0 (size 32): comm "kworker/2:1H", pid 409, jiffies 4307744490 backtrace (crc 89afd339): __kmalloc_noprof+0x5f9/0x890 sgl_alloc_order+0x7b/0x380 nvmet_req_alloc_sgls+0x290/0x4f0 [nvmet] nvmet_rdma_map_sgl_keyed+0x241/0x12e0 [nvmet_rdma] nvmet_rdma_handle_command+0x73e/0xb80 [nvmet_rdma] __ib_process_cq+0x149/0x4c0 [ib_core] ib_cq_poll_work+0x49/0x160 [ib_core] process_one_work+0x8b2/0x1640 worker_thread+0x5fd/0xfe0 kthread+0x367/0x460 ret_from_fork+0x655/0x9d0 ret_from_fork_asm+0x1a/0x30 unreferenced object 0xffff88814bd05e80 (size 64): comm "kworker/3:1H", pid 148, jiffies 4295195428 backtrace (crc e35510cb): __kmalloc_noprof+0x5f9/0x890 rdma_rw_ctx_init+0x333/0x1fa0 [ib_core] nvmet_rdma_map_sgl_keyed+0x5c8/0x12e0 [nvmet_rdma] nvmet_rdma_handle_command+0x73e/0xb80 [nvmet_rdma] __ib_process_cq+0x149/0x4c0 [ib_core] ib_cq_poll_work+0x49/0x160 [ib_core] process_one_work+0x8b2/0x1640 worker_thread+0x5fd/0xfe0 kthread+0x367/0x460 ret_from_fork+0x655/0x9d0 ret_from_fork_asm+0x1a/0x30 To avoid the memory leaks, reclaim the memory of the in-flight responses when the queue QP is torn down. Call nvmet_rdma_free_rsp_resources() that frees up the RDMA read/write context and the request SGLs of such responses.	7.5	More Details
CVE-2026-89897	In the Linux kernel, the following vulnerability has been resolved: media: cec: Serialize exclusive follower delivery cec_receive_notify() reads the exclusive follower pointer without the adapter lock. Serialize the no-follower check and message delivery against mode changes and release.	7.5	More Details
CVE-2026-92230	Apache Karaf's XmlUtils cached XML parser/transformer factories in static ThreadLocal fields on long-lived container threads. Because a ThreadLocal value outlives the OSGi bundle that created it, repeated bundle or feature install, update, or refresh operations can leave successive bundle ClassLoader's pinned in memory and unreachable for garbage collection, leading to unbounded Metaspace growth and eventual denial of service of the Karaf instance.	7.5	More Details
CVE-2026-50125	MKP is a Model Context Protocol server for Kubernetes. Prior to 0.4.1, cmd/server/main.go exposes the default HTTP endpoint and pkg/mcp/server.go registers the unauthenticated get_resource tool, which accepts attacker-controlled limitBytes and tailLines values for the pods logs subresource. buildPodLogOpts() in pkg/k8s/subresource.go parses those values as unbounded int64 parameters, and defaultGetPodLogs() copies the returned Kubernetes log stream through io.Copy into an in-memory bytes.Buffer without an application-side cap. A remote attacker who can reach the default port 8080 MCP endpoint and select a pod with sufficiently large accumulated logs can send one tools/call request that causes large allocations and additional response copies, while the request-frequency limiter does not constrain per-request volume. This can exhaust process memory, terminate the MKP server, and deny the MCP service; observed testing showed more than one GiB of RSS growth while handling a 128 MiB requested stream. This issue is fixed in version 0.4.1.	7.5	More Details
CVE-2026-50285	Pomerium is an identity and context-aware access proxy. Prior to 0.32.8, decodeQueryStringV2 in pkg/hpke/url.go performs zstd decompression of attacker-controlled data without an output-memory limit when DecryptURLValues processes HPKE V2 values for Stateless.Callback in internal/authenticateflow/stateless.go. In hosted or stateless authentication deployments, an unauthenticated attacker can obtain the receiver key from /.well-known/pomerium/hpke-public-key, provide a matching attacker-controlled sender key, and send a compressed payload to /.pomerium/callback that expands before validateSenderPublicKey rejects the sender. This can allocate hundreds of megabytes per request, exhaust proxy memory, crash or degrade the process, and block access to applications protected by the deployment. Stateful deployments are not affected because the stateful callback verifies its HMAC signature before decryption and decompression. This issue is fixed in version 0.32.8.	7.5	More Details
CVE-2026-54716	Valhalla is an open source routing engine and accompanying libraries for use with OpenStreetMap data. In 3.7.0 and earlier, a POST request to /sources_to_targets containing an exclude_polygons ring formed by three collinear points can cause unbounded memory growth in the worker. The zero-area geometry, rather than the other request options, triggers processing in src/loki/polygon_search.cc until the process is terminated by the out-of-memory killer. A single unauthenticated request can therefore stop a public-facing worker. Other endpoints that accept exclude_polygons, including /route, were not verified as affected. No fixed version is available as of this review.	7.5	More Details
CVE-2026-89863	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: edif: Fix NULL pointer deref in RX SA delete check qla_chk_edif_rx_sa_delete_pending() obtains the SCSI command via GET_CMD_SP(sp) and immediately dereferences cmd->sc_data_direction. That command pointer can be NULL: the firmware may post a status completion for a command that has already been returned or aborted. The caller qla2x00_status_entry() acknowledges this on the very same status path, re-fetching GET_CMD_SP(sp) and bailing out with the "Command already returned" message when it is NULL -- but that check runs only after qla_chk_edif_rx_sa_delete_pending() has already dereferenced the pointer, so a NULL cmd crashes the kernel in interrupt context. Return early when cmd is NULL, before touching cmd->sc_data_direction.	7.5	More Details
CVE-2026-50275	The Datadog PHP Tracer provides application performance monitoring and distributed tracing for PHP. Prior to 1.19.2, ddtrace_deserialize_baggage in ext/distributed_tracing_headers.c parses incoming W3C baggage HTTP headers without enforcing DD_TRACE_BAGGAGE_MAX_ITEMS or DD_TRACE_BAGGAGE_MAX_BYTES. A remote unauthenticated client can send an arbitrarily large number of comma-separated key-value pairs or a single oversized value, causing the tracer to allocate hash-map entries and consume unbounded CPU and memory on each request. Baggage extraction is enabled by default in most affected deployments unless baggage is removed from DD_TRACE_PROPAGATION_STYLE or DD_TRACE_PROPAGATION_STYLE_EXTRACT. This issue is fixed in version 1.19.2.	7.5	More Details

CVE-2026-50277	dd-trace-cpp is the Datadog distributed tracing library for C++. Prior to 2.1.0, dd-trace-cpp parses incoming W3C baggage headers without enforcing DD_TRACE_BAGGAGE_MAX_ITEMS or DD_TRACE_BAGGAGE_MAX_BYTES on the extraction path, even though those limits are enforced during injection. A remote unauthenticated attacker can send a header containing many comma-separated key-value pairs or one very large value, causing per-request hash-map allocation and unbounded CPU and memory consumption. Baggage extraction is enabled by default in most affected tracers unless baggage is removed from DD_TRACE_PROPAGATION_STYLE or DD_TRACE_PROPAGATION_STYLE_EXTRACT, so affected internet-facing services can be denied service. This issue is fixed in version 2.1.0.	7.5	More Details
CVE-2025-14871	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 18.4.6 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an unauthenticated user to cause denial of service due to improper resource allocation limits in the GraphQL complexity calculation logic.	7.5	More Details
CVE-2026-68523	`fulgur` converts untrusted HTML/CSS into PDF, commonly on a server that processes input supplied by many tenants. In versions prior to 0.19.0, a body-direct child whose CSS-resolved height greatly exceeds the page height was sliced into one fragment per page with no upper bound. This is fixed in 0.19.0. A `MAX_PAGES` cap bounds the slice loop — halting it even for a `+inf` height — and non-finite layout heights are sanitized so they can no longer drive the loop. As a workaround, validate or constrain untrusted CSS (in particular `height` / `vh` on body-level elements) before passing HTML to fulgur.	7.5	More Details
CVE-2026-18911	ManageEngine DataSecurity Plus versions before 6310 are vulnerable to an agent authentication bypass, allowing unenrolled agents to send requests without proper authentication.	7.5	More Details
CVE-2026-19667	If an attacker-controlled authoritative server can produce a negative answer that is exactly 65536 bytes, then a flaw in `named` results in a negative cache entry of 0 bytes. When this entry is subsequently read, `named` aborts. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-93468	The OAKlouds developed by HGiga has an Arbitrary File Read vulnerability. Unauthenticated remote attackers can exploit Relative Path Traversal to read arbitrary system files.	7.5	More Details
CVE-2026-1168	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 18.4.6 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an unauthenticated user to cause denial of service due to improper resource allocation limits in the GraphQL complexity calculation logic.	7.5	More Details
CVE-2026-27557	An unauthenticated remote attacker can exploit a path traversal vulnerability in the /index.php/view_uploaded_iodd_file endpoint allowing the SSH server's private keys to be read.	7.5	More Details
CVE-2026-79993	The `deleteContainer` opcode (0x14/20) is processed without verifying the caller's ACL permissions, allowing any authenticated client to delete specific znodes in the data tree regardless of the ACL restrictions on the znode or its parent. This opcode is considered internal-only and the official client doesn't have API for it, but a client that can open a plain TCP session on the ZooKeeper client port (2181 by default) - with NO authentication and NO ACL permissions - can delete any empty persistent znode (including regular persistent nodes, container nodes, and TTL nodes) by issuing the raw protocol OpCode deleteContainer (20). The deleteContainer request path completely skips both the session check and the DELETE ACL check that are enforced by the regular delete (OpCode 2) path. This is an authorization bypass / ACL enforcement bug. This issue affects Apache ZooKeeper: from 3.9.0 through 3.9.5, from 3.8.0 through 3.8.6. Users are recommended to upgrade to version 3.9.6 or 3.8.7, which fixes the issue.	7.5	More Details
CVE-2026-90293	In the Linux kernel, the following vulnerability has been resolved: IB/isert: post the full-feature receive buffers after session registration isert_put_login_tx() posts the full-feature receive buffers before __transport_register_session() runs, so an initiator that does not wait for the final Login Response can still have a SCSI command executed against an se_session whose se_tpg is NULL - the same oops as the previous patch, at target_submit+0xbe. Post them from isert_get_rx_pdu(), which the previous patch already uses to send that response, and post them before that send: the receive queue is filled at the moment the initiator is told it may use it. Allocating there keeps the existing property that a memory allocation failure cannot happen once the final Login Response is on the wire. The receive queue is already empty between the final Login Request and isert_post_recvm(); this moves the second point later, from a median of 92 us to 172 us over 1200 logins. Only an initiator that sends before it has been told to can reach that window, and on IB and RoCE its send is retried there until the buffers appear - isert_rdma_accept() asks for rnr_retry_count = 7. iWARP has no RNR flow control, so there the same send terminates the connection instead. Measured over rxe, 400 login cycles per run, with an initiator that does not wait: an instrumented build counted no entries to isert_recv_done() before the buffers are posted in 10 runs, where that initiator oopsed 8 of 10 unpatched runs and 5 of 10 with only the previous patch. Not tested: iWARP, discovery sessions over iSER, and real HCAs.	7.5	More Details
CVE-2026-73455	On affected platforms running Arista EOS with Open Shortest Path First version 3 (OSPFv3) configured, a specially crafted packet can cause the OSPFv3 agent to restart unexpectedly.	7.5	More Details
CVE-2026-73439	On affected platforms running Arista EOS, if OpenConfig is configured and running a gNMI server on the system, and if gNSI Pathz is configured and a gNSI Pathz policy is present on the system, then gNMI may fail to correctly enforce the rules in this policy if both a group rule and a user rule for the same path is present in the policy. Under certain conditions, this can lead to an authenticated user gaining unauthorized permission to read or write gNMI paths that the Pathz policy is intended to restrict.	7.5	More Details
CVE-2026-59969	Apache ZooKeeper quorum TLS fails to enforce peer hostname verification in FIPS-mode deployments. When sslQuorum=true, zookeeper.fips-mode=true, ssl.quorum.hostnameVerification=true, and ssl.quorum.clientHostnameVerification=true are enabled, the Java SSLSocket quorum path accepts a CA-trusted peer certificate whose SAN does not match the connected host. A malicious or misissued peer certificate can therefore join quorum traffic, participate in leader election, and enter replication flows. Users are recommended to upgrade to version 3.8.7 or 3.9.6, which fixes the issue.	7.5	More Details
CVE-2026-	snappy-java through 1.1.10.8 contains a buffer overflow vulnerability in Snappy.compress(ByteBuffer, ByteBuffer) that writes past the end of the destination buffer. Attackers can supply incompressible data that exceeds the destination buffer's	7.5	More Details

93452	remaining capacity, corrupting off-heap memory and causing JVM termination.		
CVE-2026-59739	Information disclosure via SetWatches reconnect replay in Apache ZooKeeper due to missing ACL check. An attacker can discover ACL-restricted paths by registering exists-watches on non-existent paths, then reconnecting after the paths are created with restricted ACLs. Issue is caused by incomplete fix for CVE-2024-23944 (ZOOKEEPER-4799). The fix added ACL checking to WatchManager.triggerWatch(). However, DataTree.setWatches() — the SetWatches/SetWatches2 reconnect replay handler — still calls watcher.process(event) with null ACL, bypassing the check entirely. It's important to note that only the path is exposed by this vulnerability, not the data of znode, but since znode path can contain sensitive information like user name or login ID, this issue is potentially critical. Users are recommended to upgrade to version 3.9.6, 3.8.7 which fixes the issue.	7.5	More Details
CVE-2026-93450	go-openapi/swag jsonutils before 0.27.1 contains a stack overflow vulnerability in ordered JSON parsing and serialization due to unbounded recursion with no depth limit. Remote unauthenticated attackers can submit deeply nested JSON documents to services accepting OpenAPI specifications, causing fatal stack overflow that terminates the process and all in-flight requests.	7.5	More Details
CVE-2026-81634	In NLnet Labs Unbound up to and including 1.26.0, a 255 length query name with a large TCP response can lead to a heap buffer overflow during the RRSet canonicalisation routine. This is caused by missing to add the first owner name into the buffer length check. A malicious actor operating a malicious name server or tampering with an incoming response to Unbound (canonicalisation happens before DNSSEC validation), can trigger the vulnerability.	7.5	More Details
CVE-2026-85917	Server-side request forgery (ssrf) in Azure AI Foundry allows an unauthorized attacker to elevate privileges over a network.	7.5	More Details
CVE-2026-93435	redis-parser through 3.0.0 contains a denial of service vulnerability in the RESP protocol parser that allows malicious Redis endpoints to crash the client process through unbounded recursion on nested arrays. Attackers can send crafted RESP byte streams with repeated array headers that exhaust the V8 call stack, causing an uncaught RangeError that terminates the Node.js process without triggering error handling callbacks.	7.5	More Details
CVE-2026-89063	The Online Scheduling and Appointment Booking System – Bookly plugin for WordPress is vulnerable to Insecure Direct Object Reference in all versions up to, and including, 28.1 via the 'conversation_id' parameter due to missing validation on a user controlled key. This makes it possible for unauthenticated attackers to read the full AI booking conversation transcript of any customer — leaking names, email addresses, phone numbers, and appointment details echoed by the assistant — and inject arbitrary messages into any victim conversation that are subsequently replayed to the Cloud AI worker along with the full private history. Because AI conversations are stored with no owner, user, or session identifier and conversation IDs are sequential integers, an unauthenticated attacker can enumerate all customer conversations simply by incrementing the conversation_id parameter.	7.5	More Details
CVE-2026-93436	vLLM through 0.29.0 fails to properly clean up decode-side metadata for rejected inference requests in prefill/decode disaggregated deployments. Remote attackers can submit requests with max_tokens=0 to exhaust decode-worker memory without bound until the worker restarts.	7.5	More Details
CVE-2026-89028	MikroTik RouterOS before 7.24 contains a heap memory corruption vulnerability in the userspace SMB daemon that allows remote attackers to corrupt adjacent heap memory by supplying a crafted uniPwdLen value in the SMB1 SessionSetupAndX handler. An attacker can send a malformed SMB1 request with a uniPwdLen field that triggers an integer underflow, causing the resulting value to be used as the copy length in a memory copy operation into a smaller heap buffer, corrupting adjacent heap memory.	7.5	More Details
CVE-2026-90294	In the Linux kernel, the following vulnerability has been resolved: IB/isert: delay the final Login Response until the session is registered isert_put_login_tx() puts the final Login Response on the wire before __transport_register_session(), which iscsi_post_login_handler() reaches only after iscsi_target_do_login() returns. An initiator that issues a SCSI command as soon as it sees that response can have it executed against an se_session whose se_tpg is still NULL, and the ib-comp-wq worker oopses on the NULL dereference. Oops: general protection fault, probably for non-canonical address 0xdfffffc000000000f: 0000 [#1] SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000078-0x000000000000007f] CPU: 0 UID: 0 PID: 178 Comm: kworker/0:1H Not tainted 7.2.0-rc5-V2CTL-gf5098b6bae76 #10 PREEMPT(lazy) Hardware name: QEMU Ubuntu 24.04 PC v2 (i440FX + PIIX, arch_caps fix, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Workqueue: ib-comp-wq ib_cq_poll_work RIP: 0010:target_submit+0xbe/0x390 Code: fa 48 c1 ea 03 80 3c 02 00 0f 85 89 02 00 00 48 b8 00 00 00 00 00 fc ff df 4d 8b 64 24 18 49 8d 7c 24 78 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 5a 02 00 00 48 8d 7b 78 4d 8b 6c 24 78 48 b8 00 RSP: 0018:ffff8881058cfa78 EFLAGS: 00010206 RAX: dfffffc0000000000 RBX: ffff88810c78c6f0 RCX: ffffffff964bb363 RDX: 000000000000000f RSI: 00000000fffffe00 RDI: 0000000000000078 RBP: 1ffff11020b19f52 R08: 0000000000000001 R09: ffffed1020b19f52 R10: 0000000000000003 R11: ffff88810596c000 R12: 0000000000000000 R13: ffff88810c61b000 R14: ffff88810c6a3400 R15: ffff88810c61b044 FS: 0000000000000000(0000) GS:ffff8881822b2000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f1f1b83c000 CR3: 000000006fe72001 CR4: 0000000000770ef0 PKRU: 55555554 Call Trace: <TASK> ? __pfx__raw_spin_lock_bh+0x10/0x10 ? __pfx_target_submit+0x10/0x10 ? mutex_lock+0x81/0xe0 ? __pfx_mutex_lock+0x10/0x10 ? iscsit_execute_cmd+0x650/0x850 iscsit_sequence_cmd+0x186/0x3d0 iscsit_process_scsi_cmd+0x87/0x300 isert_recv_done+0x1002/0x2390 ? __pfx_isert_recv_done+0x10/0x10 ? rxe_poll_cq+0x253/0x3d0 ? finish_task_switch.isra.0+0x1dc/0xa70 __ib_process_cq+0xe1/0x390 ib_cq_poll_work+0x46/0x150 process_one_work+0x633/0x1030 ? assign_work+0x11d/0x370 worker_thread+0x45b/0xd10 ? __pfx_worker_thread+0x10/0x10 ? __pfx_worker_thread+0x10/0x10 kthread+0x2c6/0x3b0 ? recalc_sigpending+0x15c/0x1e0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x36e/0x5a0 ? __pfx_ret_from_fork+0x10/0x10 ? __switch_to+0x572/0xdd0 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Modules linked in: ---[end trace 0000000000000000]--- Delay the final Login Response instead. isert_get_rx_pdu() runs from iscsi_target_rx_thread() after conn->rx_login_comp, completed by iscsi_post_login_handler() after __transport_register_session(); iscsi-TCP and cxgbit already take PDUs from that thread, isert alone does not. The buffers are still posted first, so the initiator's first command does not meet an empty receive queue and nothing depends on RNR flow control, and the header and payload live in isert_conn, not in the struct iscsi_login that iscsi_target_nego_release() frees first. Over rxe, 400 login cycles per run, the oops appeared in 10 of 20 unpatched runs and in none of 20 runs with this patch. An initiator that never waits is handled by the next patch. Not tested: iWARP, discovery sessions over iSER, and real HCAs.	7.5	More Details
CVE-2026-	A flaw was found in the SAML Redirect Binding implementation of Keycloak, an open-source identity and access management solution. The issue occurs because the custom DEFLATE compression and decompression helpers fail to release native zlib memory after use. An unauthenticated attacker can exploit this by sending repeated malformed SAML requests, leading to	7.5	More Details

18212	native memory exhaustion and a denial of service.		
CVE-2026-77692	An attacker can cause `named` to abort by sending a crafted DNS-over-HTTPS request with a cryptographically invalid SIG(0) record, and then closing the transport connection prematurely. This issue affects BIND 9 versions 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-92129	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not check calls from sandboxed scripts to methods added dynamically to a class at runtime, allowing attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute code outside the sandbox.	7.5	More Details
CVE-2026-19666	On a resolver configured to use ``dns64``, if an applicable answer from the authoritative server is malformed in a specific way, the resolver `named` process will exit unexpectedly. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-81736	If a BIND resolver has cached a tree of SVCB/HTTPS AliasMode records, and is then queried for the root of that tree, the resolver will spend disproportionate CPU time constructing the response. This issue affects BIND 9 versions 9.18.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.18.11-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	7.5	More Details
CVE-2026-92128	Jenkins Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier downloads a JAR file specified by URL twice, confirming the approval of the first download and loading the classpath entries from the second, allowing attackers able to define classpath entries to execute arbitrary code in the context of the Jenkins controller JVM.	7.5	More Details
CVE-2026-2380	On affected platforms running Arista EOS with OpenConfig-related services (i.e., gNMI, gNSI, RESTCONF and NETCONF), sensitive requests and responses may be unintentionally logged. These may be stored on the local EOS device or recorded on remote accounting servers. Note that gRPC-based streaming via Streaming Telemetry Agent to CloudVision is not affected by this vulnerability. Examples of sensitive information include: - Sensitive CLI commands (e.g., "username bob secret myPass") - Sensitive OpenConfig YANG leafs (e.g., "system/aaa/global/tacacs/config/secret-key") This issue was discovered internally by Arista, and the company is not aware of any malicious exploitation of this vulnerability in customer networks.	7.4	More Details
CVE-2026-61590	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, djust's observability endpoints expose live view/session state and a remote method-invocation surface (`eval_handler`). The localhost restriction was an opt-in middleware that the documented setup omits; the views themselves enforced only `DEBUG`. In the misconfigured-but-documented scenario (DEBUG on, middleware not installed) a non-localhost client could read live application state and invoke handlers remotely. This issue is fixed in djust 1.0.7. The localhost restriction is enforced in-view on every observability endpoint (no longer dependent on a separately-installed middleware), and `eval_handler` is restricted; gated requests receive a non-disclosing response. As a workaround, ensure `DEBUG=False` in production, and do not expose the observability endpoints to untrusted networks.	7.4	More Details
CVE-2026-93039	In the Linux kernel, the following vulnerability has been resolved: ASoC: meson: Keep link pointers valid on realloc failure meson_card_reallocate_links() grows the DAI link and private data arrays with two consecutive krealloc() calls and updates the owner pointers only after both calls have succeeded. A successful krealloc() may move the data: it frees the old block and returns a new one. When that happens for the link array and the second krealloc() then fails, card->dai_link still points to the block that krealloc() already freed, and the error path frees the new block too. The probe error path then calls meson_card_clean_references(), which dereferences card->dai_link and kfree()s it again, resulting in a use-after-free and a double free. Commit card->dai_link and card->num_links right after the first krealloc() succeeds, so the pointer always refers to a valid allocation that meson_card_clean_references() can walk and free. krealloc() with __GFP_ZERO zero-initializes the added entries, so walking them on the error path is safe. With both failure paths reduced to a plain return, drop the goto labels and the error message.	7.4	More Details
CVE-2026-90997	A flaw was found in Keycloak. When deployed in stateless mode with MySQL or MariaDB, a mismatch in row-count semantics between the database driver and Keycloak's application logic allows an attacker to bypass replay protection. This vulnerability enables an attacker who intercepts single-use security artifacts, such as JWT client assertions, DPoP proofs, or one-time password (TOTP) codes, to replay them. Successful exploitation grants unauthorized access to the token endpoint or login flow.	7.4	More Details
CVE-2026-61592	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, SSE sessions were keyed solely by a client-chosen `session_id` with no binding to the authenticated user — a control the WebSocket transport has but that was dropped on SSE. An attacker who learns (or a victim who leaks) a `session_id` could connect to the message endpoint and dispatch event handlers that execute with the victim's identity and state. This is fixed in djust 1.0.7. Each SSE session is bound to its owning principal at creation and cross-principal access is rejected; SSE session creation is additionally capped per principal. As a workaround, disable the SSE transport.	7.4	More Details
CVE-2026-92299	@jitsi/electron-sdk before 10.0.5 exposes getDesktopSources() via contextBridge without requiring an active getDisplayMedia() picker, allowing any script in the meeting page to enumerate screens and windows. Attackers can call the jitsi-screen-sharing-get-sources IPC route to retrieve desktop thumbnails at arbitrary resolution without user consent or operating system permission prompts.	7.4	More Details
CVE-2026-90427	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Don't fall back to a freed smmu after devm_krealloc() __tegra241_cmdqv_probe() uses devm_krealloc() to grow @smmu into the larger tegra241_cmdqv, which frees the original @smmu once it relocates. A failure after that returned NULL, and the caller then dereferenced the freed @smmu on its fallback path. Return an int and take @smmu by reference instead, then update *smmu to the reallocated pointer after devm_krealloc() succeeds, so the caller and its fallback path both use the live @smmu rather than the freed original.	7.4	More Details
CVE-2026-81446	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Server-Side Request Forgery (SSRF) vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Server-side request forgery.	7.4	More Details
CVE-2026-73459	On affected platforms running Arista EOS with IS-IS configured, an unauthenticated attacker who can inject a specially crafted IS-IS LSP PDU can cause the legitimate LSP to be unexpectedly purged from the IS-IS link-state database. This may result in traffic loss.	7.4	More Details
CVE-			

2026-24081	Transient DOS when processing a channel map with insufficient used channels and adaptive frequency hopping is fully enabled.	7.4	More Details
CVE-2026-42784	A flaw was found in sequoia-openpgp. The library incorrectly infers key flags for older certificates when a key flags subpacket is missing, leading to a discrepancy in how key capabilities are viewed. This key flag confusion allows an attacker to bypass the back-signature check. Consequently, an attacker can illegitimately bind an arbitrary subkey to their own certificate and forge signatures, completely compromising cryptographic integrity.	7.4	More Details
CVE-2026-90288	In the Linux kernel, the following vulnerability has been resolved: phy: renesas: rcar-gen2: Fix double of_node_put on phy creation failure for_each_child_of_node_scoped() releases the node reference on scope exit, so the explicit of_node_put(np) in the devm_phy_create() error path drops it twice. Drop the redundant of_node_put() and let the scoped cleanup handle it.	7.4	More Details
CVE-2026-73446	On affected platforms running Arista EOS with IS-IS configured on a broadcast interface, an unauthenticated attacker can send a crafted IS-IS Hello Protocol Data Unit (PDU) that causes the device to tear down an established IS-IS adjacency. This may result in traffic disruption and loss of IP reachability for prefixes advertised through that adjacency.	7.4	More Details
CVE-2026-78501	Improper neutralization of special elements used in a command ('command injection') in Microsoft 365 Copilot's Business Chat allows an unauthorized attacker to disclose information over a network.	7.4	More Details
CVE-2026-92913	AVideo through commit c3edcc274c389816d434acadac07ee78eaf330c1 uses a cryptographically weak pseudo-random number generator when creating account activation / login pairing codes. getRandomCode() in objects/functions.php derives the code entirely from uniqid() (sprintf('%08x%05x', seconds, microseconds)) with a single non-CSPRNG rand() character used only as padding, reducing the code space to roughly 36 x 10^6 (~2^25) values for a known generation second. Because plugin/API/set.json.php?APIName=login_code can be called without authentication, it also serves as an oracle for the server's exact microtime. An unauthenticated remote attacker who guesses a valid, unexpired code (codes expire after 10 minutes) can redeem it at plugin/API/get.json.php?APIName=login_code to obtain the target account's email address and a User::getUserHash(users_id, '+1 year') value, a credential accepted in place of the account password for one year, resulting in account takeover. No patched version is available.	7.4	More Details
CVE-2026-90229	In the Linux kernel, the following vulnerability has been resolved: nvme-apple: Destroy the admin queue on removal The admin queue is allocated with blk_mq_alloc_queue() but never destroyed. nvme_free_ctrl() only drops the last reference and blk_mq_exit_queue() and blk_sync_queue() never run: the hctx is never moved to q->unused_hctx_list and the timeout timer and work stay armed on a queue that is about to be freed which will eventually oops inside blk_mq_timeout_work(). This can only be triggered when the controller fails to come up and is then immediately torn down again which is why no one ever ran into this before. Let's just copy what the pcie driver does: unquiesce and destroy the admin queue before nvme_uninit_ctrl(). With this the following WARN followed by a panic no longer happens: WARNING: block/blk-mq.c:4390 at blk_mq_release+0x194/0x238, CPU#4: kworker/u34:4/119 CPU: 4 UID: 0 PID: 119 Comm: kworker/u34:4 Not tainted 7.2.0-rc1-dirty #248 PREEMPT Hardware name: Apple Mac mini (M1, 2020) (DT) Workqueue: nvme-wq apple_nvme_remove_dead_ctrl_work pstate: 61400005 (nZCv daif +PAN -UAO -TCO +DIT -SSBS BTYPE=--) pc : blk_mq_release+0x194/0x238 lr : blk_mq_release+0x58/0x238 sp : ffffc000833a3b50 x29: ffffc000833a3b50 x28: ffff80001d0450f8 x27: ffff800020c95200 x26: 0000000000000088 x25: 0000000000000000 x24: ffff800020f36805 x23: 0000000000000000 x22: ffffc00081a86878 x21: ffff800020be9c60 x20: 0000000000000000 x19: ffff800022501698 x18: 000000000000000a x17: 7365757165722066 x16: 666f7265776f7020 x15: 0000000000000000 x14: 0000000000000028 x13: 0000000000004def x12: 0000000000000003 x11: 0000000000000000 x10: 0000000000000000 x9 : ffffc000805b4fc8 x8 : ffffc00081915820 x7 : ffffc00081c4f3c8 x6 : 0000000000000001 x5 : 0000000000000004 x4 : ffff800022498d80 x3 : ffffc000833a3b14 x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff800022501698 Call trace: blk_mq_release+0x194/0x238 (P) blk_put_queue+0x8c/0xf0 nvme_free_ctrl+0x4c/0x260 device_release+0x44/0x128 kobject_put+0xa0/0x120 put_device+0x1c/0x40 nvme_uninit_ctrl+0x48/0x60 apple_nvme_remove+0x54/0xb0 platform_remove+0x28/0x40 device_remove+0x54/0x98 device_release_driver_internal+ device_release_driver+0x20/0x38 apple_nvme_remove_dead_ctrl_wor process_one_work+0x1f4/0x770 worker_thread+0x1b8/0x360 kthread+0x140/0x160 ret_from_fork+0x10/0x20 irq event stamp: 448 hardirqs last enabled at (447):in_unlock_irqrestore+0x74/0x80 hardirqs last disabled at (448): [<ffffc000811cf5c0>] el1_brk64+0x20/0x60 softirqs last enabled at (0): [ess+0xb28/0x2698 softirqs last disabled at (0): [<0000000000000000>] 0x0 ---[end trace 0000000000000000 Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000096000005 EC = 0x25: DABT (current EL), SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault Data abort info: ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagA GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [0000000000000000] user address Internal error: Oops: 0000000096000005 [#1] SMP CPU: 7 UID: 0 PID: 54 Comm: kwor 7.2.0-rc1-dirty #248PREEMPT Tainted: [W]=WARN Hardware name: Apple Mac mini (M1, 2020) (DT) Workqueue: kblockd blk_mq_timeou pstate: 01400005 (nzcv daif +PAN -UAO -TCO +DIT -SSBS BTYPE=--) pc : percpu_ref_tryget_many.cons lr : percpu_ref_tryget_many.constprop.0+0xc0/0x168 sp : ffffc000829cbce0 x29: ffffc000829cbce0 x28: ffff800020be9f48 x27: ffff800013e503c0 x26: 0000000000000108 x25: 000009c05 x23: 0000000000000000 x22: ffffc000819f5000 x21: ffff800020be9f48 x20: ffff8001deda4808 x19: ffff8000a x17: 00000000580e1fac x16: ffffc00082bbbb7c x15: 0000000000000000 x14: 0000000000000028 x13: 000000001 x11: 0000000000000000 x10: 0000000000000000 x9 : ffffc000829cbc20 x8 : ---truncated---	7.4	More Details
CVE-2026-25281	Transient DOS when processing large or numerous request buffers without sufficient memory allocation validation.	7.4	More Details
CVE-2026-25294	Transient DOS while parsing frame during channel usage.	7.4	More Details
CVE-2026-20222	A vulnerability in the EIGRP implementation in Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, adjacent attacker to cause the device to reload unexpectedly, resulting in a denial of service (DoS) condition. This vulnerability is due to improper resource management when handling EIGRP update messages. An attacker could exploit this vulnerability by sending crafted EIGRP updates at a high rate to an affected device. A successful exploit could allow the attacker to trigger a memory leak that will eventually cause the affected device to reload unexpectedly.	7.4	More Details

CVE-2026-76154	A stored cross-site scripting vulnerability in the Geomap panel's MapLibre base layer allows a user with the Editor role to execute arbitrary JavaScript in another user's session by hosting a malicious style configuration, enabling escalation to Org Admin.	7.3	More Details
CVE-2026-93177	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm/powerplay: bounds-check voltage index in Vega10 lookup vddInd, vddciInd and mvddInd from VBIOS-parsed tables index into vddc, vddci and vddmem lookup tables without bounds checks across nine sites. Return -EINVAL when any index is out of range.	7.3	More Details
CVE-2026-80356	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Exposure of Sensitive Information to an Unauthorized Actor vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Information exposure.	7.3	More Details
CVE-2026-89910	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Fix uninitialized stack variable issue with dmsintc Variable vector[] is declared on stack in function dmsintc_inject_irq() and sometimes it is used without initialized. Here fix this issue.	7.3	More Details
CVE-2026-92405	A security vulnerability has been detected in SourceCodester Inventory and Monitoring System 1.0. The affected element is an unknown function of the file /index.php. Such manipulation of the argument Username leads to sql injection. The attack may be launched remotely. The exploit has been disclosed publicly and may be used.	7.3	More Details
CVE-2026-92718	Nuclei versions before 3.11.1 cache template signature verification based only on file modification time without content checksums. Attackers can replace verified templates with unsigned malicious content and restore the original modification time to bypass signature checks and execute arbitrary operating system commands.	7.3	More Details
CVE-2026-92926	A vulnerability has been found in code-projects Matrimonial System 1.0. This vulnerability affects the function writepartnerprefs of the file /partner_preference.php. Such manipulation of the argument education leads to sql injection. The attack can be executed remotely. The exploit has been disclosed to the public and may be used.	7.3	More Details
CVE-2026-92522	In the Linux kernel, the following vulnerability has been resolved: ACPI: processor: validate MADT IOAPIC entry bounds The IOAPIC hotplug lookup parses both MADT and _MAT records directly. The MADT walk previously used a subtable's declared length to advance the cursor after only locating a generic header. The _MAT path likewise passed a generic header to the IOAPIC helper. Validate that a current record has a complete generic header, that its declared length is contained in the available record range, and that a typed IOAPIC record contains the full fixed IOAPIC body before reading its fields. Use the same relation for both MADT and _MAT provider paths.	7.3	More Details
CVE-2026-92406	A vulnerability was detected in SourceCodester Inventory and Monitoring System 1.0. The impacted element is an unknown function of the file /admins/assessments/databank/btn_functions.php?action=add. Performing a manipulation of the argument difficulty_id results in sql injection. Remote exploitation of the attack is possible. The exploit is now public and may be used.	7.3	More Details
CVE-2026-92362	A vulnerability was detected in ag-ui-protocol ag-ui 1.0. This impacts an unknown function of the file crates/ag-ui-client/src/sse.rs of the component SSE Frame Parser. Performing a manipulation results in resource consumption. The attack is possible to be carried out remotely. The pull request to fix this issue awaits acceptance.	7.3	More Details
CVE-2026-54634	Hamlib is a ham radio control library for radios, rotators, and amplifiers. Prior to 4.7.2, the unauthenticated rigctld send_raw command on TCP port 4532 reaches rigctl_send_raw() in tests/rigctl_parse.c, which writes a NUL byte at buf[buf_len + 1] outside its 200-byte stack buffer, and rig_send_raw() in src/rig.c, which copies reply_len - 1 bytes instead of the actual nbytes received. A remote client can send the CR terminator with a short payload to trigger both flaws in one command under the default no-password configuration. The out-of-bounds write can crash the daemon or corrupt adjacent stack memory, while the oversized copy can return up to 198 bytes of uninitialized stack data to the client. This issue is fixed in version 4.7.2.	7.3	More Details
CVE-2026-90141	In the Linux kernel, the following vulnerability has been resolved: ipvs: fix integer overflow in ftp helper port/address parsing ip_vs_ftp_get_addrport() accumulates decimal digits into a __u16 (hport) and into unsigned char (p[]) without checking for overflow. A crafted FTP PASV/EPSV response with an over-long port or address octet wraps the value, so the helper configures the data connection with a truncated port/address. The netfilter conntrack FTP helper had the same defect, fixed in commit 2b413fc689ba ("netfilter: nf_conntrack_ftp: avoid u16 overflows"). Apply the equivalent fix here: widen the port accumulator to u32 and reject values above 65535, and reject address octets above 255.	7.3	More Details
CVE-2026-92366	A vulnerability was determined in code-projects Matrimonial System 1.0. This affects an unknown part of the file /search.php of the component Regular Search. This manipulation of the argument sex/mothertongue/maritialstatus/country/state/religion/agemin/agemax causes sql injection. The attack can be initiated remotely. The exploit has been publicly disclosed and may be utilized.	7.3	More Details
CVE-2026-93331	A vulnerability was identified in GPAC 26.08-DEV. This vulnerability affects the function gf_rtp_parse_ttxt of the file src/ietf/rtp_depacketizer.c of the component RTP Depacketizer. Such manipulation of the argument size leads to out-of-bounds read. It is possible to launch the attack remotely. Upgrading to version abi-16.26 is able to resolve this issue. The name of the patch is 6bb0f64b4d1039c0fecd14ee2c1ee861d8661a68. The affected component should be upgraded.	7.3	More Details
CVE-2026-25284	Information Disclosure when a pointer is reused after being deallocated.	7.3	More Details
CVE-2026-92915	WWBN AVideo through commit e01e41ecc (no patched version available) contains a broken access control flaw in objects/userVerifyEmail.php. The script disables the login requirement ($global['ignoreUserMustBeLoggedIn'] = 1), takes users_id directly from the query string, and calls User::sendVerificationLink() with no session requirement, no CSRF/global token, no relationship check between caller and target, and no enforceRateLimit() call. The only intended throttle is keyed to the caller's own session, so cookie-less requests are never limited. An unauthenticated remote attacker can therefore cause an arbitrary number of verification emails to be sent to any account ID, and can enumerate accounts and their verification status from the three distinct JSON responses ("Verification Sent", "Already verified", "Unknown error"). In addition, createVerificationCode() invokes $user->setRecoverPass() and saves the user, so each anonymous request writes a live password-recovery token onto the targeted account; that token is embedded in base64 in the verification link emailed to the account owner and is accepted by objects/userRecoverPassSave.json.php as the credential for setting a new password.	7.3	More Details

CVE-2026-92215	A vulnerability has been found in a2ui-project a2ui up to 0.10.7. Affected by this vulnerability is the function httpx.get of the file agent_sdks/python/a2ui_agent/src/a2ui/extensions/file_resolve/file_resolver.py of the component FileResolver. The manipulation leads to server-side request forgery. It is possible to initiate the attack remotely. The identifier of the patch is 2bb8423060308bbdea8ba468dabed4fc256d18ea. To fix this issue, it is recommended to deploy a patch.	7.3	More Details
CVE-2026-92380	A flaw has been found in WuzhiCMS up to 4.1.0. The impacted element is the function ckditor::saveRemote of the file coreframe/app/attachment/index.php of the component Remote Image Fetch. This manipulation of the argument source[] causes server-side request forgery. The attack can be initiated remotely. The exploit has been published and may be used. The project was informed of the problem early through an issue report but has not responded yet.	7.3	More Details
CVE-2026-66269	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Protection mechanism bypass.	7.3	More Details
CVE-2026-92401	A vulnerability was identified in ChangeWeDer crm up to c07bd4c97141521af6475034bc58523beed51bbd. This vulnerability affects the function top.upstudy.crm.utils.LoginUserUtil.releaseUserIdFromCookie. The manipulation leads to improper authentication. The attack can be initiated remotely. Continious delivery with rolling releases is used by this product. Therefore, no version details of affected nor updated releases are available. The project was informed of the problem early through an issue report but has not responded yet.	7.3	More Details
CVE-2026-92399	A vulnerability was determined in GPAC 26.07.0. This affects the function rmt_client_handle_ws_frame of the file src/utils/rmt_ws.c of the component WebSocket Handler. Executing a manipulation of the argument payload_size can lead to heap-based buffer overflow. It is possible to launch the attack remotely. The exploit has been publicly disclosed and may be utilized. Upgrading to version abi-16.26 is able to mitigate this issue. This patch is called 37bccbb30cf53a0e1a084cea9a1ce422b3ddfe12. Upgrading the affected component is recommended.	7.3	More Details
CVE-2026-71179	Dell Update Package Framework, versions prior to 26.07.03, contains an Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of privileges.	7.3	More Details
CVE-2026-81440	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Use of Hard-coded Credentials vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Unauthorized access.	7.3	More Details
CVE-2026-85569	The Tutor LMS WordPress plugin before 4.0.8 does not correctly determine whether an incoming request is addressed to its own REST API, and does not enforce the permission recorded against an API credential, allowing the holder of a read-only key to act as the administrator account that issued it.	7.2	More Details
CVE-2026-54647	CubeCart is an ecommerce software solution. Prior to 6.7.5, admin/sources/settings.index.inc.php directly concatenates the administrator-controlled download_expire POST parameter into a raw UPDATE statement for CubeCart_downloads without numeric validation. An authenticated administrator can supply a comma-delimited value that changes the SET clause because HTML sanitization does not neutralize SQL syntax, allowing manipulation of database columns and potentially other data within the application's database privileges. This issue is fixed in version 6.7.5.	7.2	More Details
CVE-2026-81477	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Heap-based Buffer Overflow vulnerability. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Code execution.	7.2	More Details
CVE-2026-18595	The WP-Lister Lite for eBay plugin for WordPress is vulnerable to Stored Cross-Site Scripting via AJAX Cron Handler Request Parameter in all versions up to, and including, 3.8.9 due to insufficient input sanitization and output escaping. This makes it possible for unauthenticated attackers to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	7.2	More Details
CVE-2026-86865	Tanium addressed a SQL injection vulnerability in Asset.	7.2	More Details
CVE-2026-87024	Tanium addressed a SQL injection vulnerability in Asset.	7.2	More Details
CVE-2026-81480	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Stack-based Buffer Overflow vulnerability. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Code execution.	7.2	More Details
CVE-2026-17526	Keycloak is an open-source identity and access management solution. A vulnerability was discovered where a user with the impersonation role can impersonate a realm administrator. This allows the attacker to gain full administrative control over the realm, including the ability to manage users, clients, and roles.	7.2	More Details
CVE-2026-81810	The All-in-One WP Migration and Backup WordPress plugin before 7.111 does not perform any capability check on several of its AJAX actions, gating them only on an installation-wide secret which it discloses to any user permitted to export the site, allowing such a user to import an arbitrary site archive and gain administrator access. Exploitation requires an administrator to have granted the export capability to a role that does not hold the All-in-One WP Migration and Backup WordPress plugin before 7.111's own import capability, which is not a default configuration.	7.2	More Details
CVE-2026-27563	A high-privileged remote attacker can exploit a command injection vulnerability in the /api/datastorage/data endpoint by sending a crafted GET request with admin credentials allowing execution of commands with root privileges on the device.	7.2	More Details
CVE-2026-	lxc-ci contains continuous integration and image-build scripts for LXC. Prior to the 2026-05-28 Arch Linux image publication, images built from images/archlinux.yaml retain the same pacman local-signing private key in /etc/pacman.d/gnupg and redistribute it to every container or virtual machine created from that image. An attacker who controls an HTTP package mirror	7.2	More

52727	or can intercept mirror traffic can use the shared pacman signing private key to sign modified packages that affected clients accept as trusted. Installing those packages permits arbitrary code execution as root on the client system. This issue is fixed in Arch Linux images published on or after 2026-05-28.		Details
CVE-2026-76550	The WP Import Export Lite WordPress plugin before 3.9.34 does not validate a user-supplied output path when writing export files, allowing users granted its export permission to write files with arbitrary names to arbitrary locations on the server, leading to remote code execution.	7.2	More Details
CVE-2026-81445	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Privilege Management vulnerability. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Elevation of privileges.	7.2	More Details
CVE-2026-92980	HortusFox-Web prior to version 6.1 contains a remote code execution vulnerability that allows authenticated administrators to execute arbitrary OS commands as the web server user by abusing the Import/Export functionality. Attackers can leverage the Import/Export feature, which is intended solely for data portability, to deploy and execute malicious code on the underlying application server host.	7.2	More Details
CVE-2026-76424	A vulnerability in the REST API of Cisco ISE could allow an authenticated, remote attacker to upload or copy arbitrary files on an affected device. This vulnerability is due to insufficient validation in file operations. An attacker could exploit this vulnerability by uploading a file with a crafted path. A successful exploit could allow the attacker to upload files to arbitrary locations and execute arbitrary commands as root on the affected device. To exploit this vulnerability, the attacker must have valid administrative credentials.	7.2	More Details
CVE-2026-82310	Apache Airflow FAB provider: deactivating a user account does not stop tokens issued to that account before deactivation. Password authentication correctly rejects the disabled account, but the Core API continues to accept an existing, unexpired token naming it, and lets that token mint a replacement — so the account keeps its role-scoped access indefinitely after an administrator has disabled it. The user replays their own legitimate credential; no signature forgery or privilege escalation is involved, and the access stays within the roles the account already held. Affects deployments using Airflow 3 with the FAB auth manager and Core API token authentication, where an administrator deactivates an account whose row remains in the database and whose previously issued token has not expired. The trigger is administrative deactivation as a containment action, which silently fails to contain. Users of apache-airflow-providers-fab are recommended to upgrade to version 3.9.0 or later, which rejects tokens naming a deactivated account.	7.2	More Details
CVE-2026-27564	A high-privileged remote attacker can exploit a command injection vulnerability in the /api/datastorage/data endpoint by sending a PUT request with admin credentials allowing execution of commands with root privileges on the device.	7.2	More Details
CVE-2026-27560	A high-privileged remote attacker can exploit a command injection vulnerability in the /api/status/data endpoint by sending a crafted DELETE request with admin credentials allowing execution of commands with root privileges on the device.	7.2	More Details
CVE-2026-27561	A high-privileged remote attacker can exploit a command injection vulnerability in the /api/iodd/config endpoint by sending a crafted GET request with admin credentials allowing execution of commands with root privileges on the device.	7.2	More Details
CVE-2026-27562	A high-privileged remote attacker can exploit a command injection vulnerability in the /api/iodd/config endpoint by sending a crafted PUT request with admin credentials allowing execution of commands with root privileges on the device.	7.2	More Details
CVE-2026-76551	The WP Import Export Lite WordPress plugin before 3.9.33 does not restrict which PHP function may be applied to exported field values, allowing users granted its export permission to have arbitrary functions invoked on values they control, leading to remote code execution.	7.2	More Details
CVE-2026-54646	CubeCart is an ecommerce software solution. Prior to 6.7.5, admin/sources/maintenance.index.inc.php places administrator-controlled tablename values into ALTER TABLE, CHECK TABLE, and ANALYZE TABLE statements without validating the identifiers or escaping embedded backticks. An authenticated administrator can terminate the quoted identifier with a closing backtick and introduce attacker-controlled structural SQL, potentially compromising database confidentiality, integrity, and availability within the application's database privileges. This issue is fixed in version 6.7.5.	7.2	More Details
CVE-2026-90131	In the Linux kernel, the following vulnerability has been resolved: ntfs: serialize resident iomap reads with mrec_lock ntfs_read_iomap_begin_resident() walks the MFT record through ntfs_attr_lookup() -> ntfs_attr_find() without taking ni->mrec_lock, while ntfs_attr_record_resize(), ntfs_make_room_for_attr() and ntfs_resident_attr_record_add() memmove() the same base_ni->mrec buffer under that lock. map_mft_record() only takes a reference and does not serialize, so the reader can observe torn attribute length and offset fields while a writer is relocating the records. KCSAN reports the race between the mmap read fault path and both link() and unlink(): BUG: KCSAN: data-race in ntfs_attr_find / ntfs_attr_record_resize write to 0xffff888100af1018 of 4 bytes by task 96 on cpu 1: ntfs_attr_record_resize+0xd2/0x130 ntfs_attr_record_rm+0xad/0x530 ntfs_delete+0x224/0x640 ntfs_unlink+0x14d/0x280 vfs_unlink+0x157/0x520 read to 0xffff888100af1018 of 4 bytes by task 95 on cpu 0: ntfs_attr_find+0x104/0x5b0 ntfs_attr_lookup+0x39c/0x10c0 ntfs_read_iomap_begin_resident+0xc6/0x230 ntfs_read_iomap_begin+0x5d/0xa0 iomap_iter+0x2e2/0x6e0 iomap_read_folio+0x147/0x2a0 ntfs_read_folio+0x108/0x170 filemap_read_folio+0x35/0x100 filemap_fault+0x993/0x1000 value changed: 0x00000250 -> 0x000001f0 The address is mrec + 0x18, i.e. mft_record.bytes_in_use, and the change is the 96 bytes of one $FILE_NAME attribute being removed. Keep base_ni->mrec_lock from the resident read iomap lookup through iomap_end(). This protects both the attribute walk and the subsequent copy from iomap->inline_data, which points into the MFT record. The non-resident path is left alone: ntfs_lookup() already holds the directory inode's mrec_lock when it reads an index folio through read_mapping_folio(), and taking the lock in the shared wrapper deadlocks there with recursive locking on mrec_lock. The comment above the read_mapping_folio() call in fs/ntfs/dir.c notes the same hazard. The seek path uses the same lookup helper but does not dereference iomap->inline_data. Release the lock before returning from that path, whereas the regular read path records base_ni in iomap->private and releases the lock from its iomap_end() callback. Tested with a reproducer that faults in a 16-byte resident file while another thread runs link()/unlink() on it. Before: 40 KCSAN reports in about one second. After: no reports in 180 seconds over 206,090 read iterations and 423,540 link/unlink cycles. A PROVE_LOCKING build shows no lockdep splat with the same reproducer running for 60 seconds.	7.1	More Details

CVE-2025-15697	The Dictionary WordPress plugin through 1.0 does not escape user input before reflecting it back in the responses of several directly accessible scripts, allowing unauthenticated attackers to perform Reflected Cross-Site Scripting attacks against anyone they can induce to submit a crafted request.	7.1	More Details
CVE-2026-90132	In the Linux kernel, the following vulnerability has been resolved: ntfs: reject unprivileged writes to reserved $LX* xattrs Reject setxattr of the reserved $LXUID, $LXGID, $LXMOD and $LXDEV names from userspace unless the caller has CAP_SYS_ADMIN.	7.1	More Details
CVE-2026-90062	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: move hardware offload step after building the chain blob Allocate the chain blob before the ruleset offload to reduce chances of entering an inconsistent state where the offloaded ruleset in the nic and the software ruleset differ.	7.1	More Details
CVE-2026-86444	The LearnPress WordPress plugin before 4.4.7 does not escape a user supplied value before using it in an HTML attribute on a public page, allowing unauthenticated attackers to execute arbitrary JavaScript in the browser of anyone who opens a crafted link, including a logged in administrator. Only sites running a classic, non-block are affected.	7.1	More Details
CVE-2026-90089	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btnxpuart: Validate the FW dump header length nxp_process_fw_dump() pulls the ACL header off the frame and then reads seq_num and buf_len from a struct nxp_fw_dump_hdr placed at skb->data, without checking that the ACL payload is long enough to contain it. h4_recv_buf() collects HCI_ACL_HDR_SIZE bytes of header followed by the number of payload bytes named in that header, so skb->len is 4 + dlen with dlen supplied by the controller and possibly smaller than the 8 byte dump header, or zero. A short frame with connection handle 0xfff therefore reads both fields from beyond the received data. Beyond the read itself, buf_len is what terminates a dump: a value of zero makes the driver call hci_devcd_complete() and reset the controller, so a truncated frame can end a dump early. Use skb_pull_data() to validate and pull the FW dump header before accessing its fields. Warn and reject the chunk if the header is truncated.	7.1	More Details
CVE-2026-89792	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent out-of-bounds reads in share config responses Validate IPC share configuration payload sizes before consuming variable-length fields. Bound veto list parsing and account for the separator byte when deriving the path length.	7.1	More Details
CVE-2026-92773	Trigger.dev before 4.6.0 fails to verify that an authenticated user controls a GitHub App installation before binding it to their organization. Attackers can claim another user's GitHub App installation by replaying state cookies and supplying sequential installation identifiers, gaining unauthorized access to the victim's repositories.	7.1	More Details
CVE-2026-92772	Leantime before 3.9.6 contains an authorization bypass vulnerability in the HTMX plugin install endpoint that lacks permission validation. Authenticated users with limited roles can install marketplace plugins and control arbitrary properties including identifier, version, and license key to deploy malicious plugins.	7.1	More Details
CVE-2026-92804	Nango through 0.70.4 fails to validate caller-supplied connection configuration values interpolated into provider token and proxy URL templates. Authenticated attackers can supply malicious configuration values to direct server requests at internal addresses or cloud metadata endpoints, potentially exfiltrating provider credentials.	7.1	More Details
CVE-2026-90260	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: don't clobber the extent buffer when zeroing it out On a zoned filesystem a freed-but-still-dirty tree block is written out as zeros (EXTENT_BUFFER_ZONED_ZEROOUT) only to keep the zone write pointer advancing. btree_csum_one_bio() implemented this by memzeroing the extent buffer's own folios before submission. That destroys the in-memory buffer while it may still be referenced. In particular btrfs_free_tree_block() can run on it afterwards and reads the header to add a delayed reference; once the header has been zeroed it frees bytenr 0 and corrupts the extent tree (the btrfs_header_bytenr(buf) != 0 ASSERT in btrfs_free_tree_block(), or an "unable to find ref" abort). It is flaky and reproduces under fsstress, e.g. generic/461 and generic/013. Write the zeros to disk from the shared zero page instead and leave the extent buffer content untouched, so any later reference - including the delayed reference from btrfs_free_tree_block() - still sees a valid header. end_bbio_meta_write() now clears writeback on the buffer's own folios, as the bio no longer carries them.	7.1	More Details
CVE-2026-92753	PatrowlManager through 1.8.4 contains an authorization bypass vulnerability in the events and alerts API endpoints that lack ownership filtering. Authenticated attackers can read platform event history, delete arbitrary events, and modify alerts belonging to other users.	7.1	More Details
CVE-2026-90246	In the Linux kernel, the following vulnerability has been resolved: apparmor: fix integer overflow in verify_tags() bounds check verify_tags() validates the tagset table unpacked from a policy blob. For each set it reads a count and checks that advancing the index by that count stays inside sets.table[]: u32 cnt = tags->sets.table[i]; if (i+cnt >= tags->sets.size) { i, cnt and sets.size are all u32, so i+cnt is evaluated modulo 2^32. sets.table[] is filled by unpack_tagsets() with aa_unpack_u32(), so every entry is a raw unbounded 32-bit word taken from the policy blob, and verify_tags() is the function that is supposed to validate it. A count close to U32_MAX makes the sum wrap to a small value, the guard passes, and the inner loop then walks sets.table[++i] past the end of the kcalloc(size, sizeof(u32)) allocation. Note that sets.size is bounded by 65535, because unpack_tagsets() reads it with aa_unpack_array() as a u16, so the wrap cannot be reached by growing the table; it is reached purely through the attacker-supplied count. With sets.size = 2 and sets.table = { 0, 0xffffffff }: i = 0: cnt = 0, guard 0 + 0 >= 2 is false, inner loop does not run i = 1: cnt = 0xffffffff, guard (1 + 0xffffffff) mod 2^32 == 0 >= 2 is false, so the guard is bypassed and the inner loop reads sets.table[2] -- one element past a two element allocation The walk continues until an out-of-bounds value happens to be >= hdrs.size or the access faults, so a crafted policy yields an out-of-bounds read on the policy load path (aa_replace_profiles -> aa_unpack -> unpack_policydb -> unpack_tags -> verify_tags). unpack_tags() runs before the perms and DFA tables are unpacked, so no other table needs to be well formed to reach it. Policy load is gated by aa_may_manage_policy(), which checks CAP_MAC_ADMIN relative to the subject's own user namespace rather than the init user namespace, so with the default unprivileged_userns_apparmor_policy=1 the path is reachable from an unprivileged task in a matched-level nested namespace, not only by a globally privileged one. Perform the addition in u64 so that it cannot wrap, restoring the intended i + cnt < sets.size guarantee.	7.1	More Details
CVE-2026-93014	RosarioSIS versions before 12.9 fail to validate the filename request parameter in Users and Students modules, allowing authenticated users to unlink allow-listed files via path traversal. Attackers can use parent-directory sequences to escape upload directories and delete CSS, XML, JSON resources and other users' documents throughout the installation.	7.1	More Details
	AVideo (WWBN/AVideo) through 29.0 (commit e01e41ecc) is vulnerable to cross-site request forgery. objects/videoAddNew.json.php disables AVideo's automatic CSRF guard ($global['skipAutoCSRFCheck']) and the untrusted-		

CVE-2026-92582	request check ($global['bypassSameDomainCheck']) merely because 'user' and 'pass' parameters are present in the request; the values are never validated and are read from $_REQUEST, so an attacker can supply them in the query string of a cross-site request. Because User::loginFromRequestIfNotLogged() returns immediately when a session already exists, a victim who is authenticated by cookie satisfies the check while the attacker-supplied credentials are discarded. An attacker who lures an authenticated user with upload rights to visit a crafted page can therefore submit cross-origin requests that modify video records — including ownership transfer (setUsers_id), deletion of user-group access restrictions, can_download, can_share, only_for_paid, video_password, rating, status, creation date, and view count. For a victim with administrator or Permissions::canAdminVideos() rights, any video on the site can be altered, including removing group restrictions from private content. No patched version was available at the time of the advisory.	7.1	More Details
CVE-2026-61596	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, djust's per-object authorization (`get_object` + `has_object_permission`, ADR-017) was enforced on the WebSocket mount and event paths but not on three other render entry points: (a) the initial HTTP GET render, (b) SPA `url_change` navigation, and (c) `{% live_render %}` embedded child views. An authenticated user could therefore view (and on some paths act on) an object they are not authorized for by loading the page directly, navigating to it via SPA url-change, or composing it as an embedded child — a classic IDOR / broken object-level access control on object-scoped views. This is fixed in djust 1.0.7. All render entry points now route through a shared `enforce_object_permission` chokepoint: HTTP GET returns 403, `url_change` emits a `permission_denied` frame and skips the render, and `{% live_render %}` (eager + lazy) refuses the embed. Views without a custom `get_object` are unaffected (no-op). No reliable workaround short of upgrading. Do not expose object-scoped views through the HTTP-GET / url_change / live_render paths until patched.	7.1	More Details
CVE-2026-89818	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/vcn: fix integer overflow in dec_msg buffer count check If the supplied msg[2] (num_buffers) is 0x3FFFFFFF, the expression 6 + num_buffers * 4 wraps to 2 and the bounds check passes, letting the parser loop far past the end of the message BO. Triggering it additionally requires a ~4GiB mapping so that msg[1] survives the earlier "header does not fit in BO" check. Rewrite the test in division form, which is overflow-free by construction. Also update the message to reflect that msg is invalid.	7.1	More Details
CVE-2026-86049	Jupyter Server is the backend for Jupyter web applications. Prior to version 2.21.0, the 5xx request logging path in jupyter_server/log.py copies the Referer header into a JSON header block without applying the token scrubbing used for the request URI. A request that returns HTTP 500 while the Referer contains a token-bearing URL can therefore write that token to server logs in plaintext. An attacker who can read those logs can recover the token and use the affected user's Jupyter Server permissions. This issue is fixed in version 2.21.0.	7.1	More Details
CVE-2026-89826	In the Linux kernel, the following vulnerability has been resolved: drm/panthor: harden firmware build-info bounds checks panthor_fw_read_build_info() checks whether the metadata range fits in the firmware image with hdr.meta_start + hdr.meta_size. Both fields are u32, so the addition can wrap and let an out-of-bounds range pass validation. The function also reads the "git_sha: " prefix without first checking that the metadata is long enough, and meta_size == 0 can underflow the NULL terminator index. Use subtraction-based bounds checking and reject metadata that is too short to contain the expected prefix and trailing NULL byte.	7.1	More Details
CVE-2026-89838	In the Linux kernel, the following vulnerability has been resolved: f2fs: limit recovery filename logging to stored length F2FS stores recovery filenames as a length plus a fixed-size i_name buffer. The buffer is not NUL-terminated, but recover_inode() and recover_dentry() print it with %s. For a 255-byte filename, recovery logging can read past i_name into the following raw inode fields. Print the name with a precision bounded by i_namelen and F2FS_NAME_LEN.	7.1	More Details
CVE-2026-89840	In the Linux kernel, the following vulnerability has been resolved: f2fs: validate MOVE_RANGE destination size F2FS_IOC_MOVE_RANGE checks the source range, but not the destination end before updating i_size. A source hole can expose this: __clone_blkaddrs() skips NULL_ADDR entries and returns success, so the caller can still extend the destination inode with unchecked pos_out + len. Reject destination overflow and use inode_newsize_ok() before extending the destination inode.	7.1	More Details
CVE-2026-20300	A vulnerability in Cisco ISE could allow an authenticated, remote attacker to conduct SQL injection attacks on an affected device. To exploit this vulnerability, the attacker must have at least low-privileged administrative credentials. This vulnerability is due to improper validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to read or modify data in the underlying database.	7.1	More Details
CVE-2026-54510	Speakr is a personal, self-hosted web application designed for transcribing audio recordings. Prior to 0.8.21-alpha, the csrf_exempt_for_api_tokens() before_request hook in src/app.py calls csrf.exempt(view_func), permanently adding the selected view to Flask-WTF's process-global exemption set. The is_token_authenticated() function in src/utils/token_auth.py calls extract_token_from_request() and treats any present token, including request.args.get('token'), as authenticated without hashing the token, querying the database, or checking validity. A network-reachable attacker can therefore send a false token to disable CSRF protection for the targeted view for the worker lifetime. Because the exemption applies to the view function across HTTP methods, a cross-origin GET to /account with a query token can poison CSRF state for a later state-changing POST without triggering CORS preflight. This browser sequence requires attacker-controlled content on a sibling subdomain under the documented cookie conditions. The bypass can modify profile data, custom prompts, transcription settings, preferences, and administrative status through routes such as admin_toggle_admin. The change_password route also skips current-password verification when current_user.password is empty, allowing the chain to set a local password on an SSO-only account and bypass SSO. This issue is fixed in version 0.8.21-alpha.	7.1	More Details
CVE-2026-90978	The Filter Gallery WordPress plugin before 1.1.5 does not verify the nonce on several of its AJAX handlers when the nonce field is omitted, and applies no capability check, allowing low-privileged users to overwrite the content of arbitrary posts and delete the Filter Gallery WordPress plugin before 1.1.5's stored gallery options.	7.1	More Details
CVE-2026-89912	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: vgic-its: Don't dereference a NULL collection on ITT save MAPC with V=0 drops ite->collection but leaves the ITE on the device's ITT list, and vgic_its_save_ite() dereferences it unconditionally. A guest that issues MAPD, MAPTI and then MAPC(V=0) therefore oopses the host when the VMM issues KVM_DEV_ARM_ITS_SAVE_TABLES to migrate it. That sequence is UNPREDICTABLE per the architecture, but KVM already handles the resulting state in the translate, MOVI and DISCARD paths. Save a zeroed entry, which vgic_its_restore_ite() reads back as invalid. Skipping the ITE instead would leave the ITT slot holding whatever is in guest memory, and restore rejects an entry naming a collection the restored collection table does not have.	7.1	More Details
	Traccar is an open source GPS tracking system. Prior to 6.14.0, an authenticated, non-readonly user with access to an object		

CVE-2026-52851	usable in a permission pair can submit DELETE /api/permissions with an extra attacker-controlled JSON key. Permission(LinkedHashMap<String, Long>) in src/main/java/org/traccar/model/Permission.java validates only the first two keys, but DatabaseStorage.removePermission() in src/main/java/org/traccar/storage/DatabaseStorage.java concatenates every map key into the SQL WHERE clause as a column identifier. The extra key therefore becomes attacker-controlled SQL and provides a blind boolean or error oracle that can extract arbitrary database values, including administrator email, password hashes, and salts, or conditionally delete permission rows. Unauthenticated requests are rejected. This issue is fixed in 6.14.0.	7.1	More Details
CVE-2026-90401	In the Linux kernel, the following vulnerability has been resolved: md: remove REQ_NOWAIT support from raid1/10/456 REQ_NOWAIT support in md personalities that can block internally is fundamentally incomplete. While reads can avoid some blocking paths, write requests can still encounter cases where one mirror succeeds while another returns -EAGAIN. At that point md cannot distinguish queue pressure from a real device failure, so it can neither record a bad block nor safely retry the write without REQ_NOWAIT, leaving mirrors with divergent data. Rather than continue advertising REQ_NOWAIT support for personalities that cannot implement it correctly, remove it from raid1, raid10 and raid456. Keep REQ_NOWAIT for linear and raid0, which only remap bios to their underlying devices; stacked limits will still clear the feature if any component device lacks REQ_NOWAIT support.	7.1	More Details
CVE-2026-93178	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm/powerplay: bounds-check voltage index in SMU7 lookup vddInd and vddcInd fields from VBIOS-parsed tables are used to index into voltage lookup tables without a bounds check. Return -EINVAL when any index is out of range.	7.1	More Details
CVE-2026-66571	Unauthenticated Cross Site Request Forgery (CSRF) in Asset CleanUp: Page Speed Booster <= 1.4.0.5 versions.	7.1	More Details
CVE-2026-90174	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-out-of-bounds read in ksmbd_alloc_user() ksmbd_alloc_user() copies resp->hash_sz bytes out of the mountd IPC login response with user->passkey_sz = resp->hash_sz; user->passkey = kmalloc(resp->hash_sz, KSMBD_DEFAULT_GFP); if (user->passkey) memcpy(user->passkey, resp->hash, resp->hash_sz); resp->hash_sz is a __u16 supplied by the response, but resp->hash[] is only KSMBD_REQ_MAX_HASH_SZ bytes. A malformed or malicious login response can set hash_sz well beyond that (up to 65535), so the memcpy() reads past the end of the response object. ipc_validate_msg() does not bound hash_sz, so reject any response whose hash_sz exceeds the on-stack hash[] buffer before allocating and copying. [2030.238706] BUG: KASAN: slab-out-of-bounds in ksmbd_alloc_user+0x278/0x680 [2030.240549] Read of size 65535 at addr ffff888121bb6680 by task kworker/4:1/18611 [2030.242296] [2030.242710] CPU: 4 UID: 0 PID: 18611 Comm: kworker/4:1 Not tainted 7.1.0-next-20260623-virtme #96 PREEMPT(lazy) [2030.242732] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 [2030.242743] Workqueue: ksmbd-io handle_ksmbd_work [2030.242763] Call Trace: [2030.242769] <TASK> [2030.242776] dump_stack_lvl+0xa2/0xd0 [2030.242794] print_address_description+0x77/0x200 [2030.242815] ? ksmbd_alloc_user+0x278/0x680 [2030.242831] print_report+0x58/0x70 [2030.242848] kasan_report+0x117/0x150 [2030.242869] ? ksmbd_alloc_user+0x278/0x680 [2030.242888] kasan_check_range+0x3c7/0x3f0 [2030.242908] ? ksmbd_alloc_user+0x278/0x680 [2030.242925] __asan_memcpy+0x29/0x70 [2030.242942] ksmbd_alloc_user+0x278/0x680 [2030.242960] ksmbd_login_user+0xc3/0x120 [2030.242978] ntlm_authenticate+0x5e6/0x1b00 [2030.243017] ? __pfx_ntlm_authenticate+0x10/0x10 [2030.243035] ? ksmbd_session_lookup+0x188/0x1d0 [2030.243054] ? __pfx_ksmbd_session_lookup+0x10/0x10 [2030.243090] ? __sanitizer_cov_trace_switch+0x7b/0x140 [2030.243108] smb2_sess_setup+0x1e4a/0x27b0 [2030.243126] ? copy_from_kernel_nofault+0x199/0x300 [2030.243156] ? __pfx_smb2_sess_setup+0x10/0x10 [2030.243173] ? get_smb2_cmd_val+0xe3/0x1c0 [2030.243208] handle_ksmbd_work+0x954/0x1280 [2030.243230] ? __pfx_handle_ksmbd_work+0x10/0x10 [2030.243249] ? process_scheduled_works+0xa07/0x1490 [2030.243270] ? process_scheduled_works+0xa07/0x1490 [2030.243291] process_scheduled_works+0xa70/0x1490 [2030.243320] ? __pfx_process_scheduled_works+0x10/0x10 [2030.243340] ? do_raw_spin_lock+0x130/0x300 [2030.243358] ? lock_is_held_type+0x7b/0x110 [2030.243388] worker_thread+0x932/0xe20 [2030.243415] kthread+0x38a/0x470 [2030.243431] ? __pfx_worker_thread+0x10/0x10 [2030.243451] ? __pfx_kthread+0x10/0x10 [2030.243467] ret_from_fork+0x484/0x910 [2030.243485] ? __pfx_ret_from_fork+0x10/0x10 [2030.243501] ? __switch_to+0xc77/0x12c0 [2030.243523] ? __pfx_kthread+0x10/0x10 [2030.243540] ret_from_fork_asm+0x1a/0x30 [2030.243564] </TASK> [2030.243570] [2030.290164] Allocated by task 19279: [2030.290911] kasan_save_track+0x3e/0x80 [2030.292179] __kasan_kmalloc+0x72/0x90 [2030.293217] __kvmalloc_node_noprof+0x3ff/0x6b0 [2030.294467] handle_generic_event+0x59b/0x750 [2030.295345] genl_family_rcv_msg_doit+0x238/0x340 [2030.296553] genl_rcv_msg+0x606/0x7b0 [2030.297129] netlink_rcv_skb+0x22b/0x4a0 [2030.298500] genl_rcv+0x2d/0x40 [2030.299273] netlink_unicast+0x7ba/0x930 [2030.300019] netlink_sendmsg+0x8c3/0xb00 [2030.301073] __sock_sendmsg+0xec/0x140 [2030.301579] __sys_sendto+0x357/0x470 [2030.302255] __x64_sys_sendto+0xe3/0x100 [2030.303425] do_syscall_64+0x135/0x460 [2030.304763] entry_SYSCALL_64_after_hwframe+0x77/0x7f [2030.305594] [2030.305819] The buggy address belongs to the object at ffff888121bb6640 [2030.305819] which belongs to the cache kmalloc-192 of size 192 [2030.309595] The buggy address ---truncated---	7.1	More Details
CVE-2026-90986	Unauthenticated Cross Site Scripting (XSS) in Visitor Traffic Real Time Statistics Pro <= 11.21 versions.	7.1	More Details
CVE-2026-93112	In the Linux kernel, the following vulnerability has been resolved: bpf: Require a BPF cpumask for bpf_cpumask_populate() bpf_cpumask_populate() writes to its destination with bitmap_copy(), but the destination is typed as struct cpumask *. That allows the verifier to accept borrowed cpumask pointers returned by read-only kfuncs, such as scx_bpf_get_online_cpumask(), as a writable destination. Make the destination a struct bpf_cpumask * so populate follows the same ownership rule as the other mutating cpumask kfuncs. Query kfuncs continue to accept const struct cpumask * inputs.	7.1	More Details
CVE-2026-90203	In the Linux kernel, the following vulnerability has been resolved: Squashfs: check block offset is not negative If a negative offset is read off disk (for example the offset into the decompressed fragment block), this will cause squashfs_copy_data() to perform an out of bounds access. Fix by checking if offset is negative, and returning 0. This matches existing behaviour where an offset beyond the block returns 0 bytes copied. To trigger this out of bounds access requires a crafted Squashfs filesystem and CAP_SYS_ADMIN to mount it. Unprivileged users will not be able to mount such a filesystem, but once mounted, an unprivileged user can trigger the out of bounds access by reading the crafted file with the negative offset.	7.1	More Details

CVE-2026-92456	yshop-crm through 2.1.3 fails to enforce authorization on the saveRedisSet and getRedisSet endpoints in CrmCustomerController, allowing any authenticated back-office user to read and modify installation-wide lead-allocation and customer auto-recycling policy. Attackers can invoke these endpoints to manipulate shared Redis keys controlling customer auto-recycling behavior, causing mass customer data deletion, disabling lead recycling, or blocking customer creation across the deployment.	7.1	More Details
CVE-2026-90016	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: fix OOB read in rtw_restruct_wmm_ie() rtw_restruct_wmm_ie() scans in_ie for a WMM IE with: while (i < in_len) { ... if (i + 5 < in_len && in_ie[i] == 0xDD && ...) { ... break; } i += (in_ie[i + 1] + 2); /* to the next IE element */ } When the "i + 5 < in_len" match check fails simply because i is within 5 bytes of the end of the buffer (i.e. no WMM IE was found near the tail of in_ie), execution falls through to "i += (in_ie[i + 1] + 2)", which reads in_ie[i + 1]. If i == in_len - 1 at that point, this is a 1-byte out-of-bounds read of an attacker-influenced IE buffer built from association/scan data. Commit a75281626fc8f ("staging: rtl8723bs: fix potential out-of-bounds read in rtw_restruct_wmm_ie") added the "i + 5 < in_len" guard to the match condition itself, but did not add an equivalent guard before the fallthrough advance, so the same class of OOB read remained reachable through the non-matching path. Add an explicit bounds check before advancing to the next IE.	7.1	More Details
CVE-2026-90017	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: fix OOB read in rtw_action_frame_parse() rtw_action_frame_parse() takes a frame_len parameter but never actually checks it before indexing into the frame body: const u8 *frame_body = frame + sizeof(struct ieee80211_hdr_3addr); ... c = frame_body[0]; ... a = frame_body[1]; frame_body already points 24 bytes (sizeof(struct ieee80211_hdr_3addr)) into frame, so reading frame_body[0] and frame_body[1] requires frame_len >= 26. A management action frame shorter than that (e.g. exactly 24 bytes, the minimum a malicious peer can send) causes a 1-2 byte out-of-bounds read. This is reachable from rtw_cfg80211_monitor_if_xmit_entry() and cfg80211_rtw_mgmt_tx() in ioctl_cfg80211.c, both of which pass attacker/user-influenced frame buffers and lengths straight through. Add the missing length check before frame_body is dereferenced.	7.1	More Details
CVE-2026-89927	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: hyper-v: Clamp stimer deadline to avoid livelock Fix an issue where userspace or the guest can program an Hyper-V synthetic timer to have a deadline in the past via integer overflow, preventing the CPU from making progress and triggering an RCU stall. Hyper-V's SynIC exposes 4 per-vCPU synthetic timers to the guest, which are emulated by KVM. Each is programmed through the HV_X64_MSR_STIMERi_CONFIG and HV_X64_MSR_STIMERi_COUNT MSRs. Depending on CONFIG, COUNT represents either the absolute expiration time or the period of a periodic timer, both expressed in 100ns ticks. These timers may be set both by the guest (WRMSR) and the host (KVM_SET_MSRS). When the timer is enabled, stimer_start() translates COUNT to an absolute monotonic deadline and arms an hrtimer. If COUNT is set to a value close to U64_MAX, the deadline calculation can overflow. ktime_add_ns(ktime_now, 100 * (stimer->exp_time - time_now)) This can result in a CPU livelock. stimer_start() arms the timer via hrtimer_start() with a deadline in the past, which causes it to immediately fire. The stimer callback then raises KVM_RQ_HV_STIMER, with the intention of causing KVM to deliver a synthetic interrupt on the next vCPU guest enter. Then, once userspace issues KVM_RUN, vcpu_enter_guest() consumes the request, calling kvm_hv_process_stimers(). This would normally disable the timer via stimer_expiration() once the deadline is in the past. However, the deadline comparison is done between the KVM reference counter and stime->exp_time, which is a big value close to U64_MAX, so this never happens for a few thousand years. kvm_hv_process_timers() then re-arms the timer via stimer_start(), since it was not disabled, which again fires immediately. Before entering the guest, kvm_vcpu_exit_request() checks kvm_request_pending(), which returns true due to the newly raised KVM_REQ_HV_STIMER. Then vcpu_enter_guest() aborts the guest entry, returning early into vcpu_run(), which loops back again into vcpu_enter_guest(), restarting the cycle. Since there are no manual yields in this loop, a task with SCHED_FIFO may starve RCU grace-period kthreads, which exposes the stalls found by syzcaller: rcu: INFO: rcu_preempt detected stalls on CPUs/tasks: rcu: (detected by 1, t=10502 jiffies, g=14269, q=1142 ncpus=2) rcu: All QSes seen, last rcu_preempt kthread activity 10500 (4294965239-4294954739), jiffies_till_next_fqs=1, root ->qsmask 0x0 rcu: rcu_preempt kthread starved for 10500 jiffies! g14269 f0x2 RCU_GP_WAIT_FQS(5) ->state=0x0 ->cpu=0 rcu: Unless rcu_preempt kthread gets sufficient CPU time, OOM is now expected behavior. (...) Call Trace: <IRQ> __run_hrtimer kernel/time/hrtimer.c:1773 [inline] __hrtimer_run_queues+0x408/0xc30 kernel/time/hrtimer.c:1841 hrtimer_interrupt+0x45b/0xaa0 kernel/time/hrtimer.c:1903 local_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1045 [inline] __sysvec_apic_timer_interrupt+0x102/0x3e0 arch/x86/kernel/apic/apic.c:1062 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1056 [inline] sysvec_apic_timer_interrupt+0xa1/0xc0 arch/x86/kernel/apic/apic.c:1056 </IRQ> <TASK> asm_sysvec_apic_timer_interrupt+0x1a/0x20 arch/x86/include/asm/idtentry.h:697 RIP: 0010:__raw_spin_unlock_irqrestore include/linux/spinlock_api_smp.h:152 [inline] RIP: 0010:_raw_spin_unlock_irqrestore+0xa8/0x110 kernel/locking/spinlock.c:194 Code: 74 05 e8 0b f4 5f f6 48 c7 44 24 20 00 00 00 00 9c 8f 44 24 20 f6 44 24 21 02 75 4f f7 c3 00 02 00 00 74 01 fb bf 01 00 00 00 <e8> 23 6b 27 f6 65 8b 05 7c 60 5a 07 85 c0 74 40 48 c7 04 24 0e 36 RSP: 0018:ffffc900040a7320 EFLAGS: 00000206 RAX: 5de15cb931505900 RBX: 0000000000000a06 RCX: 5de15cb931505900 RDX: 0000000000000007 RSI: ffffffff8daa9dc3 RDI: 0000000000000001 RBP: ffffc900040a73b0 R08: ffffffff8fc3d0 ---truncated---	7.1	More Details
CVE-2026-90223	In the Linux kernel, the following vulnerability has been resolved: nfc: llcp: bound SNL TLV parsing to the skb and add length checks nfc_llcp_recv_snl() walked the SNL TLV list using a u16 offset/length pair derived from skb->len, without bounding reads to the actual skb data. Three problems followed: - For a short frame (skb->len < LLCP_HEADER_SIZE), tlv_len underflowed. - The per-TLV header (type, length) was read without checking that two bytes remained. - A declared TLV length could run past the end of the buffer, and an SDREQ with length == 0 made "service_name_len = length - 1" underflow (size_t), driving an out-of-bounds read in the following strncmp() / nfc_llcp_sock_from_sn(). The SDRES case likewise read tlv[2]/tlv[3] without a length check. A nearby NFC device can reach this without authentication; LLCP link activation happens automatically after NFC-DEP. Walk the TLV list by pointer, bounded by skb_tail_pointer() over the linear skb data, and validate each TLV declared length before use. Add explicit length checks for SDREQ (>= 1) and SDRES (exactly 2). Found by 0sec automated security-research tooling (https://0sec.ai).	7.1	More Details
CVE-2026-92525	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Validate num_sge/cur_sge before indexing wqe->dma.sge[] For a user QP, qp->sq.queue is a ring the application writes directly, so rxe_post_send() takes the is_user branch and only schedules send_task without validating the WQE. rxe_requester() consumes it in place via req_next_wqe() and calls copy_data(), which indexes &wqe->dma.sge[cur_sge] with the attacker-controlled num_sge/cur_sge. Only the kernel path bounds num_sge (validate_send_wr()); the user WQE is never checked, so a local unprivileged user can post a WQE with an out-of-range cur_sge or oversized num_sge and force an out-of-bounds read of the per-WQE sge array in copy_data() (vmalloc OOB read, local DoS). Bound num_sge to qp->sq.max_sge in rxe_requester() before use, the way get_srq_wqe() already guards SRQ entries, and bound cur_sge only when the WQE carries payload (dma.resid): copy_data() returns early on a zero-length copy before touching dma->sge[], so a zero-payload WQE -- the only kind a max_sge == 0 QP can post -- stays valid. Reproduced	7.1	More Details

	under KASAN; the vmalloc-out-of-bounds in copy_data() is gone.		
CVE-2026-90887	Unauthenticated Cross Site Scripting (XSS) in WP Inventory Manager <= 2.5.4 versions.	7.1	More Details
CVE-2026-85387	Concrete CMS before 9.5.4 re-authorized OAuth REST API requests from the bearer token alone and did not re-check the state of the account the token had been issued to. The resource server's authorization validator confirmed only that a token existed, had not expired, and had not been explicitly revoked, and deactivating a user did not revoke that user's outstanding tokens. As a result, a deactivated user retained full access to /ccm/api/1.0/* for the remaining lifetime of any token already issued to them. The same gap applied to accounts that had been deleted or locked pending a forced password reset. The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 2.0 with vector CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:A/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N. Thanks Myq Larson for reporting.	7.1	More Details
CVE-2026-92959	vm2 before 3.11.8 does not fully enforce the allowAsync: false option in VM and NodeVM. While localPromise.prototype.then is replaced with a handler that throws 'Async not available', the sandbox's Promise static methods (Promise.resolve, Promise.all, Promise.race, Promise.any, and Promise.allSettled) still assimilate attacker-supplied thenables: native promise resolution performs PromiseResolveThenableJob and invokes the sandboxed code's then method in a microtask without passing through the patched then, so the async restriction is never applied. As a result, sandboxed script can schedule work that runs after VM.run() or NodeVM.run() has returned and outside the configured timeout, continuing to execute after the host believes execution is complete.	7.1	More Details
CVE-2026-92602	TDuck survey form through version 5.3 fails to validate webhook URLs or verify form ownership in the WebhookConfigController. Authenticated attackers can attach webhooks to other users' forms and exfiltrate submissions to arbitrary external or internal addresses.	7.1	More Details
CVE-2026-74926	The MultiVendorX WordPress plugin before 5.0.16 does not verify that a user owns the store they are acting on in one of its REST API routes, allowing any authenticated user, such as a subscriber, to overwrite any store's details and payout settings and to replace the record of who owns it.	7.1	More Details
CVE-2026-92925	A flaw was found in Redis community. The cluster bus packet parser, responsible for handling PING, PONG, and MEET packets, fails to properly validate string-carrying extensions for null-termination. This oversight allows a remote attacker to craft a malicious packet, leading to an out-of-bounds read when the packet's payload is processed. Successful exploitation of this vulnerability could result in the disclosure of sensitive information or a remote denial of service (DoS).	7.1	More Details
CVE-2026-93203	In the Linux kernel, the following vulnerability has been resolved: batman-adv: bla: avoid CRC corruption due to parallel claim add batadv_bla_add_claim() is used to add claims and modify the backbone of claims for CLAIM frames from remote backbones and local packets. When it handles a claim, it needs to either * add the new claim's CRC to the backbone CRC * remove the already existing claim's CRC from the old backbone and add it to the new backbone But when the "new" claim code was running in parallel to the "change backbone" code, it can happen that the CRC was invalid because the backbone_gw of the claim was changed twice in the "new" claim code path: * CPU0 creates the claim for gateway A and publishes it in the claim hash. The crc16 of the address has not yet been added to A's crc at this point. * CPU1 processes a claim frame of gateway B for the same client, finds the just published claim, and performs the ownership change: it switches the pointer to B, removes the crc16 from A's crc - which never contained it - and adds it to B's crc. * CPU0 continues behind the creation branch, unconditionally switches the pointer back to A without compensating B's crc (its remove_crc is false for the creation path), and finally adds the crc16 to A's crc The CRC is then wrong for both: * claim belongs to A: but CRC is not part of backbone A's CRC * claim doesn't belong to B: CRC is still part of backbone B's CRC This wrong CRC is never recomputated from the stored claims. For local backbone claims, this can also not recovered using syncs. To avoid this, split the functionality in clear separate parts: * new claim which always adds claim CRC to the backbone CRC (but never changes the already set backbone_gw of the claim back) * update of existing claim which automatically changes the backbone_gw entry and only updates both backbone CRCs when there was an actual change	7.1	More Details
CVE-2026-90419	In the Linux kernel, the following vulnerability has been resolved: nilfs2: prevent out-of-bounds read in super root block parsing super-root inode metadata size is trusted before nilfs_read_inode_common(). Reject super-root inode sizes whose computed on-disk footprint exceeds the filesystem block size. This prevents malformed filesystem images from making nilfs_read_inode_common() read past the end of the super-root block. [ryusuke: clarify the commit title]	7.1	More Details
CVE-2026-86089	Apache NiFi 2.11.0 supports migrating the contents of a version-controlled Process Group into a Connector using REST API methods that list eligible migration sources and submit migration requests. The framework authorized both methods against the target Connector alone, without evaluating access to the Process Groups involved. The absence of Process Group authorization allowed an authenticated user with read access to a Connector to enumerate the identifiers, names, and flow registry details of version-controlled Process Groups outside the scope of granted read policies. It also allowed a user with write access to a Connector to migrate a Process Group without write access to that Process Group, copying the flow definition, referenced assets, and component state into the Connector, and leaving the source Process Group disabled and renamed. Migration excludes sensitive property values and requires the source Process Group to be stopped with empty queues, which limits the scope of exposure. Apache NiFi installations that do not implement component-level authorization policies for Process Groups are not subject to this vulnerability, because the framework enforces Connector write permissions as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which filters migration sources to Process Groups the requesting user is authorized to read, and requires write access to the source Process Group when submitting a migration request.	7.1	More Details
CVE-2026-90145	In the Linux kernel, the following vulnerability has been resolved: hinic3: Fix skb linearization mismatch and drop skb when skb_checksum_help() failed Previously, hinic3_send_one_skb() cached the skb fragment count before calling hinic3_tx_offload(). If hinic3_tx_csum() falls back to skb_checksum_help() for unsupported tunnel packets, the skb may be linearized. Continuing to build the TX descriptor with the stale fragment count leads to a descriptor mismatch, which can trigger out-of-bounds DMA reads or IOMMU faults. Furthermore, the old code ignored the return value of skb_checksum_help(), transmitting corrupted packets with incomplete checksums upon failure. Fix this by: 1. Moving the hinic3_tx_offload() call before calculating 'num_sge' to ensure the correct fragment count is used if the SKB is linearized. 2. Propagating skb_checksum_help() errors and returning HINIC3_TX_OFFLOAD_INVALID to properly drop the skb.	7.1	More Details

CVE-2026-44236	rabbitmq-c is a C-language AMQP client library for RabbitMQ. Prior to 0.16.0, a malicious AMQP server can send an undersized connection.tune.frame_max value during amqp_login(), and rabbitmq-c accepts the value in amqp_login_inner() in librabbitmq/amqp_socket.c. amqp_tune_connection() in librabbitmq/amqp_connection.c uses frame_max to reallocate the outbound buffer without enforcing AMQP_FRAME_MIN_SIZE. Immediate serialization of connection.tune-ok through amqp_frame_to_bytes() writes beyond the undersized heap allocation, causing memory corruption and likely denial of service. An on-path attacker can also trigger the flaw against plaintext AMQP traffic. Code execution is theoretically possible but was not demonstrated. This issue is fixed in version 0.16.0.	7.1	More Details
CVE-2026-91014	The Realtyna Organic IDX plugin + WPL Real Estate WordPress plugin before 5.4.2 does not sanitise and escape some of its parameters before reflecting them back in the page, allowing unauthenticated attackers to run arbitrary web scripts in a visitor's browser if they can trick the visitor into following a crafted link (reflected XSS).	7.1	More Details
CVE-2026-90161	In the Linux kernel, the following vulnerability has been resolved: erofs: fix interlaced ztailpacking pclusters On-disk sizes of interlaced pclusters should be block-aligned, and ztailpacking interlaced pclusters should be invalid at all. Currently, mkfs.erofs won't generate any interlaced pcluster with ztailpacking enabled, so this doesn't affect any existing valid filesystems. However, crafted images can contain invalid interlaced ztailpacking pclusters, resulting in an out-of-bounds read from a kmap'd page and copying irrelevant kernel memory into userspace-visible page cache.	7.1	More Details
CVE-2026-90372	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: avoid nss underflow in mt7915_mcu_get_sta_nss If a peer's VHT/HE MCS map has no supported spatial stream (all fields 0x3), the loop exits with nss == 0 and the function returned (u8)-1 (255), which was then written into the firmware sta_rec_bf beamforming fields. Clamp the result to 0.	7.1	More Details
CVE-2026-93485	Improper neutralization of input during web page generation ('cross-site scripting') vulnerability in Automattic WordPress core allows DOM-Based XSS. This issue affects WordPress versions 7.1 before 7.1.1; 7.0 through 7.0.4; 6.9 through 6.9.7; 6.8 through 6.8.8; 6.7 through 6.7.7; 6.6 through 6.6.7; 6.5 through 6.5.10; 6.4 through 6.4.10; 6.3 through 6.3.10; 6.2 through 6.2.11; 6.1 through 6.1.12; 6.0 through 6.0.14; 5.9 through 5.9.16; 5.8 through 5.8.15; 5.7 through 5.7.17; 5.6 through 5.6.19; 5.5 through 5.5.20; 5.4 through 5.4.21; 5.3 through 5.3.23; 5.2 through 5.2.26; 5.1 through 5.1.24; 5.0 through 5.0.27; 4.9 through 4.9.31; 4.8 through 4.8.30; and 4.7 through 4.7.35. The Unauthenticated Stored XSS vulnerability in the WordPress core can be reproduced on a default WordPress installation. Comment moderation is disabled by default, and the requirement for commenters to have a previously approved comment can be bypassed.	7.1	More Details
CVE-2026-68904	node-opcua is an OPC UA implementation for TypeScript and Node.js. From 2.0.0 until 2.170.0, node-opcua clients using the default keepSessionAlive setting can enter a repeated reconnection cycle when an OPC UA server's clock skew causes BadInvalidTimestamp responses. ClientSessionKeepAliveManager._ping_server treated the server-originated ServiceFault as a network outage and forced a transport reconnect, while ClientTCP_transport._on_ACK_response used socket.end() after failed HEL/ACK negotiation and could leave the connection in FIN-WAIT-2 when the peer did not close. Repetition at the keepAliveInterval accumulates file descriptors and memory until the client process or container can be terminated by resource exhaustion. This issue is fixed in version 2.170.0.	7.0	More Details
CVE-2026-92504	In the Linux kernel, the following vulnerability has been resolved: thermal: intel: int3400: clean up ODVP on probe failures evaluate_odvp() creates per-ODVP sysfs files before the thermal zone and later probe resources are registered. The current unwind path only calls cleanup_odvp() from the late sysfs failure path, so failures after evaluate_odvp() but before that label, including thermal_tripless_zone_device_register() failures, leave the ODVP files and storage behind. Move the ODVP cleanup to the common ART/TRT unwind path so every failure after evaluate_odvp() releases the ODVP state. Also clear the cached ODVP pointers in cleanup_odvp(), because evaluate_odvp() can already call it for partial setup failures while probe continues.	7.0	More Details
CVE-2026-90403	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: pci: fix error path in rtl_pci_probe() In the last error path in rtl_pci_probe(), the cleanup functions are skipped due to a wrong goto label. Moreover, the successful call to rtl_init_rfkill(), ieee80211_register_hw(), rtl_debug_add_one() have to be reverted. Fix this issue by updating the labels and adding the relevant cleanup functions to the last error path.	7.0	More Details
CVE-2026-90402	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: Fix controller cleanup on EDL sysfs failure mhi_register_controller() adds the controller device before creating the optional trigger_edl sysfs file. If sysfs_create_file() fails, the error path only drops the device reference and leaves the device registered. Hence, call device_del() in the error path before put_device().	7.0	More Details
CVE-2026-45720	Omni manages Kubernetes on bare metal, virtual machines, or in a cloud. Prior to 1.6.6 and from 1.7.0 until 1.7.3, SAML.getSession in internal/pkg/auth/interceptor/saml.go checks SAMLAssertion.Used and marks it used in separate state operations. Concurrent requests carrying the same captured saml-session token can each observe the assertion as unused and obtain authentication as the victim before either update is visible. The attacker can invoke SAML-protected gRPC endpoints, use ConfirmPublicKey to create multiple persistent credentials tied to the victim, and generate audit entries attributed to the victim, with the resulting access potentially affecting confidentiality, integrity, and availability according to the victim's privileges. This issue is fixed in versions 1.6.6 and 1.7.3.	7.0	More Details
CVE-2026-90353	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: fix ext PHY use-after-free on register error path After mt7915_register_ext_phy() succeeded, a failure of the main PHY mt7915_init_debugfs() or mt7915_coredump_register() unwound through free_phy2, which called ieee80211_free_hw() on the ext PHY hw while it was still registered with mac80211, since mt76_unregister_device() only unregisters the main hw. Unregister the ext PHY (thermal + phy + hw) first and skip the redundant free.	7.0	More Details
CVE-2026-93054	In the Linux kernel, the following vulnerability has been resolved: uio: Fix stale info pointer in failed registration path After device_add(), the UIO device is visible to userspace and /dev/uioX can be opened. If a later setup step fails, __uio_register_device() unwinds the device but leaves idev->info pointing at the caller-owned struct uio_info. That is unsafe when an opener races with the failed registration path. The open file keeps a reference to the uio_device, while the caller sees registration failure and may free its struct uio_info. Later file operations can then follow idev->info and dereference freed memory. Handle post-device_add() failures like unregister: remove UIO attributes while the info pointer is still valid, then clear idev->info under info_lock and wake existing waiters/async users before removing the device and minor. This makes already-open file descriptors observe the same "device gone" state as normal uio_unregister_device().	7.0	More Details
	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_fs: Fix fence cleanup in ffs_dmabuf_transfer()		

CVE-2026-93121	error paths The error paths for endpoint-disabled (ESHUTDOWN) and request-allocation failure (ENOMEM) in ffs_dmabuf_transfer() jump to err_fence_put which calls dma_fence_put() on the fence. However, at that point the fence has only been kmalloc'd — dma_fence_init() has not been called yet, so the refcount and the fence ops are uninitialized. Calling dma_fence_put() on such an object leads to undefined behavior. Use kfree() instead, since the fence is just a plain allocation at this stage, and rename the label to err_fence_free to reflect the actual cleanup action.	7.0	More Details
CVE-2026-93046	In the Linux kernel, the following vulnerability has been resolved: software node: Fix software_node_get_reference_args() with index -1 The bounds check for the index passed to software_node_get_reference_args() was failing when passed UINT_MAX, this in turn would lead to an out of bound access in the property array. Fix the bound check to also cover the UINT_MAX case.	7.0	More Details
CVE-2026-93116	In the Linux kernel, the following vulnerability has been resolved: platform/x86: asus-wmi: fix resource leaks on probe failure During driver initialization in asus_wmi_add(), various subsystems are registered sequentially. However, the error path labels are out of order relative to the registration sequence. Specifically: 1. If asus_wmi_custom_fan_curve_init() fails, the driver jumps to fail_custom_fan_curve. Because this label is placed below fail_sysfs, it bypasses the cleanup calls for the input device and sysfs groups, which were successfully registered before, leaking those resources. 2. If asus_screenpad_init() fails, the driver jumps to fail_screenpad. Because fail_screenpad is placed below fail_backlight, it bypasses the cleanup calls for backlight and rfkill, leaking those resources. Fix these resource leaks by reordering the error path labels in asus_wmi_add() to match the exact reverse order of the resource allocations.	7.0	More Details
CVE-2026-93176	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix dangling pointer in plane reset function amdgpu_dm_plane_drm_plane_reset() frees the old state before allocating a new one. If kzalloc() fails, the function returns without updating the state pointer, leaving a dangling pointer to already freed memory. Fix this by allocating the new state first. On allocation failure, the old state remains untouched and the function safely returns. Found by Linux Verification Center (linuxtesting.org) with SVACE. [adjust for movement around current amd-staging-drm-next]	7.0	More Details
CVE-2026-92488	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: complete object teardown when the destroy command fails erdma_destroy_qp(), erdma_destroy_cq(), erdma_dereg_mr(), and erdma_destroy_ah() returned early when erdma_post_cmd_wait() failed, leaking the queue buffers, MTTs, doorbells and the STAG, QPN, CQN and AHN identifiers. A command timeout clears ERDMA_CMDQ_STATE_OK_BIT and permanently disables the command queue, so no retry can succeed; the RDMA core keeps the object after a failed destructor and forced uverbs cleanup then nulls the pointers, making the resources unreachable. Warn on failure but release every software-owned resource and return success, since during terminal destruction the hardware command result is only diagnostic.	7.0	More Details
CVE-2026-73450	On affected platforms running Arista EOS with MLAG Dual Primary Detection configured, an unauthenticated attacker with access to the Dual Primary Detection network segment can send specially crafted packets to interfere with the dual-primary state. If the MLAG primary switch fails while these packets are present, the secondary switch incorrectly concludes it is in a dual-primary condition and err-disables its interfaces, leading to a traffic interruption.	6.9	More Details
CVE-2026-81447	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Certificate Validation vulnerability. An unauthenticated attacker with adjacent network access could potentially exploit this vulnerability, leading to Information disclosure and Information tampering.	6.8	More Details
CVE-2026-86784	The Visualizer WordPress plugin before 4.0.8 does not sanitise and escape a chart's JSON data source configuration before outputting it back in the chart editor, allowing users with the Contributor role and above to store JavaScript that executes in the browser of any higher-privileged user, such as an administrator, who reviews the affected chart.	6.8	More Details
CVE-2026-64684	RMCP is an official Rust SDK for the Model Context Protocol. Prior to 2.1.0, the rmcp crate's StreamableHttpClientTransport in crates/rmcp/src/transport/common/reqwest/streamable_http_client.rs builds its default_http_client with reqwest's automatic redirect policy and applies caller-supplied values from StreamableHttpClientTransportConfig.custom_headers without marking them as sensitive. When a malicious or compromised MCP endpoint returns a cross-origin 307 or 308 redirect, reqwest follows the redirect and apply_custom_headers causes custom API keys or authentication tokens to be replayed to the new origin, where an attacker can capture and reuse them. The separate auth_header path is not affected because it uses the standard Authorization header, which reqwest strips on cross-origin redirects. This issue is fixed in version 2.1.0.	6.8	More Details
CVE-2026-20248	A vulnerability in the DNS over TCP implementation of Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause the TCP DNS response handler to unexpectedly restart, causing the device to reload. This vulnerability is due to a logic error when parsing a DNS query and tracking the size of the incoming buffers. An attacker could exploit this vulnerability by formatting a crafted reply to a DNS query sent from the targeted device. A successful exploit could allow the attacker to cause the device to reload, causing a denial of service (DoS) condition. Note: The attacker must be able to respond to DNS queries from the device, either by controlling the DNS service or through a machine-in-the-middle attack.	6.8	More Details
CVE-2026-84088	The Xpro Addons — 140+ Widgets for Elementor WordPress plugin before 1.7.9 does not validate or sanitize a widget link setting before storing and using it in a JavaScript navigation call, allowing users with the contributor role and above to inject and store JavaScript that executes in the browser of anyone who interacts with the affected widget.	6.8	More Details
CVE-2026-76555	The WP Import Export Lite WordPress plugin before 3.9.33 does not validate a user-supplied file path before reading it and copying it into a publicly accessible directory, allowing any user whose role an administrator has granted the WP Import Export Lite WordPress plugin before 3.9.33's import permission to disclose sensitive files from the server, including files located outside the web root. The same code path also relaxes the file-system permissions of any path it is given, whether or not the copy succeeds.	6.8	More Details
CVE-2026-77401	Zope AccessControl provides a general security framework for use in Zope. Prior to 7.4, applications that allow untrusted users to create and execute AccessControl-controlled Python code do not safely guard str.format and str.format_map when those methods are reached through a str subclass. In both ImplPython.py and cAccessControl.c, Python formatting can recursively access attributes and subscriptions using unrestricted getattr and getitem behavior instead of the policy-restricted getattr and getitem operations. A controlled format string can therefore disclose objects reachable from values available to the formatting operation. This issue is fixed in version 7.4.	6.8	More Details
CVE-2026-	The WP Import Export Lite WordPress plugin before 3.9.33 does not properly sanitise and escape some export filter values before using them in SQL statements, allowing users holding its export permission, which administrators have by default and	6.8	More

76556	may also grant to lower roles, to perform SQL injection attacks.		Details
CVE-2026-91011	The EWWW Image Optimizer WordPress plugin before 8.7.7 does not properly escape image attribute values when it rewrites page output, allowing authenticated users with author-level access and above to inject arbitrary JavaScript that is stored in published content and executes in the browser of any user who later views the affected page.	6.8	More Details
CVE-2026-86788	The HT Mega Addons for Elementor WordPress plugin before 3.2.6 does not restrict the HTML tag name used to render the section headline in several of its widgets and blocks to a safe allowlist, allowing users with contributor-level access and above to store a crafted tag name that executes arbitrary JavaScript when the content is viewed, including by higher-privileged users who review or publish it.	6.8	More Details
CVE-2026-84902	The King Addons for Elementor WordPress plugin before 51.1.81 does not perform an object-level authorization check when importing template content into a page, allowing users with contributor-level access and above to overwrite the Elementor content of arbitrary posts and pages, including those owned by administrators, and to inject JavaScript through a widget setting that is output without escaping, resulting in Stored Cross-Site Scripting that executes in the session of any user who views the affected page.	6.8	More Details
CVE-2026-76557	The WP Import Export Lite WordPress plugin before 3.9.33 does not properly sanitise and escape some import configuration values before using them in SQL statements, allowing users whose role an administrator has granted the WP Import Export Lite WordPress plugin before 3.9.33's import permission to perform SQL injection attacks.	6.8	More Details
CVE-2026-92140	Jenkins Gitee Plugin 1301.v8957053c7902 and earlier does not escape the sender name from Gitee push webhook payloads in build causes, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers able to trigger builds via the Jenkins Gitee Plugin webhook endpoint.	6.8	More Details
CVE-2026-92616	FileRise before version 3.28.0 contains a privilege escalation vulnerability that allows authenticated low-privilege attackers to gain unauthorized read and write access by exploiting improper session isolation between the WebDAV interface and the web application session context. Attackers can combine valid Basic-Auth credentials with an active admin PHPSESSID cookie to bypass authorization boundaries, as the WebDAV layer incorrectly inherits elevated privileges from an ambient web session rather than enforcing independent stateless authentication per RFC 4918.	6.8	More Details
CVE-2026-76558	The WP Import Export Lite WordPress plugin before 3.9.33 does not escape custom field names retrieved from the database before inserting them into the DOM of one of its administration screens, allowing users with a role as low as contributor to perform Stored XSS attacks which will trigger in the browser of a high privileged user, such as an administrator, viewing that screen.	6.8	More Details
CVE-2026-85717	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 2.14.5 to 2.16.0 and from 3.0.9 to 3.0.11, a client configured with a client-wide Realm and redirect following can disclose credentials after a cross-origin redirect because the Interceptors authentication path falls back to the client configuration after redirect handling clears the per-exchange realm. If the attacker-controlled target returns 401, the client can send Basic or Digest credentials or a Negotiate or NTLM token to that origin. Per-request realms are stripped correctly, and this issue is a residual bypass of the earlier cross-origin credential-stripping fixes. This issue is fixed in versions 2.16.1 and 3.0.12.	6.8	More Details
CVE-2026-92952	vm2 versions 3.11.4 through 3.11.6 incompletely filter Node.js registered internal symbols across the sandbox boundary. The extraction filters in lib/setup-sandbox.js and the cross-realm symbol checks and write traps in lib/bridge.js use a fixed list of known dangerous registered symbols that omits nodejs.stream.disturbed and nodejs.stream.errored, which are exposed on host WebStream prototypes on newer Node.js releases (validated on Node.js v25.8.0). When the embedder exposes a host WebStream object and the host stream/web module to the sandbox, sandbox code can obtain the real host symbols via Object.getOwnPropertySymbols(streamWeb.ReadableStream.prototype) and use them as write keys on host stream objects, corrupting host-visible stream state — for example making stream.Readable.isDisturbed() return false for an already-consumed stream. This can bypass host logic that relies on Node's public stream-state helpers to enforce one-shot body consumption, reject errored streams, or decide whether a stream is safe to hand to another component. It is not a host code-execution primitive in the reported proof of vulnerability. This is an incomplete fix for the earlier nodejs.* symbol filtering issue. Fixed in vm2 3.11.7.	6.8	More Details
CVE-2026-92800	Docs before 5.4.1 fails to properly revoke websocket collaboration connections when access is revoked at parent documents. Attackers with revoked access can retain real-time read and write access to sub-documents through open websocket sessions that are never disconnected.	6.8	More Details
CVE-2026-92812	decap-server contains a path traversal vulnerability in the local proxy containment guard that uses plain string prefix comparison without path separator validation. Attackers can access sibling directories whose names begin with the repository directory name to read, write, or delete files outside the intended repository root.	6.8	More Details
CVE-2026-88993	The All Bootstrap Blocks WordPress plugin through 1.3.31 does not properly escape a block attribute before outputting it in HTML tag-name position, allowing users with Contributor-level access and above to inject arbitrary web scripts that execute when the affected content is viewed.	6.8	More Details
CVE-2026-26947	Dell ECS versions 3.8.1.0 through 3.8.1.7, and Dell ObjectScale versions prior to 4.4.0.0, contains an Improper Privilege Management vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of privileges.	6.7	More Details
CVE-2026-25261	Memory corruption while processing rear sensor IOCTL calls.	6.7	More Details
CVE-2026-92615	A flaw was found in flightctl. The configureRepoHTTPSClient() function in the device-render worker builds a per-repository tls.Config (which may include InsecureSkipVerify, a custom CA bundle, or tenant-supplied mTLS client certificates) and installs it into go-git's process-global client.Protocols map via gitclient.InstallProtocol("https", ...). Because the worker renders devices for multiple organizations concurrently from a shared goroutine pool, whichever tenant's repository configuration is written last wins for all in-flight git.Clone calls. This race condition can cause one tenant's TLS settings, including InsecureSkipVerify or mTLS client credentials, to leak into another tenant's git operations.	6.6	More Details

CVE-2026-88994	The All Bootstrap Blocks WordPress plugin through 1.3.31 does not validate a block attribute before using it to build a filesystem path that is included at render time, allowing users with contributor-level access and above to include arbitrary local files, disclose their contents, and execute PHP where a local file containing PHP code can be reached. Exploitation requires the plugin's Lightspeed subsystem to be enabled, which is not the default.	6.6	More Details
CVE-2026-86109	The VeloCloud Edge software update workflow may accept update bundles without properly validating their signatures because the workflow does not restrict the digest algorithm used for artifact verification. An attacker with either sufficient privileges to upload packages to VeloCloud Orchestrator or credentials permitting direct access to an Edge may be able to install unauthorized software.	6.6	More Details
CVE-2026-93455	django-page-cms through 2.0.13 fails to properly validate page permissions in admin helper views, allowing any staff account to read arbitrary page content and stored media paths. Attackers with low-privilege staff credentials can enumerate content identifiers and access unpublished drafts, page listings, and file paths without proper authorization checks.	6.5	More Details
CVE-2026-92795	Coze Studio through 0.5.1 fails to restrict the server URL supplied when registering plugin tools, allowing authenticated users to make the backend fetch internal services. Attackers can construct plugin requests to access cloud metadata endpoints and internal services reachable only from the backend network, reading responses containing sensitive information.	6.5	More Details
CVE-2026-79713	The Breeze Cache WordPress plugin before 2.5.15 does not include a set of tracking-related query parameters in its page-cache key while still caching pages requested with them, allowing unauthenticated attackers to have a page rendered under their own request context stored under, and served from, the clean URL's cache entry to every subsequent visitor.	6.5	More Details
CVE-2026-17576	The InfiniteWP Client plugin for WordPress is vulnerable to SQL Injection via the get_comments action in versions up to, and including, 1.13.9. This is due to insufficient escaping on the array-key names supplied in the JSON request body before use in a SQL statement: IWP_MMB_Comment::get_comments() calls extract() on $args (which silently skips keys that are not valid PHP variable names) but a second foreach($args as $checkbox => $checkbox_val) processes every key, strips the 'iwp_get_comments_' prefix with str_replace(), wraps the remainder in single quotes, and imploded it into an IN(...) clause that is executed via $wpdb->get_results() with no prepare(). Because the request body is read from php://input and JSON-decoded, wp_magic_quotes() never touches the data, so quote characters in keys pass through unaltered. This makes it possible for authenticated attackers, with administrator-level access and above (an administrator can register their own public key via add_site using the plugin's WP-admin-generated activation_key and then issue signed get_comments requests), to append additional SQL queries into already existing queries that can be used to extract sensitive information from the database.	6.5	More Details
CVE-2026-85078	Sanic is an opensource python web server/framework. In version 25.12.0, Sanic's core HTTP/1.1 chunked-body handling does not fully consume the trailer-part after the terminating zero chunk before reusing the keep-alive connection buffer. A remote unauthenticated client can place attacker-controlled bytes in that trailer region, causing Sanic to parse and route them as a hidden second request after the outer request. This breaks HTTP request-boundary integrity and can provide a request-smuggling primitive when Sanic is deployed behind intermediaries. This issue is fixed in version 25.12.1.	6.5	More Details
CVE-2026-78227	NLnet Labs Unbound 1.22.0 up to and including 1.26.1, has a use-after-free vulnerability when compiled for DNS-over-QUIC support with '--with-libngtcp2'. Each DoQ stream owns an output buffer that holds the DNS response. ngtcp2's retransmission buffer keeps a shallow pointer into the output buffer for as long as a STREAM frame may be resent. On a client RESET_STREAM, the output buffer is freed but ngtcp2 still holds the matching retransmission entries. The next PTO timeout makes ngtcp2 re-encode the STREAM frame and copy from the freed buffer. A malicious actor that can query Unbound over DoQ and that withholds ACKs, sends RESET_STREAM, and waits for PTO, reaches this use-after-free with no privilege. This leads to retransmissions against freed memory and eventually an abnormal server exit under a 20-query spray.	6.5	More Details
CVE-2026-85131	The WPLP Cookie Consent WordPress plugin before 4.4.4 does not perform CSRF or capability checks when processing bulk actions on its administration screens, and does not restrict the targeted items to its own records, allowing attackers to make a logged in admin permanently delete arbitrary posts and pages via a crafted request.	6.5	More Details
CVE-2026-66574	Contributor Cross Site Scripting (XSS) in Element Pack Elementor Addons <= 8.8.3 versions.	6.5	More Details
CVE-2026-92912	AVideo through c3edcc274c389816d434acadac07ee78eaf330c1 uses cryptographically weak uniqid() values for RTMP publish keys in LiveTransmition, reducing key entropy to approximately one million possibilities per creation second. Attackers who know the channel creation time can brute-force the five-digit microsecond component to forge valid stream keys and broadcast content as the channel owner.	6.5	More Details
CVE-2026-78294	Contributor Cross Site Scripting (XSS) in Geo Mashup <= 1.13.21 versions.	6.5	More Details
CVE-2026-66573	Contributor Cross Site Scripting (XSS) in JetTabs <= 2.3.3.1 versions.	6.5	More Details
CVE-2026-86465	Apache Airflow Akeyless provider: the Akeyless secrets backend's team-scope guard can be bypassed with a user-controlled key. In a multi-team deployment, a Dag author scoped to one team can supply a Variable key containing a path separator that causes the backend to resolve a secret belonging to a different team, because the lookup path is concatenated from an unvalidated key after the team-scoped lookup misses. The Execution API Variables route accepts a path-shaped key, so this is reachable from ordinary Dag code. Affects multi-team deployments using the Akeyless secrets backend. Single-team deployments are not affected, as there is no cross-team boundary to cross. This is the same class as CVE-2026-68870, CVE-2026-68871 and CVE-2026-68872 in the Azure Key Vault, Yandex Lockbox and Amazon secrets backends. Users of apache-airflow-providers-akeyless are recommended to upgrade to version 0.3.1 or later, which fixes the issue.	6.5	More Details
CVE-2026-92790	Higress before 2.2.4 panics when processing a Cookie header segment without an equals sign, causing the plugin wrapper to recover and return a continue action that bypasses AI token rate limiting. Unauthenticated attackers can craft a malformed Cookie header to skip rate limit checks and exceed thresholds intended to restrict costly model backend calls.	6.5	More Details

CVE-2026-66572	Contributor Cross Site Scripting (XSS) in JetBlog <= 2.4.10 versions.	6.5	More Details
CVE-2026-92789	Graylog through 7.1.4 validates outbound URLs against an allowlist before making requests but fails to re-validate after following HTTP redirects. Attackers with lookup table or event notification permissions can craft allowlisted endpoints that redirect to internal services, enabling the server to fetch and return internal responses.	6.5	More Details
CVE-2026-90923	The Autopay WordPress plugin before 5.0.1 does not enforce the signature on one of its payment callbacks, allowing unauthenticated users to disclose and delete the stored payment parameters of other customers' orders.	6.5	More Details
CVE-2026-93451	snappy-java through 1.1.10.8 contains a buffer overflow vulnerability in typed Snappy.uncompress*Array methods that allocate output arrays by dividing uncompressed length by element size but pass the undivided length to native code. Attackers controlling compressed input can cause misaligned length values to write past array bounds with attacker-controlled bytes, corrupting heap memory.	6.5	More Details
CVE-2026-16588	The WP Directory Kit plugin for WordPress is vulnerable to blind SQL Injection via the 'order_by' parameter in all versions up to, and including, 1.5.4 due to insufficient escaping on the user supplied parameter and lack of sufficient preparation on the existing SQL query. This makes it possible for authenticated attackers, with custom-level access and above, to append additional SQL queries into already existing queries that can be used to extract sensitive information from the database.	6.5	More Details
CVE-2026-89034	TCH QRing smart ring model R20_B006 running firmware RT09R20_1.00.00_250318 contains an unauthenticated Bluetooth Low Energy access vulnerability that allows any nearby attacker to connect to the device without pairing, authentication, or user approval by exploiting the exposed Nordic UART Service which enforces no client authentication or command authorization. Attackers within Bluetooth Low Energy range can connect directly to the ring, bypassing the official application and cloud authentication, to read battery levels, activate live heart rate monitoring, and retrieve stored historical heart rate and blood oxygen records.	6.5	More Details
CVE-2026-61588	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, when a Django `Model` instance is assigned to a public view attribute, djust serialized it to the client with no sensitive-field denylist — sending fields such as `password` (the hash), privilege flags (e.g. `is_staff` / `is_superuser`), tokens, and other PII to the browser. Because exposing model objects to templates is a normal djust pattern, this could leak credentials/PII without the developer realizing the full object crossed the wire. This is fixed in djust 1.0.7. Model serialization applies a secure-by-default sensitive-field denylist (password/hash/token/secret-style fields and known privilege flags are withheld) with an identity-subset fallback. As a workaround, keep `Model` instances on `_private` attributes and expose only the specific fields needed, until patched.	6.5	More Details
CVE-2026-92583	AVideo through 29.0 contains a race condition in the enforceRateLimit() function that fails to atomically increment rate limit counters, allowing attackers to bypass all rate limits including login brute-force protection by issuing concurrent requests. Attackers can submit parallel credential attempts to exceed the documented 30-attempts-per-5-minutes login limit by an arbitrary factor determined only by their connection concurrency.	6.5	More Details
CVE-2026-89207	A vulnerability has been identified in WTV676-HB6035 Web Interface (All versions < V3.94), WTV776-HB6035 Web Interface (All versions < V4.17). Affected devices do not properly validate input received from backend services. This could allow an unauthenticated remote attacker to force the device into protection mode, which results in losing remote connectivity functions (Web Access).	6.5	More Details
CVE-2026-76553	The WP Import Export Lite WordPress plugin before 3.9.33 does not validate a path taken from stored, user-supplied data before recursively deleting the directory it resolves to, allowing users to whom an administrator has delegated a WP Import Export Lite WordPress plugin before 3.9.33 capability to delete arbitrary directories, and every file within them, including outside the web root.	6.5	More Details
CVE-2026-27553	A low-privileged remote attacker can manipulate the schema path parameter in the /index.php/diagnostics_tab/ajax_diag_table_rows endpoint using a valid user cookie allowing disclosure of all user password hashes.	6.5	More Details
CVE-2026-92598	Nodemailer before 9.1.0 fails to apply UTS-46 normalization when encoding international domain names, causing the domain resolver to compute a different Punycode A-label than standards-compliant parsers. Attackers can craft recipient addresses with invisible characters or compatibility mappings that pass domain allow-list checks but are delivered to attacker-controlled domains via SMTP.	6.5	More Details
CVE-2026-66617	Contributor Cross Site Scripting (XSS) in PublishPress Series <= 3.1.3 versions.	6.5	More Details
CVE-2026-66579	Contributor Cross Site Scripting (XSS) in JetElements For Elementor <= 2.9.2.1 versions.	6.5	More Details
CVE-2026-85009	The RestroPress WordPress plugin through 3.4.6 does not verify ownership in its payment-recovery flow before acting on a request-supplied order identifier, allowing unauthenticated attackers to enumerate which orders are in a recoverable state and to write notes to another customer's order.	6.5	More Details
CVE-2026-66578	Contributor Cross Site Scripting (XSS) in PropertyHive <= 2.2.6 versions.	6.5	More Details
CVE-2026-	Steeltoe is an open source project that provides a collection of libraries that helps users build cloud-native applications. Prior to 4.3.0, Steeltoe.Security.Authorization.Certificate deployments using AddOrgAndSpacePolicies() and UseCertificateAuthorization() trust the public certificate supplied in the X-Client-Cert request header without proving possession of the corresponding private key. Common Cloud Foundry routers do not remove this header from inbound requests. When	6.5	More

81868	inbound requests are not restricted to a known trusted proxy source IP, an attacker who obtains the public certificate of an application instance in the target organization or space and can reach the application can spoof X-Client-Cert to bypass the SameOrg and SameSpace policies for the certificate validity period. This issue is fixed in version 4.3.0.		Details
CVE-2026-66577	Contributor Cross Site Scripting (XSS) in JetSearch <= 3.6.3 versions.	6.5	More Details
CVE-2026-81453	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Filesystem access for attacker.	6.5	More Details
CVE-2026-92811	browserless versions 1.44.0 through 2.56.7 fail to enforce file protocol restrictions in Playwright websocket endpoints, allowing authenticated token holders to read arbitrary files. Attackers can navigate Playwright-driven browsers to file scheme URLs and access files accessible to the container process despite the ALLOW_FILE_PROTOCOL setting defaulting to false.	6.5	More Details
CVE-2026-66576	Contributor Cross Site Scripting (XSS) in JetBlocks For Elementor <= 1.5.2 versions.	6.5	More Details
CVE-2026-92597	Nodemailer versions >= 6.9.16 and < 9.1.0 mis-parse RFC 5322 comments in email addresses: in lib/addressparser, a comment closed immediately before a non-break character causes the tokenizer to concatenate the atoms surrounding the comment instead of treating the comment as folding whitespace that terminates the domain. A recipient address such as user@good-corp.com(x)evil.com is therefore read by Nodemailer as the single domain good-corp.comevil.com (registrable domain comevil.com, which an attacker can register) and used for both the SMTP envelope (RCPT TO) and the emitted To:/From: headers, while a conformant RFC 5322 parser terminates the domain at the comment and reads good-corp.com. An application that validates the recipient domain with a strict RFC 5322 parser (without inspecting parse defects) or a naive prefix/substring allow-list and then hands the raw address to Nodemailer can be induced to deliver mail to a domain the attacker controls. Fixed in 9.1.0.	6.5	More Details
CVE-2026-92750	Harness through 3.3.0 omits access control validation in the infrastructure provider read endpoint, allowing authenticated users to retrieve provider configurations from spaces they do not belong to. Attackers can query the GET /api/v1/infraproviders endpoint with arbitrary space identifiers to expose sensitive provider metadata including Docker endpoints, TLS certificate paths, and cloud project identifiers.	6.5	More Details
CVE-2026-54676	Scoold is a Q&A and a knowledge sharing platform for teams. Prior to 1.69.0, users with personal API tokens can retrieve replies from questions in private spaces they cannot access because src/main/java/com/erudika/scoold/api/ApiController.java in ApiController.getPostReplies() does not apply canAccessSpace before returning data from GET /api/posts/{id}/answers. The issue is reachable when scoold.api_user_access_enabled and scoold.api_enabled are true and a token holder knows or enumerates a private question identifier. Under those conditions, the question endpoint can deny access while the answers endpoint returns the private reply bodies, exposing confidential team or project discussions. This issue is fixed in version 1.69.0.	6.5	More Details
CVE-2026-92460	yshop-crm through 2.1.3 fails to enforce authorization on the GET /admin-api/crm/operatelog/page endpoint, allowing any authenticated back-office user to access the installation-wide audit trail. Attackers can query the operation log to retrieve operator names, display nicknames, client IP addresses, User-Agent strings, request URLs, action details, and customer identifiers without proper permission checks.	6.5	More Details
CVE-2026-69147	vLLM is an inference and serving engine for large language models. Prior to 0.28.0, request bodies for Chat Completions and Responses can set media_io_kwargs.video.video_backend to pynvvideocodec, and MediaConnector.fetch_video forwards that choice to VideoMediaIO even when startup configuration selected a software decoder. The engine's _reserve_mm_ipc_gpu_memory logic budgets decoder memory only from static configuration, so the request-selected VIDEO_LOADER_REGISTRY backend can create a CUDA context, decoder surfaces, and decoded-frame allocations that were not removed from the engine's KV-cache budget. An attacker able to submit video requests to a video-capable GPU deployment with PyNvVideoCodec installed can exhaust shared GPU memory, causing request failures, worker crashes, or denial of service. The first release containing the fix is version 0.28.0.	6.5	More Details
CVE-2026-56719	MikroTik RouterOS before 7.24 contains an out-of-bounds read vulnerability in the userspace SMB daemon that allows unauthenticated attackers to read beyond the end of the request buffer by supplying a crafted uniPwdLen field value in a minimal SMB1 SessionSetupAndX frame. The out-of-bounds read occurs in the SessionSetupAndX handler before any credential validation, potentially exposing sensitive memory contents.	6.5	More Details
CVE-2026-86862	pgAdmin 4's Restore and Maintenance tools passed the client-supplied 'database' field directly as the value of the --dbname option given to pg_restore and psql. libpq expands a database name containing an equals sign into a full connection string, and connection keywords embedded in that value take precedence over the --host and --port arguments that pgAdmin supplies. A value such as 'host=attacker.example port=5432 dbname=x' therefore redirected the utility to a server chosen by the requesting user rather than the server the operation was invoked against. Because pgAdmin exports the decrypted stored database password in the PGPASSWORD environment variable before executing the utility, the redirected connection presents that credential to the attacker-nominated endpoint, which may capture it. The redirection additionally permits outbound connections from the pgAdmin host to arbitrary network addresses, including hosts not otherwise reachable by the requesting user. The behaviour is reachable by any authenticated user holding the tools_restore or tools_maintenance permission, both of which the default User role grants. The Maintenance tool was not affected in the earliest releases, where the value was wrapped by a quoting helper that incidentally prevented expansion; it became affected when that wrapper was removed. The fix supplies the target database in the PGDATABASE environment variable, which libpq treats as a literal database name and never expands as a connection string. Where pg_restore requires a --dbname argument to be present, an empty value is passed, which contains no equals sign and is therefore not expanded, while the real name is taken from the environment. This issue affects pgAdmin 4: from the introduction of the --dbname argument in the Restore and Maintenance tools before 9.18.	6.5	More Details
CVE-2026-	TDuck survey form through version 5.0 contains an authorization bypass vulnerability in the POST /user/form/data/update endpoint that allows authenticated users to overwrite other users' form submission data. Attackers can discover submission identifiers allocated in narrow ranges and modify arbitrary form responses containing personal data by sending update requests	6.5	More Details

92567	without ownership validation.		
CVE-2026-92570	reNgine through 2.2.0 contains an authorization bypass vulnerability in the GetFileContents API endpoint that allows any authenticated user to read bundled recon tool configuration files. Attackers with low-privilege Auditor roles can access files containing third-party API keys for services like SecurityTrails, Shodan, Censys, VirusTotal, BinaryEdge and Hunter by querying the endpoint without role-based permission checks.	6.5	More Details
CVE-2026-92603	ContiNew Admin through 4.1.0 contains an authorization bypass vulnerability in the personal message delete endpoint that allows authenticated users to delete other users' messages and announcements. Attackers can supply arbitrary message identifiers in the IdsReq parameter to remove any message row and purge all recipients' read receipts without ownership validation.	6.5	More Details
CVE-2026-92601	Guns through 8.3.5 contains an improper access control vulnerability in SysNoticeController where requiredPermission defaults to false and is not overridden by any action methods. Authenticated users without assigned roles can exploit this to create, edit, delete, publish and retract system-wide notices affecting arbitrary users and departments.	6.5	More Details
CVE-2026-76438	A vulnerability in the web-based management interface of Cisco BroadWorks CommPilot Application Software could allow an authenticated, remote attacker with low privileges to alter configurations on an affected device. This vulnerability is due to missing authorization checks. An attacker could exploit this vulnerability by sending a crafted HTTP request. A successful exploit could allow the attacker to alter configurations on select pages.	6.5	More Details
CVE-2026-92463	yshop-crm through 2.1.3 contains an authorization failure in the GET /admin-api/system/user/page endpoint where the @PreAuthorize annotation is commented out, allowing authenticated back-office users without system:user:list permission to enumerate all users. Attackers with valid back-office credentials and a role with data scope ALL can retrieve the complete user directory including login names, nicknames, departments, email addresses, mobile numbers, and last login information.	6.5	More Details
CVE-2026-92462	yshop-crm through 2.1.3 fails to enforce authorization checks on the CrmFlowController deleteFlowStep endpoint, allowing any authenticated back-office user to delete arbitrary approval workflow steps. Attackers can invoke the DELETE /admin-api/crm/flow/delete-step endpoint without required permissions to remove approval steps that control contract, receivable, and invoice finalization processes.	6.5	More Details
CVE-2026-57173	vLLM is an inference and serving engine for large language models. Prior to 0.24.0, the input_audio handling path for /v1/chat/completions calls AudioMediaIO.load_bytes or AudioMediaIO.load_file without passing VLLM_MAX_AUDIO_DECODE_DURATION_S to the shared audio decoder. An unauthenticated client can therefore submit a small compressed audio input that expands into a very large float32 PCM allocation, bypassing the duration guard already used by /v1/audio/transcriptions and causing an out-of-memory worker crash. Inline data URLs reach this path without being bounded by VLLM_AUDIO_FETCH_TIMEOUT. The issue affects deployments serving an audio-capable model, and authentication changes only the deployment-specific reachability. This issue is fixed in version 0.24.0.	6.5	More Details
CVE-2026-62949	AsyncSSH is a Python package which provides an asynchronous client and server implementation of the SSHv2 protocol on top of the Python asyncio framework. Prior to 2.24.0, _process_channel_open and _process_channel_open_confirmation in asyncssh/connection.py accept a peer-supplied send_pktsize value of zero. When channel data reaches SSHChannel._flush_send_buf in asyncssh/channel.py, the zero value causes each loop iteration to slice and remove zero bytes without reducing the send window, leaving the synchronous loop permanently true with no await point. A malicious SSH server can trigger the client path through SSH_MSG_CHANNEL_OPEN_CONFIRMATION before the first channel write, while an authenticated client can trigger the server path through SSH_MSG_CHANNEL_OPEN and freeze every current and future connection handled by the process. This vulnerability is fixed in 2.24.0.	6.5	More Details
CVE-2026-54677	Scoold is a Q&A and a knowledge sharing platform for teams. Prior to 1.69.0, authenticated users who are not members of a private space can create content in questions belonging to that space because src/main/java/com/erudika/scoold/controllers/QuestionController.java in QuestionController.reply() and src/main/java/com/erudika/scoold/controllers/CommentController.java in CommentController.createAjax() do not apply the canAccessSpace authorization check used by the question read path. With scoold.is_default_space_public set to false and private spaces in use, a user with a valid session and a known or enumerable question identifier can send requests to POST /question/{id} and POST /comment, causing replies and comments to be stored in a thread the user cannot read. This permits unauthorized modification of private discussions and can trigger notifications that reveal the existence or metadata of private activity. This issue is fixed in version 1.69.0.	6.5	More Details
CVE-2026-92600	Guns through 8.3.5 contains an information disclosure vulnerability in SysUserController where /sysUser/detail and /sysUser/page endpoints omit requiredPermission configuration, causing the permission interceptor to skip RBAC validation for authenticated users. Attackers with any valid login token can retrieve sensitive user information including account names, real names, email addresses, phone numbers, last login IPs, and role assignments for all users in the system.	6.5	More Details
CVE-2026-92459	yshop-crm through 2.1.3 contains a missing authorization vulnerability in the CrmCluesController receiveCustomer endpoint that allows authenticated back-office users to claim sales leads without proper permission checks. Attackers can invoke the lead-claim endpoint to reassign leads from other employees to themselves by overwriting the ownerUserId field, with no access logging or quota validation to prevent bulk lead theft.	6.5	More Details
CVE-2026-92457	yshop-crm through 2.1.3 contains a missing authorization vulnerability in the CrmInvoiceController issueInvoice endpoint that allows authenticated back-office users to issue arbitrary invoices. Attackers can call the PUT /admin-api/crm/invoice/issue endpoint without required permissions to modify invoice status, inflate contract invoiced amounts with attacker-chosen values, and trigger invoice emails to arbitrary addresses.	6.5	More Details
CVE-2026-87076	Tanium addressed an information disclosure vulnerability in Discover.	6.5	More Details
CVE-2026-	MikroORM is a TypeScript ORM for Node.js based on Data Mapper, Unit of Work and Identity Map patterns. Prior to 6.6.16 and 7.1.7, the shared SQL layer validates the field key of an orderBy clause but does not validate its direction value before AbstractSqlPlatform.getOrderByExpression concatenates it into an ORDER BY clause. Applications that bind attacker-controlled request data to the direction in em.find(), em.findOne(), em.findAndCount(), QueryBuilder.orderBy(), or	6.5	More

84993	QueryBuilderHelper.getQueryOrderFromObject() can permit a raw SQL fragment that performs blind or boolean extraction of data available to the database account. The BaseMySqlPlatform and MsSqlPlatform fallthrough paths have the same behavior, affecting SQLite, PostgreSQL, MySQL, MariaDB, MSSQL, libSQL, and Oracle drivers, while MongoDB is not affected. This issue is fixed in versions 6.6.16 and 7.1.7.		Details
CVE-2026-82561	Apache NiFi 1.5.0 through 2.11.0 provide REST API methods that replace the entire contents of a Process Group using a client-supplied flow definition, covering Process Group flow replacement together with versioned flow update and rebase operations. Framework authorization for these methods was limited to read and write privileges on the Process Group itself, unlike the corresponding asynchronous update request methods, which also authorize the components encapsulated in the Process Group along with referenced Controller Services, Parameter Contexts, and Parameter Providers. As a result of the missing authorization, an authenticated user with write access to a Process Group could supply a flow definition that modifies or removes components in descendant Process Groups protected by more restrictive access policies, and could bind components to Controller Services and Parameter Contexts without authorization for those referenced components. Existing verification checks limited the impact to stopped components, and the issue applies only to deployments that use component-level authorization policies, because the framework enforces write permissions as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which applies consistent reference resolution and component authorization across Process Group replacement and versioned flow update methods	6.5	More Details
CVE-2026-86358	Dell Update Package Framework, versions prior to 26.07.03, contains a Stack-based Buffer Overflow vulnerability. An unauthenticated attacker with adjacent network access could potentially exploit this vulnerability, leading to Remote execution.	6.5	More Details
CVE-2026-67071	HCL DevOps Deploy / HCL Launch is susceptible to an information disclosure vulnerability when processing redacted property values. If a deployment is configured with a secure property that starts with certain non-ASCII characters, the redaction engine may fail to mask subsequent ASCII secure values embedded inside insecure properties.	6.5	More Details
CVE-2026-87028	Concrete CMS 9 through 9.5.3 did not confirm that a board InstanceItem submitted to the custom-slot preview endpoint belonged to the board instance the requesting user was authorized to edit, and did not enforce page-view permission before generating page-backed summary content. As a result, an authenticated user holding edit-board-contents permission on a single board instance could submit the identifier of an item belonging to a different board instance and receive summary fields, including the page title and description, of an underlying page the same user was otherwise forbidden to view. The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 5.3 with vector CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N. Thanks Pakung for reporting.	6.5	More Details
CVE-2026-93385	Information leak in Paint in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to obtain sensitive information via a crafted HTML page. (Chromium security severity: Medium)	6.5	More Details
CVE-2026-77190	On affected platforms running Arista EOS, an unauthenticated attacker who is network-adjacent to the switch and able to connect to a device with PIM Sparse Mode and MLAG configured, can send malformed messages that cause the Pimsm agent to terminate unexpectedly. The Pimsm agent is automatically restarted, but repeated attacks can cause the agent to restart continuously, resulting in a sustained denial of service.	6.5	More Details
CVE-2026-92605	IRIS through 2.4.29 fails to properly validate case authorization in comment listing endpoints for notes, tasks, IOCs, assets, and evidence items. Attackers with access to any single case can enumerate sequential object identifiers and read comment threads from cases they have no authorization to access.	6.5	More Details
CVE-2026-20283	A vulnerability in the IPsec Open API endpoint of Cisco ISE could allow an authenticated, remote attacker to inject arbitrary commands on the underlying operating system. This vulnerability is due to insufficient validation of user-supplied input in IPsec Open API calls. An attacker could exploit this vulnerability by sending crafted input to the IPsec Open API endpoint on an affected device. A successful exploit could allow the attacker to execute arbitrary commands on the underlying operating system. To exploit this vulnerability, the attacker must have valid administrative credentials and the node must have more than one network interface, one of which must be configured as an active IPsec tunnel. Note: For CVE-2026-20283, Cisco has assigned a Security Impact Rating (SIR) of High rather than Medium as the score indicates. The reason is that it is easy to get to root from the achieved privilege level.	6.5	More Details
CVE-2026-19033	For a secondary zone with transfers restricted by TSIG, `named` may start to serve the data provided in a zone transfer before the final message with the TSIG signature arrives. This could allow an attacker that does not actually possess a valid TSIG signature to send unauthorized zone contents to a secondary server. Although no TSIG signature ever arrives, `named` does not rollback to the pre-transfer state. To exploit the vulnerability, the transfer must be a multi-message TCP IXFR, as described by RFC 8945. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	6.5	More Details
CVE-2026-92417	A vulnerability was found in Open5GS up to 2.8.0. This affects the function ogs_pfcp_parse_volume_measurement in the library lib/pfcp/types.c of the component PFCP Handler. The manipulation results in null pointer dereference. The attack may be launched remotely. The patch is identified as 8f07b507b78ff94776f2cd49276eb116ed93d7f2. A patch should be applied to remediate this issue.	6.5	More Details
CVE-2026-87116	Tanium addressed a server-side request forgery vulnerability in Threat Response.	6.5	More Details
CVE-2026-54648	CubeCart is an ecommerce software solution. Prior to 6.7.5, the GDPR tools in admin/sources/customers.gdpr.inc.php rely on page-level CC_PERM_READ access and do not require CC_PERM_DELETE for the purge, no_order_purge, or delete_guests commands. An authenticated administrator with read-only customer privileges can invoke these backend actions directly to bypass interface restrictions and irreversibly delete customer records, accounts without orders, or guest accounts, compromising data integrity and availability. This issue is fixed in version 6.7.5.	6.5	More Details
CVE-2026-92775	Wiki.js through 2.5.314 contains a server-side request forgery vulnerability in the Image Prefetch renderer that fetches arbitrary URLs without protocol, host, or address validation. Attackers with page editing permissions can inject img elements with the prefetch-candidate class to make the server request internal services and cloud metadata endpoints, with responses returned	6.5	More Details

	to the attacker.		
CVE-2026-92771	Twenty before 2.35.0 fails to validate field and row permissions in the groupBy-with-records GraphQL resolver, allowing authenticated users to bypass permission checks. Attackers with canReadObjectRecords permission but canReadFieldValue false can retrieve restricted field values through the groupBy resolver that would normally be denied.	6.5	More Details
CVE-2026-92468	zlt2000 microservices-platform through 6.0.0 contains an authorization bypass vulnerability in the search-center service that allows authenticated attackers to read any Elasticsearch index by specifying the index name in POST /search/{indexName} and GET /agg/requestStat/{indexName}/{routing} path variables. Attackers can query arbitrary indices including sys_user to retrieve sensitive user records and password hashes without proper access controls.	6.5	More Details
CVE-2026-73462	On affected platforms running Arista EOS with IGMP (Internet Group Management Protocol) snooping configured (enabled by default on all VLANs), a network-adjacent unauthenticated attacker can send malformed network packets on an affected VLAN to cause the IGMP snooping agent to terminate unexpectedly. This results in a temporary disruption of multicast traffic management, which may cause multicast traffic to be flooded to all ports of the affected VLAN until the service recovers. Repeated exploitation could result in a prolonged loss of intended multicast forwarding behavior.	6.5	More Details
CVE-2026-92765	ArcherySec through 2.0.6 fails to validate organization ownership in the WebScanVulnList endpoint, allowing authenticated users to read vulnerability findings from other organizations. Attackers can supply arbitrary scan identifiers to retrieve complete web vulnerability data including titles, severities, statuses, and analyst notes from other tenants.	6.5	More Details
CVE-2026-92770	Harbor through 2.15.2 fails to properly restrict the q query parameter filtering on scanner registration access credentials. Project administrators can exploit fuzzy filtering on the AccessCredential column to recover the scanner adapter secret one character at a time through response row counts.	6.5	More Details
CVE-2026-20287	As part of Cisco's ongoing commitment to proactive security and product quality, the Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) engineering teams have conducted a comprehensive internal security review. This review resulted in a software hardening release that addresses multiple internally discovered vulnerabilities. The vulnerabilities tracked by CVE-2026-20287 are related to improper privilege managment issues that are grouped under the Common Weakness Enumeration (CWE) Pillar CWE-269.	6.5	More Details
CVE-2026-73436	On affected platforms running Arista EOS with OSPFv2 and OSPFv2 segment routing configured, a specially crafted OSPFv2 packet from an adjacent OSPF neighbor may cause OSPF to restart unexpectedly.	6.5	More Details
CVE-2026-92760	Shlink through 5.1.6 fails to enforce API key role restrictions when issuing Mercure subscription tokens, allowing restricted keys to subscribe to all topics. Attackers with author-only or domain-only keys can access the mercure-info endpoint to receive visit data including referrer, user agent, geolocation, and full short URL objects for URLs outside their authorization boundary.	6.5	More Details
CVE-2026-92759	SecObserve versions before 1.59.1 contain an information disclosure vulnerability in the ApiConfigurationSerializer that fails to strip the basic_auth_password field from API configuration responses. View-only product members can retrieve the decrypted basic-auth password of configured scanner or integration service accounts through standard REST endpoints.	6.5	More Details
CVE-2026-92139	Jenkins Bitbucket Push and Pull Request Plugin 4.0.1 and earlier trusts values provided in the webhook payload, including certain URLs, and uses configured Bitbucket credentials to connect to those URLs, allowing attackers to capture Bitbucket credentials stored in Jenkins by sending a crafted webhook payload.	6.5	More Details
CVE-2026-73468	A specially crafted packet can cause the premature expiry of multicast forwarding state on affected interfaces, potentially resulting in temporary multicast traffic loss during the affected period.	6.5	More Details
CVE-2026-84859	ScadaLTS 2.8.1-release-candidate build 0 is affected by an Authenticated Blind SQL Injection The /api/events/search endpoint accepts a JSON body containing a sortBy array. The values in this array are concatenated directly into the SQL ORDER BY clause without any sanitization or parameterization. This allows authenticated users with the ROLE_USER role to perform time-based and boolean-based blind SQL injection to extract arbitrary data from the database, including password hashes of all users. The endpoint is accessible to any authenticated user with ROLE_USER, ROLE_ADMIN, or ROLE_PUBLIC via POST /api/events/search as defined in spring-security.xml.	6.5	More Details
CVE-2026-52852	Traccar is an open source GPS tracking system. Prior to 6.14.0, an authenticated user with permission to manage groups and request reports can create a cyclic group-parent hierarchy and request a trips or stops report for a device in that hierarchy. org.traccar.api.resource.GroupResource permits the parent cycle, while org.traccar.helper.model.AttributeUtil.lookup follows group parents without cycle detection, a visited set, or a depth limit. The storage-backed lookup reached through TripsConfig. and ReportUtils.slowTripsAndStops never terminates, pins a Jetty worker at high CPU after the client disconnects, and can exhaust the web/API worker pool when requests are repeated. The position-ingestion cache-backed path is not part of the confirmed affected scope. This issue is fixed in 6.14.0.	6.5	More Details
CVE-2026-44235	rabbitmq-c is a C-language AMQP client library for RabbitMQ. Prior to 0.16.0, a malicious AMQP server can send an undersized HEADER or METHOD frame during client login and cause unsigned size_t underflow in amqp_handle_input() in librabbitmq/amqp_connection.c. The parser subtracts HEADER_SIZE, fixed per-frame fields, and FOOTER_SIZE from state->target_size without first checking the minimum frame length. The wrapped encoded.len value is passed through amqp_decode_properties() to amqp_decode_table_internal(), where it defeats bounds checks and causes an out-of-bounds read and process crash. An on-path attacker can also trigger the issue when AMQP traffic is not protected by TLS with certificate validation. The demonstrated impact is denial of service, with no reliable memory disclosure or code execution shown. This issue is fixed in version 0.16.0.	6.5	More Details
CVE-2026-77281	Caddy is an extensible server platform that uses TLS by default. In version 2.11.3 and earlier, three configuration-dependent weaknesses affect the handler and placeholder layer. In modules/caddyhttp/rewrite/rewrite.go, Rewrite.Rewrite() can pass attacker-controlled replacement bytes through buildQueryString for a second placeholder expansion when a rewrite URI ends with a literal question mark, allowing injected environment or request-variable placeholders to disclose data and, when the file provider is registered, allowing injected file placeholders to disclose readable files. The issue is fixed in version 2.11.4.	6.5	More Details
	A flaw was found in the first broker login flow of Keycloak. When a user confirms an account-linking request from a different		

CVE-2026-92358	browser, a temporary proof is created to validate the link. However, this proof is not properly cleared after the link is established or when the user later manually removes the link. An attacker who controls the external identity can exploit this leftover proof to silently re-establish the link and gain unauthorized access to the victims account without any further confirmation.	6.4	More Details
CVE-2026-86311	The Photo Gallery by 10Web – Mobile-Friendly Image Gallery plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the Shortcode Attributes in all versions up to, and including, 1.8.44 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with Author-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2026-75016	The Magazine Blocks plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the News Ticker block's clientId attribute in versions up to, and including, 1.8.6. This is due to insufficient input sanitization and output escaping in the NewsTicker::render() method, which concatenates the clientId block attribute into an HTML class attribute without applying esc_attr(). This makes it possible for authenticated attackers, with contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2024-11222	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 13.0 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed a developer user to perform actions in the context of another user's merge request commit due to a race condition issue in pipeline creation.	6.4	More Details
CVE-2026-12106	The Auto Upload Images plugin for WordPress is vulnerable to Limited Server-Side Request Forgery in all versions up to, and including, 3.3.2 via the downloadImage function. This makes it possible for authenticated attackers, with contributor-level access and above, to make web requests to arbitrary locations originating from the web application. The plugin uses wp_remote_get() rather than wp_safe_remote_get(), and the validate() method only rejects URLs whose host matches the site's own hostname, failing to block requests to private, loopback, or link-local addresses (e.g., 127.0.0.1, 10.0.0.0/8, 169.254.169.254). Attackers can trigger this by embedding a crafted <img> tag with a src attribute pointing to internal network hosts in post content and submitting it for processing.	6.4	More Details
CVE-2026-11996	The Advanced Popups plugin for WordPress is vulnerable to Stored Cross-Site Scripting via 'Notification Button Link' Field in all versions up to, and including, 1.2.3 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with author-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2026-84909	The Custom Twitter Feeds – A Tweets Widget or X Feed Widget plugin for WordPress is vulnerable to Stored Cross-Site Scripting via 'buttoncolor' Shortcode Attribute in all versions up to, and including, 2.8.0 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page. This vulnerability is exploitable on common installs where the admin has configured access tokens with zero or multiple legacy feeds, as this causes the ctf_statuses support_legacy_shortcode option to be set to boolean true by default, activating the unfiltered legacy shortcode attribute code path.	6.4	More Details
CVE-2026-2585	The Brizy – Page Builder plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the ‘rootAttributes’ parameter in all versions up to, and including, 2.8.14 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with Contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2026-5920	The Bold Page Builder plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the 'shortcode_content' parameter of the bt_bb_shortcode shortcode in all versions up to, and including, 5.9.6. This is due to a bypassable security filter (bt_bb_save_pre) that can be circumvented via null byte injection, combined with insufficient output sanitization of base64-decoded content in the bt_bb_raw_content shortcode handler. This makes it possible for authenticated attackers, with Contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2026-15650	The RT Mega Menu – Mega Menu Builder for Elementor & Gutenberg plugin for WordPress is vulnerable to Stored Cross-Site Scripting via 'pointer_menu_item' Block Attribute in all versions up to, and including, 1.5.2 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page. The payload bypasses wp_kses_post filtering applied to post content on save because it contains no HTML tags and survives intact inside the block comment's JSON attributes, reaching the walker unescaped at render time.	6.4	More Details
CVE-2026-14855	The RT Mega Menu plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the 'css[left]' parameter in all versions up to, and including, 1.5.1 due to insufficient input sanitization and output escaping. This makes it possible for authenticated attackers, with Subscriber-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	6.4	More Details
CVE-2026-81443	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Server-Side Request Forgery (SSRF) vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Server-side request forgery.	6.4	More Details
CVE-2026-66608	Contributor Server Side Request Forgery (SSRF) in Unlimited Elements For Elementor (Free Widgets, Addons, Templates) <= 2.0.19 versions.	6.4	More Details
CVE-2026-92526	A flaw has been found in itsourcecode Leave Management System 1.0. This affects an unknown function of the file /module/leave/index.php. Executing a manipulation of the argument ID can lead to sql injection. The attack may be launched remotely. The exploit has been published and may be used.	6.3	More Details
CVE-2026-92993	A vulnerability was detected in Dromara mayfly-go up to 1.11.5. The impacted element is the function RunMachineScript of the file server/internal/machine/api/machine_script.go of the component Machine Script Feature. The manipulation of the argument params results in os command injection. The attack can be executed remotely. The exploit is now public and may be used. Exploitation needs no admin account. Any account holding machine:script:run plus tag access reaches arbitrary command	6.3	More Details

	execution on machines whose templates contain {{.param}} placeholders; the SSH exec layer (Cli.Run) also applies no input filtering to any caller. The vendor was contacted early about this disclosure but did not respond in any way.		
CVE-2026-92781	Builder.io Gen2 SDKs through versions 5.2.11 and 0.25.13 contain a prototype pollution vulnerability in the unflatten helper that processes builder.userAttributes query parameters without prototype guards. Attackers can craft preview links with __proto__ or prototype segments to pollute Object.prototype in a visitor's browser when the SDK processes the malicious URL.	6.3	More Details
CVE-2026-92184	A security flaw has been discovered in ag-ui-protocol ag-ui 0.3.0. Affected is the function urllib.request.urlopen of the file integrations/aws-strands/python/src/ag_ui_strands/utils.py of the component Multimodal Content. The manipulation of the argument Value results in server-side request forgery. The attack can be executed remotely. The patch is identified as bf0c34df34cbb4b1992bc37c9bfffe6dd54bb189. It is advisable to implement a patch to correct this issue.	6.3	More Details
CVE-2026-92364	A vulnerability has been found in itsourcecode Leave Management System 1.0. Affected by this vulnerability is an unknown functionality of the file /module/employee/index.php. The manipulation of the argument ID leads to sql injection. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used.	6.3	More Details
CVE-2026-92217	A vulnerability was determined in a2ui-project a2ui up to 0.10.6. This affects the function processMessages of the file renderers/web_core/src/v0_9/processing/message-processor.ts of the component Message Parsing. This manipulation causes dynamically-determined object attributes. The attack can be initiated remotely. The project was informed of the problem early through an issue report but has not responded yet.	6.3	More Details
CVE-2026-92527	A vulnerability has been found in chatwoot up to 4.17.1. This impacts an unknown function of the file callbacks_controller.rb of the component Shopify OAuth. The manipulation leads to server-side request forgery. Remote exploitation of the attack is possible. The exploit has been disclosed to the public and may be used. The project was informed of the problem early through an issue report but has not responded yet.	6.3	More Details
CVE-2026-93314	A vulnerability was determined in Freedesktop Poppler 26.07.0. This affects the function FoFiTrueType::mapCodeToGID of the file fofi/FoFiTrueType.cc. Executing a manipulation of the argument segCnt can lead to integer overflow. The attack can be launched remotely. The exploit has been publicly disclosed and may be utilized. This patch is called ed2a5538cf0a8d3ff908191eda9b73f91a5f952a. It is advisable to implement a patch to correct this issue.	6.3	More Details
CVE-2026-93313	A vulnerability was found in Freedesktop Poppler 26.07.0. The impacted element is the function JBIG2Stream::readCodeTableSeg of the file poppler/JBIG2Stream.cc. Performing a manipulation results in integer overflow. The attack can be initiated remotely. The exploit has been made public and could be used. The patch is named eb87cf711563894649bd0c365baa479401dc6d51. To fix this issue, it is recommended to deploy a patch.	6.3	More Details
CVE-2026-93376	Out of bounds read in DataTransfer in Google Chrome prior to 153.0.8010.52 allowed a local attacker leveraging social engineering to read memory outside the sandbox via a local program. (Chromium security severity: Medium)	6.3	More Details
CVE-2026-93015	BlueKitchen BTstack through 1.8.2 fails to validate the peer-reported endpoint count against table bounds in A2DP stream endpoint discovery. A bonded peer can send an AVDTP DISCOVER response with more endpoints than the fixed table holds, causing out-of-bounds writes that corrupt adjacent static objects and crash the process or sever event delivery.	6.3	More Details
CVE-2026-92360	A weakness has been identified in ag-ui-protocol ag-ui 1.0. The impacted element is the function prepareRunAgentInput of the file agent/agent.ts of the component Event Application Layer. This manipulation of the argument TEXT_MESSAGE_START causes origin validation error. Remote exploitation of the attack is possible. The pull request to fix this issue awaits acceptance.	6.3	More Details
CVE-2026-92992	A security vulnerability has been detected in Dromara mayfly-go up to 1.11.5. The affected element is an unknown function of the file server/internal/ai/api/ai.go of the component AI Assistant. The manipulation leads to missing authorization. Remote exploitation of the attack is possible. The exploit has been disclosed publicly and may be used. The identifier of the patch is 74bcb926eb4f5f94e7681144d7bf2168a0ec7cde. Applying a patch is the recommended action to fix this issue. The whitelist bypass is one-token wide. Any compound command containing curl, wget or sed auto-runs without approval; approval is granted by the same session user (self-approval). This issue got fixed with a silent patch.	6.3	More Details
CVE-2026-92880	A weakness has been identified in vgmstream up to r2117. Impacted is the function vadpcm_read_coefs_be of the file src/coding/vadpcm_decoder.c of the component EA SCHl parser. This manipulation of the argument entry/entries causes out-of-bounds write. Remote exploitation of the attack is possible. Patch name: ae37662ad626254ddd96ad69ac263792d7a92024. It is suggested to install a patch to address this issue.	6.3	More Details
CVE-2026-92402	A security flaw has been discovered in ChangeWeDer crm up to c07bd4c97141521af6475034bc58523beed51bbd. This issue affects the function index of the file UserController.java of the component top.upstudy.crm.controller.UserController. The manipulation results in missing authorization. The attack can be launched remotely. This product does not use versioning. This is why information about affected and unaffected releases are unavailable. The project was informed of the problem early through an issue report but has not responded yet.	6.3	More Details
CVE-2026-61589	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, the WebSocket `handle_mount` and `ViewRuntime._build_request` rebuild an `HttpRequest` via `RequestFactory().get(...)` with no `HTTP_HOST`, so `request.get_host()` defaulted to `"testserver"` on the live path. Host/subdomain/domain `TenantResolver`s then misresolved the tenant — `None` on the live path while the HTTP path resolved correctly. With `STRICT_MODE=False` the tenant-scoped managers returned unscoped rows (cross-tenant disclosure); with the default they returned an empty queryset (broken tenancy). This is fixed in djust 1.0.7. The handshake Host is extracted from the ASGI scope, validated against `ALLOWED_HOSTS` (the same logic as the CSWSH Origin gate, parsed with Django's `split_domain_port` so malformed Hosts are rejected at the boundary), and propagated — with the TLS scheme — into the reconstructed request, so live-path tenant resolution matches HTTP exactly. There is no known workaround on the live path short of upgrading. Users are most exposed when combined with `STRICT_MODE=False`.	6.3	More Details
CVE-2026-87113	Tanium addressed an improper access controls vulnerability in Threat Response.	6.3	More Details
	Verizon Cloud for Android (com.vcast.mediamanager) before 26.7.10 contains a path traversal vulnerability that allows co-		

CVE-2026-89038	resident malicious applications to write attacker-controlled bytes outside the intended staging directory by supplying a crafted _display_name value containing path-traversal sequences through exported activities OneTouchUploadActivity and PrintShopCloudActivity. Attackers can exploit the unsanitized filename concatenation in the file-staging sink via ACTION_SEND or ACTION_SEND_MULTIPLE intents to achieve arbitrary file write and inject attacker-controlled content into the authenticated user's Verizon Cloud account without user interaction.	6.2	More Details
CVE-2026-73638	Imager versions from 0.45_02 before 1.035 for Perl read outside the EXIF block via unchecked start offsets in tiff_load_ifd. tiff_load_ifd() validates an IFD entry's data by checking that `entry->offset + entry->size` stays within the EXIF block, and never checks the start offset itself. Where that sum is not the real end of the data, the check passes with the entry starting outside the block. Through 1.032 `entry->offset` is a plain int, so on the usual two's-complement implementations an offset with the high bit set converts to negative and the sum can land back inside the block. From 1.033 the field is a size_t and the addition wraps only where size_t is 32 bits. The IFD's own start offset is checked the same way and wraps where unsigned long is 32 bits, which includes 64-bit Windows. Any caller of Imager->read() on an attacker-supplied image may receive EXIF tags holding bytes from outside the block, or crash the process.	6.2	More Details
CVE-2026-92973	ansi2html versions 1.7.0a0 through 1.9.3 contain a cross-site scripting vulnerability in OSC 8 hyperlink handling that fails to validate or escape URL targets. Attackers controlling ANSI text input can inject javascript: schemes or terminate href attributes to execute arbitrary scripts in the context of pages displaying converted output.	6.1	More Details
CVE-2026-88593	kkFileView 5.0.0 through 5.0.2 allows reflected XSS via the /onlinePreview endpoint. The OnlinePreviewController passes the user-controlled page and kkagent request parameters to FreeMarker templates without sanitization, and the templates insert these values into raw JavaScript contexts.	6.1	More Details
CVE-2026-85386	Concrete CMS before 9.5.4 did not sanitize XML and XSLT documents uploaded through a public Form Block file-upload question. Plain XML uploads were validated by file extension only and stored as publicly accessible files that were served inline from the application's own origin. An unauthenticated visitor could therefore store an XML document containing an xml-stylesheet processing instruction that referenced an attacker-supplied, same-origin XSLT stylesheet. When a victim opened the stored file directly in a browser, the browser fetched the stylesheet, transformed the document into HTML, and executed attacker-controlled JavaScript in the Concrete CMS origin (stored cross-site scripting). If the victim was an authenticated administrator, the script could act with that administrator's session, and the reporter demonstrated creation of a new user in the Administrators group. The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 7.3 with vector CVSS:4.0/AV:N/AC:L/AT:P/PR:L/UI:A/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N. Thanks Valentin SARRE (Independent security researcher) for reporting.	6.1	More Details
CVE-2026-59944	Composer is a dependency Manager for the PHP language. From 1.0 until 2.2.30 and from 2.3.0 until 2.10.3, a malicious or compromised dependency can bypass the earlier CVE-2026-59946 binary-path hardening because Composer validates literal parent-directory segments only during dependency resolution, while the symlink and installed-metadata paths described by the advisory skip that validation. A package can ship an in-package binary symlink that resolves outside its installation directory, or attacker-influenced vendor/composer/installed.json metadata can provide an escaping binary path during a reinstall or regeneration of missing vendor/bin entries. The installed-metadata path is reachable only when the vendor directory was not populated by the same validated install run, such as when it is restored from an untrusted cache, copied from an earlier build stage, carried over from an older Composer run, or writable by a lower-trust build step. Composer can follow the path, change the external target's permissions to make it world-readable and executable, and create a runnable vendor/bin proxy to that external file. The issue does not directly read or transmit data and does not by itself provide remote code execution. This issue is fixed in versions 2.2.30 and 2.10.3.	6.1	More Details
CVE-2026-89330	The EmbedPress – PDF Embedder, 3D PDF FlipBook, Google Reviews, YouTube Videos, Upload & Embed PDF documents plugin for WordPress is vulnerable to Reflected Cross-Site Scripting via the 'unique' parameter in all versions up to, and including, 4.6.5 due to insufficient input sanitization and output escaping. This makes it possible for unauthenticated attackers to inject arbitrary web scripts in pages that execute if they can successfully trick a user into performing an action such as clicking on a link. This is a regression: the esc_url() wrapper that remediated the equivalent CVE-2023-5749 in version 3.9.2 was removed in version 4.3.0 during a refactor that introduced the 'unique' parameter.	6.1	More Details
CVE-2026-88976	Plate is a rich-text editor with AI and shadcn/ui. Prior to 53.3.11, and in the discontinued 54.0.0-beta.0 through 54.0.0-beta.1 builds, Plate core HTML deserialization APIs parse supplied HTML strings in the active document. When an application passes untrusted or cross-user HTML to these APIs, certain HTML attributes can trigger browser behavior before the HTML is converted into editor nodes. This can allow attacker-controlled script to execute in the consuming application's origin when another user loads the deserialized content. This issue is fixed in version 53.3.11.	6.1	More Details
CVE-2026-73460	On affected platforms running Arista EOS with IS-IS graceful restart enabled, an unauthenticated attacker who can inject a malformed IS-IS LSP PDU packet can cause the IS-IS graceful restart procedure to terminate prematurely. This may result in traffic loss following a restart event.	6.1	More Details
CVE-2026-18555	The Better Messages – Chat Rooms, Group Chat, Private Messages & AI Chat Bots plugin for WordPress is vulnerable to Reflected Cross-Site Scripting via the 'icn' parameter in all versions up to, and including, 2.15.22 due to insufficient input sanitization and output escaping. This makes it possible for unauthenticated attackers to inject arbitrary web scripts in pages that execute if they can successfully trick a user into performing an action such as clicking on a link.	6.1	More Details
CVE-2026-55946	Improper neutralization of special elements used in a command ('command injection') in Microsoft Copilot allows an unauthorized attacker to disclose information over a network.	6.1	More Details
CVE-2026-54521	FairEmail is a fully featured, open source, privacy-friendly email app for Android. Prior to 1.2319, the ActivityAMP AMP message renderer in app/src/main/java/eu/faircode/email/ActivityAMP.java enables JavaScript in its WebView but incompletely sanitizes untrusted message HTML. For non-allowlisted hosts, script.removeAttr("src") leaves inline script elements in the document and does not reject event-handler attributes or javascript: URLs on other elements. A crafted AMP email can execute arbitrary JavaScript when a recipient opens the message and enables the AMP toggle. The script can read the message DOM, exfiltrate message data, and display phishing overlays within the message-body area. Exploitation requires the recipient to enable the AMP toggle, and practical exposure is reduced because AMP email is uncommon. This issue is fixed in version 1.2319.	6.1	More Details
	AVideo through 29.0 (current revision e01e41ecc) contains a stored cross-site scripting vulnerability. The unauthenticated view-		

CVE-2026-92584	counter endpoint objects/videoAddViewCount.json.php reaches VideoStatistic::save(), which writes the caller's User-Agent (via getUserAgentInfo(), which returns unrecognized agent strings verbatim) directly into the `app` column of the videos_statistics table without invoking the sanitizing setter setApp(); normalizeApp() only truncates the value to 45 characters. The video owner's and administrator's statistics page (view/videoViewsInfo.php) renders this field in a DataTables 1.12.1 column with an unescaped renderer, and DataTables assigns cell content using innerHTML, so attacker-supplied HTML is parsed and executed in the privileged user's authenticated session when the statistics page is viewed. No patched version was available at the time of the advisory.	6.1	More Details
CVE-2026-20309	A vulnerability in the web-based management interface of Cisco Identity Services Engine (ISE) could allow an unauthenticated, remote attacker to conduct a reflected cross-site scripting (XSS) attack against a user of the interface. This vulnerability exists because the web-based management interface does not properly validate user-supplied input. An attacker could exploit this vulnerability by persuading a user of the interface to click a crafted link. A successful exploit could allow the attacker to execute arbitrary script code in the context of the affected interface or access sensitive, browser-based information.	6.1	More Details
CVE-2021-3030	Cute Editor for ASP.NET 6.4 is vulnerable to reflected cross-site scripting caused by improper validation of the Theme GET parameter in colorpicker_more.aspx. A remote, unauthenticated attacker can craft a URL that, once opened by a victim in a browser session authenticated to a site running the vulnerable component, executes arbitrary JavaScript in the security context of that site.	6.1	More Details
CVE-2026-54644	CubeCart is an ecommerce software solution. Prior to 6.7.5, the _errorMessage method in classes/gui.class.php uses strip_tags to permit anchor elements in error, information, and warning messages while retaining unsafe href values and onclick event handlers. Attacker-controlled search or input data that reaches a GUI message can carry a javascript: URI or event handler through the filter, and viewing or interacting with the rendered anchor executes JavaScript in the victim's browser session, enabling session exposure or unauthorized application actions. This issue is fixed in version 6.7.5.	6.1	More Details
CVE-2026-75523	Steeltoe is an open source project that provides a collection of libraries that helps users build cloud-native applications. Prior to 4.3.0, the Steeltoe.Management.Endpoint /actuator/httpexchanges endpoint passes recorded request URIs through MaskedUri, which masks URI user information but does not inspect query strings. When Management:Endpoints:HttpExchanges:IncludeQueryString is enabled, the HttpExchangeRequest response can disclose OAuth tokens, password-reset tokens, signed-URL signatures, API keys, and other query-string secrets from prior traffic to a caller that can reach the explicitly exposed endpoint. The Steeltoe.Management.Endpoint.Actuators.HttpExchanges DEBUG logger also records these URIs, creating a second disclosure channel for users with log access. This issue is fixed in version 4.3.0.	5.9	More Details
CVE-2026-19662	An attacker may be able to cause a `named` resolver to abort. The attack requires inducing the victim resolver to send multiple queries for a DNSSEC-signed zone hosted by an authoritative server under the control of the attacker. If the auth responds with a particular sequence of crafted answers, and those answers arrive in a particular order with particular timing, the `named` resolver will encounter a use-after-free bug, and abort. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.9	More Details
CVE-2026-92595	Nodemailer (npm package `nodemailer`) versions 9.1.0 and earlier do not honor the `disableFileAccess` and `disableUrlAccess` sandbox options when message content is resolved through the public plugin API `MailMessage.resolveContent()` using the documented legacy three-argument signature `resolveContent(data, key, callback)`. Because `shared.resolveContent()` normalizes the missing `options` argument to an empty object, the message-level flags copied into `mail.data` by the MailMessage constructor are discarded, and `resolveContentValue()` skips both access-control checks, reaching `nmfetch(url)` or `fs.createReadStream(path)`. As a result, plugin or application code that resolves untrusted message content (html, text, attachment `path` or `href`) via this API can be induced to read arbitrary local files or issue outbound HTTP(S) requests (server-side request forgery), bypassing the sandbox the application enabled. The internal paths used by `transporter.sendMail()` (`resolveAll()`, `_convertDataImages()`, and the MIME streaming path) are not affected. Fixed in version 9.1.1.	5.9	More Details
CVE-2026-92591	Craft CMS 5.0.0 through 5.10.12 treats a database connection failure as meaning that Craft is not installed, which makes anonymous installer actions — including install/validate-site — reachable on an installed production site whenever PHP remains available but the configured MySQL endpoint does not. The action accepts a site name, serializes it through Site::getName(), and expands ${NAME} expressions using App::env(). An unauthenticated attacker who obtained a guest session cookie and matching CSRF token before the outage and whose session remains valid during it can submit a predictable variable name (for example Craft's conventional CRAFT_SECURITY_KEY) and receive its value, disclosing Craft secrets, process environment variables, $_SERVER entries, or PHP constants such as the security key, database credentials, or API keys. The issue requires an independently occurring database outage; the vulnerability itself provides no way to induce it. Fixed in 5.10.13.	5.9	More Details
CVE-2026-18120	Concrete CMS before 9.5.3 exposed a legacy Express entry search endpoint that returned entry result JSON without invoking the canViewExpressEntries() permission check applied by the normal dashboard and CSV Export flow. An unauthenticated visitor who knew or discovered an Express entity identifier could enumerate that entity's entry search results, disclosing attribute values intended to be restricted to privileged users. For Express entities that do not support entry-specific permissions (i.e., supportsEntrySpecificPermissions() returns false), per-entry permission filtering is additionally disabled via EntryList::ignorePermissions(). The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 6.3 with vector CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N. Thanks Daniel Powell for reporting.	5.9	More Details
CVE-2026-86107	The VeloCloud Edge and Gateway exhibit an out-of-bounds write vulnerability when processing tunneled IP fragments between authenticated overlay neighbors. This vulnerability impacts the VeloCloud VCMP tunnel protocol only. A successful exploit can cause the affected process to terminate and restart, leading to a temporary disruption of traffic. Hosts on the internet that are unauthenticated and unable to form an overlay peer relationship can not trigger the vulnerable logic.	5.9	More Details
CVE-2026-85720	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 2.0.0 until 2.16.1 and 3.0.12, a request using an HTTP proxy to reach an HTTPS origin can expose preemptive origin credentials because NettyRequestFactory and NettyRequestSender.sendRequestWithNewChannel attach Authorization to the plaintext CONNECT request before the TLS tunnel exists. Basic or Digest credentials and per-connection NTLM, Kerberos, or SPNEGO tokens intended for the origin are therefore visible to the proxy and to observers on the client-to-proxy hop. The tunneled request still receives origin Authorization after the tunnel is established, while Proxy-Authorization remains on CONNECT for its intended proxy recipient. This issue is fixed in versions 2.16.1 and 3.0.12.	5.9	More Details
CVE-	An inapplicable NSEC record may be accepted by a `named` resolver as proof that no wildcard exists, which could allow an		

2026-19941	attacker at the same or an upstream level of the zone name to mask the existence of a victim's wildcard record. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.9	More Details
CVE-2026-86861	pgAdmin 4's File Manager save_file endpoint, which backs saving from the Query Tool and ERD, validated the requested path with Filemanager.check_access_permission() and then opened the file for writing with a plain open() call. CVE-2026-7819 had previously hardened the separate file upload path by opening its target with O_NOFOLLOW, so that the kernel refuses to follow a symbolic link at the final path component, but save_file was left on an unprotected open(). A symbolic link already present when the check ran was rejected by the check itself; the remaining exposure was a link substituted at the final path component in the interval between the check and the write, at which point the write followed the link and landed outside the user's storage directory, creating or overwriting an arbitrary file as the operating-system account running pgAdmin. Exploitation requires the ability to create or replace a symbolic link inside the requesting user's storage directory. pgAdmin exposes no interface that creates symbolic links, so this requires filesystem access to the pgAdmin host or to a shared or network-mounted storage backend, together with winning the timing window. The reporter did not achieve a write against a released version, having attempted approximately 63,000 racing requests without success; the reported defect is the uncovered write path rather than a demonstrated sandbox escape. The fix routes save_file's write through the same O_NOFOLLOW-protected helper already used by the upload path, so both File Manager write sinks carry identical kernel-enforced protection, and reports a symbolic-link refusal explicitly instead of surfacing the underlying system error text. Note that files created by save_file are now given mode 0600, inheriting the behaviour introduced for uploads. The protection covers the final path component; substitution of an intermediate directory component is not addressed by this change. This issue affects pgAdmin 4: from the introduction of the containment check in the File Manager save path before 9.18.	5.9	More Details
CVE-2026-92091	A flaw was found in jwcrypto. The JWK.import_key() function validates the key_ops JWK member for duplicate values using an algorithm with O(n^2) time complexity, and the length of key_ops is not bounded. A remote, unauthenticated attacker can supply a JWK with a large key_ops array to an application that passes attacker-controlled key material to a public key-import API (reachable via ECDH-ES key agreement, OIDC dynamic client registration, DPoP, or ACME account key registration, among others) to consume excessive CPU time, resulting in a denial of service.	5.9	More Details
CVE-2026-85718	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 3.0.8 until 3.0.12, a client with maxConnections or maxConnectionsPerHost set above zero leaks one connection permit whenever TLS connection establishment fails before the handshake completes. NettyConnectListener removes the partitionKeyLock permit from NettyResponseFuture before every failure path is bound to the channel closeFuture, so an abort can leave the permit unreleased. Repeated failures can permanently lock out one host under a per-host limit or drain the shared pool under a global limit, blocking later requests even when no connection remains open. The default unlimited connection setting is not affected. This issue is fixed in version 3.0.12.	5.9	More Details
CVE-2026-92081	fastify is a fast and low-overhead web framework for Node.js. In versions before 5.12.5, when a route registers a response trailer via reply.trailer() and is served over HTTP/2, fastify unconditionally sets the Transfer-Encoding: chunked header, which is forbidden on HTTP/2, so Node.js throws while serializing the response headers. The exception is not caught and becomes an uncaughtException, so a single unauthenticated HTTP/2 request to any route that uses trailers crashes the server process and drops all in-flight requests, and it can be repeated on every restart. The issue is fixed in fastify 5.12.5, and users should upgrade to 5.12.5 or later. As a workaround, avoid registering response trailers with reply.trailer() on routes served over HTTP/2 until upgrading.	5.9	More Details
CVE-2026-82720	NLnet Labs Unbound 1.12.0 up to and including 1.26.0 has a use-after-free vulnerability when compiled for DNS-over-HTTPs support with '--with-libnghttp2'. During failure code paths (i.e., RPZ drop query, jostle due to heavy traffic), a dropped DoH stream brings down the whole DoH session and does not account properly for other DoH streams in the same session. This leads to use-after-free in those code paths. If the prerequisites are satisfied (possible RPZ drop or heavy client traffic), a malicious actor can trigger the vulnerability with a single DoH connection and the appropriate traffic. Impact is limited as the reads are not user controlled and the use-after-free leads to early returns. However, a hardened allocator can catch the use-after-free and controllably terminate the process resulting to denial of service.	5.9	More Details
CVE-2026-77119	A validly signed NSEC3 from an unrelated sibling zone may be accepted as an insecurity proof, downgrading a secure delegation and letting a forged unsigned answer through. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.9	More Details
CVE-2026-92936	vm2 versions 3.11.0 through 3.11.6 leak absolute host filesystem paths to sandboxed code through error stack formatting. Attacker-supplied code can force the host-realm source transformer to throw a SyntaxError (for example by calling eval with malformed source) and then read the error's .stack property; the bridge forwards the .stack read to the host-realm formatter, bypassing the sandbox-side host-path redaction introduced for GHSA-v27g-jcqj-v8rw. The returned stack string discloses absolute paths from vm2, Node.js internals, and the embedding application's own source tree, along with host function names. Default new VM() and new NodeVM() configurations are affected without any special options, and the issue persists when string eval is disabled because the host-side transformer throws before eval is handled. The impact is information disclosure only; no code execution results. Fixed in vm2 3.11.7.	5.8	More Details
CVE-2026-81479	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Partial String Comparison vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Denial of service.	5.8	More Details
CVE-2026-73469	When specific platforms are using Arista EOS with a loose Unicast Reverse Path Forwarding (uRPF) configuration, certain traffic may not be subjected to the intended verification drop. Consequently, traffic that should be dropped based on these routes could still be processed and forwarded by the device. This issue was discovered internally by Arista and the company is not aware of any malicious uses of this issue in customer networks.	5.8	More Details
CVE-2026-78301	A malformed zone may contain an NS or DNAME node above its origin, which `named` treats as a zone cut. If an attacker inserts a malformed zone into a BIND authoritative server (e.g., via zone transfer), queries for names inside the configured zone then lose authoritative status and return an out-of-zone delegation. On a server that also provides recursion BIND can follow this locally sourced cut and cache attacker-supplied data, affecting names outside the configured zone. This situation persists as long as the malformed zone remains in the zone database. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.8	More Details

CVE-2026-20120	A vulnerability in the access control list (ACL) Object Group Search (OGS) implementation of Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass configured access controls. This vulnerability is due to a logic error in populating group access control policies (ACPs) with OGS configured. An attacker could exploit this vulnerability by sending traffic that should be blocked through the device. A successful exploit could allow the attacker to bypass access controls and reach devices in protected networks.	5.8	More Details
CVE-2026-92933	vm2 is a sandbox for running untrusted Node.js code. In versions <= 3.11.7, NodeVM exposes the host `util` module to the sandbox as an unfiltered shallow copy (`Object.assign({}, util)` in `defaultBuiltinLoaderUtil`), and the deprecated `sys` builtin (an alias of host `util`) is exposed through the generic builtin loader. On Node.js >= 22.9 this hands sandboxed code `util.getCallSites()`, a programmatic stack-introspection API that returns the host process's full call stack, including absolute file paths, function names, and line numbers for vm2 bridge internals and the embedding application's entrypoint. This bypasses the host-frame redaction introduced for GHSA-v27g-jcqj-v8rw, which only applies to the `Error.prepareStackTrace` formatting channel. The issue is fixed in vm2 3.11.8.	5.8	More Details
CVE-2026-90984	The Generate PDF using Contact Form 7 WordPress plugin before 4.2.2 does not restrict the destination of the image fetch its PDF renderer performs on submitted form content, allowing unauthenticated users to make the server request internal resources and read the response back through the generated PDF.	5.8	More Details
CVE-2026-20290	A vulnerability in SSL/TLS certificate parsing in the Snort 2 Detection Engine of Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause the Snort 2 Detection Engine to restart. This vulnerability is due to incomplete validation of the SSL certificate. An attacker could exploit this vulnerability by sending a crafted SSL connection setup request to be parsed by Snort 2. A successful exploit could allow the attacker to cause the Snort 2 Detection Engine to restart unexpectedly, resulting in a denial of service (DoS) condition.	5.8	More Details
CVE-2026-50022	Metacat is data repository software that helps researchers preserve, share, and discover data. Prior to 3.4.2, MetacatSolrIndex.query forwards the client-controlled qt parameter through Apache SolrJ from search endpoints such as /d1/mn/v2/query/solr/ to its privileged Solr backend. An unauthenticated client can select the /admin/file handler, and SolrJ reformats the parameter into a request accepted even when handleSelect=false is configured on Solr 7.0 or later. When Solr returns the selected core configuration file, Metacat embeds the raw content in an XML processing error response, disclosing internal files such as solrconfig.xml and enabling infrastructure profiling. This issue is fixed in version 3.4.2.	5.8	More Details
CVE-2026-87828	The Seraphinite Accelerator WordPress plugin before 2.29.24 does not perform a capability check on one of its state-update AJAX actions, allowing authenticated users such as subscribers to write a malformed value that causes an uncaught error on every subsequent admin page load, making the entire admin area inaccessible to all administrators (denial of service).	5.7	More Details
CVE-2026-44940	The rancher-extension-stackstate extension in SUSE Observability exposes service tokens in plain configuration or insecure locations rather than managing them securely. An attacker with minimal access could obtain the token to gain unauthorized access or escalate privileges within the observability environment.	5.7	More Details
CVE-2026-92758	If logging mode is set to DEBUG or a malformed MongoDB connection string is used, application logs may collect sensitive information (if in use) such as passwords and AWS secure access keys.	5.5	More Details
CVE-2026-50291	OpenImageIO is a toolset for reading, writing, and manipulating image files of any image file format relevant to VFX / animation. Prior to versions 3.0.16.0 and 3.1.11.0, processing a crafted BMP file through oiiotool or an application linked to OpenImageIO can reach BMP palette handling in src/bmp.imageio/bmpinput.cpp with an empty color table. BmpInput::read_native_scanline then performs an invalid palette read while decoding an RLE-compressed scanline, causing a process crash and denial of service. This issue is fixed in versions 3.0.16.0 and 3.1.11.0.	5.5	More Details
CVE-2026-76781	A flaw was found in libxml2. A local user or an attacker providing a specially crafted XML catalog can trigger a NULL pointer dereference during XML catalog parsing. This occurs when a `nextCatalog` element lacks its mandatory `catalog` attribute, leading to the application crashing and causing a Denial of Service (DoS).	5.5	More Details
CVE-2026-76104	Dell ObjectScale, versions prior to 4.4.0.0, contains an Incorrect Permission Assignment for Critical Resource vulnerability in the OS. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Denial of service.	5.5	More Details
CVE-2026-92213	A vulnerability was detected in a2ui-project a2ui up to 0.10.6. This impacts the function z.any of the file renderers/web_core/src/v0_9/schema/server-to-client.ts of the component Angular Renderer. Performing a manipulation of the argument primaryColor results in injection. The attack is possible to be carried out remotely. The patch is named fb8e85aec78d04e81feb9992a57638ca1ec4dc1b. It is suggested to install a patch to address this issue.	5.5	More Details
CVE-2026-92757	Applications built on MongoDB Entity Framework Core Provider which place a database name in the connection string may inadvertently disable field level encryption.	5.5	More Details
CVE-2026-92756	Applications built on MongoDB Entity Framework Core Provider which combine independent encryption settings and this provider's encryption settings may silently lose TLS and schema-map settings leading to protected fields being stored unencrypted in the database.	5.5	More Details
CVE-2026-92132	Jenkins Gradle Plugin 2.19.1252.v15196b_5a_6e10 and earlier requests build scan data from the build scan link detected in the build log, even when a Develocity server URL is configured in the global configuration, allowing attackers able to control the build log to capture the Develocity access key configured in the global configuration by having Jenkins connect to an attacker-specified URL.	5.4	More Details
CVE-2026-92133	Jenkins GitLab Plugin 1.2149.vcfc32c82b_f7f and earlier caches the GitLab API client built for alternative GitLab API token credentials under a cache key derived from the credentials ID alone, omitting the folder in which the credentials are resolved, allowing attackers with Item/Configure permission to access GitLab API token credentials they are not entitled to use.	5.4	More Details
CVE-	Adenion Blog2Social plugin for WordPress before 9.1.0 allows low-privileged users to modify the scheduled post records of other users. The b2s_calendar_move_post AJAX handler in includes/Ajax/Post.php issues an UPDATE against the b2s_posts table using		More

2026-89031	only the attacker-supplied b2s_id primary key with no blog_user_id ownership constraint, allowing any user with the edit_posts capability to reschedule, suppress, or alter the publication state of any other user's scheduled social media post.	5.4	Details
CVE-2026-54643	CubeCart is an ecommerce software solution. Prior to 6.7.5, the delete-note handler in admin/sources/orders.index.inc.php verifies only the presence of order_id and delete-note parameters before deleting records from CubeCart_order_notes, without requiring CC_PERM_DELETE for orders. An authenticated administrator lacking order modification privileges can directly invoke the handler with valid identifiers and delete order-history notes, removing operational records and audit-trail data. This issue is fixed in version 6.7.5.	5.4	More Details
CVE-2026-73999	Contributor Insecure Direct Object References (IDOR) in Cooked <= 1.16.0 versions.	5.4	More Details
CVE-2026-92920	admin3 through 3.0.0 fails to invalidate existing sessions when disabling a user account, allowing attackers to retain authenticated access with original permissions. Attackers can continue using bearer tokens issued before account disablement to authenticate requests, as the AuthInterceptor never re-validates the user's locked status and session expiry resets on each request.	5.4	More Details
CVE-2026-93454	Aureus ERP through 1.6.0 stores the Payment Term note field unsanitized and renders it as raw HTML in the Accounting plugin. Authenticated users with payment-term create permission can submit arbitrary JavaScript to the payment-terms endpoint, which persists to the database and executes in browsers of all users viewing that Payment Term record.	5.4	More Details
CVE-2026-74005	Unauthenticated Cross Site Request Forgery (CSRF) in PublishPress Series <= 3.1.3 versions.	5.4	More Details
CVE-2026-93386	UI misrepresentation in WebAppInstalls in Google Chrome prior to 153.0.8010.52 allowed a remote attacker leveraging social engineering to spoof UI elements via a crafted HTML page. (Chromium security severity: Low)	5.4	More Details
CVE-2026-92568	MLRun through 1.11.0 contains a server-side request forgery vulnerability in the WebhookNotification handler that allows authenticated users to make the API server send arbitrary HTTP requests to internal addresses. Attackers can update a run with a malicious webhook notification that executes when the run reaches a terminal state, enabling requests to internal services, Kubernetes APIs, or cloud metadata endpoints from within the cluster.	5.4	More Details
CVE-2026-92778	CMAK through 3.0.0.6 fails to apply the scheduled leader election feature toggle to HTML form routes, allowing attackers to bypass the feature gate. Attackers can access the form endpoints to start and stop the recurring election scheduler, disrupting leadership across managed Kafka clusters.	5.4	More Details
CVE-2026-92579	In AVideo through 29.0, the autoCSRFGuard() function maintains a hardcoded allowlist of exempt basenames tested without directory context, allowing plugin files matching core filenames to inherit CSRF exemptions. The LoginWordPress plugin file login.json.php inherits an exemption and unconditionally logs out authenticated users on cross-site POST requests before validating credentials.	5.4	More Details
CVE-2026-80355	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Cross-Site Request Forgery (CSRF) vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Remote execution.	5.4	More Details
CVE-2026-54613	Vvveb is a powerful and easy to use CMS with page builder to build websites, blogs or ecommerce stores. Prior to 1.0.8.5, getThemeFolder() in admin/controller/editor/revisions.php returns the attacker-controlled theme parameter without sanitization, and backupFolder() concatenates it beneath DIR_THEMES before editor/revisions/load or editor/revisions/delete operates on a named .html file. sanitizeBackupFileName() strips traversal characters only from the separate file parameter and does not protect the theme directory component. An authenticated user with the default Editor role and editor/* permission can submit traversal sequences that redirect file_get_contents() or unlink() to a reachable backup subdirectory outside the web root. A valid admin session and CSRF token are required, the read is limited to .html files in backup directories, and deletion additionally requires filesystem write permission. This can disclose sensitive exported site content or remove backup data. This issue is fixed in version 1.0.8.5.	5.4	More Details
CVE-2026-84397	Adobe Experience Manager is affected by a stored Cross-Site Scripting (XSS) vulnerability that could be abused by a low-privileged attacker to inject malicious scripts into vulnerable form fields. Malicious JavaScript may be executed in a victim's browser when they browse to the page containing the vulnerable field. Scope is changed.	5.4	More Details
CVE-2026-13407	The Royal Elementor Addons WordPress plugin before 1.7.1067 does not properly sanitize and escape values submitted through its form widget before including them in the body of administrator notification emails, allowing unauthenticated attackers to inject arbitrary HTML into emails sent to the site administrator on form submission.	5.4	More Details
CVE-2026-92991	The Biggop Library is vulnerable to Cross-Site Scripting via the ‘display_id’ parameter from the Sigmative API in various versions due to insufficient output escaping. This makes it possible for attackers who can compromise the Sigmative API server to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page.	5.4	More Details
CVE-2026-14311	The Booking for Appointments and Events Calendar – Amelia plugin for WordPress is vulnerable to unauthorized access and modification of data due to a missing ownership verification on /users/customers/<id> endpoint in all versions up to, and including, 2.4.4. This makes it possible for authenticated attackers, with wpamelia-provider role, to view and modify arbitrary customers, including password reset. Takeover of WordPress user accounts, with the roles up to Editor, is also possible if that user had made an Amelia booking. This vulnerability affects only the Premium version of the plugin, where the Employee Panel is present.	5.4	More Details
CVE-2026-92590	Craft CMS versions from 5.7.0 before 5.10.13 contain a stored cross-site scripting vulnerability in the Generated Fields feature that disables Twig autoescaping and fails to encode cached values. Content editors can inject malicious JavaScript through editable fields that executes in authenticated Control Panel sessions of higher-privileged users viewing element indexes.	5.4	More Details
CVE-	The WPBot WordPress plugin before 8.7.6 does not perform a capability check on the AJAX action that saves its Claude AI		

2026-87959	provider settings, allowing users with subscriber-level access to overwrite those settings, including the API key used for the WPBot WordPress plugin before 8.7.6's outgoing AI requests.	5.4	More Details
CVE-2026-85999	Soup Sieve is a CSS selector library designed to be used with Beautiful Soup 4. Prior to 2.9, selector_iter in src/soupsieve/css_parser.py trims the raw selector with RE_WS_END, an end-anchored WSC whitespace-and-comment expression used with search(), so the regular expression engine retries a greedy scan at every starting offset. An attacker-controlled valid selector containing a long internal whitespace run, or a selector containing a long CSS comment run followed by another token, causes quadratic CPU work before tokenization. User-controlled selectors can reach the path through soupsieve.compile() and BeautifulSoup.select(), while applications using only hard-coded selectors are unaffected. This root cause is separate from the IDENTIFIER and VALUE backtracking vulnerability because the cost occurs in RE_WS_END.search during trimming rather than token matching. The resulting CPU consumption can hold the Python GIL, exhaust workers, and stall a service without causing memory corruption or code execution. The issue is fixed in version 2.9.	5.3	More Details
CVE-2026-81829	A flaw was found in SmallRye JWT's AwsAlbKeyResolver, which is used by applications to verify JSON Web Tokens signed by AWS Application Load Balancers. When the AWS_ALB key provider is configured, the resolver constructs the key-fetch URL by directly concatenating the attacker-controlled kid header value from an inbound JWT without sanitizing path traversal characters or query-string separators. This allows an unauthenticated remote attacker to force the application server to issue GET requests to arbitrary paths on the same origin as the configured key endpoint. As a result, non-public endpoints or internal data reachable on that origin may be read by the attacker before JWT signature verification takes place.	5.3	More Details
CVE-2026-86000	Soup Sieve is a CSS selector library designed to be used with Beautiful Soup 4. Prior to 2.9, the selector parser in src/soupsieve/css_parser.py defines IDENTIFIER with adjacent quantified groups over overlapping character classes, and VALUE embeds IDENTIFIER for attribute selectors. When an attacker-controlled selector contains a long identifier or unquoted attribute-value run followed by input that makes the overall match fail, the regular expression engine explores quadratically many splits between the overlapping groups. User-controlled selectors can reach this path through soupsieve.compile(), soupsieve.select(), or BeautifulSoup.select(), while applications using only hard-coded selectors are unaffected. The resulting CPU consumption can hold the Python GIL, exhaust application workers, and stall a service; successful plain identifier matches are linear, and the issue does not cause memory corruption or code execution. The issue is fixed in version 2.9.	5.3	More Details
CVE-2026-54918	NetBox Device Type Library is a collection of community-sourced device type definitions for import into NetBox. In the affected repository revisions, NETBOX_DT_LIBRARY_URL in tests/test_configuration.py is a free-form tracked constant that an unauthenticated pull-request author can change before the validation test harness runs. During pytest collection, tests/definitions_test.py passes the value to Repo.clone_from and create_remote("upstream").fetch(), causing blind Git smart-HTTP requests to an attacker-selected host or loading attacker-controlled tests/known-*.json validation caches. The blind request cannot set arbitrary metadata-service headers or return response bodies, and this path does not execute remote Git hooks, but substituted known data can bypass slug, module, and rack uniqueness validation. This vulnerability is fixed in commit 8980c690097e92f5028c7e6df402b327d827ecd5.	5.3	More Details
CVE-2026-85350	The UpsellWP WordPress plugin before 2.2.10 does not check that products added to the cart through a Frequently Bought Together campaign belong to that campaign, allowing unauthenticated users to buy arbitrary products at the campaign's discounted price.	5.3	More Details
CVE-2026-63461	Vendure is an open-source headless commerce platform. Prior to 3.6.5, the public Shop API products, collections, and facets queries combine mandatory visibility guards with caller-supplied filters using the caller-controlled filterOperator. When filterOperator is OR, a predicate matching a hidden entity can bypass the Product.enabled, Collection.isPrivate, or Facet.isPrivate guard. An unauthenticated caller can therefore retrieve disabled products and private collections or facets. This issue is fixed in version 3.6.5.	5.3	More Details
CVE-2026-90976	The Clean Login WordPress plugin before 1.19 does not check whether user registration is enabled before creating an account in its registration handler, allowing unauthenticated users to create accounts even when the site has registration disabled.	5.3	More Details
CVE-2026-93310	A vulnerability was identified in O-RAN-SC SMO OAM 2025-06-10. This affects an unknown part of the component VES Collector. The manipulation leads to allocation of resources. Remote exploitation of the attack is possible. The exploit is publicly available and might be used. The project was informed of the problem early through a bug report but has not responded yet.	5.3	More Details
CVE-2026-16582	The Booking for Appointments and Events Calendar – Amelia plugin for WordPress is vulnerable to unauthorized modification of data in all versions up to, and including, 2.4.5. This is due to the plugin accepting a client-supplied package-redemption identifier as proof of payment without validating it. This makes it possible for unauthenticated attackers to create approved appointment bookings without completing payment	5.3	More Details
CVE-2026-92927	A vulnerability was found in SourceCodester Drug Recommendation System 1.0. This issue affects some unknown processing of the file /db/drug_recommendor.sql. Performing a manipulation results in information disclosure. The attack is possible to be carried out remotely. The exploit has been made public and could be used.	5.3	More Details
CVE-2026-90977	The Clean Login WordPress plugin before 1.19 does not verify its registration CAPTCHA when the stored session value is empty, allowing unauthenticated users to bypass the anti-automation control on the registration form and create accounts without solving it.	5.3	More Details
CVE-2026-16750	The Motors – Car Dealership & Classified Listings Plugin plugin for WordPress is vulnerable to unauthorized access of data due to missing authorization checks in mvl_ajax_dealer_load_cars() function in all versions up to, and including, 1.4.120. This makes it possible for unauthenticated attackers to retrieve draft, pending, private, and future car listings belonging to arbitrary users.	5.3	More Details
CVE-2026-54907	Caddy Proxy Manager is a web interface for managing Caddy Server reverse proxies and certificates. Prior to 1.5.1, Caddy Proxy Manager enables email and password self-registration by default at /api/auth/sign-up/email, allowing an unauthenticated remote actor to create an active account with the user role without administrator approval. The user role cannot view or modify proxy data, so the direct impact is limited to unauthorized creation of low-privilege accounts. The fixed configuration in src/lib/config.ts and src/lib/auth-server.ts requires AUTH_ALLOW_SELF_REGISTRATION=true before the authentication library's disableSignUp control permits sign-up. This issue is fixed in version 1.5.1.	5.3	More Details
CVE-			

2026-71568	In BMCtest, Ironic is started without authentication and TLS for the duration of the test. Exploiting the problem requires winning the race with bmctest itself, which reduces the attack window and significantly increases its complexity.	5.3	More Details
CVE-2026-88798	The Really Simple Security WordPress plugin before 9.8.3 does not validate a client-supplied address value before using it as a storage key in one of its own options, allowing unauthenticated attackers to grow that option without bound and to slow the site's handling of missing pages.	5.3	More Details
CVE-2026-89278	The GPTranslate – Multilingual AI Translation Agent for WordPress: Translate Your Site with AI plugin for WordPress is vulnerable to Sensitive Information Exposure in all versions up to, and including, 2.34.6 via the enqueue_frontend_scripts. This makes it possible for unauthenticated attackers to extract the plaintext third-party AI provider API key (OpenAI, DeepL, xAI/Grok, Gemini, Claude, or Google Cloud Translation) — a credential granting billed account access — by fetching any public page and applying the inverse transformation bundled in the plugin's own public JavaScript asset. This exposure affects the default configuration (gpt-3.5-turbo in client mode) and all supported non-DeepSeek providers; only deepseek-* models and gpt-* models configured in server-proxy mode correctly suppress key emission.	5.3	More Details
CVE-2026-66676	Unauthenticated Broken Access Control in Easy Invoice <= 2.3.8 versions.	5.3	More Details
CVE-2026-74000	Contributor Broken Access Control in Simple Membership <= 4.8.2 versions.	5.3	More Details
CVE-2026-74002	Unauthenticated Broken Access Control in Booking Calendar <= 11.7 versions.	5.3	More Details
CVE-2026-78528	Unauthenticated Broken Access Control in BerqWP <= 4.1.15 versions.	5.3	More Details
CVE-2026-87966	The Easy Appointments WordPress plugin before 4.0.2.2 does not perform an ownership or authorization check on its unauthenticated appointment-reservation endpoint before updating an existing appointment identified by a request-supplied id, allowing unauthenticated attackers to overwrite, and through a follow-on cleanup delete, arbitrary appointments.	5.3	More Details
CVE-2026-8674	Initializing the DNS stub resolver from an /etc/resolv.conf file, or a LOCALDOMAIN environment variable, whose search list contains a domain of roughly 200 characters or more in the GNU C Library version 2.26 to 2.44 results in an assertion failure which aborts the process. The resolver truncates the search list when copying it into the fixed-size _res.defdname buffer, then asserts that the copy is consistent with the full configuration. The consistency check compared against the wrong size and did not handle a first entry that does not fit, so a correctly truncated list failed the assertion. Any process that resolves names through the library is affected, including long-running processes that reload /etc/resolv.conf on the next query after it changes. Search domains are commonly written to /etc/resolv.conf from data received over DHCP or from a VPN server, so an attacker on the local network may be able to trigger this without privileges on the target system, subject to validation by the network configuration software.	5.3	More Details
CVE-2026-92963	vm2 versions before 3.11.2 fail to properly restrict access to the VM2_INTERNAL_STATE_DO_NOT_USE_OR_PROGRAM_WILL_FAIL global variable. Attackers can access this internal state object through globalThis to retrieve sensitive sandbox internals.	5.3	More Details
CVE-2026-74017	Unauthenticated Broken Access Control in User Registration <= 5.2.7 versions.	5.3	More Details
CVE-2026-86796	The Hide My WP Ghost WordPress plugin before 7.0.11 does not verify that a request is a genuine WooCommerce request before disabling its firewall, threat-detection and login/URL-hiding protections, treating the mere presence of an attacker-suppliable request parameter as sufficient, which allows unauthenticated attackers to disable those protections and re-expose the concealed login and admin URLs on any request.	5.3	More Details
CVE-2026-86800	The Hide My WP Ghost WordPress plugin before 7.0.11 does not properly validate a loopback security-check request before disabling its login and URL hiding protection, dropping that protection precisely when the request's verification value is missing or incorrect, which any visitor can arrange, allowing unauthenticated attackers to re-expose the concealed WordPress login page location.	5.3	More Details
CVE-2026-85123	The Easy Form Builder by WhiteStudio WordPress plugin before 4.2.0 does not validate a submitted value against the stored configuration for some of its form types, allowing unauthenticated users to create WordPress accounts on a site whose owner has disabled registration.	5.3	More Details
CVE-2026-93395	A missing lower-bound validation in the bson_new_from_buffer() function of libbson allows an integer underflow when processing BSON data with a zero-length prefix. The function reads a 32-bit document length from the input buffer but does not verify that the value is at least 5 (the minimum valid BSON document size) before using it in an array index calculation. When the length field is zero, the expression used to check the document's null terminator wraps to UINT32_MAX, causing a heap out-of-bounds read that crashes the process. An unauthorized party who can supply crafted BSON input to an application using this API can cause a denial of service.	5.3	More Details
CVE-2026-91707	The The Divi theme for WordPress is vulnerable to arbitrary shortcode execution in all versions up to, and including, 5.11.1. This is due to the software allowing users to execute an action that does not properly validate a value before running do_shortcode. This makes it possible for unauthenticated attackers to execute arbitrary shortcodes. When the 'Force Enable D4 Shortcode Framework' option is enabled, this includes invoking the et_pb_contact_form shortcode to send email to an attacker-selected recipient with attacker-controlled content.	5.3	More Details
CVE-			More

2026-66575	Unauthenticated Insecure Direct Object References (IDOR) in King Addons for Elementor <= 51.1.81 versions.	5.3	Details
CVE-2026-87854	The Subscriptions for WooCommerce WordPress plugin before 2.0.3 does not correctly validate the shared secret protecting one of its REST endpoints, allowing unauthenticated users to retrieve the store's full list of subscriptions, including customer usernames, product names, recurring amounts and payment dates.	5.3	More Details
CVE-2026-76444	A vulnerability in an internal service of Cisco ISE and Cisco ISE-PIC could allow an unauthenticated, remote attacker to retrieve sensitive configuration information from an affected device. This vulnerability is due to missing authentication on the Policy Runtime Repository Table (PRRT) service. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to obtain sensitive configuration information from the affected device.	5.3	More Details
CVE-2026-76447	A vulnerability in the Online Certificate Status Protocol (OCSP) responder of Cisco ISE and Cisco ISE-PIC could allow an unauthenticated, remote attacker to cause an administrative reload of the OCSP responder certificate and key material. This vulnerability is due to missing authentication on a function of the OCSP responder. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to cause the OCSP responder to reload certificate and key material on demand.	5.3	More Details
CVE-2026-88910	The kboard WordPress plugin before 6.7 does not verify ownership or context before deleting board media, allowing unauthenticated attackers to permanently delete its uploaded media files and their database records by iterating identifiers.	5.3	More Details
CVE-2026-92803	LibreTranslate through 1.9.6 omits the access_check decorator from the download_file route, allowing unauthenticated access to translated files. Attackers can bypass API key requirements and abuse ban lists to download files without authentication on protected instances.	5.3	More Details
CVE-2026-87907	The Rox Appointment Booking WordPress plugin before 1.2.8 does not perform any authorization check on the endpoints that return booking service and category records, allowing unauthenticated attackers to read the private internal notes stored on each service and category.	5.3	More Details
CVE-2026-87896	The Rox Appointment Booking WordPress plugin before 1.2.8 does not perform any authorization check on the endpoint that returns booking agent (staff) records, allowing unauthenticated attackers to read staff email addresses, phone numbers, private internal notes and the linked WordPress account name for every agent.	5.3	More Details
CVE-2026-89064	The All-in-One WP Migration and Backup plugin for WordPress is vulnerable to Insufficient Credential Protection in versions up to, and including, 7.110. This is due to the `Ai1wm_Main_Controller::init()` method — registered on the `admin_init` hook, which fires unauthenticated on `admin-ajax.php` and `admin-post.php` requests — reading `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']` from any incoming request and writing them to the `ai1wm_auth_header` option via `update_option()` as a reversible base64-encoded string, with no capability check, nonce verification, `is_user_logged_in()` check, or confirmation that Basic authentication actually succeeded. This makes it possible for unauthenticated attackers to capture into the database, in reversible base64 form, any WordPress Application Password or HTTP Basic credential presented to `/wp-admin/` by a legitimate integration, or to overwrite the stored credential with an attacker-chosen value by sending an anonymous request carrying a crafted `Authorization: Basic` header. This is particularly impactful in environments using WordPress Application Passwords for REST API or third-party integrations, as those credentials are transmitted as HTTP Basic auth to `/wp-admin/` and will be silently harvested via this unauthenticated write path.	5.3	More Details
CVE-2026-90982	@fastify/static is a Fastify plugin that serves static files from a configured root directory. In versions before 10.1.4, on a case-insensitive filesystem such as Windows or the default macOS volume, a route guard or allowedPath restriction can be bypassed by altering the letter case of a path segment. The route matcher is case-sensitive while the filesystem is not, so a request that changes the case of a protected segment does not match the guarded route and falls through to the static handler, yet the filesystem resolves it to the same protected file. As a result, an unauthenticated request can read a file that a route guard or allowedPath was configured to protect. The issue does not affect case-sensitive filesystems and is not a directory traversal, since nothing is served from outside the configured root. The issue is fixed in @fastify/static 10.1.4, which validates the requested path against its actual on-disk spelling and rejects case-aliased paths before authorization. As a workaround, serve static files from a case-sensitive filesystem, or ensure route guards and allowedPath rules account for every letter-case variant of the protected paths.	5.3	More Details
CVE-2026-86823	The Newsletter WordPress plugin before 9.3.7 does not validate the destination of the redirect performed after a public subscription action, allowing unauthenticated attackers to redirect users to arbitrary external sites and to disclose a subscriber token that grants access to that subscriber record's front-end actions.	5.3	More Details
CVE-2026-86449	The LearnPress WordPress plugin before 4.4.7 does not check the user's capabilities before applying a user supplied post status filter in one of its REST routes, allowing unauthenticated attackers to list courses that are not published, including draft, pending, private, scheduled and trashed ones.	5.3	More Details
CVE-2026-90922	The Paid Membership Subscriptions WordPress plugin before 3.0.9 does not verify that the amount and currency reported by the payment provider match the pending payment before completing it, allowing unauthenticated users to obtain a paid membership by paying an arbitrary lower amount.	5.3	More Details
CVE-2026-86447	The LearnPress WordPress plugin before 4.4.7 does not check the user's capabilities in one of its administrative course tools, allowing unauthenticated attackers to list every enrolled student's display name and user identifier against the course they are enrolled on, and to recover their email addresses through the same handler's search filter.	5.3	More Details
CVE-2026-86445	The LearnPress WordPress plugin before 4.4.7 does not check the user's capabilities in one of its administrative template handlers, allowing unauthenticated attackers to retrieve the text, identifier and type of every published quiz question on the site, along with a keyword search over them, which is content the LearnPress WordPress plugin before 4.4.7 otherwise keeps non-public.	5.3	More Details
CVE-2026-91015	The Master Addons for Elementor WordPress plugin before 3.1.9 does not perform an authorization check on the AJAX action that deactivates its Popup Builder popups, relying only on a nonce that is publicly output to every visitor, allowing unauthenticated attackers to permanently disable any popup on the site.	5.3	More Details

CVE-2026-84906	The Eventin WordPress plugin before 4.1.24 does not verify that a completed payment corresponds to the order it is applied to, confirming only that the payment gateway reports the transaction as successful, not its amount, currency, or which order it belongs to, allowing unauthenticated visitors to mark unpaid orders of any value as paid by replaying the transaction of a single genuine low-value payment.	5.3	More Details
CVE-2026-86475	The Appointment Hour Booking WordPress plugin before 1.5.95 does not check every appointment in a booking submission against the capacity configured for its own slot, allowing unauthenticated visitors to take slots that are already fully booked.	5.3	More Details
CVE-2026-76439	A vulnerability in the endpoint posture status reporting functionality of the guest portal web application of Cisco ISE could allow an unauthenticated, remote attacker to submit forged posture status events into the endpoint posture pipeline. This vulnerability is due to insufficient authentication on an internal interface that is exposed through the guest portal. An attacker could exploit this vulnerability by sending a crafted request to an affected system. A successful exploit could allow the attacker to manipulate the posture status on the affected system.	5.3	More Details
CVE-2026-76433	A vulnerability in the client provisioning download feature of Cisco ISE and Cisco ISE-PIC could allow an unauthenticated, remote attacker to access protected files on an affected device. This vulnerability is due to insufficient validation of directory traversal character sequences in a user-supplied path when the software processes provisioning resource requests. An attacker could exploit this vulnerability by sending a crafted request to the provisioning download service. A successful exploit could allow the attacker to access protected files without authentication, potentially exposing sensitive information.	5.3	More Details
CVE-2026-73463	On affected platforms running Arista EOS, when multiple gRPC Network Security Interface (gNSI) transports are configured, a race condition in the gNSI Authz service may cause a policy rotation to fail silently. An authenticated user whose access was revoked by the new policy may retain unauthorized access to gRPC interfaces. This does not affect Bootz. This issue was discovered internally by Arista, and the company is not aware of any malicious exploitation of this vulnerability in customer networks.	5.3	More Details
CVE-2026-80225	In NLnetLabs Unbound up to and including 1.26.0, a degradation of service vulnerability is present in the TCP/DoT reading procedure where there is no limit on consecutive reads. A malicious actor that can stream and sustain a rate of distinct uncached names over the TCP/DoT connection, monopolizes a single worker's entire event loop for as long as its writes stay ahead of the drain.	5.3	More Details
CVE-2026-85501	Novel vulnerabilities to launch algorithmic complexity attacks on DNSSEC have been researched under the term 'ReTrap'. These result in degradation of service when malicious zones are used to serve the algorithmic complexity vulnerabilities. NLnet Labs Unbound up to and including 1.26.0 is vulnerable to some of them. TagTrap, where the triple(Zone, Algo, KeyTag) matching mechanism introduces a significant attack vector when resolvers handle malicious responses containing numerous mismatched DNSKEY, RRSIG, and DS record. DelegationTrap, where constructing the chain-of-trust requires iterative validation of DNSKEY and DS records from the root zone downward. For deeply nested domains, this results in significant computational overhead. NsecTrap, where responses with excessive invalid NSEC records compel the resolver to validate each one. AdditionalTrap, where Unbound by default would try to DNSSEC validate the ADDITIONAL section as well. This can be exploited to waste validation resources by malicious users.	5.3	More Details
CVE-2026-20121	A vulnerability in the access control list (ACL) Object Group Search (OGS) implementation of Cisco Secure Firewall Adaptive Security Appliance (ASA) Software and Cisco Secure Firewall Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass configured access controls. This vulnerability is due to a logic error in populating group access control policies (ACPs) with OGS configured. An attacker could exploit this vulnerability by sending traffic that should be blocked through the device. A successful exploit could allow the attacker to bypass access controls and reach devices in protected networks.	5.3	More Details
CVE-2026-73438	On affected platforms running Arista EOS with Open Shortest Path First version 3 (OSPFv3) configured, an unauthenticated attacker on the same OSPFv3 broadcast domain can send a specially crafted set of packets that can cause the Ospf3 agent to restart unexpectedly. The crash results in the loss of all OSPFv3 adjacencies on the affected device and may disrupt routing across the broader OSPF domain until the agent recovers. This issue was reported externally by Dravanet Inc., and Arista is not aware of any malicious exploitation of this vulnerability in customer networks.	5.3	More Details
CVE-2026-84439	When audit logging is enabled (zookeeper.audit.enable=true), an unauthenticated attacker can inject arbitrary fields into Apache ZooKeeper's audit log by sending a digest authentication request with tab characters (\t) embedded in the username. Because the audit log uses tab-separated key=value format, the injected tabs are parsed as legitimate field separators, allowing the attacker to spoof audit results (e.g., injecting result=success), forge operation types, and corrupt forensic evidence. A log injection vulnerability in Apache ZooKeeper allows a client that can call setACL to inject forged key-value fields into zookeeper_audit.log. When audit logging is enabled, the server serializes attacker-controlled digest ACL ids into the acl= audit field without escaping tab characters. Because audit events are emitted as tab-separated key=value records, a crafted ACL id can make one successful setAcl event appear to contain forged fields such as operation=delete and znode=/forged. This undermines the integrity of downstream audit parsing, alerting, and incident response. This issue affects Apache ZooKeeper: from 3.9.0 through 3.9.5, from 3.8.0 through 3.8.6. Users are recommended to upgrade to version 3.9.6 or 3.8.7, which fixes the issue.	5.3	More Details
CVE-2026-92475	A weakness has been identified in GPAC 26.08-DEV. This impacts the function wait_for_header_and_parse of the file src/utils/downloader.c. This manipulation of the argument Content-Range causes out-of-bounds read. The attack requires local access. The exploit has been made available to the public and could be used for attacks. Upgrading to version abi-16.26 will fix this issue. Patch name: c74a3065038ede35c1c7b75fa493a69ef6bcdb84. It is recommended to upgrade the affected component.	5.3	More Details
CVE-2026-81176	Svelte devalue is a JavaScript library that serializes values into strings when JSON.stringify isn't sufficient for the job. Prior to 5.9.2, devalue.parse does not reject out-of-bounds indices that are greater than or equal to values.length in src/parse.js. A specially crafted untrusted payload can make the parser alternate between array representations, producing quadratic work as the payload grows and causing denial of service in applications that parse untrusted devalue data. This issue is fixed in version 5.9.2.	5.3	More Details
CVE-2026-	Under certain circumstances on affected platforms running Arista EOS with gRPC Network Packet Sampling Interface (gNPSI)	5.3	More

73457	enabled, the gNPSI client credentials might be logged in clear text in local or remote accounting logs to authenticated users.		Details
CVE-2026-19607	A flaw was found in the first-broker-login flow of the keycloak-services component. This component handles the initial authentication and account linking when a user logs in via an external identity provider. The issue allows an attacker to register a matching username on an external provider to trigger a collision in Keycloak, which results in the legitimate user being locked out of their account.	5.3	More Details
CVE-2026-92565	Rallly before 4.15.0 contains an information disclosure vulnerability in the polls.get tRPC procedure that returns scheduled-event invitee names and email addresses to unauthenticated callers. Attackers can access a poll's urlId from public invite links to retrieve sensitive invitee information regardless of privacy settings.	5.3	More Details
CVE-2026-75029	In a query response, an attacker may send `named` multiple copies of a record that should only exist once (such as an SOA record). If the RDATA is the same on all the copies, the record is appended to the in-memory RDATA set, which can cause increased memory usage of the negative cache and possibly lead to other memory attack vectors. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.3	More Details
CVE-2026-61709	OpenFGA is an authorization and permission engine built for developers. Prior to 1.18.1, the ListUsers API could return a user that should have been excluded when an authorization relation used an intersection containing a base but not excluded operand, the base was granted through a type-bound public wildcard, and the excluded user also had a concrete tuple through another intersection operand. In pkg/server/commands/listusers/list_users_rpc.go, expandIntersection counted the concrete tuple and wildcard without first rejecting entries in excludedUsersMap. Applications that used ListUsers to enumerate or enforce access could therefore treat an excluded user as authorized. This issue is fixed in version 1.18.1.	5.3	More Details
CVE-2026-91016	The Motors WordPress plugin before 1.4.121 does not verify that a request is authorized to view a user's non-published listings before returning them, allowing unauthenticated attackers to read any author's draft, pending and private car listings - including titles, prices, media URLs and seller notes - by supplying only the target's numeric user id.	5.3	More Details
CVE-2026-84501	An unauthenticated attacker can inject arbitrary fake log lines into Apache ZooKeeper's operational log by sending a crafted add_auth("ensemble", ...) request containing newline characters (\n). When the ensemble name doesn't match, EnsembleAuthenticationProvider.handleAuthentication() logs the raw, unsanitized name via LOG.warn(). Because SLF4J's {} placeholder preserves embedded newlines, the attacker can forge complete log entries — with arbitrary timestamps, log levels, class names, and messages — that are visually indistinguishable from genuine ZooKeeper log output. This issue affects Apache ZooKeeper: from 3.9.0 through 3.9.5, from 3.8.0 through 3.8.6. Users are recommended to upgrade to version 3.8.7 or 3.9.6, which fixes the issue.	5.3	More Details
CVE-2026-78296	Insufficient Verification of Data Authenticity vulnerability in WP ManageNinja LLC FluentAuth allows Identity Spoofing. This issue affects FluentAuth: from n/a through 2.1.2.	5.3	More Details
CVE-2026-82125	The Schema & Structured Data for WP & AMP WordPress plugin before 1.66 does not correctly verify the ownership or the moderation status of a comment before returning its content, allowing unauthenticated users to read the content of comments still awaiting moderation or marked as spam.	5.3	More Details
CVE-2026-92220	A vulnerability was found in vllm-project vLLM 0.26.0/0.27.0. Affected is the function MoRIIOConnectorScheduler.request_finished/MoRIIOConnectorWorker.get_finished/MoRIIOWrapper._handle_release_message of the file vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py of the component MoRIIO Acknowledgement Handler. Performing a manipulation of the argument request_id/kv_transfer_params results in resource consumption. It is possible to initiate the attack remotely. The project was informed of the problem early through a pull request but has not reacted yet.	5.3	More Details
CVE-2026-11984	The Ad Inserter – Ad Manager & AdSense Ads plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 2.8.16 due to a missing capability check on the `ai-debug-code` URL-parameter. This makes it possible for unauthenticated attackers to view administrator-configured header and footer code blocks that have been disabled from public display.	5.3	More Details
CVE-2026-77702	The Eventin WordPress plugin before 4.1.24 does not prevent the token issued to a guest at checkout from being used to change that order's tickets afterwards, allowing unauthenticated users to replace a paid ticket with a free one and complete the order at no charge.	5.3	More Details
CVE-2026-78474	The Ni WooCommerce Sales Report WordPress plugin before 4.2.0 does not have any authentication or authorisation checks on one of its report-printing routines, allowing unauthenticated users to retrieve WooCommerce order details and customer contact information, to target an individual order, and to search the store's orders by customer name or email address.	5.3	More Details
CVE-2026-82124	The Schema & Structured Data for WP & AMP WordPress plugin before 1.66 does not check whether a post is password protected before including its content in the structured data it generates, allowing unauthenticated users to obtain the content of password protected posts via more than one public output route.	5.3	More Details
CVE-2026-19668	A BIND recursive resolver may experience excessive resource consumption if it encounters large numbers of a particular kind of invalid DNSSEC record. Default limits on "max-records-per-type" and "max-types-per-name" help mitigate the exposure. This issue affects BIND 9 versions 9.11.0 through 9.18.50, 9.20.0 through 9.20.27, 9.21.0 through 9.21.25, 9.11.3-S1 through 9.18.50-S1, and 9.20.9-S1 through 9.20.27-S1.	5.3	More Details
CVE-2026-92587	n8n is a workflow automation platform. In versions before 1.123.76, 2.37.7, and 2.38.2, the Git node validated a relative remote URL against the configured repositoryPath but then invoked git with that path as its working directory; git walked up to the enclosing repository's top level and resolved the same relative URL from there. An authenticated user (member) who nested the repository one level below the configured path could therefore make an identical URL string pass the file-access check while git resolved it outside the sandbox. A subsequent fetch or pull read a git repository outside N8N_RESTRICT_FILE_ACCESS_TO and merged its objects into the user's own repository, where their contents could be read back. The issue is fixed in n8n 1.123.76, 2.37.7, and 2.38.2, which resolve the remote reference from the directory git actually operates in before applying the sandbox check. As a workaround, the Git node can be disabled by adding n8n-nodes-base.git to NODES_EXCLUDE.	5.0	More Details

CVE-2026-54546	CloudTAK is a browser-based Common Operating Picture and situational awareness tool compatible with TAK. Prior to 13.22.1, the authenticated PUT /api/basemap endpoint passes an attacker-controlled URL through importBasemapURL() in api/routes/basemap.ts to fetch(url) without resolved-address classification or redirect revalidation. BasemapProtocol.isValidURL in api/lib/interface-basemap.ts checks only the HTTP or HTTPS scheme and is not applied on the vulnerable import path. Direct internal addresses, alternate IP encodings, and redirects to internal addresses can reach cloud metadata, loopback, private, and CGNAT HTTP services. The OptionalTileJSON response reflects fields including name, attribution, and tiles[0] to the caller, making the request forgery full-read rather than blind and enabling cloud credential theft and internal service disclosure. This issue is fixed in version 13.22.1.	5.0	More Details
CVE-2026-76448	A vulnerability in Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) could allow an authenticated, remote attacker to conduct an SQL or HQL injection attack on an affected device. This vulnerability is due to insufficient validation of user-supplied input to the affected APIs before it is used to build database queries. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to execute arbitrary SQL or HQL queries against the underlying database, which could allow the attacker to view or modify data that they are not authorized to access. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-20235	A vulnerability in the API of Cisco Identity Services Engine (ISE) could allow an authenticated, remote attacker to view sensitive information on an affected device. To exploit this vulnerability, the attacker must have valid administrative credentials. This vulnerability is due to insufficient validation of user-supplied parameters in API requests. An attacker could exploit this vulnerability by sending a crafted API request to an affected device. A successful exploit could allow the attacker to gain access to sensitive information, including hashed credentials that could be used in future attacks.	4.9	More Details
CVE-2026-20282	A vulnerability in Cisco ISE could allow an authenticated, remote attacker to obtain write access on the underlying operating system of an affected device. This vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to obtain write access to the underlying operating system. To exploit this vulnerability, the attacker must have valid administrative credentials. Note: For CVE-2026-20282, Cisco has assigned a Security Impact Rating (SIR) of High rather than Medium as the score indicates. The reason is that it is easy to get to root from the achieved privilege level.	4.9	More Details
CVE-2026-92813	Metabase through 0.63.18 fails to properly validate the unspecified address 0.0.0.0 in custom GeoJSON URLs, allowing unauthenticated attackers to reach loopback services. Attackers can save a malicious GeoJSON entry with 0.0.0.0 and trigger requests that return loopback service responses to unauthenticated callers.	4.9	More Details
CVE-2026-76451	A vulnerability in Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) could allow an authenticated, remote attacker to conduct an SQL or HQL injection attack on an affected device. This vulnerability is due to insufficient validation of user-supplied input to the affected APIs before it is used to build database queries. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to execute arbitrary SQL or HQL queries against the underlying database, which could allow the attacker to view or modify data that they are not authorized to access. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-76450	A vulnerability in Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) could allow an authenticated, remote attacker to conduct an SQL or HQL injection attack on an affected device. This vulnerability is due to insufficient validation of user-supplied input to the affected APIs before it is used to build database queries. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to execute arbitrary SQL or HQL queries against the underlying database, which could allow the attacker to view or modify data that they are not authorized to access. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-76449	A vulnerability in Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) could allow an authenticated, remote attacker to conduct an SQL or HQL injection attack on an affected device. This vulnerability is due to insufficient validation of user-supplied input to the affected APIs before it is used to build database queries. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to execute arbitrary SQL or HQL queries against the underlying database, which could allow the attacker to view or modify data that they are not authorized to access. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-76432	A vulnerability in the web-based management interface of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker with administrative-level privileges to write arbitrary files on an affected device. This vulnerability exists because the affected software does not properly validate directory traversal character sequences in a user-supplied file path during the upload process. An attacker could exploit this vulnerability by uploading a crafted file to the affected system. A successful exploit could allow the attacker to write files to an arbitrary location on the affected system.	4.9	More Details
CVE-2026-20072	A vulnerability in the web-based management interface of Cisco ISE could allow an authenticated, remote attacker to obtain sensitive information from network users that are outside the security group that the attacker is assigned to. This vulnerability exists because certain files lack proper authorization enforcement. An attacker with administrative privileges and management rights over network users could exploit this vulnerability by exporting the users. A successful exploit could allow the attacker to view passwords that are normally not visible to administrators.	4.9	More Details
CVE-2026-76426	A vulnerability in the REST API of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker to conduct SQL injection attacks against the monitoring database. This vulnerability is due to insufficient validation of specific parameters that are then concatenated into an SQL statement. An attacker could exploit this vulnerability by sending a crafted request that contains SQL statements in one of the affected parameters. A successful exploit could allow the attacker to read information from the monitoring database. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-76446	A vulnerability in an API of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker to read specific files on the underlying operating system of an affected device. This vulnerability is due to improper restriction of XML external entity references. An attacker could exploit this vulnerability by sending a crafted request to an affected device. A successful exploit could allow the attacker to read specific files on the affected system that the underlying process has permission to access.	4.9	More Details
CVE-	A vulnerability in the offline profiler feed service of Cisco ISE could allow an authenticated, remote attacker to read arbitrary files that are stored on an affected device. This vulnerability is due to the parsing of attacker-controlled feed metadata with an XML parser that does not disable external entity resolution. An attacker could exploit this vulnerability by uploading a crafted		More

2026-76427	offline feed package through the administrative interface. A successful exploit could allow the attacker to read arbitrary files from the file system and issue requests to internal systems from the affected device. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	Details
CVE-2026-76428	A vulnerability in the REST APIs of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker to conduct SQL injection attacks against the session database. This vulnerability is due to certain parameters being concatenated directly into SQL clauses without parameterization. An attacker could exploit this vulnerability by sending a crafted request that contains SQL statements in one of the affected parameters. A successful exploit could allow the attacker to read information from the session database. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.9	More Details
CVE-2026-76431	A vulnerability in the file management function of the web-based management interface of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker to delete arbitrary files and directories on an affected device. To exploit this vulnerability, the attacker must have valid administrative credentials. This vulnerability is due to improper validation of directory traversal character sequences in a user-supplied file path before the request is validated. An attacker could exploit this vulnerability by sending a crafted request to the web-based management interface of an affected device. A successful exploit could allow the attacker to delete arbitrary files and directories on the underlying operating system of the affected device.	4.9	More Details
CVE-2026-73445	On affected platforms running Arista EOS, an issue with the gRPC Network Security Interface (gNSI) Authz Rotate RPC may cause an incorrect Authz policy which was uploaded in the ongoing RPC stream to become active. This does not affect Bootz. This issue was discovered internally by Arista and the company is not aware of any malicious uses of this issue in customer networks.	4.9	More Details
CVE-2026-76434	A vulnerability in the certificate import functionality of the web-based management interface of Cisco ISE and Cisco ISE-PIC could allow an authenticated, remote attacker to read arbitrary files from the affected system. To exploit this vulnerability, the attacker must have valid administrative credentials. This vulnerability is due to insufficient validation of user-supplied input by the affected feature. An attacker could exploit this vulnerability by sending a crafted request to the web-based management interface of an affected device. A successful exploit could allow the attacker to read arbitrary files from the affected device, which could contain sensitive information.	4.9	More Details
CVE-2026-91019	The Event Booking Manager for WooCommerce WordPress plugin before 5.6.0 does not restrict who can view its stored payment gateway configuration, allowing users with Contributor-level access and above to read the site's PayPal and Stripe credentials, including their secret keys.	4.9	More Details
CVE-2026-92921	admin3 through 3.0.0 stores account passwords using single-round MD5 with only the username as salt and no key derivation function. Attackers with database access can recover plaintext passwords through offline dictionary or brute-force attacks due to negligible computational effort.	4.9	More Details
CVE-2026-54645	CubeCart is an ecommerce software solution. Prior to 6.7.5, admin/sources/products.index.inc.php reads the description, description_short, and spec_copy rich-text fields from $GLOBALS['RAW']['POST'] and removes only script elements before the values are stored and rendered through Smarty templates. An administrator with product-editing rights can store event-handler attributes, SVG content, or javascript: URIs that bypass this filter, causing persistent JavaScript execution when a storefront visitor or another administrator views the product content and enabling session exposure or unauthorized browser-context actions. This issue is fixed in version 6.7.5.	4.8	More Details
CVE-2026-86824	The Newsletter WordPress plugin before 9.3.8 does not generate its email tracking signing key with sufficient entropy and signs its tracking links with an unkeyed hash, allowing an unauthenticated attacker who recovers that key offline to forge tracking links, obtain any subscriber's session token, and read and modify that subscriber's stored personal data.	4.8	More Details
CVE-2026-87965	The Easy Appointments WordPress plugin before 4.0.2.2 does not use an unguessable token to authorize its mail-link appointment cancellation and confirmation action, deriving the token from a hardcoded source-embedded salt and the appointment's creation timestamp, so unauthenticated attackers who know or guess that timestamp can cancel or confirm arbitrary appointments.	4.8	More Details
CVE-2026-92298	EspoCRM through 10.0.8 uses PHP's rand() function to generate tokens for lead-capture opt-in, event invitation, and campaign URLs instead of a cryptographically secure generator. Remote unauthenticated attackers can guess these roughly 31-bit tokens to confirm opt-ins, accept or decline event invitations on behalf of other contacts, and access event details.	4.8	More Details
CVE-2024-38639	An improper authentication vulnerability has been reported to affect product. The remote attackers can then exploit the vulnerability to compromise the security of the system. QTS is not affected. We have already fixed the vulnerability in the following version:	4.8	More Details
CVE-2026-19857	The Formidable Forms WordPress plugin before 6.35 does not prevent a request-derived value from reaching the WordPress shortcode parser when it substitutes a supported token into a form's custom HTML, allowing unauthenticated visitors to have arbitrary shortcodes, with attacker-chosen attributes, executed server-side on any page displaying an affected form.	4.8	More Details
CVE-2026-85732	oras-go is a Go library for managing OCI artifacts. Prior to 2.6.2, the parseLink function in registry/remote/utils.go accepts an absolute URL from a registry-controlled Link response header without validating its scheme, host, or port. Tags, Referrers, and Repositories pagination operations then issue a GET request to the attacker-selected URL from the victim's network, allowing blind server-side request forgery against internal services. The response body is not returned to the attacker, but timing and error differences can reveal service reachability, and credentials may be attached when the credential store has an entry for the target host. Exploitation requires a victim to perform a pagination-based listing operation against a malicious registry. The maintainer identifies this report as a duplicate of GHSA-3hr5-mjrr-hfjh and states that remediation is consolidated in that earlier advisory. The consolidated issue is fixed in version 2.6.2.	4.7	More Details
CVE-2026-20350	A vulnerability in the web-based management interface of Cisco ThousandEyes Virtual Appliance could allow an authenticated, remote attacker to inject arbitrary operating system commands. This vulnerability is due to improper validation of user-supplied input to the web-based management interface. An attacker could exploit this vulnerability by saving configuration details that contain malicious values. A successful exploit could allow the attacker to execute arbitrary operating system commands with root privileges. To exploit this vulnerability, the attacker must have valid administrative credentials.	4.7	More Details

CVE-2026-54565	rhwp is an HWP viewer and editor implemented in Rust and WebAssembly. Prior to rhwp 0.7.15 and rhwp Chrome and Firefox extension 0.2.4, the browser extensions use an all-URLs host permission to detect HWP and HWPX links on visited pages, but the service workers do not validate message senders, URL schemes, or destination addresses before privileged fetches. The affected paths are rhwp-chrome/manifest.json, rhwp-chrome/content-script.js, rhwp-chrome/sw/message-router.js, rhwp-chrome/sw/thumbnail-extractor.js, rhwp-firefox/manifest.json, rhwp-firefox/content-script.js, rhwp-firefox/sw/message-router.js, and rhwp-firefox/sw/thumbnail-extractor.js. An untrusted page can make the fetch-file and extract-thumbnail handlers request localhost or private-network resources. When a target HWP or HWPX file contains an extractable PrvImage, the extension returns the preview as a data URI in page-readable DOM, allowing page script to read it. The flaw also permits internal-resource existence and port probing and extension presence or version fingerprinting. Exploitation requires a user to visit an untrusted page while the extension is enabled, and preview disclosure is limited to an extractable PrvImage. This issue is fixed in rhwp 0.7.15 and rhwp Chrome and Firefox extension 0.2.4.	4.7	More Details
CVE-2026-92247	A security vulnerability has been detected in synaptikcms synaptik-cms up to 1.3.4.4. This affects the function rename of the file admin/file-manager.php of the component Admin File Manager. The manipulation leads to unrestricted upload. The attack can be initiated remotely. The exploit has been disclosed publicly and may be used. Upgrading to version 1.3.5 is able to mitigate this issue. It is suggested to upgrade the affected component.	4.7	More Details
CVE-2026-92221	A vulnerability was determined in gedelumbung HospitalManagement up to c2d45543789a3887067d3915f69d44cfc2cf76a8. Affected by this vulnerability is the function generate_index_pasien of the file application/models/app_global_admin_model.php. Executing a manipulation of the argument cari can lead to sql injection. It is possible to launch the attack remotely. The exploit has been publicly disclosed and may be utilized. This product takes the approach of rolling releases to provide continious delivery. Therefore, version details for affected and updated releases are not available. The project was informed of the problem early through an issue report but has not responded yet.	4.7	More Details
CVE-2026-73443	On affected platforms running Arista EOS with VRRPv2 IP-AH authentication configured, an unauthenticated attacker within the same layer 2 network segment on which VRRP is running can capture a legitimate authenticated VRRP advertisement and replay it indefinitely. Replayed advertisements can be used to advertise stale VRRP state, for example to prevent a backup router from taking over the virtual gateway after the original master has gone down, resulting in a denial of service for hosts using the virtual gateway address.	4.7	More Details
CVE-2026-75025	Mattermost Desktop App versions <=6.2 6.2.2.0 Fixed an issue where Mattermost Desktop did not sufficiently restrict server-rendered content from accessing local or private network resources. Thanks to game0v3r for contributing to this improvement under the Mattermost responsible disclosure policy. Mattermost Advisory ID: MMSA-2026-00698	4.7	More Details
CVE-2026-19619	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 19.0 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an unauthenticated user to execute arbitrary JavaScript in the context of a targeted user's session due to improper sanitization of pasted HTML content in the Content Editor.	4.7	More Details
CVE-2025-56566	MikroTik firmware 7.19.4 stores sensitive authentication credentials and network state in cleartext within non-volatile storage. An attacker with physical access to the device can extract this material from an SPI flash dump, without authenticating to the device and without knowledge of the administrative password.	4.6	More Details
CVE-2026-86341	GitLab has remediated an issue in GitLab EE affecting all versions from 17.1 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that, under certain conditions, an authenticated user with Owner or Maintainer permissions could have silently disabled protected environment deployment approval requirements, allowing unapproved deployments to reach production, due to improper access control checks performed after the protected resource was modified.	4.4	More Details
CVE-2026-77955	In NLnet Labs Unbound 1.13.2 up to and including 1.26.1, a vulnerability in ZONEMD configured zones (zonemd-check: yes) which are located below (but not at) a trust anchor allow for an attack window where (tampered with) zone contents are served (or stored to disk) prior to the ZONEMD integrity check. This is caused by the needed DS/DNSKEY asynchronous resolution that needs to happen before the ZONEMD check completes. If a zonefile is written to disk (zonefile: option) while the ZONEMD check failed, the tampered data are reloaded on startup and available until ZONEMD verification concludes again. If verification fails, the data is not served any more but still persists on disk for future reloads.	4.4	More Details
CVE-2026-92588	n8n is a workflow automation platform. In n8n versions before 1.123.76, 2.37.7, and 2.38.2, the source control push endpoint derived the set of files to push from the file paths and status supplied in the client request payload instead of from the server-side status computed for the requesting user. An authenticated project-scoped user (e.g., a project admin) could therefore reference files belonging to projects they have no access to and push a deletion of those projects' workflows and credentials, resulting in cross-project data destruction. Exploitation requires the Source Control (Environments) enterprise feature to be licensed, enabled, and connected to a remote repository. The issue is fixed in 1.123.76, 2.37.7, and 2.38.2.	4.4	More Details
CVE-2026-63225	Redocly CLI makes OpenAPI validation, linting, and documentation workflows easier. Prior to @redocly/cli 2.33.2, the split command constructs output paths under --outDir from untrusted OpenAPI or AsyncAPI component names and x-codeSamples lang values without verifying that the resolved target remains inside the selected directory. A description containing literal ../ traversal segments in those fields can cause the split command to place or overwrite files outside --outDir. Component data remains constrained to YAML or JSON, and code-sample filenames remain based on the HTTP method, so this is not an unrestricted arbitrary-content file write. This issue is fixed in @redocly/cli version 2.33.2.	4.4	More Details
CVE-2025-36591	Dell ECS versions 3.8.1.0 through 3.8.1.7, and Dell ObjectScale versions prior to 4.4.0.0, contains an Use of a Broken or Risky Cryptographic Algorithm vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Information exposure.	4.4	More Details
CVE-2026-92383	A security vulnerability has been detected in PbootCMS up to 3.2.24. This vulnerability affects the function UserController::del/UserController::mod of the file apps/admin/controller/system/UserController.php of the component User Management. Such manipulation leads to cross-site request forgery. The attack may be performed from remote. The exploit has been disclosed publicly and may be used. Upgrading to version 3.2.25 is able to resolve this issue. The name of the patch is c25241a0964742cefb7f698efbb6c38b868d6ff7. It is advisable to upgrade the affected component.	4.3	More Details
	Apache NiFi 2.9.0 through 2.11.0 provide Connector configuration update and verification REST API methods that do not enforce authorization checking on Assets and Secrets referenced in proposed configuration. Updating or verifying a Connector configuration step can apply Asset and Secret references, but framework authorization was limited to write privileges on the		

CVE-2026-81866	Connector itself. As a result of the missing authorization, an authenticated user authorized to modify a Connector, but not authorized to read a referenced Parameter Provider, could apply Secret values backed by that Parameter Provider. The same methods also accepted Asset identifiers without verifying that the Asset belonged to the Connector being configured. Apache NiFi installations that do not implement different levels of authorization across Connectors and Parameter Providers are not subject to this vulnerability, because the framework enforces write permissions on the Connector as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which authorizes read access to referenced Parameter Providers and verifies Connector ownership of referenced Assets during configuration update and verification.	4.3	More Details
CVE-2026-93387	Improper state validation in Skia in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to obtain cross-origin data via a crafted HTML page. (Chromium security severity: High)	4.3	More Details
CVE-2026-92904	A flaw was found in the foreman_remote_execution plugin's template invocations controller. The show_template_invocation_by_host action resolves the job invocation by ID without evaluating the caller's view_job_invocations permission filter against the record. An authenticated user whose job invocation visibility is restricted by a permission filter can enumerate job invocation IDs and read the live output, rendered script, and input values for other users' job invocations within their own organizations.	4.3	More Details
CVE-2026-92569	Hippo4j through 1.5.0 contains a server-side request forgery vulnerability in four ThreadPoolController endpoints that fail to validate the clientAddress parameter. Authenticated attackers can supply arbitrary hostnames and ports to trigger outbound GET requests to internal networks and cloud metadata services.	4.3	More Details
CVE-2026-89029	Adenion Blog2Social plugin for WordPress before 9.1.0 allows low-privileged users to enumerate WordPress user accounts. The b2s_get_select_mandant_user AJAX handler in includes/Ajax/Get.php resolves arbitrary user IDs supplied in the owner parameter to display names without verifying that the caller is authorized to read user account data, allowing any user with the edit_posts capability to map WordPress user IDs to display names and confirm account existence for arbitrary IDs.	4.3	More Details
CVE-2026-54551	WireGuard Portal, or wg-portal, is a web-based configuration portal for WireGuard server management. From 2.2.0 until 2.3.0, the authenticated GET /api/v0/ws statistics WebSocket in internal/app/api/v0/handlers/endpoint_websocket.go subscribes to TopicPeerStatsUpdated and TopicInterfaceStatsUpdated and forwards every TrafficDelta event without per-user authorization in handleWebsocket(). A low-privilege user can enumerate peer public keys through EntityId and monitor BytesReceived and BytesTransmitted values for peers belonging to other users. The same connection exposes interface_stats and interface names that the REST API limits to administrators. Tunnel content, AllowedIPs, and user identities remain authorization-gated. This issue is fixed in version 2.3.0.	4.3	More Details
CVE-2026-92361	A security vulnerability has been detected in ag-ui-protocol ag-ui 1.0. This affects an unknown function of the file sdks/community/go/pkg/client/sse/client.go of the component SSE Client. Such manipulation leads to resource consumption. The attack can be executed remotely. The pull request to fix this issue awaits acceptance.	4.3	More Details
CVE-2026-93312	A flaw has been found in Freedesktop Poppler 26.07.0. Impacted is the function JBIG2Stream::rewind of the file poppler/JBIG2Stream.cc. This manipulation causes null pointer dereference. It is possible to initiate the attack remotely. The exploit has been published and may be used. Upgrading to version 26.08.0 is recommended to address this issue. Patch name: 5e49250f13b0390edeb3f90eb4c02c9941f97067. Upgrading the affected component is advised.	4.3	More Details
CVE-2026-85641	The Formidable Forms WordPress plugin before 6.35 does not restrict who can set the identifier recording which user last edited a form entry, and relies on that identifier when deciding whether to strip HTML from stored entry values, allowing unauthenticated visitors to have markup rendered in the admin entry view that would otherwise be removed, and to attribute their submission to an administrator who never made it.	4.3	More Details
CVE-2026-93311	A vulnerability was detected in Freedesktop Poppler 26.07.0. This issue affects the function SampledFunction::SampledFunction of the file poppler/Function.cc of the component SampledFunction. The manipulation of the argument BitsPerSample results in integer overflow. The attack may be performed from remote. The exploit is now public and may be used. The project was informed of the problem early through a bug report but has not responded yet.	4.3	More Details
CVE-2026-92357	A vulnerability was identified in a2ui-project a2ui 0.8/0.9/1.0. Impacted is an unknown function of the file model-processor.ts of the component Model Processor. The manipulation of the argument current[segment] leads to information disclosure. The attack may be initiated remotely. The identifier of the patch is 1b3bff234661ce922cbc3771be642b23ec9fd0fa. To fix this issue, it is recommended to deploy a patch.	4.3	More Details
CVE-2026-92893	A flaw was found in the foreman_ansible plugin's Ansible inventory API. The controller builds its host query using an unscoped Host.where call that does not enforce the search filter associated with the caller's view_hosts permission. An authenticated user whose host visibility is restricted by a permission filter can supply arbitrary host IDs within their organization and receive the full Ansible inventory for those hosts, including parameter values marked as hidden.	4.3	More Details
CVE-2026-93383	Information leak in Permissions in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	4.3	More Details
CVE-2026-92356	A vulnerability was determined in a2ui-project a2ui 0.9/0.9.1. This issue affects the function updateComponents of the file basic_functions.ts of the component Update Components. Executing a manipulation can lead to resource consumption. The attack can be launched remotely. The project was informed of the problem early through an issue report but has not responded yet.	4.3	More Details
CVE-2026-89138	The Filter Gallery plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 1.1.4. This is due to the plugin not properly verifying that a user is authorized to perform an action. This makes it possible for authenticated attackers, with subscriber-level access and above, to overwrite the title and content of arbitrary WordPress posts, write the _wp_attachment_image_alt meta key on arbitrary posts, and create or overwrite arbitrary ufg_gallery_* options.	4.3	More Details
CVE-2026-92461	yshop-crm through 2.1.3 contains a missing authorization vulnerability in the GET /admin-api/crm/flow/flow-users endpoint that allows any logged-in back-office user to access approval workflow data. Attackers can retrieve approval chain topology, step ordering, approver identifiers, and personal information including login names, nicknames, departments, email addresses, mobile numbers and last login IP addresses.	4.3	More Details

CVE-2026-92216	A vulnerability was found in a2ui-project a2ui up to 0.10.7. Affected by this issue is the function openUrl of the file renderers/web_core/src/v0_9/rendering/generic-binder.ts of the component Binder. The manipulation results in open redirect. It is possible to launch the attack remotely. The project was informed of the problem early through an issue report but has not responded yet.	4.3	More Details
CVE-2026-93307	A vulnerability has been found in O-RAN-SC SMO OAM 2025-06-10. Affected is an unknown function of the component VES Collector. Such manipulation of the argument additionalFields.padding leads to uncontrolled memory allocation. The attack can be launched remotely. The exploit has been disclosed to the public and may be used. The project was informed of the problem early through a bug report but has not responded yet.	4.3	More Details
CVE-2026-20285	A vulnerability in the web-based management interface of Cisco Identity Services Engine (ISE) and Cisco ISE Passive Identity Connector (ISE-PIC) could allow an authenticated, remote attacker to modify parts of the configuration on an affected device. This vulnerability is due to the lack of server-side validation of Administrator permissions. An attacker could exploit this vulnerability by submitting a crafted HTTP request to an affected system. A successful exploit could allow the attacker to modify descriptions of files on a specific page. To exploit this vulnerability, an attacker would need valid Administrator credentials.	4.3	More Details
CVE-2026-20286	A vulnerability in the web-based management interface of Cisco Identify Services Engine (ISE) could allow an authenticated, remote attacker to modify parts of the configuration on an affected device. This vulnerability is due to the lack of server-side validation of Administrator permissions. An attacker could exploit this vulnerability by submitting a crafted HTTP request to an affected system. A successful exploit could allow the attacker to modify descriptions of files on a specific page. To exploit this vulnerability, an attacker would need valid Administrator credentials.	4.3	More Details
CVE-2026-89030	Adenion Blog2Social plugin for WordPress before 9.1.0 exposes the email addresses of all registered WordPress users to low-privileged accounts. The b2s_search_user AJAX handler in includes/Ajax/Get.php invokes B2S_Tools::searchUser() in includes/Tools.php, which returns the email address of every matching user without restricting access to callers holding the list_users capability, allowing any user with the edit_posts capability to retrieve user email addresses including those of administrators.	4.3	More Details
CVE-2026-92413	A flaw has been found in Artifex MuPDF up to b6d17493700c621c0e70036980a6ebd06d2202c9. Affected by this vulnerability is the function pdf_open_filter of the file pdf-stream.c of the component PDF Xref Loading. Executing a manipulation can lead to null pointer dereference. The attack can be launched remotely. The exploit has been published and may be used. This patch is called 3df1e30f9d7b77260e13bd0dbe1928ddeba8386e. Applying a patch is advised to resolve this issue.	4.3	More Details
CVE-2026-92455	yshop-crm through 2.1.3 fails to enforce authorization on the sendSms and sendMail endpoints in CrmCustomerController, allowing any authenticated back-office user to send SMS and email to arbitrary customers. Attackers can invoke POST /admin-api/crm/customer/send-sms and POST /admin-api/crm/customer/send-mail with arbitrary customerIds, templateCode, and templateParams to deliver unauthorized messages through the organization's SMS and email channels.	4.3	More Details
CVE-2026-54495	The OpenFeature Operator allows users to expose feature flags to applications. In version 0.9.2 and earlier, a tenant who can create a controller-owned workload can use the openfeature.dev/featureflagsource annotation with NAMESPACE/NAME syntax to reference a FeatureFlagSource or InProcessConfiguration in another namespace. On multi-tenant clusters that use namespaces as trust boundaries, the cluster-scoped operator reads that resource and materializes spec.envVars literal values, spec.httpSyncBearerToken, sync URIs, and supporting ConfigMaps into the tenant's workload. Single-tenant clusters are not impacted, secretKeyRef and configMapKeyRef values remain namespace-local, and creating a FeatureFlagSource is not required.	4.3	More Details
CVE-2026-18441	The Appointment Booking Plugin – LatePoint | Calendar & Scheduling for WordPress plugin for WordPress is vulnerable to Insecure Direct Object Reference in all versions up to, and including, 5.6.9 via the set_customer_object due to missing validation on a user controlled key. This makes it possible for unauthenticated attackers to enumerate arbitrary customer records and disclose personally identifiable information - including first name, last name, email address, and phone number - by iterating the customer[id] parameter. This issue is exploitable only when the site is configured with customer authentication disabled (guest checkout enabled).	4.3	More Details
CVE-2026-93308	A vulnerability was found in O-RAN-SC SMO OAM 2025-06-10. Affected by this vulnerability is an unknown functionality of the component VES Collector. Performing a manipulation results in allocation of resources. The attack may be initiated remotely. The exploit has been made public and could be used. The project was informed of the problem early through a bug report but has not responded yet.	4.3	More Details
CVE-2026-93309	A vulnerability was determined in O-RAN-SC SMO OAM 2025-06-10. Affected by this issue is some unknown functionality of the component VES Collector. Executing a manipulation can lead to allocation of resources. The attack may be launched remotely. The exploit has been publicly disclosed and may be utilized. The project was informed of the problem early through a bug report but has not responded yet.	4.3	More Details
CVE-2026-92458	yshop-crm through 2.1.3 contains a missing authorization vulnerability in the StoreProductController onSale handler that allows authenticated back-office users to modify product sale status. Attackers can invoke the GET /admin-api/product/store-product/sale endpoint with sequential product IDs to withdraw entire product catalogs from sale or re-enable withdrawn products without proper permission checks.	4.3	More Details
CVE-2026-93379	Incorrect authorization in ORB in Google Chrome prior to 153.0.8010.52 allowed a remote attacker to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	4.3	More Details
CVE-2026-92416	A vulnerability has been found in Open5GS up to 2.8.0. Affected by this issue is the function smf_n4_handle_session_report_request of the file src/smf/n4-handler.c of the component PFCP Session Report Request Handler. The manipulation leads to reachable assertion. The attack may be initiated remotely. The identifier of the patch is e5f0c06d0f2d9613b003daa1cfa3ba8a4bd157e9. It is suggested to install a patch to address this issue.	4.3	More Details
CVE-2026-85572	The Tutor LMS WordPress plugin before 4.0.8 does not check that a user has access to a course before returning its lesson discussion content, allowing any authenticated user, such as a subscriber, to read comments from courses they are not enrolled in, including comments awaiting moderation.	4.3	More Details

CVE-2026-92581	In AVideo through 29.0, Like::__construct() performs counter arithmetic on raw request values before validation, allowing array-typed parameters to desynchronize stored votes from counters. Authenticated attackers can send array-typed like parameters followed by ordinary requests to drive video like counts arbitrarily negative, with the corruption persisting in the denormalized counter until manual repair.	4.3	More Details
CVE-2026-92879	A security flaw has been discovered in vgmstream up to r2117. This issue affects the function parse_mus of the file src/meta/mus_acm.c. The manipulation results in resource consumption. The attack may be launched remotely. The patch is identified as ae37662ad626254ddd96ad69ac263792d7a92024. Applying a patch is advised to resolve this issue.	4.3	More Details
CVE-2026-92894	A flaw was found in the foreman_ansible plugin's Ansible override values API. The destroy action resolves the target LookupValue record by ID without verifying it belongs to an AnsibleVariable the caller is authorized to edit. An authenticated user with the edit_ansible_variables permission can delete any LookupValue by ID, including override values for Ansible variables outside their permission filter scope and override values belonging to Puppet smart class parameters.	4.3	More Details
CVE-2026-92764	OpenCVE before 3.1.0 fails to properly scope the organizations API endpoint to the token's organization, instead returning the token creator's memberships. Attackers with organization-scoped tokens can list and retrieve every organization their creator belongs to, bypassing intended token isolation boundaries.	4.3	More Details
CVE-2026-92774	Wiki.js through 2.5.314 omits page tags from authorization checks in multiple GraphQL resolvers, allowing tag-based access restrictions to be bypassed. Attackers can query the list, tree, tags, searchTags, and links resolvers to retrieve restricted page metadata including titles, descriptions, paths, and tag information without proper authorization.	4.3	More Details
CVE-2026-78427	The NeuVector admission webhook silently excludes containers from policy evaluation when their image path matches one of three hardcoded service mesh sidecar images. Since the image path is entirely controlled by the workload author, any user capable of deploying workloads can evade admission deny rules simply by naming their image path after one of these sidecar images.	4.3	More Details
CVE-2026-92802	kan through 0.6.0 fails to properly validate board creation permissions in the GitHub project import endpoint, allowing guests to create boards despite lacking board:create permission. Attackers can bypass authorization checks by using the importProjects mutation to create boards while remaining blocked on direct creation paths.	4.3	More Details
CVE-2026-87829	The Checkout Field Manager (Checkout Manager) for WooCommerce WordPress plugin before 7.9.7 does not properly validate the ownership of an attachment before deleting it, allowing any authenticated user such as a customer to delete arbitrary media attachments belonging to other users.	4.3	More Details
CVE-2026-92809	PrestaShop psgdpr versions through 1.4.3 fail to validate that GDPR consent log entries are attributed to the authenticated customer. Authenticated attackers can submit arbitrary customer identifiers to create forged consent records for other customers, corrupting audit logs.	4.3	More Details
CVE-2026-92810	PrestaShop blockwishlist through 3.0.2 fails to validate wishlist ownership in the getUrlByIdWishListAction method, allowing authenticated customers to retrieve share tokens for any wishlist by identifier. Attackers can supply sequential wishlist identifiers to obtain valid share links and read other customers' private wishlist contents.	4.3	More Details
CVE-2026-92141	Jenkins Keycloak Authentication Plugin 2.4.1 and earlier does not restrict the redirect URL after login, allowing attackers to perform phishing attacks.	4.3	More Details
CVE-2026-91009	The Active Woot Products Tables for WooCommerce. 100% FREE WordPress plugin before 2.1.3 does not have authorisation and CSRF checks in some of its AJAX actions, allowing any authenticated users, such as subscriber, to change the title of arbitrary posts, pages and products.	4.3	More Details
CVE-2026-92585	AVideo through 29.0 (commit c3edcc274c389816d434acadac07ee78eaf330c1) fails to validate video access permissions in the API like endpoint, allowing logged-in users to vote on password-protected and group-restricted videos. Attackers can submit like and dislike requests to increment vote counters on videos they cannot watch by calling the set.json.php endpoint with APIName parameters.	4.3	More Details
CVE-2026-92363	A flaw has been found in ag-ui-protocol ag-ui 1.0. Affected is an unknown function of the file src/stream/sse_parser.cpp of the component JSON Parser. Executing a manipulation can lead to resource consumption. The attack may be performed from remote. This patch is called ab6e0bc298996caac2b4b0b3ec0bd8d32a15a186. Applying a patch is advised to resolve this issue.	4.3	More Details
CVE-2026-92586	AVideo through 29.0 (commit c3edcc274c389816d434acadac07ee78eaf330c1) fails to verify video access permissions in the set_api_comment function, allowing authenticated users to post comments on password-protected and group-restricted videos. Attackers can submit POST requests to the comment API endpoint with arbitrary video IDs to write comments on videos they cannot watch.	4.3	More Details
CVE-2026-92589	Craft CMS 5.0.0 through 5.10.12 (fixed in 5.10.13) contains a broken access control flaw in the nested-elements reorder endpoint. When an authenticated control panel user with viewEntries and viewPeerEntries (but without savePeerEntries) opens another author's entry in read-only mode, Craft unconditionally grants that session a `manageNestedElements::<ownerId>::field:<handle>` authorization flag for the entry's Matrix/Address fields. Unlike the corresponding delete endpoint, actions/nested-elements/reorder trusts this session flag alone and never rechecks the caller's save permission for the owner element. As a result, a view-only user can POST to actions/nested-elements/reorder using the ownerElementType, ownerId, ownerSiteId, attribute, elementIds, and offset parameters present in the read-only page source and rewrite the sort order of Matrix blocks or Addresses belonging to content they are explicitly denied save access to.	4.3	More Details
CVE-2026-87860	The Subscriptions for WooCommerce WordPress plugin before 2.0.3 does not verify the security token on the request that cancels a subscription, allowing attackers to make a logged-in customer cancel their own active subscription through a crafted request they are tricked into making.	4.3	More Details
CVE-2026-92365	A vulnerability was found in vllm-project vllm up to 0.29.0. Affected by this issue is some unknown functionality of the file vllm/v1/sample/thinking_budget_state.py. The manipulation results in inefficient algorithmic complexity. It is possible to launch the attack remotely. The pull request to fix this issue awaits acceptance.	4.3	More Details

CVE-2026-85349	The FluentBoards WordPress plugin before 2.0.15 does not properly verify authorization when returning the list of boards a user belongs to, allowing any authenticated user, including a Subscriber with no board access, to disclose the private board memberships of arbitrary users by referencing their user ID.	4.3	More Details
CVE-2026-92839	Canva Desktop before v1.125.0 performed double decoding in the deeplink handler. A threat actor could cause the application to load arbitrary same-origin content under the user's session.	4.3	More Details
CVE-2026-18317	The Foxtool All-in-One: Contact chat button, Custom login, Media optimize images plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 2.5.3. This is due to the plugin not properly verifying that a user is authorized to perform an action. This makes it possible for authenticated attackers, with subscriber-level access and above, to modify arbitrary subkeys of the foxtool_settings option, including enabling site-wide SVG uploads by toggling the media-up3 key, which can facilitate stored cross-site scripting via malicious SVG files.	4.3	More Details
CVE-2026-92754	PatrowlManager through 1.8.4 contains an improper access control vulnerability in the user listing API endpoint where the authorization decorator is commented out. Authenticated attackers with low-privilege accounts can enumerate all users and their privilege flags including superuser and staff status by accessing the endpoint.	4.3	More Details
CVE-2026-16794	GitLab has remediated an issue in GitLab EE affecting all versions from 18.11 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an authenticated user with the Security Manager role to execute arbitrary CI/CD jobs and access protected variables within group projects due to improper authorization controls on compliance framework management.	4.3	More Details
CVE-2026-8030	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 13.0 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an authenticated user to prevent another user from modifying their group settings due to improper validation of group URL slugs during namespace transfers.	4.3	More Details
CVE-2026-91010	The Invisible Anti-Spam & CAPTCHA — reCAPTCHA Alternative for All Forms WordPress plugin before 5.1.1 does not check the user's capabilities in its message deletion AJAX action, and only tests that a nonce parameter is present rather than validating it, allowing any authenticated user, such as a subscriber, to permanently delete every form submission the Invisible Anti-Spam & CAPTCHA — reCAPTCHA Alternative for All Forms WordPress plugin before 5.1.1 has stored.	4.3	More Details
CVE-2026-92881	A security vulnerability has been detected in vgmstream. The affected element is the function init_vgmstream_awb_memory of the file src/meta/awb.c of the component AWB parser. Such manipulation leads to divide by zero. The attack can be executed remotely. The name of the patch is ae37662ad626254ddd96ad69ac263792d7a92024. A patch should be applied to remediate this issue.	4.3	More Details
CVE-2026-93013	RAGFlow through 0.27.2 contains a path traversal vulnerability in the dev_insert_chunks_from_file and dev_insert_metadata_from_file endpoints that allows authenticated attackers to read arbitrary files by supplying absolute file paths in the file_path parameter. Attackers with valid access tokens can exploit missing path validation to read any file accessible to the service, with disclosure limited to files matching expected JSON structures that are then written to datasets.	4.3	More Details
CVE-2026-87831	The Checkout Field Manager (Checkout Manager) for WooCommerce WordPress plugin before 7.9.7 does not properly validate the ownership of an attachment before deleting it, allowing any authenticated user such as a customer to delete arbitrary media attachments belonging to other users.	4.3	More Details
CVE-2026-75017	The Magazine Blocks – Blog Designer, Magazine & Newspaper Website Builder, Page Builder with Posts Blocks, Post Grid plugin for WordPress is vulnerable to authorization bypass in all versions up to, and including, 1.8.6. This is due to the plugin not properly verifying that a user is authorized to perform an action. This makes it possible for authenticated attackers, with contributor-level access and above, to demote administrator-owned published builder templates (header, footer, front page, single, archive, 404, and search) to draft status and replace them with attacker-authored block content rendered site-wide, enabling defacement, phishing, and SEO spam. This is possible because the mzb-builder-template post type is registered with capability_type='post' and exposed via the REST API, and the _mzb_template meta key is accessible to any user with edit_posts capability, meaning Contributor-level users and above can trigger the vulnerable save_post() hook.	4.3	More Details
CVE-2026-7514	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 13.9 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that an authenticated user with developer-role permissions could substitute package file content and hide packages from their owners due to improper authorization checks in the Generic Package Registry.	4.3	More Details
CVE-2026-92138	The OAuth authorization endpoint in Jenkins Bitbucket Server Integration Plugin 6.0.1 and earlier reads the `oauth_callback` URL from the submitted form rather than from the server-side stored request token, allowing attackers to hijack the OAuth flow and obtain an access token on behalf of the victim.	4.2	More Details
CVE-2026-73440	On affected platforms running Arista EOS with Simple Network Management Protocol (SNMP) configured, SNMPv3 local or remote user credentials may be exposed as a one-way hashed, localized key value within the device's running and sanitized configurations. An authenticated user who gains access to this sensitive information could leverage it to perform unauthorized read operations on SNMP tables or to send fraudulent trap notifications to the Network Management System (NMS). This issue was discovered internally by Arista, and the company is not aware of any malicious exploitation of this vulnerability in customer networks.	4.2	More Details
CVE-2026-92131	Jenkins Pipeline: Groovy Libraries Plugin 805.va_fc79344957d and earlier does not restrict the library path provided to the library Pipeline step to a relative path inside the SCM checkout, and follows symbolic links to locations outside of the SCM checkout when retrieving the library, resulting in a path traversal vulnerability, allowing attackers able to configure Pipelines to read files in a resources directory and to delete files in a test directory on the Jenkins controller file system.	4.2	More Details
CVE-2026-92945	vm2 before 3.11.7 contains a module allowlist bypass vulnerability in isPathAllowedForModule that uses raw string prefix matching instead of boundary-anchored comparison. Attackers can reach non-allowlisted packages sharing a prefix with allowlisted modules by performing relative requires from allowlisted packages when transitive loading is disabled.	4.2	More Details
CVE-2026-92814	changedetection.io through 0.60.6 fails to escape the scraped page title in HTML notifications, allowing arbitrary markup injection. Attackers can place malicious markup in monitored page titles that reaches notification channels like email and Telegram as live content when the watch_title token is used in templates.	4.2	More Details

CVE-2026-19640	On affected platforms running Arista EOS, an authenticated user with access to the gNMI (gRPC Network Management Interface) may receive incorrect authorization results, potentially allowing access beyond their currently assigned permissions. This issue was discovered internally by Arista and the company is not aware of any malicious uses of this issue in customer networks.	4.2	More Details
CVE-2026-76559	The WP Import Export Lite WordPress plugin before 3.9.33 does not properly validate URLs before requesting them during the import process, allowing users with the import capability, which administrators hold by default, to make the site issue requests to internal hosts and services and read their responses. This is an incomplete fix for CVE-2026-11397.	4.1	More Details
CVE-2026-81441	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains a Missing Authentication for Critical Function vulnerability. An unauthenticated attacker with local access could potentially exploit this vulnerability, leading to Denial of service.	4.0	More Details
CVE-2026-92949	vm2 versions from 3.9.6 before 3.11.7 fail to properly restrict access to accessor properties on frozen objects, allowing sandboxed scripts to bypass vm.freeze() and vm.readonly() protections. Attackers can use Object.getOwnPropertyDescriptor() or __lookupSetter__() to extract and invoke host object setters directly, mutating properties the embedder explicitly marked read-only.	4.0	More Details
CVE-2026-87026	Tanium addressed an improper access controls vulnerability in Threat Response.	3.8	More Details
CVE-2026-89327	The FluentBoards WordPress plugin before 2.0.15 does not verify that a board member submitting a comment is the user the comment is attributed to, allowing any board member to post comments that appear to be authored by another user, including administrators.	3.8	More Details
CVE-2026-20071	A vulnerability in the SSID bring-your-own-device (BYOD) onboarding workflow of Cisco ISE could allow an unauthenticated, adjacent attacker to hijack the onboarding session of another user and access protected 802.1X networks. This vulnerability is due to insufficient authentication checks that are performed while a user is being onboarded. An attacker could exploit this vulnerability by spoofing the legitimate user and triggering a redirection to the guest web portal. A successful exploit could allow the attacker to take over the user session and gain access to the protected 802.1X network.	3.8	More Details
CVE-2026-81340	The MasterStudy LMS WordPress Plugin WordPress plugin before 3.7.50 does not perform per-object ownership or capability checks when updating orders through its REST API, allowing users with the Instructor role to modify any order on the site, granting free course enrolment, revoking other users' paid enrolments, and tampering with order notes.	3.8	More Details
CVE-2026-89328	The FluentBoards WordPress plugin before 2.0.15 does not properly verify that a user holds board-manager privileges before performing several board-management operations, checking only board membership. This allows any member of a board to carry out manager-only actions on it, including adding or removing members and enabling public access to a private board.	3.8	More Details
CVE-2026-84904	The King Addons for Elementor WordPress plugin before 51.1.81 does not perform per-object authorization checks on a group of image-optimization actions, gating them only on a coarse capability that lower-privileged users also hold and never confirming ownership of the targeted object, allowing authenticated users with author-level access and above to disclose absolute file paths for, overwrite the bytes of, and site-wide re-reference media belonging to other users, including administrators.	3.8	More Details
CVE-2026-91008	The Event Booking Manager for WooCommerce WordPress plugin before 5.3.8 does not perform an ownership or authorization check before rendering booking confirmation details, allowing unauthenticated attackers to retrieve registered attendees' personal information (full name, email address, phone number, and custom registration fields) by supplying an enumerable booking reference. Exploitation is limited to sites configured to use the Event Booking Manager for WooCommerce WordPress plugin before 5.3.8's native (non-WooCommerce) checkout, which is not the default.	3.7	More Details
CVE-2026-86448	The LearnPress WordPress plugin before 4.4.7 does not perform any authentication, capability or nonce check before serving a previously generated order export file, allowing unauthenticated attackers who can determine its identifier to download customer names, purchases, amounts and guest email addresses.	3.7	More Details
CVE-2026-86071	Junrar is an open source Java RAR archive library. Prior to version 7.6.1, LocalFolderExtractor in src/main/java/com/github/junrar/LocalFolderExtractor.java can create directories outside the intended extraction root when processing a crafted archive entry. LocalFolderExtractor.createFile() validates only the final canonical file path, while LocalFolderExtractor.makeFile() creates intermediate path segments with unchecked mkdir() calls. An entry can therefore make the final path resolve inside the destination while causing intermediate directory creation outside it, enabling filesystem pollution or file-versus-directory squatting that can make later security-sensitive writes fail. The demonstrated impact is directory creation, not unconditional arbitrary file-content write. This issue is fixed in version 7.6.1.	3.7	More Details
CVE-2026-93384	Server-side request forgery in Omnibox in Google Chrome on on Android prior to 153.0.8010.52 allowed a remote attacker leveraging social engineering to bypass system access restrictions via crafted network traffic. (Chromium security severity: Medium)	3.7	More Details
CVE-2026-86446	The LearnPress WordPress plugin before 4.4.7 does not restrict the correctness flags it returns when a quiz answer is checked, allowing unauthenticated attackers to obtain the correct answer to every option of a question, along with the instructor's explanation, on courses configured to be taken without enrolling.	3.7	More Details
CVE-2026-93394	A flaw in libmongoc's SCRAM authentication implementation caused the client to continue the authentication handshake and transmit the client proof even when a nonce mismatch was detected in the server's first message. An unauthorized party with a man-in-the-middle position could exploit this by injecting a crafted server-first-message containing a controlled salt and low iteration count, then capturing the resulting client proof to perform offline password cracking. This vulnerability is mitigated by TLS, which is standard in production deployments.	3.7	More Details
CVE-2026-91017	The Robokassa payment gateway for Woocommerce WordPress plugin before 1.8.9 does not verify the authenticity of incoming payment notifications when its non-default deferred-payment feature is enabled, allowing unauthenticated attackers to forge a notification and mark arbitrary WooCommerce orders as paid or on-hold without payment or a valid signature.	3.7	More Details
CVE-	The NeuVector JWT verifier accepts noncanonical Base64URL encodings of the same RSA signature field. An attacker holding a		

2026-78426	valid JWT that has not expired, but was logged out of NeuVector, can continue using the non-expired token with equivalent spelling of the RSA signature field until the token validity expires.	3.7	More Details
CVE-2026-85716	The AsyncHttpClient (AHC) library allows Java applications to easily execute HTTP requests and asynchronously process HTTP responses. From 3.0.8 until 3.0.12, processScramAuthenticationInfo and processAuthenticationInfo compute the SCRAM ServerSignature or Digest rspauth verification result but log a mismatch and still deliver the response as authenticated. On a non-TLS or compromised transport, a peer that has not proved knowledge of the shared secret can therefore be accepted as the server. The fix rejects a present invalid value and computes Digest rspauth from the Authorization parameters actually sent, but verification remains unenforced when the value is absent, the sent parameters cannot be recovered, or Digest uses qop=auth-int. This issue is fixed in version 3.0.12.	3.7	More Details
CVE-2026-81438	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains Use of a Broken or Risky Cryptographic Algorithm vulnerability. An unauthenticated attacker with remote access could potentially exploit this vulnerability, leading to Information disclosure.	3.7	More Details
CVE-2026-12284	Mattermost Desktop App versions <=6.2 6.2.2.0 fails to validate the IPC sender in the leaveCall handler which allows a malicious or compromised Mattermost server (or a user with script access to a connected server view) to disconnect an active call belonging to a different connected server via the desktopAPI.leaveCall IPC message. Mattermost Advisory ID: MMSA-2026-00699	3.7	More Details
CVE-2026-61700	MariaDB Connector/J is used to connect applications developed in Java to MariaDB and MySQL databases. Prior to 2.7.14, 3.3.5, 3.4.3, and 3.5.9, ClientMessage.readPacket processes a server-initiated LOCAL INFILE protocol packet 0xfb without enforcing allowLocalInfile=false. When an application sends a LOAD DATA LOCAL INFILE COM_QUERY, a rogue or man-in-the-middle server can echo the same filename and cause the connector to transmit that file despite the disabled option. The server cannot redirect the request to an arbitrary path and can receive only the exact file already selected by the application, so exploitation requires an application that actively loads sensitive data over an untrusted connection. This issue is fixed in versions 2.7.14, 3.3.5, 3.4.3, and 3.5.9.	3.7	More Details
CVE-2026-84907	The Eventin WordPress plugin before 4.1.24 does not properly authorise order finalisation when its offline (local) payment method is enabled, relying on a nonce that is exposed to unauthenticated visitors and never checking that the caller owns the order, allowing unauthenticated attackers to reset any existing order and its attendees to a pending state and thereby invalidate paid tickets.	3.7	More Details
CVE-2026-81439	Dell OpenManage Server Administrator, versions prior to 11.1.0.3, contains an Incorrect Authorization vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Protection mechanism bypass.	3.7	More Details
CVE-2026-69200	node-opcua is an OPC UA implementation for TypeScript and Node.js. Prior to node-opcua-client 2.145.0, the internal fieldsToJson method in packages/node-opcua-client/source/alarms_and_conditions/client_alarm.ts directly assigns unsanitized field names and allows a __proto__.pollutedKey path to modify Object.prototype. Successful exploitation requires an application to expose attacker-controlled event fields to fieldsToJson and may cause denial of service or application logic corruption. This vulnerability is fixed in 2.145.0.	3.7	More Details
CVE-2026-77860	In NLnetLabs Unbound 1.20.0 up to and including 1.26.0, a vulnerability on the 'serve-expired' code path can cause a double decrement on the 'wait-limit' counter per client IP essentially bypassing one of the counter measures that was introduced for DNSBomb (CVE-2024-33655). A malicious actor can exploit this by controlling an authoritative zone with short TTL, so cached entries expire quickly. Each 'slow' query, one the attacker's authoritative never answers, is followed by one query for an expired cached name, which is answered immediately via the 'serve-expired' path and decrements the counter twice. This second query was named 'pump'. By alternating slow queries and pumps, the attacker keeps the per-client counter at or below the configured 'wait-limit' indefinitely, and can hold an arbitrary number of pending queries from a single source IP, up to the global mesh quota (num-queries-per-thread); eventually bypassing one of the counter measures introduced for DNSBomb (CVE-2024-33655). This vulnerability is present on the 'serve-expired' code path.	3.7	More Details
CVE-2026-54471	Dell SmartFabric Manager, versions prior to 2.2.1, contains an Improper Handling of Insufficient Permissions or Privileges vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Information disclosure.	3.5	More Details
CVE-2026-92381	A weakness has been identified in PbootCMS up to 3.2.22. This affects the function decode_string of the file apps/admin/controller/content/ContentController.php of the component Template Rendering. This manipulation of the argument Title causes cross site scripting. The attack is possible to be carried out remotely. The exploit has been made available to the public and could be used for attacks. The reported GitHub issue was closed with the reason "completed".	3.5	More Details
CVE-2026-92214	A flaw has been found in a2ui-project a2ui up to 0.10.7. Affected is an unknown function of the file samples/community/client/angular/projects/a2a-chat-canvas/src/lib/services/sanitizer-markdown-renderer-service.ts of the component a2a-chat-canvas. Executing a manipulation can lead to cross site scripting. The attack may be performed from remote.	3.5	More Details
CVE-2026-92418	A vulnerability was determined in ChangeWeDer crm up to c07bd4c97141521af6475034bc58523beed51bbd. This vulnerability affects unknown code of the file src/main/resources/public/js/customerServe/customer.serve.js of the component Save Endpoint. This manipulation of the argument customerName causes cross site scripting. Remote exploitation of the attack is possible. The exploit has been publicly disclosed and may be utilized. This product follows a rolling release approach for continuous delivery, so version details for affected or updated releases are not provided. The project was informed of the problem early through an issue report but has not responded yet.	3.5	More Details
CVE-2026-92473	A vulnerability was identified in GPAC 26.08-DEV. The impacted element is the function gf_sg_command_del of the file src/scenegraph/commands.c of the component BIFS Handler. The manipulation leads to use after free. The attack needs to be performed locally. The exploit is publicly available and might be used. Upgrading to version abi-16.24 is sufficient to resolve this issue. The identifier of the patch is e34f4ba349d55cd1849f0bcf4cf46552732e2db7. It is suggested to upgrade the affected component.	3.3	More Details
	A vulnerability was determined in GPAC 26.08-DEV. The affected element is the function gf_node_deactivate_ex of the file		

CVE-2026-92472	src/scenegraph/base_scenegraph.c of the component MP4Box. Executing a manipulation can lead to use after free. The attack needs to be launched locally. The exploit has been publicly disclosed and may be utilized. Upgrading to version abi-16.24 is sufficient to fix this issue. This patch is called e34f4ba349d55cd1849f0bcf4cf46552732e2db7. The affected component should be upgraded. This issue is distinct from CVE-2026-90827.	3.3	More Details
CVE-2026-92474	A security flaw has been discovered in GPAC 26.08-DEV. This affects the function gf_inline_get_proto_lib of the file src/compositor/mpeg4_inline.c of the component Proto Link Handler. The manipulation results in use after free. The attack requires a local approach. The exploit has been released to the public and may be used for attacks. Upgrading to version abi-16.24 mitigates this issue. The patch is identified as e34f4ba349d55cd1849f0bcf4cf46552732e2db7. Upgrading the affected component is recommended.	3.3	More Details
CVE-2026-93380	Race condition in FileSystem in Google Chrome prior to 153.0.8010.52 allowed a remote attacker who had compromised the renderer process and leveraged social engineering to bypass system access restrictions via a crafted HTML page. (Chromium security severity: Medium)	3.1	More Details
CVE-2026-93378	Missing authorization in Storage in Google Chrome prior to 153.0.8010.52 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted PDF file. (Chromium security severity: Medium)	3.1	More Details
CVE-2026-92359	A security flaw has been discovered in ag-ui-protocol ag-ui 0.3.0. The affected element is the function create_strands_app of the file integrations/aws-strands/python/src/ag_ui_strands/utils.py of the component CORSMiddleware. The manipulation results in permissive cross-domain policy with untrusted domains. The attack may be launched remotely. The attack requires a high level of complexity. The exploitability is described as difficult. Upgrading to version AGUI.Abstractions@0.0.6 is sufficient to fix this issue. The patch is identified as 9b143b9668fa52c2054ede9d34a45ac4b4401089. It is suggested to upgrade the affected component.	3.1	More Details
CVE-2026-3855	GitLab has remediated an issue in GitLab CE/EE affecting all versions from 18.2.7 before 19.1.8, 19.2 before 19.2.6, and 19.3 before 19.3.2 that under certain conditions could have allowed an authenticated user with project-level permissions to access restricted file contents on the server or cause denial of service due to improper validation of parameters in the Terraform state upload functionality.	3.1	More Details
CVE-2026-92130	Jenkins Pipeline: Multibranch Plugin 841.vec5b_9e1806ec and earlier does not set the appropriate context for credentials lookup in the resolveScm Pipeline step, allowing attackers with Item/Configure permission to access and capture credentials they are not entitled to.	3.1	More Details
CVE-2026-71181	Dell Update Package Framework, versions prior to 26.07.03, contains an Improper Link Resolution Before File Access ('Link Following') vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Filesystem access for attacker.	3.0	More Details
CVE-2026-71182	Dell Update Package Framework, versions prior to 26.07.03, contains an Improper Link Resolution Before File Access ('Link Following') vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Filesystem access for attacker.	3.0	More Details
CVE-2026-73442	On affected platforms running Arista EOS with VRRP enabled, the peer device VRRP authentication credentials are logged in cleartext on the switch, allowing an authenticated user with sufficient privileges to view agent trace logs (or a system receiving forwarded log output) to obtain the peer device VRRP authentication credentials without having access to the network segment on which VRRP is running.	3.0	More Details
CVE-2026-45723	Omni manages Kubernetes on bare metal, virtual machines, or in a cloud. Prior to 1.6.6 and 1.7.3, managementServer.CreateSchematic in internal/backend/grpc/schematics.go passes the caller-controlled TalosVersion field to imageFactoryClient.OverlaysVersions without validating it as a version. An authenticated Operator can submit traversal segments in TalosVersion, and url.URL.JoinPath normalizes them into unintended paths on the configured image-factory host. Omni then issues HTTP GET requests to those paths and reflects error-body content, enabling same-host endpoint probing and possible disclosure of internal diagnostics while preventing redirection to another host or write requests. This issue is fixed in versions 1.6.6 and 1.7.3.	2.7	More Details
CVE-2026-87031	n Concrete CMS 9.2.0 through 9.5.3, the REST API user creation endpoint (POST /ccm/api/1.0/users, the add() method of concrete/src/Api/Controller/Users.php) did not perform a permission check before creating an account. As a result, any valid OAuth token carrying the users:add scope, including a client_credentials token with no associated user context, could create active, validated user accounts, bypassing email verification and administrator approval. Under default registration settings the created accounts could then edit page content, providing a path to stored cross-site scripting and further compromise. The Concrete CMS security team gave this vulnerability a CVSS v4.0 score of 2.1 with vector CVSS:4.0/AV:N/AC:L/AT:P/PR:H/UI:N/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N. Thanks Winston Crooker for reporting.	2.7	More Details
CVE-2026-89008	The Bookit — Booking & Appointment Calendar WordPress plugin before 2.6.0.5 does not perform an authorization check on one of its appointment-retrieval actions, allowing users with a low-privilege Bookit — Booking & Appointment Calendar WordPress plugin before 2.6.0.5-specific role to read other users' appointment records, including customer names, email addresses, phone numbers and private booking comments.	2.7	More Details
CVE-2026-84905	The Eventin WordPress plugin before 4.1.24 does not verify a user's capability to create accounts when adding a speaker, allowing users with contributor-level access and above to create new WordPress user accounts that carry capabilities beyond their own, including publishing content and uploading files, and, by supplying an email address they control, to obtain a working login to the created account.	2.7	More Details
CVE-2026-84903	The King Addons for Elementor WordPress plugin before 51.1.81 does not perform any capability, post-status, or password check before rendering the content of a user-supplied post, allowing users with Contributor-level access and above to read the content of private, draft, pending, and password-protected posts they are not authorized to access.	2.7	More Details
CVE-2026-82126	The Schema & Structured Data for WP & AMP WordPress plugin before 1.66 does not check that a user is allowed to edit the specific post they request schema generation for, allowing users with the contributor role and above to obtain the content of other users' draft, pending, private and password protected posts.	2.7	More Details

CVE-2026-88844	The MasterStudy LMS WordPress Plugin WordPress plugin before 3.7.50 does not verify that the requesting user owns the course before returning its enrolled-student data, allowing users with the MasterStudy LMS WordPress Plugin WordPress plugin before 3.7.50's Instructor role to disclose the names and email addresses of students enrolled in other instructors' courses.	2.7	More Details
CVE-2026-89007	The Bookit — Booking & Appointment Calendar WordPress plugin before 2.6.0.5 does not perform a capability check in one of its appointment-deletion functions, allowing users with its low-privileged custom Staff role to delete arbitrary appointments.	2.7	More Details
CVE-2026-87836	The Comments Import & Export WordPress plugin before 2.5.4 does not restrict its comment export to users able to moderate comments, nor scope the export to content owned by the requesting user, allowing users with the Author role and above to retrieve every comment on the site, including commenter email addresses, IP addresses, unapproved comment content and comment meta.	2.7	More Details
CVE-2026-75588	Mattermost Desktop App versions <=6.2 6.2.2.0 fail to validate the URL scheme when checking whether a target URL is internal to the connected server, which allows a network-positioned attacker to load a plugin popout window over an insecure connection via a link using a downgraded URL scheme. Mattermost Advisory ID: MMSA-2026-00717	2.6	More Details
CVE-2026-92385	A vulnerability has been found in SourceCodester Online Food Ordering System 1.0. The affected element is an unknown function of the file /admin/update_category.php of the component Category Update. The manipulation leads to cross site scripting. The attack can be initiated remotely. The exploit has been disclosed to the public and may be used.	2.4	More Details
CVE-2026-49292	Kiwi TCMS is an open source test management system. Prior to 16.0, the unauthenticated /init-db/ page handled by InitDBView in tcms/core/views.py remains reachable after initial setup and proxies repeated requests to Kiwi/manage.py migrate. The migration command is reentrant, so repeated access reports that no migrations are available and does not cause data loss, alter application state, reveal confidential information, or produce a documented availability impact. This issue is fixed in version 16.0.	0.0	More Details
CVE-2026-92521	In the Linux kernel, the following vulnerability has been resolved: ACPI: PCI: Clear driver_data on all paths that free the acpi_pci_root acpi_pci_root_add() assigns the freshly allocated root to device->driver_data before dmar_device_add() and pci_acpi_scan_root(). Both failure paths reach the end: label where root is kfree()'d, but only the pci_acpi_scan_root() path clears driver_data first. When dmar_device_add() fails during a hot-add, root is freed while device->driver_data still points at it. The ACPI core does not clear driver_data on attach failure, so a later acpi_pci_find_root() call may dereference this dangling pointer. acpi_pci_root_remove() has the same problem: it frees root without clearing device->driver_data, leaving a dangling pointer behind after the root bridge is removed. Move the NULL assignment to the shared end: label so every error path in acpi_pci_root_add() clears driver_data before freeing root, and clear it in acpi_pci_root_remove() as well, so the object is never left reachable through driver_data after being freed.	N/A	More Details
CVE-2026-92523	In the Linux kernel, the following vulnerability has been resolved: RDMA/nldev: validate dynamic counter attribute length RDMA_NLDEV_ATTR_STAT_HWCOUNTERS is a nested attribute whose children are consumed directly with nla_get_u32(). The top-level policy validates only the container, so it does not establish the fixed shape of each child. Require every child payload to be exactly one u32 before reading it.	N/A	More Details
CVE-2026-93136	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: ep: Fix device refcount leak in the error path of MHI device creation mhi_ep_create_device() takes one device reference for the UL channel and another for the DL channel after allocating the transfer device. These references are normally released by mhi_ep_destroy_device() before the device itself is removed. If dev_set_name() or device_add() fails, the error path currently drops only one reference. The remaining channel references keep the device from being released and leave the channels associated with a device that was never registered. Route both failures through a common unwind path that drops the DL channel reference, the UL channel reference, and the initial reference from device_initialize().	N/A	More Details
CVE-2026-89963	In the Linux kernel, the following vulnerability has been resolved: powerpc/kexec_file: Fix null-ptr-def in extra size calculation A static Sashiko AI review identified a potential NULL pointer dereference in kexec_extra_fdt_size_ppc64(). On platforms without any reserved memory regions, get_reserved_memory_ranges() can return 0 while leaving 'rmem' unallocated as NULL. Passing it directly leads to a kernel panic when evaluating 'rmem->nr_ranges'. Add a NULL check for 'rmem' to prevent this crash.	N/A	More Details
CVE-2026-90021	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_midi: initialize work in f_midi_alloc() f_midi_alloc initializes free_ref to 1 and it can only be incremented when a sound card is registered via f_midi_register_card(). f_midi_register_card() is only called in f_midi_bind() which actually performs INIT_WORK. If f_midi_bind() is never run, work is not initialized and the if condition in f_midi_free becomes true, this results in a warning later in __flush_work as work->func = 0. Fix this by moving INIT_WORK from f_midi_bind() to f_midi_alloc().	N/A	More Details
CVE-2026-93068	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix DM I2C teardown race DM I2C adapters can remain visible to userspace while DM teardown is already in progress. A concurrent i2c-dev transfer may then enter amdgpu_dm_i2c_xfer() after the backing DM state has been torn down, leading to a NULL pointer dereference. Create a devres group around the DM I2C adapter lifetime and release it at the start of dm_hw_fini(), before HPD, IRQ, and DM state are torn down. This removes the I2C adapters first and waits for in-flight users to drain before the structures used by amdgpu_dm_i2c_xfer() disappear. This fixes a teardown ordering race seen during device removal: BUG: kernel NULL pointer dereference RIP: amdgpu_dm_i2c_xfer+0x122/0x1c0 [amdgpu] Call Trace: __i2c_transfer i2c_transfer i2cdev_ioctl_rdwr	N/A	More Details
CVE-2026-89962	In the Linux kernel, the following vulnerability has been resolved: powerpc/kexec_file: Prevent kexec range truncation Sashiko AI review pointed out the following issue. The __merge_memory_ranges() function incorrectly handles overlapping memory ranges when merging them. Although sort_memory_ranges() sorts all ranges by their start address in ascending order beforehand, the merge logic remains defective in two ways: 1. It compares the current range's start against the previous element (i-1) instead of the running target index (idx) 2. It unconditionally overwrites 'ranges[idx].end' with 'ranges[i].end'. This logic flaw leads to critical memory truncation when a larger memory range completely subsumes subsequent smaller ranges. For example, consider a sorted input array with three ranges: Range A (idx=0): [0x1000 - 0x9000] Range B (i=1): [0x2000 - 0x5000] (completely inside Range A) Range C (i=2): [0x6000 - 0x8000] (completely inside Range A) 1. When i=1 (Range B): ranges[1].start (0x2000) <= ranges[0].end + 1 (0x9001) is TRUE. The code executes: ranges[0].end = ranges[1].end, which erroneously shrinks Range A's end from 0x9000 down to 0x5000. 2. When i=2 (Range C): ranges[2].start (0x6000) <= ranges[1].end + 1 (0x5001) is FALSE. The code falls into the else block, creating a broken new range. As a result, valid memory fragments [0x5001 - 0x5fff] and [0x8001 - 0x9000] are completely lost from the kexec exclude lists, potentially allowing the	N/A	More Details

	crash kernel to overwrite active memory, causing data corruption or crashes. Fix this by ensuring the start of the current range is compared against the end of the active merged range (idx), and use max() to safely prevent the outer boundary from being truncated.		
CVE-2026-93135	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject programs with inlined helpers if JIT is not available When an architecture (such as LoongArch, ARM64, and RISC-V) implements bpf_jit_inlines_helper_call(), the verifier skips rewriting the helper call offset (insn->imm) in bpf_do_misc_fixups(). This is because the helper is expected to be inlined by the JIT compiler later. Therefore, insn->imm remains as the raw helper enum ID. However, if JIT is disabled at runtime (net.core.bpf_jit_enable=0) or if JIT compilation fails dynamically (e.g., due to OOM), the program falls back to the BPF interpreter. When the interpreter executes (__bpf_call_base + insn->imm) with the unpatched raw ID, it jumps into an invalid address space, triggering an instruction alignment fault or a kernel panic. Although these helpers have valid C implementations in the kernel, the omission of offset rewriting makes runtime interpreter fallback fatal. Fix this by setting 'prog->jit_required = 1' when helper call rewriting is skipped for JIT inlining. This ensures that such programs are safely rejected if JIT is not available, preventing the runtime kernel panic.	N/A	More Details
CVE-2026-92520	In the Linux kernel, the following vulnerability has been resolved: bpf: Zero queue and stack outputs on lock failure Queue and stack pop/peek helpers accept an uninitialized output buffer because the verifier expects the helper to initialize it. The empty-map error path clears the buffer, but a failed lock acquisition returns -EBUSY without writing it. Clear the output before returning -EBUSY so BPF programs cannot observe uninitialized stack contents after a failed helper call.	N/A	More Details
CVE-2026-93140	In the Linux kernel, the following vulnerability has been resolved: udf: Mark LVID buffer as uptodate before marking it dirty When an I/O error occurs while writing the Logical Volume Integrity Descriptor (LVID) buffer to the block device, the block layer's completion handler (`end_buffer_write_sync()`) clears the `BH_Uptodate` flag on the buffer. However, the buffer still contains valid LVID data in memory. If the filesystem is subsequently remounted read-write or synced, `udf_open_lvid()` or `udf_sync_fs()` will modify the LVID buffer and call `mark_buffer_dirty()`. This triggers a spurious `WARN_ON_ONCE(!buffer_uptodate(bh))` warning in `mark_buffer_dirty()` because the buffer is not marked uptodate, even though its in-memory contents are valid and are about to be overwritten. To prevent this spurious warning, unconditionally set the `BH_Uptodate` flag before calling `mark_buffer_dirty()` in `udf_open_lvid()` and `udf_sync_fs()`. This acknowledges that the in-memory buffer is valid and matches the workaround previously applied to `udf_close_lvid()` in commit 853a0c25baf9 ("udf: Mark LVID buffer as uptodate before marking it dirty"). Extending this workaround ensures consistent behavior across all LVID updates. Buffer I/O error on dev loop0, logical block 128, lost sync page write ------------[cut here]------------ !buffer_uptodate(bh) WARNING: fs/buffer.c:1087 at mark_buffer_dirty+0x299/0x410 fs/buffer.c:1087 ... Call Trace: <TASK> udf_open_lvid+0x369/0x5b0 fs/udf/super.c:2078 udf_reconfigure+0x336/0x540 fs/udf/super.c:679 reconfigure_super+0x232/0x8f0 fs/super.c:1080 vfs_cmd_reconfigure fs/fsopen.c:268 [inline] vfs_fsconfig_locked+0x171/0x320 fs/fsopen.c:297 __do_sys_fsconfig fs/fsopen.c:463 [inline] __se_sys_fsconfig+0x6b9/0x810 fs/fsopen.c:350 do_syscall_64+0x174/0x580 arch/x86/entry/syscall_64.c:94 </TASK>	N/A	More Details
CVE-2026-93141	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: r8a66597: avoid double free of ep0_req in probe error path If usb_add_gadget_udc() fails, r8a66597_probe() jumps to err_add_udc and frees ep0_req, then falls through to clean_up2 where ep0_req is freed again when it is non-NULL. Remove the redundant free from err_add_udc and keep the cleanup in clean_up2 so the request is released exactly once. Issue found using a prototype static analysis tool and confirmed by code review.	N/A	More Details
CVE-2026-93142	In the Linux kernel, the following vulnerability has been resolved: thermal/drivers/rcar: Fix error checking in probe() This code accidentally calls thermal_zone_device_enable() before checking whether thermal_zone_device_register_with_trips() failed. Move the call until later to avoid an error pointer dereference of "priv->zone". The driver works differently depending on if we are using OF thermal or not. We use thermal_add_hwmon_sysfs() if we are using OF thermal and call thermal_zone_device_enable() if not. We can share same error check for if either of these fail. Moving the thermal_zone_device_enable() call is a bit cleaner as well. The original code used a three step process to cleanup: 1. Call thermal_zone_device_unregister() to cleanup. 2. Set priv->zone to an error pointer to preserve the error code. 3. Set priv->zone to NULL to avoid a second call to thermal_zone_device_unregister() in the rcar_thermal_remove() function. Now we can just do a direct goto error_unregister and rcar_thermal_remove() handles the cleanup properly.	N/A	More Details
CVE-2026-93143	In the Linux kernel, the following vulnerability has been resolved: staging: media: ipu7: fix pm_runtime refcount leak in ipu7_resume() ipu7_resume() calls pm_runtime_get_sync() before resuming the device. If the runtime PM resume fails, the usage count remains incremented, but the error path returns without dropping the reference. Use pm_runtime_resume_and_get() instead, which balances the usage count on failure and avoids the leak. Keep returning 0 on error, as resume callbacks should not propagate failures to the PM core, matching the behaviour of the ipu6 driver.	N/A	More Details
CVE-2026-89958	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: Fix dereference matrix_mdev->kvm without checking for NULL The ap_driver structure has two fields which are function pointers to callbacks: * .on_config_changed: called at the start of the AP bus scan function to notify the device driver that the host AP configuration has changed and the associated AP devices will be added or removed accordingly. This gives the implementor a chance to evaluate the configuration changes and respond to them before the associated devices are added or removed. * .on_scan_complete: Called at the end of the AP bus scan function to notify the device driver that the host AP configuration has changed and the AP devices have been added or removed accordingly. This gives the implementor the opportunity to respond to the changes after the associated devices are added or removed. These two callbacks are implemented in the vfio_ap device driver via the vfio_ap_on_cfg_changed and vfio_ap_on_scan_complete functions respectively. Within the call stack of these two callback functions the matrix_mdev->kvm->lock mutex is taken without checking whether matrix_mdev->kvm is NULL or not. If matrix_mdev->kvm has never been set, trying to take the lock will trigger a NULL pointer dereference. This patch adds checks for matrix_mdev->kvm == NULL before taking the matrix_mdev->kvm->lock mutex. Note that the matrix_mdev->kvm->lock mutex taken in the vfio_ap_mdev_hot_plug_config function is moved to the calling function along with the matrix_dev->mdevs_lock which is needed there to access the fields of the matrix_mdev. It makes little sense to make the change the check for matrix_mdev->kvm there before taking the kvm->lock mutex only to have to move it out via another patch, so it is done in this patch. It is important to make note of the following: 1. The matrix_dev->guests_lock is acquired at the start of both callback functions. This ensures that matrix_mdev will not be removed via the vfio_ap_mdev_remove function because it too takes matrix_dev_guests_lock before removing the object; so, matrix_mdev will be available for the duration of the callback functions. 2. The matrix_dev->mdevs_lock mutex must be taken in order to access fields within the matrix_mdev structure 3. matrix_mdev->kvm->lock mutex must be taken before the matrix_dev->mdevs_lock to prevent a lockdep splat. 4: The kvm->lock must be held while plugging the guest's AP configuration into its SIE state description via the	N/A	More Details

	vfio_ap_mdev_update_guest_apcb function. 5. The vfio_ap_mdev_update_guest_apcb checks matrix_mdev->kvm to verify it is not NULL before doing the hot plug of the guest's AP configuration.		
CVE-2026-93145	In the Linux kernel, the following vulnerability has been resolved: clk: qcom: gdsc: tear down per-domain genpds in gdsc_unregister() gdsc_unregister() removes the OF provider entry and tears down the parent/subdomain wiring, but never calls pm_genpd_remove() on the individual generic_pm_domain structures registered by gdsc_init(): void gdsc_unregister(struct gdsc_desc *desc) { struct device *dev = desc->dev; size_t num = desc->num; gdsc_pm_subdomain_remove(desc, num); of_genpd_del_provider(dev->of_node); } That leaves dangling entries on the global gpd_list. After a provider unbind/rebind cycle (deferred-probe replay during early boot, real module unload of a clk driver that owns GDSCs, or an OF-overlay tear-down) the next gdsc_init() will end up trying to re-register a name that is still in the list and pm_genpd_init() returns -EEXIST. While we are here, flip the order so the consumer-facing OF provider entry is the first thing removed -- otherwise a fresh of_genpd_get_from_provider() call racing with the teardown could attach to a domain that is mid-removal. Iterate the scs[] array and pm_genpd_remove() each registered domain after the subdomain links are torn down. The regulators stay devm-managed (devm_regulator_get_optional() in gdsc_register()), so the release happens automatically when the underlying device is unbound; just the genpd accounting needs to be undone explicitly.	N/A	More Details
CVE-2026-93139	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/mes: Fix hung_queue_db_array loop limit for multi-XCC The loop iterated only AMDGPU_MAX_MES_PIPES times, leaving entries uninitialized for multi-XCC GPUs. This causes null pointer dereferences when accessing arrays indexed by XCC ID >= 2. Extend the loop to cover all XCCs (AMDGPU_MAX_MES_PIPES * num_xcc), matching other per-XCC arrays.	N/A	More Details
CVE-2026-93133	In the Linux kernel, the following vulnerability has been resolved: ACPI: RISC-V: Check acpi_get_handle() status in riscv_acpi_add_prt_dep() In riscv_acpi_add_prt_dep(), the acpi_get_handle() call can fail which would leave link_handle uninitialized. Fix it by checking the acpi_get_handle() return status and skip the entry if it fails.	N/A	More Details
CVE-2026-92524	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3-its: Prevent leak in its_vpe_irq_domain_alloc() When its_irq_gic_domain_alloc() fails, the following its_vpe_irq_domain_free() fails to invoke its_vep_teardown() for the corresponding interrupt, which leaks the resource. Invoke its_vpe_teardown() in the error handling path to avoid the leak. [tglx: Massaged change log]	N/A	More Details
CVE-2026-93134	In the Linux kernel, the following vulnerability has been resolved: printk: Fix possible console use-after-free When emitting a record via legacy printing, it is possible that a handover to another legacy printing context occurs. When a context has performed a handover, the console SRCU read lock is released and the pointer to the console struct might now be invalid. Therefore, after calling nbcon_legacy_emit_next_record() or console_emit_next_record(), it is necessary to check if a handover occurred _before_ further @con usage. Sashiko pointed out that console_flush_one_record() was not doing this. In console_flush_one_record(), after emitting a record, move the further usage of @con after the handover check.	N/A	More Details
CVE-2026-89966	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb_cma: fix null nodemask dereference in hugetlb_cma_alloc_frozen_folio alloc_buddy_hugetlb_folio_with_mpol() can pass a NULL nodemask to alloc_fresh_hugetlb_folio() as a fallback to allocate from all nodes. If order is gigantic, alloc_fresh_hugetlb_folio() propagates the NULL nodemask down to hugetlb_cma_alloc_frozen_folio() via alloc_gigantic_frozen_folio(). Additionally, hugetlb_cma_alloc_frozen_folio() previously attempted allocation on hugetlb_cma[nid] without verifying if nid is included in the caller's nodemask. Adding a node_isset(nid, *nodemask) check ensures the initial preferred node allocation honors the memory policy / nodemask. However, hugetlb_cma_alloc_frozen_folio() dereferences the nodemask in node_isset(nid, *nodemask) and for_each_node_mask(node, *nodemask), leading to a null pointer dereference kernel panic when nodemask is NULL. Fix this by checking if nodemask is NULL in hugetlb_cma_alloc_frozen_folio() and defaulting it to cpuset_current_mems_allowed. Enclose the allocation attempts within the cpuset seqcount retry loop so that if the cpuset changes concurrently during allocation, the attempts are retried using the updated nodemask. This ensures that the initial node check and fallback loop safely honor the task's cpuset without violating cpuset constraints or causing NULL pointer dereferences or unexpected allocation failures. From a userspace perspective, this bug allows an unprivileged user to crash the kernel (trigger a panic) by requesting a gigantic hugepage allocation with MPOL_PREFERRED_MANY on a system where CMA is only configured on a subset of NUMA nodes. This can be reproduced by booting a VM with two NUMA nodes, restricting CMA to Node 1 (e.g., hugetlb_cma=1:1G default_hugepagesz=1G hugepagesz=1G hugepages=0), and running a program that allocates a 1GB hugepage area without reserving, restricts allocation to Node 0 using mbind() with MPOL_PREFERRED_MANY, and triggers a page fault: void *ptr = mmap(NULL, 1UL << 30, PROT_READ | PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS | MAP_HUGETLB | MAP_HUGE_1GB | MAP_NORESERVE, -1, 0); unsigned long nodemask = 1; /* Node 0 */ mbind(ptr, 1UL << 30, MPOL_PREFERRED_MANY, &nodemask, sizeof(nodemask) * 8, 0); memset(ptr, 0, 1UL << 30); /* Trigger fault */ This results in a NULL pointer dereference: BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page Oops: Oops: 0000 [#1] SMP NOPTI RIP: 0010:hugetlb_cma_alloc_frozen_folio+0x75/0x120 Call Trace: <TASK> only_alloc_fresh_hugetlb_folio.isra.0+0x2c/0x160 alloc_surplus_hugetlb_folio+0x6d/0x100 alloc_hugetlb_folio+0x3c5/0x660 hugetlb_no_page+0x3d9/0x650	N/A	More Details
CVE-2026-93040	In the Linux kernel, the following vulnerability has been resolved: dmaengine: dw-edma: Serialize channel state checks pause() and resume() read and update channel state without holding vc.lock, while the interrupt handlers update the same state under it. Take the same lock around those state checks so that request, status, and configured stay consistent. For example, pause() can observe EDMA_ST_BUSY right before the interrupt handler completes the final descriptor and moves the channel to EDMA_ST_IDLE, and then record EDMA_REQ_PAUSE on an already idle channel. No further interrupt will acknowledge the request, and since issue_pending() requires EDMA_REQ_NONE, the channel is wedged for good: terminate_all() leaves the stale request behind, so even reconfiguring the channel does not recover it. issue_pending() already runs under vc.lock, but it tests configured before taking it. Move that test under the lock as well, so configured, request, and status are evaluated as one channel-state snapshot.	N/A	More Details
CVE-2026-90006	In the Linux kernel, the following vulnerability has been resolved: samples/damon/mtier: handle damon_stop() failure damon_sample_mtier_stop() assumes its damon_stop() call will always successfully stops the two DAMON contexts. Hence it deallocates the two DAMON contexts after the damon_stop() call. However, if a given context is already stopped, damon_stop() fails and returns an error while letting the DAMON contexts that have not yet stopped keep running. This kind of unexpected early DAMON context stops could happen due to memory allocation failures in kdamond_fn(). Because damon_sample_mtier_stop() just deallocates all DAMON contexts with damon_target and damon_region objects that are linked to the contexts, the execution of the unstopped DAMON context (kdamond) ends up using the memory that freed (use-after-free). Fix the issue by separating the damon_stop() to be invoked per context. Note that DAMON_SYSFS also allows multiple	N/A	More Details

	DAMON contexts execution. But, it calls damon_stop() for each context one by one. Hence this issue is only in mtier. For the long term, it would be better to refactor damon_stop() to always ensure stopping all contexts regardless of the failures in the middle. Make this fix in the current way, though, to keep it simple and easy to backport. I will do the refactoring later. The issue was discovered [1] by Sashiko.		
CVE-2026-93126	In the Linux kernel, the following vulnerability has been resolved: remoteproc: qcom_q6v5_adsp: Fix reference leak for device node When calling of_parse_phandle_with_args(), the caller is responsible to call of_node_put() to release the reference of device node. In adsp_map_carveout, it does not release the reference.	N/A	More Details
CVE-2026-89964	In the Linux kernel, the following vulnerability has been resolved: parisc: eisa: Fix infinite loop when parsing invalid IRQ value When an invalid value is passed via the "eisa_irq_edge=" kernel command line parameter (e.g. "eisa_irq_edge=16,5"), eisa_irq_setup() prints an error message and continues without advancing the current position. As a result the same invalid value is parsed again and again, causing an infinite loop while the kernel boots. Advance to the next comma-separated entry, or stop parsing when there is no next entry, before continuing so that the remaining entries are processed normally.	N/A	More Details
CVE-2026-93128	In the Linux kernel, the following vulnerability has been resolved: platform/x86: lg-laptop: Fix LED resource handling The event notification callback might access kbd_backlight even when it was not successfully registered with the LED subsystem. The same happens inside acpi_remove(), where the LED devices are unregistered unconditionally. Fix this by tracking the availability of the kbd_backlight LED device and use devm_led_classdev_register() to let devres take care of unregistering the LED devices during removal. For this the parent device of the LED devices is changed to the native platform device.	N/A	More Details
CVE-2026-93129	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-wmi-base: Fix handling of ultra performance key The commit message of commit 5fbd827eb9c2 ("platform/x86: dell-wmi: Recognise or support new switches") states that the ultra performance key contains additional data after the type and code fields. The event data passed to dell_wmi_process_key() is already parsed, so "buffer" already starts after those two fields. Use the correct index for accessing the first data field to avoid a potential buffer overread.	N/A	More Details
CVE-2026-93038	In the Linux kernel, the following vulnerability has been resolved: iio: dac: ad5686: missing NULL check on match data Verify that chip_info pointer is not NULL. If a user binds the driver using driver_override via sysfs with a device name not present in the id_table or of_match_table, match data will be NULL.	N/A	More Details
CVE-2026-90015	In the Linux kernel, the following vulnerability has been resolved: xhci: fix lost bounce buffers on TDs spanning several ring segments When a TD reaches a link TRB with data that is not aligned to the endpoint's wMaxPacketSize, xhci_align_td() stages the unalignable tail through the bounce buffer of the ring segment holding that link TRB. xhci_unmap_td_bounce_buffer() later unmaps it and, for IN transfers, copies the data back into the URB's buffer. The enqueue path records the segment that was bounced in td->bounce_seg, under the assumption that a TD never spans more than two ring segments. That assumption does not hold: a TD large enough to span three or more segments crosses several link TRBs and can be bounced at each of them. Only the last one survives in td->bounce_seg, so every earlier bounce buffer is neither copied back nor DMA unmapped. The URB still completes with actual_length equal to the requested length and no error, so the transfer looks successful while a wMaxPacketSize sized hole in the destination buffer silently keeps its previous contents. It also leaks a DMA mapping per dropped bounce. Any sufficiently large and fragmented bulk transfer can hit this. It was found with a USB mass storage device behind xHCI backing a dm-verity target with 512 byte hash blocks, where the stale data is detected rather than silently consumed. The device enumerates as SuperSpeed, so wMaxPacketSize is 1024, while dm-bufio issues one 512 byte bio per hash block. verity_prefetch_io() makes the block layer merge hundreds of them into a single request of up to 512 scatterlist entries of 512 bytes each. At 256 TRBs per ring segment such a TD spans three segments, and every segment boundary falls on an odd multiple of 512, i.e. unaligned to wMaxPacketSize. dm-bufio then caches a hash block holding stale data and dm-verity declares the metadata block corrupted: device-mapper: verity: 8:2: metadata block 10850 is corrupted A reproducer running this under qemu is available at https://github.com/baloo/xhci-verity The bounce state (bounce_buf, bounce_dma, bounce_len, bounce_offs) already lives on the ring segment, so there is nothing extra to track. Keep recording the last bounced segment in td->bounce_seg and, on completion, walk the segments from td->start_seg up to it, unmapping every segment that still has a pending bounce. Stopping at td->bounce_seg rather than td->end_seg matters: a bounce implies the TD continues past that segment's link TRB, so bounce_seg is always strictly before end_seg, and a later TD may already have started in end_seg and been bounced there. Walking that far would copy a foreign bounce buffer into this URB and unmap it twice. It also keeps the walk correct if a TD ever wraps the whole ring so that end_seg == start_seg. [mn: Add ring->num_segs check to prevent unlikely infinite for loop.]	N/A	More Details
CVE-2026-90019	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: fix null pointer dereference in usb_put_function_instance() usb_put_function_instance() attempts to dereference fd inside fi struct to get mod in uvc_alloc_inst() error path. However, fd is not allocated until later in try_get_usb_function_instance() after allocating fi in uvc_alloc_inst() and thus guranteed to be null in error path. Fix this by adding a null check for fi->fd that returns if fd is null.	N/A	More Details
CVE-2026-93130	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-wmi-base: Fix resource leak on module load failure We need to properly clean up the SMBIOS request and the privacy driver when the module load fails.	N/A	More Details
CVE-2026-93131	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-privacy: Fix race condition Accessing priv->features_present needs to happen with the list mutex being held, otherwise priv can be freed at any moment.	N/A	More Details
CVE-2026-90020	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: fix NULL pointer dereference in gadget_dev_ioctl() gadget_dev_ioctl() reads dev->gadget before acquiring dev->lock, but dev->state is checked after acquiring the lock. Therefore a concurrent bind can change the device state between these operations, which can leave ioctl with a stale NULL gadget pointer and causing a NULL pointer dereference at gadget->ops->ioctl. Read dev->gadget while holding dev->lock so that the gadget pointer and device state are sampled consistently.	N/A	More Details
CVE-2026-93132	In the Linux kernel, the following vulnerability has been resolved: ACPI: RISC-V: Fix riscv_acpi_add_prt_dep() loop handling The loop in riscv_acpi_add_prt_dep() includes error conditions that are handled in a dubious - if not outright wrong - way, by continuining the loop (which skips and misses the entry pointer update to point to the next entry). Rewrite the loop as a for loop (that handles the continuation correctly) and wrap the condition and update statements using helper functions to make it cleaner.	N/A	More Details

CVE-2026-92519	In the Linux kernel, the following vulnerability has been resolved: riscv, bpf: Fix memory leak in bpf_jit_free When bpf_int_jit_compile() is called for subprograms, it returns early during the first pass (!prog->is_func || extra_pass is false), keeping ctx->offset alive for the subsequent extra pass. If JIT compilation fails for a later subprogram, the BPF core aborts and calls bpf_jit_free() to clean up the first subprogram. However, bpf_jit_free() fails to free jit_data->ctx.offset, which causes a memory leak of the JIT context offsets array. Fix this by adding the missing kfree(jit_data->ctx.offset) in bpf_jit_free().	N/A	More Details
CVE-2026-93146	In the Linux kernel, the following vulnerability has been resolved: time/namespace: Validate nanosecond field in proc_timens_set_offset() The function validates tv_sec to be within [-KTIME_SEC_MAX, KTIME_SEC_MAX] but never validates that tv_nsec is within the valid range of [0, NSEC_PER_SEC-1] before using it in timespec64_add(). timespec64_add() expects both timespec64 structures to have normalized values with tv_nsec in the range [0, 999999999]. If off->val.tv_nsec contains invalid values (negative or >= NSEC_PER_SEC), it could lead to incorrect calculations or unexpected behavior. Add validation to ensure tv_nsec is within the valid range before performing the addition.	N/A	More Details
CVE-2026-92517	In the Linux kernel, the following vulnerability has been resolved: bpf, riscv: Fix extable handling for arena load_acquire emit_atomic_ld_st() returns 1 to have build_body() skip the zext after a sub-word load_acquire. The caller does "ret = ret ?: add_exception_handler(...)", which skips add_exception_handler() on any non-zero ret, so the extable entry is missing and a faulting PROBE_ATOMIC load_acquire oopses. REG_DONT_CLEAR_MARKER leaves rd stale on fault, and the verifier still thinks the load overwrote it, so a program can leak it through a map. Check ret >= 0 before calling add_exception_handler(), and pass rd for LOAD_ACQ so the fault zeroes rd like a PROBE_MEM load. Return ret unchanged for the zext skip.	N/A	More Details
CVE-2026-90023	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_mass_storage: fix null pointer dereference in fsg_common_set_num_buffers() Previously fsg_num_buffers_validate() was removed as it was not necessary due to Kconfig setting the limits for n from 2 to 256 with default as 2. However, setting the page content in such a way that kstrtou8() reflects n value as either 0 or 1 bypasses these restrictions leading to a null pointer dereference if n is 0. Fix this by adding a check for n < 2 and returning -EINVAL if n is either 0 or 1 consistent with Kconfig logic.	N/A	More Details
CVE-2026-89956	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: Fix missing lock required to access list of ap_matrix_mdev objects In order to traverse or add/remove ap_matrix_mdev objects in the matrix_dev->mdev_list, the matrix_dev->guests_lock mutex must be held. There are two functions that access the list without holding the mutex: vfio_ap_mdev_probe function ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ The vfio_ap_mdev_probe function uses the matrix_dev->mdevs_lock mutex to guard the add of a newly created ap_matrix_mdev object to the matrix_dev->mdev_list. This mutex does not protect list access; its purpose is to guard against concurrent access to fields contained in an ap_matrix_mdev object. This could lead to kernel memory corruption or use-after-free if another mdev is created or removed concurrently. The adding of an ap_matrix_mdev object to matrix_dev->mdev_list is now guarded by the matrix_dev->guests_lock which is the correct way to protect against concurrent mdev_list access. Also removed the following two lines of code because the matrix_mdev is allocated via vfio_alloc_device macro which uses kzalloc, so req_trigger and cfg_chg_trigger are already zero-initialised when the struct is allocated before the call to vfio_register_emulated_iommu_dev. This prevents a window whereby these triggers are set to NULL after the device is exposed to userspace. matrix_mdev->req_trigger = NULL; matrix_mdev->cfg_chg_trigger = NULL; vfio_ap_mdev_for_queue function ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ The status_show function that supports display of the status attribute of the devices in /sys/bus/ap/devices calls the vfio_ap_mdev_for_queue function which iterates the matrix_dev->mdev_list to find the object representing the queue device whose status is to be displayed. In order to traverse this list, the matrix_dev->guests_lock mutex must be held. To fix this, the guests_lock mutex is taken prior to taking the matrix_dev->mdevs_lock mutex in the status_show function. It is taken there rather than the vfio_ap_mdev_for_queue function - where it is needed - because it must be taken prior to the mdevs_lock mutex in order to adhere to the proper locking order and prevent a lockdep splat; also because the mdevs_lock is needed there to access fields within the matrix_mdev object in that function. See the vfio-ap-locking.rst in the linux kernel tree.	N/A	More Details
CVE-2026-93161	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - clear AES key schedule from stack qat_alg_xts_reverse_key() expands the forward XTS AES key on the stack. That schedule contains key material and can remain in the stack frame. Clear the temporary crypto_aes_ctx with memzero_explicit() after the copy.	N/A	More Details
CVE-2026-93162	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - cancel work on re-enable SR-IOV timeout The QAT reset worker queues SR-IOV reenable work using a work_struct and completion embedded in an on-stack adf_sriov_dev_data. If the completion wait times out, the reset worker can return while device_sriov_wq still holds or executes the stack-backed work item. Cancel the work on the device_sriov_wq on timeout before the stack frame unwinds.	N/A	More Details
CVE-2026-93163	In the Linux kernel, the following vulnerability has been resolved: hwrng: core - fix rng list on registration error hwrng_register(rng) does the following: 1. Checks if rng has name and read methods set 2. Checks if the name already exists 3. Adds rng to global rng_list 4. May try to set rng to current_rng If step 4 fails, it returns an error. However, it does not remove the rng from rng_list, causing a dangling reference which can result in use-after-free if the caller frees rng, since registration failed. Add a list_del_init() cleanup step.	N/A	More Details
CVE-2026-92513	In the Linux kernel, the following vulnerability has been resolved: RDMA/mana_ib: drain QP references after partial table insertion mana_table_store_ud_qp() publishes a QP at its send-queue id before inserting the receive-queue id, dropping the XArray lock between the two xa_insert_irq() calls. A concurrent completion handler can look up the QP and take a transient reference. When the second insertion fails, the rollback erased only the send-queue entry and returned, leaving both the initial table reference and the transient reference outstanding while RDMA core frees the QP, causing a use-after-free. Drain the reference as normal destruction does: drop the initial reference and wait for qp->free, releasing the QP only after every concurrent lookup returns its reference.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: uprobes/x86: Move optimized uprobe from nop5 to nop10 Andrii reported an issue with optimized uprobes [1] that can clobber redzone area with call instruction storing return address on stack where user code may keep temporary data without adjusting rsp. Fixing this by moving the optimized uprobes on top of 10-bytes nop instruction, so we can squeeze another instruction to escape the redzone area before doing the call, like: lea -0x80(%rsp), %rsp call tramp Note the lea instruction is used to adjust the rsp register without changing the flags. We use nop10 and following transformation to optimized instructions above and back as suggested by Peterz [2]. Optimize path (int3_update_optimize): 1) Initial state after set_swbp() installed the uprobe: cc 2e 0f 1f 84 00 00 00 00 00 From offset 0 this is INT3 followed by the tail of the original 10-byte NOP. After a previous unoptimization bytes 5..9 may still contain the old call instruction, which remains valid for threads already there. 2) Rewrite the LEA tail and call displacement: cc [8d 64 24 80 e8 d0 d1 d2 d3] From offset 0 this traps on the uprobe INT3. Bytes 1..9 are not executable entry points while byte 0 is trapped. 3)		

CVE-2026-93164	Publish the first LEA byte: [48] 8d 64 24 80 e8 d0 d1 d2 d3 From offset 0 this is: lea -0x80(%rsp), %rsp call <uprobe-trampoline> Unoptimize path (int3_update_unoptimize): 1) Initial optimized state: 48 8d 64 24 80 e8 d0 d1 d2 d3 Same as 3) above. 2) Trap new entries before restoring the NOP bytes: [cc] 8d 64 24 80 e8 d0 d1 d2 d3 From offset 0 this traps. A thread that had already executed the LEA can still reach the intact CALL at offset 5. 3) Restore bytes 1..4 of the original NOP while keeping byte 0 trapped and byte 5 as CALL. cc [2e 0f 1f 84] e8 d0 d1 d2 d3 From offset 0 this still traps. Offset 5 is still the CALL for any thread that was already past the first LEA byte. 4) Publish the first byte of the original NOP: [66] 2e 0f 1f 84 e8 d0 d1 d2 d3 From offset 0 this is the restored 10-byte NOP; the CALL opcode and displacement are now only NOP operands. Offset 5 still decodes as CALL for a thread that was already there. Tthere is only a single target uprobe-trampoline for the given nop10 instruction address, so the CALL instruction will not be changed across unoptimization/optimization cycles. Therefore, any task that is preempted at the CALL instruction is guaranteed to observe that CALL and not anything else. Note as explained in [2] we need to use following nop10: PF1 PF2 ESC NOPL MOD SIB DISP32 NOP10: 0x66, 0x2e, 0x0f, 0x1f, 0x84, 0x00, 0x00, 0x00, 0x00, 0x00 -- cs nopw 0x00000000(%rax,%rax,1) which means we need to allow 0x2e prefix which maps to INAT_PFX_CS attribute in is_prefix_bad function. Also changing the uprobe syscall error when called out of uprobe trampoline to -EPROTO, so we are able to detect the fixed kernel. The optimized uprobe performance stays the same: uprobe-nop : 3.129 ± 0.013M/s uprobe-push : 3.045 ± 0.006M/s uprobe-ret : 1.095 ± 0.004M/s --> uprobe-nop10 : 7.170 ± 0.020M/s uretprobe-nop : 2.143 ± 0.021M/s uretprobe-push : 2.090 ± 0.000M/s uretprobe-ret : 0.942 ± 0.000M/s --> uretprobe-nop10: 3.381 ± 0.003M/s usdt-nop : 3.245 ± 0.004M/s --> usdt-nop10 : 7.256 ± 0.023M/s [1] https://lore.kernel.org/bpf/20260509003146.976844-1-andrii@kernel.org/ [2] https://lore.kernel.org/bpf/20260518104306.GU3102624@noisy.programming.kicks-ass.net/#t	N/A	More Details
CVE-2026-89950	In the Linux kernel, the following vulnerability has been resolved: batman-adv: mcast: linearize skbuff for packet generation batadv_mcast_forw_packet() and batadv_mcast_forw_scrape() is not only called (indirectly) by the unsharing+linearizing batadv_recv_mcast_packet() handler. When it is called (indirectly) by batadv_mcast_forw_mcsend() then it will be unshared but not linearized. The SKB_LINEAR_ASSERT() can therefore cause a fatal BUG(). The linearization should happen during the expansion of the head because the scrape function can be hit already during the initial batadv_mcast_forw_mode() selection code: * batadv_interface_tx * batadv_mcast_forw_mode * batadv_mcast_forw_mode_by_count() * batadv_mcast_forw_push() -> calls batadv_mcast_forw_expand_head() before everything else * batadv_mcast_forw_push_tvlvs() * batadv_mcast_forw_push_dests() * batadv_mcast_forw_push_adjust_padding() * batadv_mcast_forw_scrape()	N/A	More Details
CVE-2026-93166	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: debug: fix off by on in rtw89_ppdu_str() This > comparison should be >= to avoid an out of bounds access.	N/A	More Details
CVE-2026-93167	In the Linux kernel, the following vulnerability has been resolved: csky: Fix a4/a5 restoration in syscall trace path The syscall trace path reloads syscall arguments from pt_regs before calling the syscall handler. On C-SKY ABIv2, the 5th and 6th syscall arguments are prepared as stack arguments before invoking syscallid. The current code adjusts sp before loading LSAVE_A4 and LSAVE_A5. Since those offsets are relative to the original pt_regs base, loading them after changing sp fetches the wrong slots. As a result, traced syscalls that use the 5th or 6th argument may receive corrupted arguments. This is visible with mmap2(), which takes six arguments. A small PTRACE_SYSCALL reproducer opens a file and maps one page with: mmap(NULL, 4096, PROT_READ | PROT_EXEC, MAP_PRIVATE, fd, 0) Before the fix, the traced child fails the mmap and exits with 12. After the fix, the mapping succeeds and the child exits with 0. Fix the trace path by loading a4/a5 from pt_regs before changing sp. Tested on: ck860f, linux-4.19.15, C-SKY abiv2	N/A	More Details
CVE-2026-93168	In the Linux kernel, the following vulnerability has been resolved: dmaengine: xilinx_dma: Fix CPU stall in xilinx_dma_poll_timeout Currently when calling xilinx_dma_poll_timeout with delay_us=0 and a condition that is never fulfilled, the CPU busy-waits for prolonged time and the timeout triggers only with a massive delay causing a CPU stall. This happens due to a huge underestimation of wall clock time in poll_timeout_us_atomic. Commit 7349a69cf312 ("iopoll: Do not use timekeeping in read_poll_timeout_atomic()") changed the behavior to no longer use ktime_get at the expense of underestimation of wall clock time which appears to be very large for delay_us=0. Instead of timing out after approximately XILINX_DMA_LOOP_COUNT microseconds, the timeout takes XILINX_DMA_LOOP_COUNT * 1000 * (time that the overhead of the for loop in poll_timeout_us_atomic takes) which is in the range of several minutes for XILINX_DMA_LOOP_COUNT=1000000. Fix this by using a non-zero value for delay_us. Use delay_us=10 to keep the delay in the hot path of starting DMA transfers minimal but still avoid CPU stalls in case of unexpected hardware failures. One-off measurement with delay_us=0 causes the cpu to busy wait around 7 minutes in the timeout case. After applying this patch with delay_us=10 the measured timeout was 1053428 microseconds which is roughly equivalent to the expected 1000000 microseconds specified in XILINX_DMA_LOOP_COUNT. Add a constant XILINX_DMA_POLL_DELAY_US for delay_us value.	N/A	More Details
CVE-2026-93169	In the Linux kernel, the following vulnerability has been resolved: dmaengine: zynqmp_dma: fix race between runtime PM and device removal In zynqmp_dma_remove(), runtime PM was disabled only after checking state and doing a manual suspend. This can race with runtime PM in the remove/unbind (rmmod) path. Disable runtime PM first, then suspend only if the device is not already suspended. To prevent any further runtime PM transitions.	N/A	More Details
CVE-2026-89949	In the Linux kernel, the following vulnerability has been resolved: batman-adv: dat: avoid unaligned fault in IP extraction Independent of the alignment of the ARP packet in the SKB, either the batadv_arp_ip_src or the batadv_arp_ip_dst will have an unaligned access (on HW without native unaligned read support). Use get_unaligned() to handle this properly on all architectures.	N/A	More Details
CVE-2026-93171	In the Linux kernel, the following vulnerability has been resolved: leds: lp5860: Fix a potential double-unlock In lp5860_device_init(), if lp5860_init_dt() fails, an already unlocked mutex is unlocked another time. Slightly rework how the lock is taken/released to avoid this potential double unlock.	N/A	More Details
CVE-2026-93172	In the Linux kernel, the following vulnerability has been resolved: mm/mm_init: handle alloc_percpu failure in free_area_init_core_hotplug We miss a failed allocation check for pgdat->per_cpu_nodestats, which results in a NULL deref when we offset into the per-cpu area. Propagate -ENOMEM up the stack and leave per_cpu_nodestats pointing at boot_nodestats so a later online can retry the allocation. hotadd_init_pgdat() returns NULL on failure, which __try_online_node() already maps to -ENOMEM. On failure nothing needs to be unwound: - the node is never marked online - per_cpu_nodestats is left pointing at boot_nodestats - __add_memory_resource() cleans up pending memblock resources - later online attempts retry the per_cpu_nodestats allocation	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: bpf,lsm: Drop bpf_prog_free from sleepable_lsm_hooks __bpf_prog_put_rcu() is the call_rcu() callback for non-sleepable programs. security_bpf_prog_free() called from there fires bpf_prog_free in softirq; if a sleepable LSM prog is attached to that hook, might_fault() BUGs: BUG: sleeping function called from		

CVE-2026-93173	invalid context in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 5038 preempt_count: 101, expected: 0 Call Trace: <IRQ> __bpf_prog_enter_sleepable+0x1cd/0x320 kernel/bpf/trampoline.c:1255 bpf_trampoline_6442549705+0x53/0xd7 security_bpf_prog_free+0xde/0x130 security/security.c:5465 __bpf_prog_put_rcu+0xab/0xd0 kernel/bpf/syscall.c:2365 rcu_do_batch kernel/rcu/tree.c:2617 [inline] handle_softirqs+0x236/0x800 kernel/softirq.c:622 </IRQ> The call_rcu/call_rcu_tasks_trace split reflects the freed program's sleepability, not that of any attached observer. security_bpf_prog_free() also frees prog->aux->security, which has to stay after the grace period, so drop bpf_prog_free from sleepable_lsm_hooks rather than move the call. Non-sleepable observers still run there.	N/A	More Details
CVE-2026-90024	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: midi2: Fix null-pointer dereference in f_midi2_free_ep_reqs A null-pointer dereference occurs in f_midi2_free_ep_reqs() when attempting to clean up an endpoint that was never initialized. When configuring the MIDI 2.0 gadget via configfs and setting the block direction to SNDRV_UMP_DIR_INPUT, the initialization of the midi1_ep_out endpoint is explicitly skipped during the gadget bind phase (f_midi2_bind()). As a result, the usb_ep->card field remains NULL. Later, when the host sets the alternate setting, f_midi2_set_alt() unconditionally stops both the IN and OUT endpoints by calling f_midi2_stop_eps(), which in turn calls f_midi2_free_ep_reqs() for both endpoints. When f_midi2_free_ep_reqs() is called for the uninitialized midi1_ep_out, it attempts to dereference usb_ep->card to determine the number of requests to free, leading to a crash. Fix this by using usb_ep->num_reqs instead of usb_ep->card->info.num_reqs in f_midi2_free_ep_reqs(). usb_ep->num_reqs is correctly set during f_midi2_init_ep() and remains 0 if the endpoint was never initialized, safely avoiding the loop. For consistency, apply the same change to f_midi2_alloc_ep_reqs(). Oops: general protection fault, probably for non-canonical address 0xdffffc00000000ee: 0000 [#1] SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000770-0x0000000000000777] ... RIP: 0010:f_midi2_free_ep_reqs drivers/usb/gadget/function/f_midi2.c:1166 [inline] RIP: 0010:f_midi2_stop_eps+0x28e/0x4d0 drivers/usb/gadget/function/f_midi2.c:1246 ... Call Trace: <TASK> f_midi2_set_alt+0x11c/0xf00 drivers/usb/gadget/function/f_midi2.c:1296 composite_setup+0x1ffd/0x3480 drivers/usb/gadget/composite.c:1933 configfs_composite_setup+0xbd/0x100 drivers/usb/gadget/configfs.c:1877	N/A	More Details
CVE-2026-93160	In the Linux kernel, the following vulnerability has been resolved: crypto: atmel-ecc - reject hardware ECDH without a public key The hardware ECDH path in atmel_ecdh_compute_shared_secret() uses the private key stored in the device. However, the public key is cached only after atmel_ecdh_set_secret() successfully generated that private key for the current tfm. atmel_ecdh_generate_public_key() already rejects requests when no public key is cached. Add the same check to atmel_ecdh_compute_shared_secret() to prevent the device from using a private key that was not generated for the current tfm.	N/A	More Details
CVE-2026-93159	In the Linux kernel, the following vulnerability has been resolved: crypto: atmel-sha204a - fix heap info leak on I2C transfer failure The nonblocking RNG path allocates a work_data structure to track the state of an in-flight asynchronous I2C request. This pointer is stored in rng->priv and later consumed by the read path once the transaction completes. If the underlying I2C transfer fails, the completion callback is invoked with a non-zero status. In this case, the allocated work_data is not usable for producing RNG output and must not remain associated with the hwrng state. Previously, the failure path only logged a warning but left the pointer state uncleared, which can result in subsequent read attempts observing stale state and interpreting it as valid completion data. Fix this by freeing the pending work_data. The I2C transaction reports an error. This ensures that failed requests do not leave residual state behind that could be interpreted as valid RNG data on later reads. Clearing rng->priv is done at the subsequent call to nonblocking read.	N/A	More Details
CVE-2026-92514	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: Fix CEQ tasklet use-after-free on removal Each CEQ interrupt handler only schedules eqc->tasklet. The tasklet calls erdma_ceq_completion_handler(), which reads the DMA-coherent EQ ring through get_next_valid_eqe() and updates eq->dbrec through notify_eq(). erdma_ceqs_uninit() frees each CEQ IRQ and then destroys its EQ. free_irq() prevents another hard IRQ and waits for an in-flight handler, but it does not drain a tasklet that the handler already scheduled. The tasklet can therefore access eq->qbuf or eq->dbrec after erdma_eq_destroy() frees them. Clearing ceq_cb->ready does not synchronize with a tasklet that already passed the check at the start of erdma_ceq_completion_handler(). Kill the tasklet after free_irq(), when no handler can schedule it again, and before erdma_ceq_uninit_one() releases the EQ buffers.	N/A	More Details
CVE-2026-93153	In the Linux kernel, the following vulnerability has been resolved: RDMA/bng_re: return a timeout when firmware responses stall __wait_for_resp() documents that it returns a non-zero error when a firmware command does not complete, and bng_re_rcfw_send_message() already marks the firmware as stalled when the helper returns -ENODEV. However, the helper ignores wait_event_timeout() expiry. If the response slot remains in use after the timeout and after the polled CREQ service attempt, the loop starts another full timeout period and can repeat forever. Return -ENODEV after a timed out wait that still has no response. The existing caller then marks FIRMWARE_STALL_DETECTED and returns -ETIMEDOUT to the command issuer.	N/A	More Details
CVE-2026-93123	In the Linux kernel, the following vulnerability has been resolved: serial: qcom-geni: do not advance stale DMA completions The qcom GENI serial DMA TX completion path advances the transmit fifo by the number of bytes recorded in port->tx_remaining. If uart_flush_buffer() runs after the hardware has completed a DMA transfer but before the DMA completion interrupt has been handled, the serial core resets the transmit fifo while port->tx_remaining still describes the old DMA transfer. A previous fix avoided advancing an empty fifo by checking that the fifo length is at least tx_remaining. That still does not distinguish the old DMA payload from new bytes written after the flush. If userspace writes new data before the stale DMA completion interrupt is handled, the fifo can again contain at least tx_remaining bytes and the stale completion can advance and discard those new bytes. Mark an in-flight DMA transfer stale when the transmit fifo is flushed. The later completion still unprepares the original DMA mapping using the saved length, but it no longer advances the transmit fifo.	N/A	More Details
CVE-2026-89955	In the Linux kernel, the following vulnerability has been resolved: s390/vfio-ap: Fix NULL deref in status_show() during queue probe When vfio_ap_mdev_probe_queue() creates the sysfs attribute group, the queue's driver data has not yet been set. A concurrent read of the 'status' attribute can therefore call dev_get_drvdata() and get NULL, which is then passed directly to vfio_ap_mdev_for_queue() where q->apqn is unconditionally dereferenced, causing a NULL pointer dereference. Fix this by acquiring the update locks before calling sysfs_create_group(). The status_show() function acquires guests_lock before reading the driver data, so any concurrent read will block until after dev_set_drvdata() has been called and the update locks are released. As a bonus, the APQN no longer needs to be read from the queue struct after allocation — it can be read directly from apdev before allocation and stored in a local variable, which is then assigned to q->apqn once the allocation succeeds.	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211_hwsim: avoid NULL skb in stop queue drain mac80211_hwsim_stop() drops any frames left in data->pending. The loop currently checks skb_queue_empty() and then dequeues separately. That split is racy with TX status handling, which can remove a pending frame under the queue lock. If the	N/A	More

93149	last entry is removed after the empty check, skb_dequeue() returns NULL and the stop path passes that NULL skb to ieee80211_free_txskb(). Use skb_dequeue() as the loop condition instead. The dequeue result is the object that stop owns and frees, and a concurrent status completion that empties the queue simply makes the loop terminate.		Details
CVE-2026-93150	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: Make nr_deadline_tasks an atomic_t The nr_deadline_tasks variable in the cpuset structure was introduced by commit 6c24849f5515 ("sched/cpuset: Keep track of SCHED_DEADLINE task in cpusets"). It is reported by sashiko [1] that nr_deadline_tasks can currently be modified by inc_dl_tasks_cs() under rq->lock and by cpuset_attach() under cpuset_mutex. So if both updates happen simultaneously, the nr_deadline_tasks variable can be corrupted leading to incorrect operations down the road. Fix that by changing its type to atomic_t so that nr_deadline_tasks are always atomically updated. This fix patch is a low hanging fruit. It can handle some of the races between a concurrent sched_setscheduler() and cpuset_can_attach()/cpuset_attach() calls, but not all of them like the other issue raised by sashiko [2]. This will be handled hopefully in a future follow up patch. [1] https://sashiko.dev/#/patchset/20260626181923.133658-1-longman%40redhat.com [2] https://sashiko.dev/#/patchset/20260630033344.352702-1-longman%40redhat.com	N/A	More Details
CVE-2026-89953	In the Linux kernel, the following vulnerability has been resolved: mtd: mtdoops: free page bitmap when the backing MTD is removed mtdoops_notify_add() allocates oops_page_used when the configured MTD device is registered. mtdoops_notify_remove() detaches from that device but leaves the bitmap allocated. If the same MTD device is later registered again, the add path allocates a new bitmap and overwrites the old pointer, leaking one vmalloc allocation per remove/add cycle. This is only visible when the backing MTD device can disappear and be registered again while mtdoops remains loaded, so the usual static MTD case does not expose it. Free the bitmap after unregistering the dumper and flushing the pending workers, then clear the pointer and page count before a later attach can allocate fresh state. Clearing the pointer also keeps the module exit path from freeing the same bitmap a second time after a remove event.	N/A	More Details
CVE-2026-93152	In the Linux kernel, the following vulnerability has been resolved: nvme-apple: Use acquire/release for queue enabled state apple_nvme_init_queue() initializes queue state and then marks the queue enabled. The interrupt and request paths check enabled before using that queue state. The old wmb() after WRITE_ONCE(enabled, true) does not publish the earlier initialization before enabled becomes visible. Use a release store when enabling the queue and acquire loads when testing it. Although the shutdown-side enabled accesses are not used for publishing queue initialization, use helpers for them as well for consistency.	N/A	More Details
CVE-2026-89952	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: validate ONFI extended parameter page sections nand_flash_detect_ext_param_page() allocates the length declared by the ONFI parameter page, then treats the data as a fixed header followed by variable-length sections. It reads that header and advances over sections without first proving that the fixed page and each current section fit in the allocation. Reject pages shorter than the fixed header, track the remaining variable area while walking sections, and require the ECC section to contain every field read from struct onfi_ext_ecc_info. Use device-scoped diagnostics that identify the malformed ONFI section.	N/A	More Details
CVE-2026-93158	In the Linux kernel, the following vulnerability has been resolved: crypto: sa2ul - stop probe if context pool creation fails sa_ul_probe() calls sa_init_mem() to create the DMA pool used for security context buffers, but ignores its return value. If pool creation fails, probe still continues with DMA setup, algorithm registration and child population even though later request setup depends on that pool. Stop probing when sa_init_mem() fails, and route that failure to the PM cleanup path without attempting to destroy an uncreated DMA pool.	N/A	More Details
CVE-2026-92516	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix offset warn check for bpf_res_spin_lock Sashiko pointed out correctly that the case statement for BPF_RES_SPIN_LOCK incorrectly checks offset for BPF_SPIN_LOCK. Fix it by checking res_spin_lock_off instead.	N/A	More Details
CVE-2026-93155	In the Linux kernel, the following vulnerability has been resolved: crypto: keembay - Fix AEAD unregister count in error path register_aes_algs() registers the AEAD algorithms before registering the skcipher algorithms. If skcipher registration fails, the function unwinds the earlier AEAD registration with crypto_engine_unregister_aeads(), but it passes ARRAY_SIZE(algs), which is the skcipher table size. Use ARRAY_SIZE(algs_aead) for the AEAD unwind path so the unregister helper iterates over the same table that was registered. Also clarify the nearby comment: the crypto registration helpers clean up algorithms registered within the same call, while this function must still unwind earlier successful registration steps.	N/A	More Details
CVE-2026-93156	In the Linux kernel, the following vulnerability has been resolved: crypto: rk3288 - fail ahash requests on HASH idle timeout rk_hash_run() waits for RK_CRYPTO_HASH_STS to become idle after the final DMA transfer, but ignores the poll result. If the hash engine never becomes idle, the driver still reads the digest registers and finalizes the request with the previous success value. Store the poll result and finalize the request with the timeout error before reading the digest registers.	N/A	More Details
CVE-2026-92512	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix use after free in ib_query_qp() When querying a QP via the netlink flow the only synchronization mechanism for the said QP is rdma_restrack_get(), meanwhile during the QP destroy path rdma_restrack_del() is called at the end of the ib_destroy_qp_user() function which is too late, since by then the vendor specific resources for said QP would already be destroyed, and till the rdma_restrack_del() is called this QP can still be accessed, which could cause the use after free below. Fix this by moving the rdma_restrack_begin_del() to the start of the ib_destroy_qp_user(), which in turn waits for all usages of the QP to be done then removes it from the database to prevent access to it while it is being destroyed. RIP: 0010:ib_query_qp+0x15/0x50 [ib_core] Code: 48 83 05 5d 8e b9 ff 01 eb b5 66 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 44 00 00 48 c7 46 40 00 00 00 00 48 c7 46 78 00 00 00 00 <48> 8b 07 48 8b 80 88 01 00 00 48 85 c0 74 1a 48 83 05 54 91 b9 ff RSP: 0018:ff11000108a8f2f0 EFLAGS: 00010202 RAX: 0000000000000000 RBX: ff11000108a8f370 RCX: ff11000108a8f370 RDX: 0000000000000000 RSI: ff11000108a8f3d8 RDI: 0000000000000000 RBP: ff1100010de5a000 R08: 0000000000000e80 R09: 0000000000000004 R10: ff110001057a604c R11: 0000000000000000 R12: ff11000108a8f370 R13: ff110001090e8000 R14: 0000000000000000 R15: ff110001057a602c FS: 00007f2ffd8db6c0(0000) GS:ff110008dc90b000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 000000010b9a7004 CR4: 0000000000373eb0 Call Trace: <TASK> mlx5_ib_gsi_query_qp+0x21/0x50 [mlx5_ib] mlx5_ib_query_qp+0x689/0x9d0 [mlx5_ib] ib_query_qp+0x35/0x50 [ib_core] fill_res_qp_entry_query.isra.0+0x47/0x280 [ib_core] ? __wake_up+0x40/0x50 ? netlink_broadcast_filtered+0x15a/0x550 ? kobject_uevent_env+0x562/0x710 ? ep_poll_callback+0x242/0x270 ? __nla_put+0xc/0x20 ? nla_put+0x28/0x40 ? nla_put_string+0x2e/0x40 [ib_core] fill_res_qp_entry+0x138/0x190 [ib_core] res_get_common_dumpit+0x4a5/0x800 [ib_core] ? fill_res_qp_entry_query.isra.0+0x280/0x280 [ib_core] nldev_res_get_qp_dumpit+0x1e/0x30 [ib_core] netlink_dump+0x16f/0x450 __netlink_dump_start+0x1ce/0x2e0 rdma_nl_rcv_msg+0x1d3/0x330 [ib_core] ? nldev_res_get_qp_raw_dumpit+0x30/0x30 [ib_core] rdma_nl_rcv_skb.constprop.0.isra.0+0x108/0x180 [ib_core]	N/A	More Details

	rdma_nl_rcv+0x12/0x20 [ib_core] netlink_unicast+0x255/0x380 ? __alloc_skb+0xfa/0x1e0 netlink_sendmsg+0x1f3/0x420 __sock_sendmsg+0x38/0x60 ____sys_sendmsg+0x1e8/0x230 ? copy_msghdr_from_user+0xea/0x170 ___sys_sendmsg+0x7c/0xb0 ? __futex_wait+0x95/0xf0 ? __futex_wake_mark+0x40/0x40 ? futex_wait+0x67/0x100 ? futex_wake+0xac/0x1b0 __sys_sendmsg+0x5f/0xb0 do_syscall_64+0x55/0xb90 entry_SYSCALL_64_after_hwframe+0x4b/0x53		
CVE-2026-92515	In the Linux kernel, the following vulnerability has been resolved: bpf: Preserve unique-field state across nested structs btf_find_struct_field() initializes a fresh seen mask for every recursive descent. Unique special fields in different levels of the same aggregate therefore do not see one another. The duplicate fields can reach btf_parse_fields(), where they trigger an invariant WARN_ON_ONCE(). A crafted user BTF can consequently trigger the warning before map creation checks capabilities. Initialize the seen mask once in btf_find_field() and pass the same pointer through struct, datasec, and nested-struct walks. This gives the entire field traversal one shared uniqueness state.	N/A	More Details
CVE-2026-93157	In the Linux kernel, the following vulnerability has been resolved: hwrng: xilinx-trng - propagate timeout before any data is read xtrng_readblock32() polls for 16-byte chunks but returns the number of bytes read even when the first poll times out. Its caller then treats a zero return as a short successful read, and partial reads for full 32-byte blocks can make the tail copy use a fixed block offset rather than the amount already produced. Return the poll error when no data has been read, preserve partial positive returns after some data is available, stop the generator on all collection exits, and append tail bytes at the current output count.	N/A	More Details
CVE-2026-93124	In the Linux kernel, the following vulnerability has been resolved: platform/x86: asus-wireless: Fail probe when there is no ACPI match Every platform driver can be forced to match a device that does not match its list of device IDs because of device_match_driver_override(), so platform drivers that rely on the existence of a device ACPI companion object need to verify its presence. asus_wireless_probe() returns success when acpi_match_acpi_device() finds no match, leaving behind an input device that never reports anything because the notify handler is not installed. Worse, when the driver is force-bound to a device without an ACPI companion, probe still succeeds and stores a NULL companion pointer, which asus_wireless_remove() later passes to acpi_dev_remove_notify_handler(), leading to a NULL pointer dereference on unbind. Return -ENODEV when the device does not match the ID table. This also covers the missing-companion case, because acpi_match_acpi_device() rejects a NULL device. Perform the check before allocating any driver state, instead of after the input device has already been registered.	N/A	More Details
CVE-2026-93041	In the Linux kernel, the following vulnerability has been resolved: dmaengine: dw-edma: Serialize abort state updates dw_edma_abort_interrupt() drops vc.lock before changing request and status. issue_pending() can acquire the lock in that small window, observe the old busy state, and skip starting queued descriptors. Then the abort handler overwrites the channel status as idle, leaving the new descriptors stranded for good. Keep descriptor completion and the state transition in the same critical section.	N/A	More Details
CVE-2026-93067	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: tc358767: clamp the reported AUX read size to the request tc_aux_transfer() clamps an AUX read to the payload limit: size_t size = min_t(size_t, DP_AUX_MAX_PAYLOAD_BYTES - 1, msg->size); After the transfer it replaces size with the byte count the controller reports in AUX_BYTES: if (size) size = FIELD_GET(AUX_BYTES, auxstatus); AUX_BYTES is GENMASK(15, 8), so it can be up to 255. Nothing clamps it back to the request. tc_aux_read_data() reads that many bytes into the 16-byte auxrdata stack buffer, then copies them into the caller buffer. A reported count of 255 makes the read run to 256 bytes and overruns both. The controller should never report more than it was asked to transfer, so this is defense in depth rather than a live hole. The reported count is only lightly trusted, and the check is cheap. Clamp it back to the request, the same way ti-sn65dsi86 does in commit aca58eac52b8 ("drm/bridge: ti-sn65dsi86: Never store more than msg->size bytes in AUX xfer").	N/A	More Details
CVE-2026-93052	In the Linux kernel, the following vulnerability has been resolved: misc: bcm-vk: Use acquire/release for msgq_inited bcm_vk_sync_msgq() fills the message queue information and then sets msgq_inited. Readers call bcm_vk_drv_access_ok() before accessing the message queues and their cached queue information. atomic_set()/atomic_read() do not order those accesses. A reader can see msgq_inited set while still seeing stale queue information. Use release when publishing the initialized queues and acquire when checking the gate. Keep the clear in bcm_vk_blk_drv_access() as atomic_set(). It closes the gate and does not publish queue state to readers.	N/A	More Details
CVE-2026-93055	In the Linux kernel, the following vulnerability has been resolved: UDF symlink pathComponent header OOB read udf_symlink_filler() can enter udf_pc_to_char() with a partial pathComponent header. Validate that enough input remains for a complete pathComponent header before accessing it. Reject malformed symlink data that would otherwise make udf_pc_to_char() perform an out-of-bounds read.	N/A	More Details
CVE-2026-93174	In the Linux kernel, the following vulnerability has been resolved: bpf: Copy per-CPU map value padding in copy_map_value_long() In kernel, per-CPU map elements are stored with round_up(map->value_size, 8) bytes. On UAPI lookup paths, it copies the rounded size for each CPU into a temporary buffer. However, copy_map_value_long() passes 'map->value_size' to bpf_obj_memcpy(). When the map has special fields, bpf_obj_memcpy() copies around those fields with memcpy(), and does not copy the tail padding between 'map->value_size' and round_up(map->value_size, 8). The temporary UAPI lookup buffers are allocated without __GFP_ZERO. As a result, when the per-CPU map's value size is not equal to round_up(map->value_size, 8), UAPI LOOKUP_ELEM and its variants can return stale heap contents from that padding to user space. The same issue applies to bpf_iter for per-CPU maps. Pass round_up(map->value_size, 8) to bpf_obj_memcpy() from copy_map_value_long(), so per-CPU maps both with and without special fields copy the entire per-CPU slot. Remove the now redundant round_up() from bpf_obj_memcpy()'s long_memcpy path.	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: transfer the pmd dirty bit to the folio on zap zap_huge_pmd_folio() propagates the pmd young bit to the folio for the file case, but not the dirty bit. The pte path does propagate it, in zap_present_folio_ptes() and so does the pmd split path, in __split_huge_pmd_locked(). For most file mappings the omission is harmless, because writing to a shared file mapping goes through page_mkwrite(), which dirties the folio. tmpfs is different: it has no page_mkwrite(), and vma_wants_writenotify() is false for it, so a *read* fault on a MAP_SHARED tmpfs mapping installs a writable pmd via do_read_fault(). do_read_fault() does not call fault_dirty_shared_page(), so subsequent stores through that mapping set only the hardware dirty bit in the pmd and never call folio_mark_dirty(). A shmem folio allocated by a fault is marked uptodate but not dirty (see the clear: block in shmem_get_folio_gfp()), so PG_dirty is never set at all. Unmapping such a folio - munmap(), or exit_mmap() when the process dies - then loses the only record that it was written, because zap_huge_pmd() drops the pmd without transferring the dirty bit. Reclaim afterwards sees a clean shmem folio: the whole swap-out block in shrink_folio_list() is inside "if (folio_test_dirty(folio))", so pageout() is skipped and the folio falls into __remove_mapping(). There, folio_is_file_lru() is false for a swapbacked folio, so no shadow entry is created and	N/A	More

89987	__filemap_remove_folio(folio, NULL) simply empties the i_pages slot. The data is freed without ever being written to swap, and the next fault on that index returns a freshly zeroed folio. This is silent data loss for any process that keeps state in a MAP_SHARED tmpfs segment across an unmap - for example a cache handed from one process generation to the next through /dev/shm. It requires the folio to be PMD-mapped, so it only shows up once shmem THP is enabled (which is what we did in Meta fleet and started noticing crashes); with THP off the pte path transfers the dirty bit correctly. It also only becomes visible when swap is enabled, because with no swap device shmem folios (which are on the anon LRU) are not scanned by reclaim at all, so the clean folio is never dropped. Reproduced on x86_64 with a tmpfs mounted huge=within_size: read-fault a 2MB-backed region, write a known pattern through the resulting mapping, munmap, force reclaim of the cgroup, then re-map and read back. Without this patch the region reads back as zeros and vmstat shows zswpout 0 - the data was discarded rather than swapped. With this patch the region reads back correctly and the pages are swapped out as expected. With huge=never, or when the first touch is a write, the test passes either way.		Details
CVE-2026-93080	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix transport device teardown lookup SCMI transport devices are deliberately excluded from normal SCMI bus matching so protocol drivers cannot bind to the internal transport children. However, scmi_device_destroy() uses the same protocol/name lookup to find devices that must be unregistered during channel teardown. Split the match helper so driver matching still skips transport devices, while explicit child lookup can find them for teardown. Use a shared transport-device name prefix macro for both matching and name generation. Since transport-device names are derived from direction and protocol ID, reject duplicate protocol channel setup before creating or finding a transport device. This prevents malformed firmware with duplicate protocol child nodes from reusing an existing transport device and then destroying it when the duplicate IDR insertion fails.	N/A	More Details
CVE-2026-89996	In the Linux kernel, the following vulnerability has been resolved: dma-buf: dma-heap: don't publish fd before copy_to_user() succeeds DMA_HEAP_IOCTL_ALLOC allocates a dma-buf and installs an fd into the caller's fd table via dma_buf_fd() -> fd_install() before dma_heap_ioctl() copies the result back to userspace. If the trailing copy_to_user() fails, userspace never learns the fd number, but the fd (and the underlying dma-buf reference) are already visible to other threads in the same process and are leaked for the lifetime of the process. The obvious "close it on the failure path" fix is unsafe: once fd_install() has run, another thread can already dup() the fd, send it via SCM_RIGHTS, or close() it and let its number be reused, so a subsequent close_fd() from the ioctl path can operate on an unrelated file. This was pointed out by Christian König on v1 [1]. Restructure the allocation path so that fd_install() is the last, unfailable step of a successful ioctl: 1. heap->ops->allocate() creates the dma_buf. 2. get_unused_fd_flags() reserves an fd number in the caller's fd table without publishing it, so no other thread can observe it. 3. copy_to_user() delivers the fd number to userspace; on failure the fd is returned with put_unused_fd() and the dma_buf reference is dropped with dma_buf_put(), leaving no user- visible state behind. 4. dma_buf_fd_install() publishes the fd and emits the trace_dma_buf_fd tracepoint -- from here on the ioctl cannot fail. A new dma_buf_fd_install() helper is introduced in dma-buf.c to wrap fd_install() together with the DMA_BUF_TRACE() call, preserving the export tracing that dma_buf_fd() provides. dma_heap_ioctl_allocate() is refactored to return the struct dma_buf * directly (returning ERR_PTR on failure) so the caller holds the dmabuf reference across steps 3 and 4. The failure at step 3 is easily reachable from userspace: pass a struct dma_heap_allocation_data that lives in a page whose protection is flipped to PROT_READ between copy_from_user() and copy_to_user() (e.g. via mprotect()). Before this change each such ioctl leaks one dmabuf fd; after it, the fd table is unchanged on failure and only /dev/dma_heap/<name> remains open. No UAPI or heap-driver interface change. [1] https://lore.kernel.org/dri-devel/175e98de-f414-47d7-81c1-c0fe0a8f7f62@amd.com/	N/A	More Details
CVE-2026-93081	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix SCMI device destroy lifetimes scmi_child_dev_find() drops the reference returned by device_find_child() before returning the scmi_device pointer. A concurrent unregister can then release the device while the destroy path is still using the returned pointer. Make the lookup helper return the device_find_child() reference and keep it until scmi_device_destroy() has finished unregistering the child. Also split device_unregister() in __scmi_device_destroy() so the SCMI bus ID is not made reusable until after device_del() has removed the old scmi_dev.N name from sysfs. This avoids a new SCMI device reusing the same ID while the old device is still registered. The final device release callback is also a possible cleanup path when SCMI children are deleted by driver core recursion rather than __scmi_device_destroy(). Release the SCMI bus ID from a common helper used by destroy, register-failure and final-release paths, and clear scmi_dev->id after freeing it so the final release cannot free the same ID again.	N/A	More Details
CVE-2026-93053	In the Linux kernel, the following vulnerability has been resolved: speakup: keyhelp: guard letter_offsets possible out-of-range indexing help_init() builds letter_offsets[] by using the first byte of each function name as an index via `(start & 31) - 1`. If function_names are overridden from sysfs (root) with a name starting outside [a–z], the index underflows or exceeds the array, leading to OOB write. Function names can be overridden with the following commands as root: modprobe speakup_soft echo "0 _bad" > /sys/accessibility/speakup/i18n/function_names # then press Insert+2 on /dev/tty This fix checks the first letter in help_init(), and if it is not in the [a–z] range the function returns an error to the caller. Eventually this error is propagated to drivers/accessibility/speakup/main.c:2217, which causes a bleep sound.	N/A	More Details
CVE-2026-93051	In the Linux kernel, the following vulnerability has been resolved: misc: ad525x_dpot: use driver core groups for sysfs files ad_dpot_probe() creates per-RDAC sysfs files manually and then optionally creates the command sysfs group. This leaves probe responsible for rolling back partial sysfs state and makes remove responsible for matching every file that probe created. Move the device attributes into driver core dev_groups for the I2C and SPI drivers and use an is_visible() callback to expose only the attributes supported by the probed device. With this shape, the driver core creates the sysfs files only after probe succeeds and removes them before the remove callback frees the driver data.	N/A	More Details
CVE-2026-93057	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Avoid possible memory reclaim deadlock in TX EQTR context TX EQTR may run while devfreq gear scaling has quiesced the UFS tagset. In that context, functions ufshcd_tx_eqtr(), __ufshcd_tx_eqtr() and ufs_qcom_get_rx_fom() allocate memory with GFP_KERNEL. If direct reclaim is triggered, reclaim/writeback can depend on I/O to UFS device. Because the queue is quiesced, this can cause deadlock. Use memalloc_noio_save/restore() in ufshcd_tx_eqtr() to cover all allocations in the TX EQTR call tree, including: - params->eqtr_record in ufshcd_tx_eqtr() - eqtr_data in __ufshcd_tx_eqtr() - params in ufs_qcom_get_rx_fom() This is preferred over tagging individual call sites with GFP_NOIO, as it automatically covers any future allocations added anywhere in the call tree without requiring each caller to be aware of this constraint. [mkp: fix label as suggested by Bart]	N/A	More Details
CVE-2026-93082	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Unwind P2A receiver mailbox setup failure mailbox_chan_setup() can request an additional P2A receiver channel after successfully acquiring the primary P2A channel. If that later request fails, the function returns immediately and leaves the primary channel allocated. Unwind the primary mailbox channel before returning the error so probe deferral or other setup failures do not leave the channel busy for	N/A	More Details

	later probe attempts.		
CVE-2026-93083	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Unwind TX receiver mailbox setup failure mailbox_chan_setup() can request an additional unidirectional TX receiver channel after successfully acquiring the primary channel. If that second request fails, the function returns immediately and leaves the primary channel allocated. Unwind the primary mailbox channel before returning the error so probe deferral or other setup failures do not leave the channel busy for later probe attempts.	N/A	More Details
CVE-2026-93084	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Drop handle on protocol bind failures The SCMI bus notifier acquires an SCMI handle when the driver core emits BUS_NOTIFY_BIND_DRIVER, before invoking the protocol driver probe callback. The protocol probe path only checks whether sdev->handle is set. If device_link_add() fails after the handle has been acquired, the protocol device can still bind with a valid handle but without the dependency link to the SCMI parent. A concurrent parent unbind can then miss the child and tear down the SCMI instance while the child still holds a handle into it. If the protocol driver probe later fails, for example with -EPROBE_DEFER, the driver core emits BUS_NOTIFY_DRIVER_NOT_BOUND rather than BUS_NOTIFY_UNBOUND_DRIVER. The SCMI notifier only released the handle on BUS_NOTIFY_UNBOUND_DRIVER, so each failed protocol-device bind leaked the SCMI instance users refcount and left sdev->handle set after the failed probe. Make the link helper report failure and drop the acquired handle if the link cannot be created. Also handle BUS_NOTIFY_DRIVER_NOT_BOUND in the same cleanup path used for unbind so failed probes balance the earlier BUS_NOTIFY_BIND_DRIVER acquisition.	N/A	More Details
CVE-2026-93085	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Reject out of range DT protocol IDs SCMI protocol IDs carried in message headers are limited by MSG_PROTOCOL_ID_MASK. The DT parsing paths noticed protocol IDs outside that range, but only logged an error and then kept processing the invalid value. That lets a malformed 32-bit DT reg value reach helpers which take a u8 protocol ID, where it can be truncated and/or treated as a different protocol. For channel setup, two different out-of-range values can also be used as distinct IDR keys while aliasing the generated SCMI protocol identity. Skip DT protocol nodes whose reg value does not fit the SCMI protocol ID field before setting up channels or creating protocol devices.	N/A	More Details
CVE-2026-93086	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Avoid IDR updates while cleaning channels scmi_cleanup_channels() walks the TX/RX channel IDRs with idr_for_each() to free transport resources and destroy the dedicated transport devices before calling idr_destroy(). The destroy callback removed each entry from the same IDR being walked. That is not needed for this cleanup path, and it is unsafe because idr_for_each() has not advanced its radix-tree iterator while the callback is running. Removing the current entry from the callback can invalidate the iterator state. The callback also cannot be protected by rcu_read_lock(), because scmi_device_destroy() may sleep. Leave IDR teardown to the following idr_destroy() call and keep the callback limited to device destruction.	N/A	More Details
CVE-2026-93089	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Free transport channel on IDR failure If transport channel setup succeeds but the following IDR insertion fails, the error path destroys the transport device and frees the channel info without invoking the transport cleanup callback. Call chan_free() before destroying the device so transport specific resources such as IRQs, mailbox channels and mapped shared memory are released consistently with the normal teardown path.	N/A	More Details
CVE-2026-93090	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Clean up channels on setup failure scmi_channels_setup() can fail after the common BASE channel or earlier protocol channels have already been registered in the TX/RX IDRs. Route this failure through the existing channel cleanup label so the transport channels, transport devices and IDR state created before the failure are released before the probe error path frees the SCMI instance ID.	N/A	More Details
CVE-2026-93056	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_uac1_legacy: remove broken string configfs attributes The UAC1_STR_ATTRIBUTE macro defines configfs show/store handlers for the fn_play, fn_cap, and fn_cntl string options. The store function contains an inverted null check on the kstrndup() return value. This means every write attempt returns -ENOMEM on success and dereferences a NULL pointer on allocation failure. The attributes have been broken and unused for many years. Remove the UAC1_STR_ATTRIBUTE macro and the three attributes it generated. The internal defaults (FILE_PCM_PLAYBACK, FILE_PCM_CAPTURE, FILE_CONTROL) set in f_audio_alloc_inst() are unaffected.	N/A	More Details
CVE-2026-93078	In the Linux kernel, the following vulnerability has been resolved: cxl/features: Reject Set Features output buffer smaller than the header cxlctl_set_feature() sizes its output buffer from the user's fwctl_rpc.out_len but never checks it is large enough to hold even the fwctl_rpc_cxl_out header. With out_len == 0 , kvzalloc() returns ZERO_SIZE_PTR, which passes the !rpc_out check, the subsequent rpc_out->size = 0 then writes through the poison pointer. Reject requests whose output buffer can't hold the response header, before allocating. The Set Feature reply carries no payload, so the header is all that is required.	N/A	More Details
CVE-2026-93092	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Unregister device notifier before IDR teardown The requested-devices notifier looks up protocol fwnodes from the active_protocols IDR. During remove, unregister the notifier before releasing and destroying active_protocols so no notifier callback can race with the IDR teardown. Keep the bus notifier registered until after the protocol state is torn down, matching the existing remove ordering for SCMI bus users.	N/A	More Details
CVE-2026-93064	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: fix off-by-one in TXF key sanitiser iwl_mvm_frob_txf_key_iter() tracks the last matched byte position in loop variable 'i'. When a full key match is found (match == keylen), 'i' points at the last byte of the matched key. The memset start offset should therefore be i + 1 - keylen, not i - keylen; the current code zeroes one byte before the match and leaves the final key byte un-sanitised.	N/A	More Details
CVE-2026-93069	In the Linux kernel, the following vulnerability has been resolved: OPP: Fix cleanup ordering Commit 173e02d67494 ("OPP: Initialize scope-based pointers inline") added initialization for all pointers. In some cases, the ordering was changed so that *opp_table was initialized after *opp. This also changes the order of the registered cleanup functions. When the cleanup happens, this can cause use-after-free errors when the last reference is released and the release function _opp_kref_release tries to access the already freed opp->opp_table. Initialize *opp_table before *opp again to fix this and ensure the correct cleanup order.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix infinite loop in pcpu_freelist push with one possible CPU __pcpu_freelist_push() can loop forever when only one CPU is possible and an NMI re-enters pcpu_freelist_push() while the interrupted context holds that CPU's freelist lock. After the current-CPU fast path fails, the fallback loop walks cpu_possible_mask while skipping the current CPU. With CONFIG_SMP=n, or when an SMP kernel is limited to one possible CPU		

CVE-2026-89991	with nr_cpus=1 or possible_cpus=1, there are no other possible CPUs to examine. The loop therefore makes no lock acquisition attempt and can never make progress. The following stack was observed on a UP system: NMI context: pcpu_freelist_push free_htab_elem htab_map_delete_elem [perf-event BPF program] __perf_event_overflow perf_event_nmi_handler exc_nmi Interrupted context: __pcpu_freelist_push pcpu_freelist_push free_htab_elem htab_map_delete_elem [raw_tp/sys_enter BPF program] __bpf_trace_sys_enter do_syscall_64 raw_res_spin_lock() detects the same-CPU recursive acquisition and returns -EDEADLK, but the subsequent fallback loop has no candidate head on a system with one possible CPU. Restore the extra fallback head that existed before the rqspinlock conversion. Keep the current-CPU fast path, then try the other possible CPUs and finally the extra head. The additional head lets a push, which cannot fail without losing a preallocated element, make progress when the only per-CPU head is held by the interrupted context. Also check the extra head from the pop path so that nodes placed there can be reused.	N/A	More Details
CVE-2026-93066	In the Linux kernel, the following vulnerability has been resolved: x86/mm/pat: Take cpa_lock around large-page collapse Loading and unloading modules concurrently on several CPUs on a KASAN build, with a short delay injected at the CPA page-table lookup to widen the window, faults within minutes: BUG: KASAN: use-after-free in __change_page_attr+0x7cc/0x7e0 Write of size 8 at addr ffff888181139718 by task modprobe ... The buggy address belongs to the physical page: pfn:0x181139 ... page_type: f2(table) cpa_collapse_large_pages() rebuilds a leaf PMD from its 4K PTEs and frees the old PTE-table pages, while __change_page_attr() fetches a PTE pointer from a lockless lookup_address_in_pgd_attr() and writes it with set_pte_atomic() only later. When module text is served from a shared large ROX mapping the two run on the same PMD: CPU A (module load) CPU B (module finalize) ---------- ---------- execmem_make_temp_rw set_memory_nx __change_page_attr split 2M -> 4K table P kpte = &P[i] (lockless) execmem_restore_rox set_memory_rox (CPA_COLLAPSE) cpa_collapse_large_pages rebuild leaf PMD flush_tlb_all pagetable_free(P) set_pte_atomic(kpte, ...) -> writes into freed P P is a page-table page (page_type: table), reused at once, so the write corrupts whatever got the page next: a bad-pte or bad-page splat, or a fatal fault once P has been turned into read-only text. The flush_tlb_all() before the free does not close this: its IPI only serializes against page-table walkers that run with interrupts off (e.g. GUP-fast); the walk in __change_page_attr() runs with interrupts on, so nothing stops it from holding a stale pointer into P. Serialize the collapse - the PMD rebuild, TLB flush and PTE-table free - under cpa_lock, the same lock __change_page_attr() now takes unconditionally since commit ("x86/mm/pat: stop gating cpa_lock on debug_pagealloc_enabled()"), so a concurrent walker can no longer hold a pointer into a table the collapse is about to free.	N/A	More Details
CVE-2026-93072	In the Linux kernel, the following vulnerability has been resolved: irqchip/renesas-irqc: Fix generic interrupt chip leak on remove The driver allocates domain generic chips probe. However, on driver removal, the generic chips are not automatically freed when the interrupt domain is removed because the domain flags do not include IRQ_DOMAIN_FLAG_DESTROY_GC. This causes both the domain generic chips structure and the associated generic chips to be leaked. Additionally, the generic chips remain on the global list and may later be accessed by generic interrupt chip suspend, resume, or shutdown callbacks after the driver has been removed, potentially resulting in a use-after-free and kernel crash. Fix the resource leak by setting IRQ_DOMAIN_FLAG_DESTROY_GC on the interrupt domain; this lets the interrupt domain core automatically release all generic chips when irq_domain_remove() is invoked, removing the need for manual cleanup calls in error paths and remove callback.	N/A	More Details
CVE-2026-93073	In the Linux kernel, the following vulnerability has been resolved: dax: read holder_ops once in dax_holder_notify_failure() dax_holder_notify_failure() reads dax_dev->holder_ops twice without READ_ONCE() -- once for the NULL check and once for the indirect notify_failure() call. A concurrent fs_put_dax() can clear holder_ops between the two reads, so the check can observe a non-NULL pointer while the call dereferences NULL. (kill_dax() also clears holder_ops, but only after synchronize_srcu(), so it cannot race a reader that is inside dax_read_lock(); fs_put_dax() does no such synchronization.) Fetch holder_ops once into a local with READ_ONCE() so the NULL check and the indirect call observe the same value.	N/A	More Details
CVE-2026-93065	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: fix counter type in iwl_fwrt_dump_error_logs The loop counter 'count' was declared as u8 while num_pc is u32. If firmware advertises more than 255 PC entries the counter wraps back to zero and the loop never terminates potentially causing an infinite loop or reading past the allocated pc_data array. Change the declaration to u32 to match num_pc.	N/A	More Details
CVE-2026-89989	In the Linux kernel, the following vulnerability has been resolved: ima: Check for ERR_PTR from dentry_path() in validate_hash_algo() dentry_path() returns ERR_PTR(-ENAMETOOLONG) when the path exceeds the buffer. validate_hash_algo() passes the result straight to integrity_audit_msg() without checking. ERR_PTR is not NULL, so integrity_audit_message() sees a valid pointer and calls strlen() on it, which faults: BUG: unable to handle page fault for address: ffffffffffffffdc RIP: 0010:strlen+0x30/0xa0 Call Trace: audit_log_untrustedstring+0x19/0x30 integrity_audit_message+0x366/0x4f0 ima_inode_setxattr+0x512/0x5f0 Check for IS_ERR() and use NULL instead, which makes the audit message skip the name= field instead of crashing.	N/A	More Details
CVE-2026-93075	In the Linux kernel, the following vulnerability has been resolved: dax/fsdev: clear pgmap ops and owner on unbind fsdev_dax_probe() sets pgmap->ops = &fsdev_pagemap_ops and pgmap->owner = dev_dax, but nothing ever clears them. For a dynamic device the pgmap is devm-allocated and freed on unbind, so this is harmless. For a static device the pgmap is the shared, long-lived one owned by the dax bus (kill_dev_dax() only NULLs dev_dax->pgmap for the non-static case), and device.c's probe sets only pgmap->type, never clearing ops/owner. So after fsdev unbinds a static device the stale fsdev_pagemap_ops survives on the shared pgmap. If the device is then rebound to device_dax (MEMORY_DEVICE_GENERIC, which installs no ->memory_failure), or the fsdev_dax module is unloaded, a subsequent memory_failure on that pgmap dispatches through the stale -- and possibly freed -- handler. Register a devm action that clears pgmap->ops and pgmap->owner on unbind, symmetric with setting them at probe, so the pgmap carries no fsdev state once fsdev is detached.	N/A	More Details
CVE-2026-93058	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Only fini scheduler after successful init msm_ringbuffer_new() destroys a partially initialized ring through msm_ringbuffer_destroy() when an allocation or scheduler setup step fails. If drm_sched_init() fails before it finishes initializing the scheduler, the failure path still calls drm_sched_fini(). That teardown path assumes the scheduler work items, lists, and workqueue state were initialized. Track successful scheduler initialization and call drm_sched_fini() only after drm_sched_init() returned 0. This issue was found by a static analysis checker and confirmed by manual source review. Patchwork: https://patchwork.freedesktop.org/patch/738905/	N/A	More Details
CVE-2026-89993	In the Linux kernel, the following vulnerability has been resolved: dmaengine: dw-edma: Initialize IRQ data before requesting IRQs dw_edma_irq_request() passes struct dw_edma_irq to request_irq() before dw_edma_channel_setup() fills the back pointer. A shared interrupt can therefore enter the handler with dw_irq->dw still NULL, leading to a NULL pointer dereference. Set the back pointer before installing each handler.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: dax/fsdev: clear vmemmap_shift when binding static pgmap		

2026-93076	Clear pgmap->vmemmap_shift for static DAX devices. When rebinding a static device from device_dax (which may set vmemmap_shift based on alignment) to fsdev_dax, the stale vmemmap_shift persists on the shared pgmap. Explicitly zero it before devm_memremap_pages() so the vmemmap is built for order-0 folios as fsdev requires.	N/A	More Details
CVE-2026-93062	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: guard against division by zero in iwl_dbg_tlv_alloc_fragments Make sure we don't end-up with a num_frags = 0 situation. For that, check that the required size is not 0 and put a checker on num_frags as well.	N/A	More Details
CVE-2026-93061	In the Linux kernel, the following vulnerability has been resolved: gpu: host1x: Avoid stack over-read in debug output helpers host1x_debug_output() and host1x_debug_cont() used vsnprintf(), which returns the length the formatted string would have reached with an unbounded buffer. That return value was passed straight to o->fn as the number of bytes to emit. This could cause a read past end of the output buffer if a call to host1x_debug_* produced a string longer than 256 bytes. This only affected the debugfs files as the printk debug sink ignores the number of bytes. In practice, this is very unlikely to occur. Fix by switching to vscnprintf(), which returns the number of bytes actually written.	N/A	More Details
CVE-2026-93060	In the Linux kernel, the following vulnerability has been resolved: drm/msm/adreno: fix use after free on error path in a6xx_gpu_init() The a6xx_destroy() function frees "a6xx_gpu" and so "adreno_gpu" points to freed memory. Preserve the error code before freeing the memory to avoid a use after free. Patchwork: https://patchwork.freedesktop.org/patch/732275/	N/A	More Details
CVE-2026-93059	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Fix task_struct reference leak in recover_worker get_pid_task() increments the task reference count, but the corresponding put_task_struct() was missing in the else branch, leaking a reference on every GPU hang recovery. Patchwork: https://patchwork.freedesktop.org/patch/730662/	N/A	More Details
CVE-2026-93077	In the Linux kernel, the following vulnerability has been resolved: cxl/features: Clamp Get Feature output size to the remaining buffer cxl_get_feature() reads a feature in a loop but passes a fixed size_out as the output capacity every iteration. On the last partial iteration the buffer has less room left, so a device that returns more than asked can overflow feat_out. Use the per-iter size data_to_rd_size, which already tracks the remaining room, as the output capacity.	N/A	More Details
CVE-2026-93091	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Quiesce notifications before teardown scmi_notification_exit() clears and releases the notification instance, but transport callbacks can still deliver incoming notifications until the TX/RX channels are freed. During remove, an RX interrupt in that window can enter scmi_notify() while notification state is being torn down and then dereference freed memory. The same ordering exists on the probe error path after notification initialization. The notification late-init worker has a separate lifetime issue: protocol event registration queues ni->init_work on the system workqueue, so destroying ni->notify_wq does not drain that work. If the devres group is released while init_work is still pending or running, the late-init worker can dereference the freed notification instance. Quiesce the notification core before TX/RX channels are torn down, then clean up the channels before releasing the notification core resources. Use disable_work_sync() so future late-init queueing is rejected and any already queued or running late-init work has completed before channel teardown starts.	N/A	More Details
CVE-2026-93093	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Publish channel state before callbacks Transport setup can enable callbacks before the setup routine returns. mailbox_chan_setup() registers the mailbox client with mbox_request_channel(), and the mailbox controller startup path can enable interrupt delivery before SCMI mailbox channel state has been published. Similarly, smc_chan_setup() requests the optional A2P completion IRQ before the SMC transport has made its cinfo pointer visible. If a pending or spurious callback fires in those windows, the transport RX callback can dereference a NULL transport cinfo pointer. Publishing only the transport-private pointer is not sufficient either: an early callback can enter the SCMI core before scmi_chan_setup() has assigned cinfo->handle. The core derives scmi_info from cinfo->handle in the RX path, so a NULL handle can still fault even when the transport-private cinfo is valid. Assign cinfo->handle before invoking the transport setup callback. Publish the mailbox and SMC transport-private channel state before requesting the mailbox channels or IRQ, and clear the early-published pointers again on setup failure. Also unwind mailbox setup devres resources on failure so an optional RX setup error that is ignored by the core does not leave stale transport state behind.	N/A	More Details
CVE-2026-89975	In the Linux kernel, the following vulnerability has been resolved: nvme-fabrics: fix DHCHAP secret leak on parse failure nvmf_parse_options() duplicates dhchap_secret and dhchap_ctrl_secret with match_strdup() before validating the DHHC-1: representation. If validation fails, the parser returns -EINVAL before the temporary string in p is assigned to opts->dhchap_secret or opts->dhchap_ctrl_secret. nvmf_create_ctrl() subsequently frees opts, but nvmf_free_options() cannot release the unassigned temporary string. Each rejected option therefore leaks one allocation. This is easy to miss because valid secrets transfer ownership to opts and are freed normally, while the malformed-secret path still returns the expected -EINVAL to userspace. With CONFIG_NVME_HOST_AUTH enabled, the leak is reachable before the required-option checks and transport lookup. No NVMe-oF target or working transport connection is required; for example, repeatedly writing dhchap_secret=BAD or dhchap_ctrl_secret=BAD to /dev/nvme-fabrics deterministically takes the leaking parse path. Free the temporary string before leaving both validation error paths. Use kfree_sensitive() because the copied option may contain secret material even when its representation is rejected, matching the sensitive cleanup used for stored DHCHAP secrets.	N/A	More Details
CVE-2026-90004	In the Linux kernel, the following vulnerability has been resolved: mm/damon/core: handle region split failure in apply_min_nr_regions() damon_apply_min_nr_regions() repeatedly split each region until its size becomes small enough to meet the user-defined low limit of the number of regions. The loop assumes the split operation (damon_split_region_at()) will always succeed and create the new region. But the operation could silently fail for memory allocation failures, for example. If such failure happens and the region was the last region, the linked list-based next region fetching returns invalid pointer. As a result, invalid memory dereference and corruption could happen. Even if the corner case is handled, it imposes stress to the allocator by trying split regions for other targets. Fix the issue by breaking all the loops for any region split failure. This means there could be a min_nr_regions violation. It will only rarely happen since the allocation is arguably too small to fail. Even if it happens, it is only temporal. damon_apply_min_nr_regions() will be called again after the aggregation interval. The user impact of the issue should be minor, since the allocation is arguably too small to fail. But, it could still theoretically happen, and the consequence is very bad. This issue was discovered [1] by Sashiko.	N/A	More Details
CVE-2026-89981	In the Linux kernel, the following vulnerability has been resolved: arm64: Don't read GMID_EL1 when MTE is disabled __cpuinfo_store_cpu() gates the GMID_EL1 read on the raw ID_AA64PFR1_EL1, so it reads the register even when the kernel has disabled MTE (CONFIG_ARM64_MTE=n or arm64.nomte). KVM sets HCR_EL2.TID5 in that case, and pKVM injects an UNDEF the host cannot handle: Internal error: Oops - Undefined instruction: 0000000002000000 [#1] SMP pc : __cpuinfo_store_cpu+0xf4/0x264 Kernel panic - not syncing: Attempted to kill the idle task! Only pKVM reaches it, and only after a CPU is offlined and brought back online: its CPU_ON relay sets the host HCR before the CPU enters EL1, while plain nVHE	N/A	More Details

	sets it at CPUHP_AP_KVM_ONLINE. Gate the read on the CPU's own ID_AA64PFR1_EL1 with the command-line override applied, and on CONFIG_ARM64_MTE, which no register reflects. The boot CPU stores its registers before init_cpu_features() strips an unsafe override, so clamp against the hardware value here too.		
CVE-2026-89978	In the Linux kernel, the following vulnerability has been resolved: accel/amdxdna: return early from a zero-length flush SYNC_BO does not constrain its size, so a request for zero bytes reaches drm_clflush_virt_range(), which ends with an unconditional clflushopt(end - 1). For an empty range that is the byte before the mapping, and abo->mem.kva comes from vmap(), so the access lands in the guard page below the vmalloc area and faults: BUG: unable to handle page fault for address: ffffd16fbbc70fff #PF: supervisor read access in kernel mode Oops: Oops: 0000 [#1] SMP NOPTI CPU: 7 UID: 1000 Comm: sync_bo_probe RIP: 0010:drm_clflush_virt_range+0x3c/0x70 Call Trace: amdxdna_drm_sync_bo_ioctl+0x124/0x430 [amdxdna] drm_ioctl+0x301/0x4c0 __x64_sys_ioctl+0x115/0x2f0 do_syscall_64+0xa6/0x3d0 Any process that can open the render node can do this. Reproduced 3 of 3 times on a Strix Point NPU (1022:17f0), by calling SYNC_BO with size 0 on an AMDXDNA_BO_SHARE object. The import arm takes the same request but flushes the whole scatterlist, so it survives it. Nothing needs flushing for an empty range, so answer before choosing a path.	N/A	More Details
CVE-2026-93113	In the Linux kernel, the following vulnerability has been resolved: clk: qcom: camcc-sc8280xp: unregister CAMCC_GDSC_CLK With the introduction of sync_state support in the clk and pmdomain subsystems, the following warning happens when the unused clocks are shutdown in camcc-sc8280xp: [15.408367] titan_top_gdsc status stuck at 'on' [15.408429] WARNING: drivers/clk/qcom/gdsc.c:178 at gdsc_toggle_logic+0x14c/0x160, CPU#2: kworker/u32:1/14 [15.408462] Modules linked in: bnep vfat fat ath11k_pci(+) ath11k mac80211 cfg80211 mhi libarc4 snd_soc_wcd938x snd_soc_wcd938x_sdw snd_soc_wcd_classh hci_uart snd_soc_wcd_common snd_soc_sc8280xp soundwire_qcom snd_soc_wcd_mbhc snd_soc_qcom_sdw slimbus snd_soc_qcom_common regmap_sdw btqca btrtl qcom_camss soundwire_bus btbcm btintel snd_soc_sdca snd_soc_lpass_wsa_macro bluetooth snd_soc_lpass_tx_macro snd_soc_lpass_va_macro snd_soc_lpass_rx_macro snd_soc_hdmi_codec snd_soc_lpass_macro_common videobuf2_dma_sg ov5675 v4l2_fwnode videobuf2_memops qcom_spmi_adc5 snd_soc_core qcom_spmi_adc_tm5 videobuf2_v4l2 snd_seq snd_seq_device videobuf2_common v4l2_async qcom_vadc_common qcom_spmi_temp_alarm pm8941_pwrkey industrialio videodev snd_compress rfkill ac97_bus snd_pcm_dmaengine qcom_tsens mc qcom_edac snd_pcm pci_pwrctrl_pwrseq qcom_cpufreq_hw snd_timer snd qcomtee soundcore tee leds_gpio joydev binfmt_misc zram lz4hc_compress governor_simpleondemand panel_edp msm xhci_plat_hcd nvme nvme_core dwc3 qcom_pm8008_regulator [15.408688] ucsi_glink nvme_keyring nvme_auth pmic_glink_altmode udc_core typec_ucsi aux_hpd_bridge qcom_battmgr ulpi ubwc_config socinfo ocmem drm_gpuvm qcom_q6v5_pas drm_exec qcom_pil_info leds_qcom_lpg gpu_sched led_class_multicolor rtc_pm8xxx qcom_pbs qcom_common drm_display_helper qcom_pon qcom_glink_smem qcom_glink ghash_ce pwrseq_qcom_wcn gpio_sbu_mux qcom_stats phy_qcom_qmp_combo qcom_q6v5 gf128mul cec dispcc_sc8280xp phy_qcom_edp camcc_sc8280xp i2c_qcom_cci qcom_sysmon drm_dp_aux_bus mdt_loader aux_bridge qcom_pm8008 i2c_hid_of_elan dwc3_qcom_legacy llcc_qcom icc_bwmon gpi typec qcom_refgen_regulator phy_qcom_qmp_usb nvmem_qfprom qcom_ipcc phy_qcom_snps_femto_v2 gpucc_sc8280xp pinctrl_sc8280xp_lpass_lpi qcom_hwspinlock pinctrl_lpass_lpi lpasscc_sc8280xp qrtr qcom_aoss pmic_glink pdr_interface phy_qcom_qmp_pcie qcom_smd qcom_pdr_msg icc_osm_l3 qcom_wdt qmi_helpers qcom_rng smp2p rpmsg_core gpio_keys pwm_bl smem hid_multitouch fuse i2c_dev [15.408928] CPU: 2 UID: 0 PID: 14 Comm: kworker/u32:1 Not tainted 7.1.0+ #2 PREEMPT(lazy) [15.408937] Hardware name: LENOVO 21BX0016US/21BX0016US, BIOS N3HET88W (1.60) 03/14/2024 [15.408942] Workqueue: pm pm_runtime_work [15.408959] pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [15.408967] pc : gdsc_toggle_logic+0x14c/0x160 [15.408978] lr : gdsc_toggle_logic+0x14c/0x160 [15.408987] sp : ffff8000800f3b40 [15.408991] x29: ffff8000800f3b40 x28: 0000000000000000 x27: 0000000000000000 [15.409003] x26: 0000000000000000 x25: 0000000000000000 x24: 0000000000000000 [15.409014] x23: 0000000000000000 x22: 0000000000000001 x21: ffffa33f298fca88 [15.409024] x20: 0000000000000000 x19: ffffa33f298fc5b0 x18: 00cd15db75dacefd [15.409035] x17: 000000040044ffff x16: ffffa33f3b1a3d88 x15: 726f776b80000002 [15.409045] x14: ffffffffffffffff x13: 0000000000000028 x12: 0101010101010101 [15.409056] x11: 7f7f7f7f7f7f7f7f x10: fefeff3039313274 x9 : ffffa33f3a5edafc [15.409067] x8 : ffff8000800f3780 x7 : 0000000000000001 x6 : 0000000000000001 [15.409078] x5 : ffff000bf3ca1288 x4 : 0000000000000000 x3 : ffff5cccb6a3f000 [15.409088] x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff000080ae0000 [15.409098] Call trace: [15.409103] gdsc_toggle_logic+0x14c/0x160 (P) [15.409115] gdsc_disable+0x4c/0x190 [15.409126] _genp ---truncated---	N/A	More Details
CVE-2026-8462	SQL injection in ClickHouse-backed meter definitions in OpenMeter OpenMeter before v1.0.0-beta.228 on all platforms allows a remote unauthenticated attacker to access or modify metering event data, and potentially cause denial of service, via crafted user-controlled JSONPath values submitted to meters API.	N/A	More Details
CVE-2026-93114	In the Linux kernel, the following vulnerability has been resolved: platform/surface: acpi-notify: Check ACPI companion before use Since every platform driver can be forced to match a device that doesn't match its list of device IDs because of device_match_driver_override(), platform drivers that rely on the existence of a device's ACPI companion object should verify its presence. san_probe() dereferences the result of ACPI_COMPANION() when installing the GSBUS address space handler, so force-binding the driver to a device without an ACPI companion leads to a NULL pointer dereference. The dereference was introduced when the probe function was switched from ACPI_HANDLE() to ACPI_COMPANION(). Check the ACPI companion against NULL and return -ENODEV when it is missing, like commit e4865a56d013 ("ACPI: driver: Check ACPI_COMPANION() against NULL during probe") does for the core ACPI platform drivers.	N/A	More Details
CVE-2026-93115	In the Linux kernel, the following vulnerability has been resolved: platform/mellanox: mlxbf-pmc: Check ACPI_COMPANION() against NULL Every platform driver can be forced to match a device that doesn't match its list of device IDs because of device_match_driver_override(), so platform drivers that rely on the existence of a device's ACPI companion object need to verify its presence. mlxbf_pmc_probe() passes the result of ACPI_COMPANION() to acpi_device_hid(), which dereferences it, so force-binding the driver to a device without an ACPI companion leads to a NULL pointer dereference. Accordingly, add a requisite ACPI_COMPANION() check against NULL to the mlxbf-pmc driver and return -ENODEV when the companion is missing.	N/A	More Details
CVE-2026-89977	In the Linux kernel, the following vulnerability has been resolved: accel/ethosu: check MMIO mapping errors in probe devm_platform_ioremap_resource() returns an error pointer when the register resource cannot be mapped. ethosu_probe() stores it and continues until initialization dereferences it through MMIO accessors. Return the mapping error before initializing the device.	N/A	More Details
CVE-2026-93117	In the Linux kernel, the following vulnerability has been resolved: usb: fix UAF when probe runs concurrent to dyn ID removal Dynamic IDs are only guaranteed to be valid when usb_dynids_lock is held, as remove_id_store can free the node. Thus, make a copy in usb_probe_interface. Clarify the documentation that the id parameter is only valid during the probe. USB serial has the	N/A	More Details

	same pattern, but it does not need fixing as the IDs cannot be removed via sysfs.		
CVE-2026-93109	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Drain RCU callbacks during module teardown devx_free_subscription() can remain queued after the last DevX event file drops its module reference or an auxiliary driver detaches its devices. mlx5_ib can then unload before the callback runs. Registration error unwind has the same risk because driver registration can attach existing devices before failing. Wait after all drivers have stopped.	N/A	More Details
CVE-2026-93044	In the Linux kernel, the following vulnerability has been resolved: bpf: Disallow interpreter fallback for arena-related insns Since the interpreter does not support the arena-related insns, interpreter fallback should not be allowed for these insns in core.c::__bpf_prog_select_runtime(). Currently, when the interpreter executes the arena ST/LDX/STX insns, it would hit the BUG_ON() in ___bpf_prog_run() at run time. [2.579196] BPF interpreter: unknown opcode a2 (imm: 0x0) [2.579998] ------------[cut here]------------ [2.580652] kernel BUG at kernel/bpf/core.c:2349! [2.581314] Oops: invalid opcode: 0000 [#1] SMP PTI Set jit_required as true when arena map is used in the prog to disallow interpreter fallback for arena-related insns.	N/A	More Details
CVE-2026-93118	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: aspeed_udc: check endpoint DMA allocation ast_udc_probe() allocates a coherent DMA buffer used as the backing store for endpoint buffers. ast_udc_init_ep() derives per-endpoint buffer pointers from udc->ep0_buf, so a failed allocation is dereferenced during probe. Check the allocation before endpoint setup. The existing probe error path called ast_udc_remove(), which unregisters the gadget unconditionally and is not safe before usb_add_gadget_udc() succeeds. Add a local cleanup helper for probe failures so pre-registration failures only unwind the resources that were actually initialized. This was found by a local static analysis checker for unchecked allocator returns while scanning Linux 6.16. The change was checked by applying it to current mainline and by running checkpatch. I do not have access to Aspeed UDC hardware, so no runtime testing was performed.	N/A	More Details
CVE-2026-93043	In the Linux kernel, the following vulnerability has been resolved: bpf: Disallow interpreter fallback for gotox insn The interpreter does not recognize the BPF_JMP|BPF_JA|BPF_X insn, which is used for insn_array map. Thereafter, it would hit the BUG_ON() in ___bpf_prog_run() at run time. [2.563726] BPF interpreter: unknown opcode 0d (imm: 0x0) [2.564557] ------------[cut here]------------ [2.565206] kernel BUG at kernel/bpf/core.c:2349! [2.565882] Oops: invalid opcode: 0000 [#1] SMP PTI Set jit_required as true when insn_array map is used in the prog in order to disallow interpreter fallback for gotox insn in core.c::__bpf_prog_select_runtime().	N/A	More Details
CVE-2026-93119	In the Linux kernel, the following vulnerability has been resolved: usb: ljca: bound bank_num in ljca_enumerate_gpio() ljca_enumerate_gpio() reads desc->bank_num from the device and loops valid_pin[i] = get_unaligned_le32(...) for i < bank_num. valid_pin[] holds only LJCA_MAX_GPIO_NUM / 32 = 2 entries. Two checks run before the loop. The reply length must match struct_size(desc, bank_desc, bank_num). The product pins_per_bank * bank_num must not exceed LJCA_MAX_GPIO_NUM. Neither one bounds bank_num against the size of valid_pin[]. The reply is capped at LJCA_MAX_PAYLOAD_SIZE (60) bytes, so the struct_size check limits bank_num to 9. A device that reports bank_num 9 with pins_per_bank 7 still passes both checks. gpio_num is 63 and the reply is 56 bytes. The loop then writes nine u32 into the two entry array and overruns valid_pin[] on the stack. A broken or malicious LJCA device can therefore overflow the stack. Reject a bank_num that does not fit valid_pin[].	N/A	More Details
CVE-2026-93120	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: configfs: fix out-of-bounds read of qw_sign os_desc_qw_sign_show() passes OS_STRING_QW_SIGN_LEN as the input length to utf16s_to_utf8s(), but that argument counts UTF-16 code units while OS_STRING_QW_SIGN_LEN (14) is the byte size of qw_sign[]. The array holds only OS_STRING_QW_SIGN_LEN / 2 (7) code units, so the conversion reads up to 7 units (14 bytes) past the end of qw_sign[] into the following members of struct gadget_info when the stored signature fills the array without a NUL terminator, exposing those bytes through the configfs attribute. The store path halves the count for its input bound but passes the full byte count as the utf8s_to_utf16s() output limit; use the destination code-unit count in both directions.	N/A	More Details
CVE-2026-90005	In the Linux kernel, the following vulnerability has been resolved: samples/damon/wsse: handle damon_start() failure Patch series "samples/damon: handle damon_{start,stop}() failures". All DAMON sample modules are not correctly handling failures from damon_start(). Among those, mtier also has an additional problem for handling of damon_stop() failures. wsse and prcl also have a problem in their damon_call() failure handling. As a result, memory leaks, next DAMON operation disruptions, and use-after-free can happen. Fix those. Note that only the damon_start() failure caused issues can reliably be reproduced. Reproducing those issues require the admin permission, though. This patch (of 6): damon_sample_wsse_start() callers assume it will clean up resources when it fails. And the function does the cleanup for context buildup failures. However, it is not doing the cleanup for damon_start() failure. As a result, when damon_start() fails, it leaks the memory for DAMON context. Free the context in case of the failure to fix the issues. Note that the issue can reliably be reproduced because the module calls damon_start() in the exclusive mode. For example, $ sudo damo start $ echo $$ | sudo tee /sys/module/damon_sample_wsse/parameters/target_pid $ echo Y | sudo tee /sys/module/damon_sample_wsse/parameters/enabled $ sudo cat /proc/allocinfo | grep damon_new_ctx Because the first command is running another DAMON instance, the third command fails the damon_start() call because the new DAMON instance cannot exclusively run. And without this fix, by repeating the third and the fourth commands above, we can show the memory consumption is only increasing due to the leaks. It requires the sudo permission though. The issue was discovered [1] by Sashiko.	N/A	More Details
CVE-2026-89976	In the Linux kernel, the following vulnerability has been resolved: accel/ethosu: fix job completion fence cleanup ethosu_ioctl_submit_job() allocates done_fence before validating buffer handles. Errors after allocation call ethosu_job_err_cleanup(), which frees the job but leaks the uninitialized fence. A scheduler dependency error also lets ethosu_job_run() return before dma_fence_init(). Normal cleanup then passes a zeroed refcount to dma_fence_put(). Release done_fence in the common cleanup path and use dma_fence_was_initialized() to distinguish initialized fences from raw allocations. [robh: also fix goto]	N/A	More Details
CVE-2026-93110	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Wait for RCU callbacks before unloading ib_core put_gid_ndev() is queued with call_rcu() and implemented in ib_core. Stopping the workqueues does not drain callbacks already queued, so RCU could invoke it after the module code has been unloaded. synchronize_rcu() does not wait for callbacks. Wait for them after all producers have stopped.	N/A	More Details
CVE-2026-93108	In the Linux kernel, the following vulnerability has been resolved: RDMA/ipoib: Drain RCU callbacks during module teardown IPoIB reclamation completions can be signaled from inside an RCU callback. Teardown can wake before the callback returns and unload ib_ipoib while its code is still executing. Client registration failure can also remove already-added devices and queue callbacks. Wait after client and workqueue teardown.	N/A	More Details

CVE-2026-93094	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix dp_link_peer dangling references on AP vdev rollback ath12k_mac_vdev_create() for an AP vdev creates the bss self-peer via ath12k_peer_create(), which finishes by calling ath12k_dp_link_peer_assign() to publish the dp_link_peer in the dp_hw->dp_peers[peerid_index] RCU table, in the dp_peer's link_peers[] array, and in the per-addr rhashtable. If a step after ath12k_peer_create() fails the function jumps to err_peer_del, which open-codes a WMI peer_delete and waits for the unmap / delete_resp events. The wait_for_peer_delete_done() path relies on ath12k_dp_link_peer_unmap_event() freeing the dp_link_peer when the unmap arrives, but err_peer_del never calls ath12k_dp_link_peer_unassign() first. The published references in the dp_hw RCU table, dp_peer->link_peers[] and the rhashtable are left pointing at the dp_link_peer that unmap_event then frees, producing dangling pointers and use-after-free on subsequent lookups. Replace the open-coded sequence with a call to ath12k_peer_delete(), which already does ath12k_dp_link_peer_unassign() before sending the WMI command. This drops the published references before the dp_link_peer is freed, in the same order as the normal teardown path in ath12k_mac_remove_link_interface(). Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.1.c7-00108-QCAHMTSWPL_V1.0_V2.0_SILICONZ_UPSTREAM-3	N/A	More Details
CVE-2026-93099	In the Linux kernel, the following vulnerability has been resolved: fs/resctrl: Fix UAF from worker threads when domains are removed The mbm_handle_overflow() and cqm_handle_limbo() workers read event counters and may sleep while doing so. They are scheduled via delayed_work embedded in struct rdt_l3_mon_domain. Architecture allocates and frees these domains from CPU hotplug callbacks under cpus_write_lock(), and the workers acquire cpus_read_lock() to keep the domain alive across their access. A use-after-free can occur when a worker is blocked waiting for cpus_read_lock() while the hotplug core holds cpus_write_lock(): the architecture frees the rdt_l3_mon_domain that contains the worker's work_struct. When the worker unblocks, the container_of() it performs on the embedded work pointer dereferences freed memory. Drop cpus_read_lock() from the workers and instead drain pending and in-flight work synchronously before the architecture can free the domain. Since architecture offlines the domain under cpus_write_lock() after it has been unlinked from the RCU list and a grace period has elapsed, no new work can be scheduled. The cancel only needs to wait out existing work. Drop rdtgroup_mutex during CPU offline around cancel_delayed_work_sync() so that a worker waiting on the mutex can complete before re-pinning the work on a different CPU. When offlining a CPU the architecture may iterate over resources in any order. For example, the MBA control domain may be offlined before or after a corresponding L3 monitor domain. Ensure that resctrl fs cancels the workers no matter what order the architecture offlines the domains.	N/A	More Details
CVE-2026-93050	In the Linux kernel, the following vulnerability has been resolved: ipack: ipoctal: fix UAF, null-ptr-deref, and use-after-free in cleanup on remove Three issues arise when the device is removed while a tty session is still active: 1. UAF of struct ipoctal: the remove callback frees ipoctal via kfree() while tty ops may still access it. Fix by introducing kref-based lifetime management — kref is taken in install() when a tty is opened and released in cleanup() when the tty is finally destroyed; remove() uses kref_put() instead of kfree(). 2. NULL dereference in ipoctal_write_tty(): __ipoctal_remove() frees xmit_buf via tty_port_free_xmit_buf() while a userspace process may still hold the tty fd and call write(). Fix by checking for NULL xmit_buf in ipoctal_write_tty(). 3. UAF in ipoctal_cleanup(): ipack_put_carrier(ipoctal->dev) dereferences ipoctal->dev after the ipack_device has been freed by ipack_device_del(). Fix by caching ipoctal->carrier_owner during probe() and calling module_put() on the cached pointer directly in cleanup(), avoiding any access to ipoctal->dev. Also introduce a "removed" flag in struct ipoctal, set at the start of __ipoctal_remove(), and checked in every tty op that accesses hardware resources (port_activate, write_tty, set_termios, hangup, shutdown). This prevents page faults when devm_ioremap() regions are unmapped after remove() returns.	N/A	More Details
CVE-2026-93049	In the Linux kernel, the following vulnerability has been resolved: mtd: mtdswap: Avoid freeing registered blktrans device twice In mtdswap_add_mtd(), debugfs setup failure after successful blktrans registration can free mbd_dev twice. add_mtd_blktrans_dev() initializes the blktrans device reference and publishes the disk. Once that succeeds, del_mtd_blktrans_dev() tears the disk down and drops the blktrans reference; when that reference reaches zero, blktrans_dev_release() frees the mtd_blktrans_dev. The debugfs failure path called del_mtd_blktrans_dev(mbd_dev), then fell through the common cleanup label and called kfree(mbd_dev) again. Clear the local pointer after deregistration so the common cleanup can still release the mtdswap state without freeing the blktrans object twice. This issue was found by a static analysis checker and confirmed by manual source review.	N/A	More Details
CVE-2026-93048	In the Linux kernel, the following vulnerability has been resolved: mtd: part: reject MTDPART_OFS_RETAIN in mtd_add_partition() mtd_add_partition() does not reject the special offset value MTDPART_OFS_RETAIN (-3), which leads to a WARN_ON in add_mtd_device() when called through the BLKPG ioctl on NAND devices. The RETAIN value depends on cur_offset being the end of the previous partition, but in the dynamic partition path cur_offset equals the offset argument itself, causing undefined behavior. Commit 5daa7b21496a ("mtd: prepare partition add and del functions for ioctl requests") introduced mtd_add_partition() and correctly rejected MTDPART_OFS_APPEND (-1) and MTDPART_OFS_NXTBLK (-2), since those special offsets rely on cur_offset tracking the previous partition's end. However, commit 1a31368bf92e ("mtd: add a flags for partitions which should just leave smth. after them") later added MTDPART_OFS_RETAIN (-3) for the static partition table path without updating mtd_add_partition() to also reject this value. With offset=-3 passed via BLKPG, the RETAIN size calculation in allocate_partition() underflows (parent_size - 0xFFFFFFFFFFFFFFFD = parent_size + 3). If the underflow result does not appear to leave enough space, allocate_partition() jumps to out_register via goto, skipping erasesize initialization. This results in erasesize=0, which triggers: WARN_ON((!mtd->erasesize || !master->_erase) && !(mtd->flags & MTD_NO_ERASE)) in add_mtd_device(). If the underflow result appears to leave enough space, a bogus partition size is calculated, but the "out of reach" sanity check catches the invalid offset and creates a disabled empty partition (offset=0, size=0) instead of returning an error. Fix this by adding MTDPART_OFS_RETAIN to the rejection list in mtd_add_partition(), consistent with the existing handling of APPEND and NXTBLK.	N/A	More Details
CVE-2026-89984	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: Fix kernel address leakages in LBR stack Before Arch LBR gained CPL filtering support, a user-only branch stack could still contain kernel addresses. As a result, kernel branch records may be exposed to user space even when PERF_SAMPLE_BRANCH_USER is requested. For example, on Intel Tiger Lake, the following command can still report SYSRET/ERET entries with kernel-space from addresses: $ ./perf record -e cycles:p -o - --branch-filter any,save_type,u -- \ ./perf bench syscall basic --loop 1000 | \ ./perf script -i - --fields brstack|tr ' ' '\n'| \ grep -E '0x[89a-f][0-9a-f]{15}' Total time: 0.000 [sec] 0.219000 usecs/op 4,566,210 ops/sec [perf record: Woken up 1 times to write data] [perf record: Captured and wrote 0.551 MB -] 0xffffffff93c001c8/0x7f12a2b1d647/P/-/-/16959/SYSRET/- 0xffffffff93c001c8/0x7f12a2b1d5c2/P/-/-/17535/SYSRET/- 0xffffffff93c01928/0x7f12a2861000/P/-/-/6719/ERET/- 0xffffffff93c01928/0x7f12a297a000/P/-/-/8575/ERET/- The problem is that intel_pmu_lbr_filter() does not fully validate the privilege level of sampled entries. It filters some mismatches based on the branch type and the to address, but it does not reject entries whose from address violates the requested branch privilege filter. Fix this by extending software filtering to validate both from and to addresses against br_sel. Any LBR entry contains kernel address does not match the requested user	N/A	More Details

	filter is dropped. This prevents kernel addresses from appearing in user-only branch stacks.		
CVE-2026-93096	In the Linux kernel, the following vulnerability has been resolved: cxl/features: Serialize multi-part Get/Set Feature transfers A Get or Set Feature payload larger than the mailbox payload size is split into several mailbox commands. mbox_mutex only serializes individual mailbox commands and is dropped between iterations of these loops. Nothing serializes the multi-part transfer as a whole. cxl_get_feature() and cxl_set_feature() are reachable concurrently from fwctl (per-fd RPCs run under a read-held registration lock) and from the EDAC scrub/ECS/repair paths, so two transfers to the same mailbox can interleave their parts and corrupt the device's transfer context. Add a per-mailbox feat_mutex and hold it across the whole transfer in both functions. It nests outside mbox_mutex (which is taken inside cxl_internal_send_cmd()), and is taken nowhere else, so no lock-ordering inversion is introduced.	N/A	More Details
CVE-2026-93097	In the Linux kernel, the following vulnerability has been resolved: cxl/mbox: Break poison list loop on an empty payload A device that returns count == 0 with CXL_POISON_FLAG_MORE set on every iteration never advances nr_records, so the max_errors guard never trips and the do/while loops forever while holding poison.mutex. That hangs the sysfs-triggered scan thread and blocks all subsequent poison operations on the device. The existing "Protect against an uncleared _FLAG_MORE" guard was intended to bound a misbehaving device but does not cover the count == 0 case. Stop the loop on an empty payload so a malfunctioning or malicious device cannot wedge the poison scan.	N/A	More Details
CVE-2026-93098	In the Linux kernel, the following vulnerability has been resolved: rpmsg: glink: fix deadlock in endpoint destroy during driver detach During driver detach, the device core holds the device mutex throughout the driver's remove callback chain. When the rpmsg endpoint is destroyed as part of that teardown, the GLINK endpoint destroy implementation attempts to unregister the underlying rpmsg device. That unregistration calls device_del(), which tries to re-acquire the same device mutex already held higher up the stack, causing rmmod to hang indefinitely. The deadlock manifests with the following call chain: [<0>] device_del+0x44/0x414 <- tries to acquire same mutex [<0>] device_unregister+0x18/0x34 [<0>] rpmsg_unregister_device+0x28/0x4c [<0>] qcom_glink_remove_rpmsg_device+0x70/0xc0 [<0>] qcom_glink_destroy_ept+0x58/0xbc [<0>] rpmsg_dev_remove+0x50/0x60 [<0>] device_remove+0x4c/0x80 [<0>] device_release_driver_internal+0x1cc/0x228 <- acquires device mutex [<0>] driver_detach+0x4c/0x98 [<0>] bus_remove_driver+0x6c/0xbc [<0>] driver_unregister+0x30/0x60 [<0>] unregister_rpmsg_driver+0x10/0x1c [<0>] fastrpc_exit+0x28/0x38 [fastrpc] [<0>] __arm64_sys_delete_module+0x1b8/0x294 [<0>] invoke_syscall+0x48/0x10c [<0>] el0_svc_common.constprop.0+0xc0/0xe0 [<0>] do_el0_svc+0x1c/0x28 [<0>] el0_svc+0x34/0x108 [<0>] el0t_64_sync_handler+0xa0/0xe4 [<0>] el0t_64_sync+0x198/0x19c The rpmsg device unregistration inside endpoint destroy is redundant. In both contexts where endpoint destruction is triggered: - Driver detach path: the driver core already tears down the rpmsg device. - Channel close path: the rpmsg device is already unregistered before endpoint destruction is reached. Remove the redundant unregistration to fix the deadlock.	N/A	More Details
CVE-2026-93100	In the Linux kernel, the following vulnerability has been resolved: fs/resctrl: Prevent use-after-free in rdtgroup_kn_put() A struct rdtgroup is reference counted via rdtgroup::waitcount. Callers that need the structure to remain valid across a sleep (while waiting on acquiring rdtgroup_mutex) take a reference with rdtgroup_kn_get() and release it with rdtgroup_kn_put(). The release path is intended to serve as the fallback freer: if the count drops to zero and the group has already been marked RDT_DELETED, rdtgroup_kn_put() frees the structure. The bulk teardown paths free_all_child_rdtgrp() and rmdir_all_sub() resulting from a resctrl directory remove or resctrl fs unmount act as the primary freer: they hold rdtgroup_mutex and free each rdtgroup whose waitcount is zero, otherwise they set RDT_DELETED and leave the freeing to the last waiter. These two freers race. rdtgroup_kn_put() commits waitcount == 0 with atomic_dec_and_test() outside rdtgroup_mutex, then reads rdtgroup::flags. Between those two operations a concurrent caller of free_all_child_rdtgrp() or rmdir_all_sub() (which holds the mutex) can observe waitcount == 0 via atomic_read(), call rdtgroup_remove(), and kfree() the structure. The subsequent read of rdtgroup::flags in rdtgroup_kn_put() is then a use-after-free, and the structure may even be freed twice if the freed memory happens to satisfy the RDT_DELETED flag check. Replace the bare atomic_dec_and_test() with atomic_dec_and_mutex_lock() so that the decrement-to-zero takes rdtgroup_mutex before the count becomes globally visible. The inspection of rdtgroup::flags then runs under the same mutex held by the bulk freers, making the two paths mutually exclusive. The common case where the count does not reach zero remains lock-free. Defer kernfs_unbreak_active_protection() until after the mutex is dropped since kernfs active protections functionally wrap rdtgroup_mutex. Remove resource group, which in turn drops its kernfs reference, after kernfs protection is restored. [bp: Split the commit messsages into smaller, easier-parseable paragraphs.]	N/A	More Details
CVE-2026-89982	In the Linux kernel, the following vulnerability has been resolved: i2c: mux: Fix channel node leak on adapter add failure i2c_mux_add_adapter() takes a reference to the Device Tree channel node before registering the new adapter. If adapter registration fails, the error path frees the private data without dropping that reference. Release the channel node before freeing the private data.	N/A	More Details
CVE-2026-93101	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-async: Unregister sub-device if asc_list is empty When my em28xx USB device that uses the i2c tvp5150 driver is disconnected, it crashes. The cause is that the tvp5150 i2c module uses v4l2_async, but the em28xx driver does not since it predates v4l2_async. In that corner case sd->asc_list is empty, so v4l2_async_unregister_subdev() never calls v4l2_device_unregister_subdev(). Modify the code so that, if sd->asc_list is empty, v4l2_device_unregister_subdev() is still called.	N/A	More Details
CVE-2026-93102	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Free RX data on late probe failure hfi1_init_dd() allocates the shared AIP/VNIC RX support before returning. If hfi1_init() or hfi1_register_ib_device() later fails, init_one() tears down the device data without calling hfi1_free_rx(). This leaks netdev_rx and its dummy netdev. Free the RX support after IB unregistration and before postinit_cleanup(), as done on normal device removal.	N/A	More Details
CVE-2026-93103	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Preserve unit 0 on allocation failure hfi1_free_devdata() assumes that the device was inserted into the unit table and unconditionally erases dd->unit. If xa_alloc_irq() fails, the zero-initialized unit remains zero, so full cleanup can remove an unrelated device from index 0. Release only the rdmavt allocation and return immediately while the unit table has not acquired the device.	N/A	More Details
CVE-2026-93104	In the Linux kernel, the following vulnerability has been resolved: RDMA/rvt: Return NULL after port allocation failure rvt_alloc_device() deallocates the IB device when its port array cannot be allocated but then returns the pointer to the released allocation. Callers treat any non-NULL value as valid and dereference it, resulting in a use-after-free. Return NULL immediately after deallocation so callers can propagate the allocation failure.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: i2c: core: fix debugfs UAF on adapter removal		

CVE-2026-89983	i2c_del_adapter() frees the adapter's debugfs directory before it unregisters the adapter device, but the new_device sysfs attribute stays writable until device_del(). A write racing with removal still reaches i2c_device_probe(), which passes the freed adap->debugfs to debugfs_create_dir() as the new client's parent: BUG: KASAN: slab-use-after-free in lookup_noperm_common+0x407/0x430 Read of size 4 at addr ffff88803ef87810 by task syz.0.61/6090 lookup_noperm_common+0x407/0x430 simple_start_creating+0x9c/0x110 debugfs_start_creating+0xdb/0x1a0 debugfs_create_dir+0x24/0x350 i2c_device_probe+0x814/0xbf0 It's technically possible to create a client after i2c_deregister_clients has run. That client will never be unregistered and make wait_for_completion hang. Close the window by removing the new_device attribute at the start of i2c_del_adapter(). device_remove_file() will drain any clients left.	N/A	More Details
CVE-2026-93106	In the Linux kernel, the following vulnerability has been resolved: crash_dump: release keyring reference at the correct time restore_dm_crypt_keys_to_thread_keyring() gets a reference to the user keyring before restoring the saved dm-crypt keys. The same keyring reference is then passed to add_key_to_keyring() for each saved key, but add_key_to_keyring() drops that reference on every call. This is only balanced when exactly one key is restored. With multiple keys, the keyring reference is dropped too many times and may trigger a refcount underflow or use-after-free. When more than five keys are restored, a refcount underflow/use-after-free warning can be triggered. The early error paths after lookup_user_key() also return without dropping the keyring reference. Keep ownership of the keyring reference in restore_dm_crypt_keys_to_thread_keyring(), drop it once on all exit paths, and make add_key_to_keyring() only use the reference without consuming it.	N/A	More Details
CVE-2026-93047	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Associate BOs with every job that accesses them A submission can expand into a chain of jobs (e.g. bin + render + cache clean). Implicit synchronization in v3d_submit_lock_reservations() is gated on each job's bo[], but the BO list was only ever attached to the last job of the chain. When that last job is a trailing CACHE_CLEAN job, the job that actually consumes the BOs (that is, a RENDER or CSD job) was left with bo_count == 0 and picked up no implicit dependencies. It could therefore be dispatched to the hardware and read a BO while another context was still writing it, leading to data corruption. Attach the BOs to the job that consumes them, so (1) it acquires the correct implicit dependencies during reservation locking and (2) they are kept mapped until the end of the submission. Give it references to all consuming job's BOs through v3d_job_reference_bos() instead of looking the handles up a second time; that avoids a redundant lookup and guarantees both jobs reference the exact same objects. As the CACHE_CLEAN job now carries a BO array as well, add a per-job `has_implicit_dep` flag so that only the consuming jobs take implicit dependencies. The CACHE_CLEAN job (a global flush) and the BIN job (binning waiting on another context is not a realistic scenario) are excluded.	N/A	More Details
CVE-2026-93071	In the Linux kernel, the following vulnerability has been resolved: media: bcm2835-unicam: Fix asc leaked in error/remove path v4l2_async_nf_add_fwnode_remote() allocates the asc, which is freed when v4l2_async_nf_cleanup() is called. Call v4l2_async_nf_cleanup() properly in the driver paths. Discovered with kmemleak after rmmod: unreferenced object 0xffff000084526b80 (size 64): comm "modprobe", pid 185, jiffies 4295013512 hex dump (first 32 bytes): 01 00 00 00 00 00 00 00 e8 0d ff bf 00 00 ff ff ..... 40 83 bc 84 00 00 ff ff 60 83 bc 84 00 00 ff ff @.....`..... backtrace (crc ac584083): [<00000000ffb081a7>] kmemleak_alloc+0x38/0x44 [<00000000d2fd9301>] __kmalloc+0x1b0/0x250 [<000000004dd5354d>] __v4l2_async_nf_add_fwnode+0x28/0x9c [<0000000067587657>] __v4l2_async_nf_add_fwnode_remote+0x3c/0x64	N/A	More Details
CVE-2026-89865	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Zero SFP DMA buffer in FRU/I2C bsg handlers The FRU and I2C bsg handlers stage their transfer in a DMA_POOL_SIZE (256-byte) bounce buffer obtained from dma_pool_alloc(), which does not zero the allocation. They initialize only a few leading bytes before handing the buffer to qla2x00_write_sfp(). qla2x00_write_sfp() can override the transfer length with a user-supplied value: if (len == 1) opt |= BIT_0; if (opt & BIT_0) len = *sfp; *sfp is the first byte of the (user-controlled) payload, so len can grow up to 255. The device then DMA-reads len bytes from the 256-byte pool buffer. Since only a small prefix was written (e.g. MAX_FRU_SIZE == 36 bytes for a FRU version, one byte for a FRU status register), the hardware reads past the initialized region and writes up to ~219 bytes of stale DMA-pool heap memory to the device flash. Allocate the buffer with dma_pool_zalloc() in all five FRU/I2C handlers so any bytes beyond the initialized data are zero rather than stale heap contents.	N/A	More Details
CVE-2026-89948	In the Linux kernel, the following vulnerability has been resolved: batman-adv: bla: fix freeing of claims on meshif deletion When the mesh interface is getting deleted, then batadv_bla_del_backbone_claims() (via batadv_bla_purge_backbone_gw()) could make sure that all claims gets removed. But this function is only executed when bat_priv->bla.claim_hash is not NULL. And since batadv_bla_free() is always setting it to NULL before it is (indirectly) called, it was never actually executed. But the batadv_bla_purge_claims() -> batadv_handle_unclaim() is at the moment too fragile because the BLA code is not handling the rehashing in batadv_bla_update_orig_address(). The stored backbone address doesn't have to be the one actually used for the hash bucket selection during the initial adding of the backbone. The batadv_handle_unclaim() can therefore fail to find the respective backbone for the unclaim and then stop the deletion. But the actual backbone_gw object is not needed for the unclaim because all relevant information is always provided by the caller. And the check for the existence of the backbone_gw doesn't provide any additional security check for the deletion of a claim.	N/A	More Details
CVE-2026-89798	In the Linux kernel, the following vulnerability has been resolved: rpcrdma: arm rn_done before publishing the notification rpcrdma_rn_register() inserts @rn into rd_xa with xa_alloc() before storing the caller's callback in rn->rn_done. The xarray makes @rn reachable to rpcrdma_remove_one(), which walks rd_xa and invokes rn->rn_done(rn) for every registered notification. A device removal that races a fresh registration can therefore observe @rn with rn_done still NULL, because the notification objects are zero allocated by their owners, and call through a NULL function pointer. Store rn->rn_done before xa_alloc() publishes @rn. The xarray's store-side and load-side ordering then guarantees that any CPU which finds @rn in rd_xa also observes the armed callback. rpcrdma_rn_unregister() treats a non-NULL rn_done as the sentinel for a completed registration, so the early store must not survive a failed registration. Clear rn_done again when xa_alloc() fails. Were it left set, the failed-accept cleanup path would call rpcrdma_rn_unregister() on an @rn that was never inserted, erasing an unrelated rd_xa slot and underflowing rd_kref.	N/A	More Details
CVE-2026-89816	In the Linux kernel, the following vulnerability has been resolved: drm: Fix drm_crtc_commit leak if signaled when PAGE_FLIP_EVENT is used Commit 1c6ceeee6ebb ("drm/atomic: Fix memleak on ERESTARTSYS during non-blocking commits") fixed a very similar issue when the event was allocated by drm_atomic_helper_setup_commit() itself. However, if the event is allocated in prepare_signaling(), it will also be set to NULL in complete_signaling(), which prevents drm_crtc_commit from being put in __drm_atomic_helper_crtc_destroy_state(). Dropping the reference when the event is set to NULL at complete_signaling() fixes the leak. The leak can be reproduced by sending a signal to the thread using DRM_MODE_PAGE_FLIP_EVENT and using a sw_sync fence to cause the atomic ioctl to block at drm_atomic_helper_wait_for_fences(). It happened both with amdgpu and vkms.	N/A	More Details

CVE-2026-89813	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: force complete the KIQ ring fences on reset Like the MES scheduler ring, the KIQ ring sets no_scheduler = true and uses a polling fence, so it is skipped by the force-completion loop in amdgpu_device_pre_asic_reset(). Its hw fence value lives in wb (GTT) memory and survives a MODE1 reset while fence_drv.sync_seq keeps advancing, so after a reset the first KIQ submission can poll forever on a seq that is never written back. Force complete the KIQ ring fences too so their hw fence is realigned to sync_seq.	N/A	More Details
CVE-2026-89812	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: force complete the MES ring fences on reset The MES scheduler ring has no drm scheduler (no_scheduler = true), so it is skipped by the force-completion loop in amdgpu_device_pre_asic_reset(). It uses a polling fence whose hw value lives in wb (GTT) memory and survives a MODE1 reset, while fence_drv.sync_seq keeps advancing for every packet. When the reset is triggered because MES itself stopped responding, the timed-out packets advance sync_seq past the last hw fence value MES wrote. After resume the first MES submission polls forever on a seq that is never written back, failing the resume and wedging the box on a second reset: amdgpu: MES ring buffer is full. amdgpu: *ERROR* ring gfx_0.0.0 test failed (-110) amdgpu: resume of IP block <gfx_v11_0> failed -110 amdgpu: GPU reset end with ret = -110 Force complete the MES scheduler ring fences together with the scheduler rings so their hw fence is realigned to sync_seq. v2: cover all XCCs (one scheduler ring each), not just mes.ring[0].	N/A	More Details
CVE-2026-89809	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: fix scope of mqd_mgr dereference in pqm_debugfs_mqds Reading /sys/kernel/debug/kfd/mqds while a process holds an active KFD queue triggers a NULL pointer dereference because the for loop that calls mqd_mgr->debugfs_show_mqd() is incorrectly placed outside the if (pqn->q) block that initializes mqd_mgr. The queue list can contain entries where pqn->q is NULL (kernel queues where only pqn->kq is valid). In the original code: if (pqn->q) { ... mqd_mgr = q->device->dqm->mqd_mgrs[mqd_type]; size = mqd_mgr->mqd_stride(...); } for (xcc = 0; xcc < num_xccs; xcc++) { // WRONG: outside if block mqd = q->mqd + size * xcc; r = mqd_mgr->debugfs_show_mqd(m, mqd); } When iterating over a queue node where pqn->q is NULL: 1. The if (pqn->q) block is skipped 2. mqd_mgr remains uninitialized (NULL from declaration) 3. The for loop executes anyway 4. mqd_mgr->debugfs_show_mqd(m, mqd) dereferences NULL The crash manifests as: BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor instruction fetch in kernel mode RIP: 0010:0x0 Call Trace: pqm_debugfs_mqds+0x10c/0x1d0 [amdgpu] kfd_debugfs_mqds_by_process+0x9b/0x110 [amdgpu] seq_read_iter+0x132/0x4b0 ... Fix by moving the for loop inside the if (pqn->q) block, so mqd_mgr and related variables are only used when properly initialized. (cherry picked from commit 8bfe29d5c798940f797aa24135d2734c3ffce9de)	N/A	More Details
CVE-2026-89807	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: guard against NULL restore_mqd in CRIU queue restore Both create_queue_cpsch() and create_queue_nocpsch() unconditionally call mqd_mgr->restore_mqd() when a CRIU restore is in progress (qd != NULL), with no NULL guard. On any system where restore_mqd is not implemented for the given queue type, a user holding CAP_CHECKPOINT_RESTORE can trigger a kernel NULL pointer dereference and panic the machine by issuing KFD_IOC_CRIU_OP_RESTORE with a crafted queue restore object. Note that checkpoint_mqd is likewise unimplemented on GFX12, so no legitimate CRIU image can reach this path — only a hand-crafted restore payload. Add a NULL guard for restore_mqd immediately after mqd_mgr is resolved, unwinding via the existing error labels and returning -EOPNOTSUPP if the callback is not implemented. This mirrors the existing checkpoint_mqd guard in checkpoint_mqd().	N/A	More Details
CVE-2026-89802	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/uvmm: fix NULL deref unwinding an OP_MAP_SPARSE op Each bind_job_op is zeroed by kzalloc_obj() in bind_job_op_from_uop(), and the OP_MAP_SPARSE case in nouveau_uvmm_bind_job_submit() only creates a region, so op->ops stays NULL for a successfully processed sparse map. If a later op in the same job fails, the reverse unwind loop revisits that op and calls drm_gpuva_ops_free(&uvmm->base, op->ops) unconditionally. drm_gpuva_ops_free() dereferences its argument right away (list_for_each_entry_safe on &ops->list), so a NULL op->ops oopses. The path is reachable by any render-node fd holder, since NOUVEAU_VM_BIND is DRM_RENDER_ALLOW. Guard the free with IS_ERR_OR_NULL(), as nouveau_uvmm_bind_job_cleanup() already does for the identical free.	N/A	More Details
CVE-2026-11432	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89800	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/uvmm: clear the dirty flag when unwinding an OP_UNMAP_SPARSE A successful OP_UNMAP_SPARSE marks its region dirty with nouveau_uvma_region_dirty() and defers the teardown to nouveau_uvmm_bind_job_cleanup(); it does not remove the region from uvmm->region_mt. If a later op in the job fails, the unwind path never clears reg->dirty (set in one place, cleared nowhere) and sets op->reg = NULL, so cleanup skips the teardown. The region is left in the tree with dirty set and its completion never signalled. Later binds over that range then fail permanently -- -ENOENT or -EINVAL from the dirty checks, or an unkillable wait_for_completion() in bind_validate_region() -- for the lifetime of the uvmm. Clear reg->dirty when the unwind reverts the sparse unmap, restoring the region to the state it was found in.	N/A	More Details
CVE-2026-89797	In the Linux kernel, the following vulnerability has been resolved: power: supply: ab8500_fg: fix use-after-free on remove ab8500_fg_remove() destroys the driver workqueue while the threaded interrupt handlers are still armed; they are devm-managed and freed only after ->remove() returns, so a handler that fires in that window queues work on the freed workqueue. Tear the workqueue down through devm instead, registering its cleanup after the power supply and before the interrupt requests. devm then frees the interrupts first, so the handlers can no longer queue work, before disabling the delayed and plain work items and destroying the workqueue. Disabling the items, rather than cancelling them, keeps them disabled so no producer (including the power-supply external_power_changed callback) can requeue them. Found by an in-house static analysis tool.	N/A	More Details
CVE-2026-89851	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix FCE trace enable parsing in debugfs qla2x00_dfs_fce_write() called kstrtoul() with a NULL result pointer, so a successful parse would dereference NULL and oops. Worse, the int return value (0 on success, negative errno on failure) was assigned to the unsigned long enable flag, inverting the intended logic: a valid number was treated as "disable" while a parse failure enabled FCE. Parse the value into enable and propagate parse errors to userspace.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: mm/damon/core: avoid infinite kdamond_merge_regions() internal loop Patch series "mm/damon: unurgent fixes for infinite loop, NULL de-ref and races", v1.1. Sashiko found a few issues in DAMON that could cause infinite loop, NULL dereference and monitoring results degradation. The first two sounds scary but the infinite loop happens only under unreasonable user setup. The NULL dereference is only in a unit test. Monitoring results degradation is trivial since it is only best-effort, and those happens from only unlikely races. Still those are bugs that better to		

CVE-2026-89796	fix if possible. Fix those. This patch (of 6): Due to online parameter update like events, the number of DAMON regions could be higher than the user-set upper limit. kdamond_merge_regions() repeats merge regions until the number meets the limit, while doubling the merge threshold up to the theoretical maximum threshold. It is tried only up to the theoretical maximum threshold because even the aggressive merging can fail from reducing the number of regions under the user-defined upper limit. For example, there could be many user-defined non-contiguous regions that cannot be merged. The threshold based loop break condition is evaluated by comparing the threshold for the next merging try against the theoretical maximum threshold. If max_thres is larger than UINT_MAX / 2, doubling the threshold could make it overflow, and bypass the loop break condition. In the case, if the number of regions cannot be reduced under the upper limit like explained above, the loop will run infinitely. Prevent the case by doing the break condition check before doubling the threshold. Also, prevent the threshold exceeding the maximum threshold, as it could overflow and apply the wrong merge threshold. This issue is unlikely to occur in real world, since having the max_thres higher than UINT_MAX / 2 require unrealistically large aggregation intervals compared to the sampling interval. Also, it requires an unrealistically large number of uncontiguous regions setup. Nonetheless, the consequence is bad and the fix is simple. The issue was discovered [1] by Sashiko.	N/A	More Details
CVE-2026-89794	In the Linux kernel, the following vulnerability has been resolved: ksmbd: zero pipe read compound padding Compound response handling extends the last response iov to an eight-byte boundary. smb2_read_pipe() allocates only the payload size, so the alignment padding can expose up to seven bytes of uninitialized kernel heap memory. Allocate the aligned size and clear the unused tail before pinning the response buffer.	N/A	More Details
CVE-2026-53534	JabRef is a desktop application for managing BibTeX and BibLaTeX libraries. Prior to 6.0-alpha.6, when jabsrv or JabRef's built-in HTTP server is enabled, the GET /better-bibtex/cayw endpoint accepts an external command query parameter and CAYWQueryParams.getCommand() passes it through CAYWResource.getCitation() into PushToSublimeText.getCommandLine(). On Unix-like systems, PushToSublimeText combines this untrusted cite-command prefix and citation keys into a string executed through sh -c by ProcessBuilder without shell escaping. A client that can cause a localhost request with application=sublime can inject shell metacharacters and execute operating-system commands as the JabRef user when a valid Sublime Text command path is configured and the victim completes the CAYW selection dialog. The built-in server is disabled by default, so exploitation requires the victim to enable it or run jabsrv. This issue is fixed in version 6.0-alpha.6.	N/A	More Details
CVE-2026-53554	SQLBot is an intelligent Text-to-SQL system based on large language models and RAG. Prior to 1.9.0, the POST /api/v1/datasource/parseExcel endpoint in backend/apps/datasource/api/datasource.py uses attacker-controlled multipart filename data when selecting where uploaded content is stored, writes the content before spreadsheet parsing and validation finish, and can transform a double-extension filename into a Python source file. An attacker able to submit a crafted multipart upload can use these behaviors to place attacker-controlled content in /opt/sqlbot/app/alembic/versions/ even when a spreadsheet parsing failure after the file write causes the endpoint to return an error. The planted file remains on disk, and subsequent SQLBot startup or migration processing causes Alembic to import the module and execute its module-level statements in the SQLBot application runtime. This issue is fixed in version 1.9.0.	N/A	More Details
CVE-2026-53555	SQLBot is an intelligent Text-to-SQL system based on large language models and RAG. Prior to 1.9.0, an authenticated uploader can submit an image/svg+xml assistant UI logo through PATCH /api/v1/system/assistant/ui, and SQLBot stores the SVG without sanitizing or validating embedded active content. SQLBot later serves the file inline from the same application origin through GET /api/v1/system/assistant/picture/{filename}. When another user loads that generated resource, JavaScript embedded in the SVG executes in the SQLBot web application context, resulting in stored cross-site scripting with access to data and actions available to the victim's session. This issue is fixed in version 1.9.0.	N/A	More Details
CVE-2026-53556	SQLBot is an intelligent Text-to-SQL system based on large language models and RAG. Prior to 1.9.0, the POST /api/v1/datasource/previewData endpoint in backend/apps/datasource/crud/datasource.py incorporates the client-controlled table_name value into generated SQL without safe identifier handling. An authenticated user can configure a datasource for SQLBot's internal PostgreSQL service and submit a crafted table_name that remains a SELECT operation under the read-only policy while invoking pg_read_file(), pg_read_binary_file(), or pg_ls_dir(). In the default tested trusted loopback authentication configuration, the internal connection accepts invalid credentials and executes with PostgreSQL superuser privileges, allowing filesystem content such as /etc/hosts and /etc/passwd to be returned in the previewData API response and potentially exposing configuration, credentials, authentication secrets, and source code. This issue is fixed in version 1.9.0.	N/A	More Details
CVE-2026-53557	SQLBot is an intelligent Text-to-SQL system based on large language models and RAG. Prior to 1.9.0, an authenticated user can supply a crafted sheet["tableName"] value in the Excel datasource configuration submitted through POST /api/v1/datasource/, and SQLBot stores that value without safe identifier handling. When the same datasource is later removed through DELETE /api/v1/datasource/{id}, the stored value is interpolated into datasource cleanup SQL and executed by PostgreSQL. This second-order SQL injection can invoke PostgreSQL COPY TO PROGRAM and execute arbitrary operating-system commands with the privileges of the postgres process inside the SQLBot container. This issue is fixed in version 1.9.0.	N/A	More Details
CVE-2026-54343	Frappe Learning Management System (LMS) is a learning system that helps users structure their content. Prior to version 2.52.1, a remote attacker can request a traversal path handled by SCORMRenderer.render in lms/page_renderers.py. The renderer constructs and opens a server-side path without first confirming that its real path remains within public/scorm, allowing files outside the SCORM directory to be read when they are accessible to the server process. This issue is fixed in version 2.52.1.	N/A	More Details
CVE-2026-89817	In the Linux kernel, the following vulnerability has been resolved: drm/gud: NUL-terminate TV mode names read from the device gud_connector_add_tv_mode() reads a buffer of fixed-size mode names from the USB device and passes pointers into it to drm_mode_create_tv_properties_legacy(), which calls strlen() on each one. Nothing guarantees the device NUL-terminates a name, so strlen() can run past the end of a slot and, for the last mode, past the end of the allocation. Terminate each name at the end of its slot before use.	N/A	More Details
CVE-2026-89820	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix dc_lock leak on GPU reset error paths On GPU reset, dm_suspend() takes dc_lock and leaves it for dm_resume() to drop. If amdgpu_dm_commit_zero_streams() or dm_dmub_hw_init() fails, the function returns with the lock still held. The matching resume path is then skipped, so every later dc_lock take hangs. Release the cached DC state and unlock before returning the error.	N/A	More Details
CVE-2026-89821	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: avoid divide-by-zero in __is_lut_linear() __is_lut_linear() computes the expected value of each entry with expected = i * MAX_DRM_LUT_VALUE / (size - 1); If it is ever called with a single-entry LUT, size - 1 is zero and the kernel takes a divide error (#DE). A LUT with fewer than two entries cannot describe a linear mapping anyway, so return false early instead of dividing by zero.	N/A	More Details

CVE-2026-89822	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Guard against NULL driver_data in i915_pci_probe() pci_match_device() can return the dummy pci_device_id_any entry when a device is force-bound via sysfs driver_override, in which case ->driver_data is unset (NULL). i915_pci_probe() casts it to struct intel_device_info * unconditionally and dereferences intel_info->require_force_probe, causing a NULL-ptr-deref. (cherry picked from commit 2727922084672cc274ecea726ea00363c2893731)	N/A	More Details
CVE-2026-54355	MapServer is a system for developing web-based GIS applications. From 6.0 until 8.6.4, MapServer's OpenLayers HTML output for SERVICE=WMS&REQUEST=GetMap&FORMAT=application/openlayers reflects an attacker-controlled X-Forwarded-Host value received as HTTP_X_FORWARDED_HOST through msBuildOnlineResource(), processLine(), and the [mapserv_onlineresource] substitution in src/maputil.c and src/maptemplate.c without escaping it for a single-quoted JavaScript string. When the deployment trusts the forwarded header and does not configure a fixed ows_onlineresource or MS_ONLINERESOURCE value, embedded single quotes can escape the generated URL string. An unauthenticated attacker can craft a URL that executes arbitrary JavaScript in the MapServer site origin when opened by a victim, enabling access to sessions or tokens, same-origin data and requests, and actions as the victim. This issue is fixed in version 8.6.4.	N/A	More Details
CVE-2026-54501	Browsertrix is a high-fidelity, browser-based crawling service for web archiving that can be self-hosted or used through Webrecorder's hosted instance. From 1.15.0 until 1.22.8, Browsertrix improperly sanitizes Git URLs specified as Custom Behaviors, allowing command injection through /api/orgs/*/crawlconfigs/validate/custom-behavior. A user with crawler or administrator permission on the specific instance can supply a crafted Git URL that executes arbitrary operating-system commands in the backend pod. Open registration or hosted free-trial access can make the required role broadly obtainable. Successful exploitation can expose, modify, or delete application database records, archived items, browser profiles, storage data, proxy credentials, and other configured service data. This issue is fixed in version 1.22.8.	N/A	More Details
CVE-2026-89845	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Avoid req_q_map double-read in qla2x00_error_entry() qla2x00_error_entry() reads ha->req_q_map[que] twice: once for the NULL check and again when assigning it to req. The map slot is cleared by qla25xx_free_req_que() (ha->req_q_map[que_id] = NULL under mq_lock) during queue teardown, while the response-queue interrupt that drives qla2x00_error_entry() is still registered (the IRQ is released later in qla25xx_free_rsp_que()). If the slot is set to NULL between the two reads, req becomes NULL and is dereferenced. Read the slot once into req and NULL-check the local before use. mq_lock is a mutex and cannot be taken from interrupt context, so the single read plus local check is the appropriate fix for the reported NULL dereference.	N/A	More Details
CVE-2026-89843	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Zero-init bsg stack buffers to avoid info leak Several bsg handlers stage their request/reply in an uninitialized 256-byte on-stack buffer (uint8_t bsg[DMA_POOL_SIZE]) and fill it via sg_copy_to_buffer(), which only copies as many bytes as the user-supplied request payload. When the request is shorter than the structure, the remainder of the buffer is left holding stale stack data. qla2x00_read_fru_status() and qla2x00_read_i2c() then copy the full structure back to the reply payload with sg_copy_from_buffer(), leaking the uninitialized stack bytes to user space. The write/update paths do not copy the buffer back, but can feed uninitialized fields to the device. Zero the stack buffer at declaration in all five handlers, mirroring the heap kzalloc() approach, so short requests can no longer expose stale memory.	N/A	More Details
CVE-2026-89842	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Skip NVMe LS reject IOCB when FW not started qla_nvme_xmt_ls_rsp() bails out to the out: label when firmware is not started (!ha->flags.fw_started), but the out: path unconditionally calls qla_nvme_ls_reject_iocb(), which ends in qla2x00_start_iocbs() and an unconditional doorbell write to the request queue in-pointer register. This rings the firmware doorbell and queues an IOCB that stopped or resetting firmware cannot consume, and touches MMIO during the reset/EEH window where fw_started is also clear. Only emit the LS reject IOCB (and ring the doorbell) when fw_started is set; otherwise just clean up and return. The post-allocation failure cases (SRB alloc / qla2x00_start_sp() failure) run with firmware started and still send the reject. Apply the same guard to the reject emission in qla2xxx_process_purls_pkt().	N/A	More Details
CVE-2026-89839	In the Linux kernel, the following vulnerability has been resolved: f2fs: use the mount idmap for the owner check in f2fs_xattr_advise_set() f2fs_xattr_advise_set() calls inode_owner_or_capable() with &nop_mnt_idmap before allowing the "system.advise" xattr to be set, instead of the idmap that the VFS passes to the ->set() handler. f2fs supports idmapped mounts, so on such a mount this checks the caller's fsuid against the unmapped on-disk owner rather than the mapped owner: the actual owner can be wrongly denied with -EPERM and an unrelated caller wrongly allowed. Pass the handler's idmap instead.	N/A	More Details
CVE-2026-54604	OpenSlide is a C library for reading whole slide image files. Prior to 4.0.1, a behavior change in libtiff 4.7.1 causes the indirect TIFF tile path in src/openslide-decode-tiff.c and _openslide_tiff_read_tile() to request a full-height destination for a partial bottom tile row, allowing uninitialized heap memory to enter pixel data returned by openslide_read_region(). A crafted but valid slide can trigger the issue in TIFF-based formats other than Hamamatsu NDPI when a network service renders attacker-provided slides. Successful extraction depends on pixel transparency handling and recompression, OpenSlide with libtiff 4.7.0 or earlier is not affected, and official binary builds 4.0.0.10 and 4.0.0.11 are affected. This issue is fixed in OpenSlide 4.0.1 and official binary build 4.0.0.12.	N/A	More Details
CVE-2026-89837	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix dentry folio leak in find_in_level find_in_level() gets a dentry folio with f2fs_find_data_folio() before calling find_in_block(). If find_in_block() returns an error, the function stores the error in res_folio and breaks out of the loop without dropping the dentry folio. This leaks the folio reference on the find_in_block() error path. Drop the dentry folio before returning the error to the caller.	N/A	More Details
CVE-2026-89835	In the Linux kernel, the following vulnerability has been resolved: f2fs: avoid NULL checkpoint thread access in sysfs checkpoint_merge can be enabled even when no checkpoint merge thread is running. A read-only mount is one case: f2fs does not start f2fs_issue_ckpt there, but ckpt_thread_ioprio is still writable through sysfs. The ckpt_thread_ioprio store path updates the saved ioprio value and, when checkpoint_merge is enabled, calls set_task_ioprio() for the checkpoint thread. If cprc->f2fs_issue_ckpt is NULL, that dereferences a NULL task pointer. Protect ckpt_thread_ioprio sysfs writes with s_umount as well, so the checkpoint thread cannot disappear under the store path while updating its ioprio.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to migrate all curseg types during free_segment_range In free_segment_range(), the curseg evacuation loop only iterates up to NR_CURSEG_PERSIST_TYPE (0..5), missing non-persistent in-memory curseg types such as CURSEG_COLD_DATA_PINNED and CURSEG_ALL_DATA_ATGC. Even though these in-memory curseg types are not saved in the on-disk checkpoint header, they still occupy active physical		More

2026-89834	segments at runtime. If an active in-memory curseg happens to be allocated within the segment range being truncated during filesystem shrink, failing to evacuate it will cause subsequent writes to the curseg attempting out-of-bounds I/O on the truncated storage range. Fix this by expanding the curseg evacuation loop upper bound to NR_CURSEG_TYPE to ensure all active curseg types are safely migrated out of the target range.	N/A	Details
CVE-2026-57847	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89833	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid potential deadloop in f2fs_fsync_node_pages() There is potential deadloop in race condition: Thread A Thread B - fsync - f2fs_do_sync_file - f2fs_fsync_node_pages - last_fsync_dnode - folio_get(last_folio) - f2fs_setattr - f2fs_truncate - f2fs_truncate_blocks - f2fs_do_truncate_blocks - f2fs_truncate_inode_blocks - truncate_dnode - truncate_node - invalidate_mapping_pages - folio->mapping = NULL - is_node_folio alwasy return false - atomic && !marked is always true, then goto retry	N/A	More Details
CVE-2026-89831	In the Linux kernel, the following vulnerability has been resolved: f2fs: protect critical_task_priority updates with s_umount The sysfs store path already takes s_umount for GC thread control entries, and ckpt_thread_ioprio is covered as well. critical_task_priority also updates checkpoint or GC kthread scheduling state, but it is not covered by that serialization. It can race with remount or teardown paths that are stopping those threads. Protect critical_task_priority sysfs writes with s_umount too.	N/A	More Details
CVE-2026-89830	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix valid block count leak on data block allocation failure In __allocate_data_block(), when allocating a new data block (dn->data_blkaddr == NULL_ADDR), inc_valid_block_count() is called first to increment total_valid_block_count and i_blocks. If the subsequent f2fs_allocate_data_block() fails, the function returns the error directly without rolling back the already-incremented block counts, causing a permanent leak. Fix this by calling dec_valid_block_count() to undo the increment before returning the error. The condition old_blkaddr == NULL_ADDR precisely identifies the case where inc_valid_block_count() was called.	N/A	More Details
CVE-2026-89828	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix init ordering in amdgpu_vram_mgr_init() drmm_cgroup_register_region() is called before INIT_LIST_HEAD() and gpu_buddy_init() in amdgpu_vram_mgr_init(). If it fails, the function returns early and bypasses those initializations. Since adev->mman.initialized is set to true before amdgpu_vram_mgr_init() is called, a failure triggers amdgpu_ttm_fini(), which calls amdgpu_vram_mgr_fini(), which then: - Calls list_for_each_entry_safe() on reservations_pending and reserved_pages, whose list_head::next pointers are zero-initialized (NULL). The loop does not recognize them as empty and dereferences NULL. - Calls gpu_buddy_fini(), which iterates free_trees[] unconditionally via for_each_free_tree(). Since mm->free_trees is NULL (never allocated), this dereferences NULL. Both result in a kernel panic on the module load error path. Fix by moving drmm_cgroup_register_region() to after the list and buddy allocator are fully initialized, so the teardown path is safe to run.	N/A	More Details
CVE-2026-89827	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: avoid force-completing uninitialized UVD rings uvd_v7_0_sw_init() does not initialize the UVD decode ring for an SR-IOV VF. However, amdgpu_uvd_resume() unconditionally force-completes the decode ring when restoring its fence sequence. Skip fence completion when the fence driver is not initialized.	N/A	More Details
CVE-2026-89824	In the Linux kernel, the following vulnerability has been resolved: drm/panel-edp: fix i2c adapter leak on probe failure Make sure to drop the i2c adapter reference on probe failure (e.g. probe deferral) and on driver unbind also if a devicetree redundantly uses the 'ddc-i2c-bus' property to point to the aux ddc bus.	N/A	More Details
CVE-2026-86585	The lack of signature verification of firmware update packages in VEO and VEO-XS Wi-Fi monitors, in versions prior to 01.48.001, allows an attacker who controls the delivery of an update to install unauthorised firmware.	N/A	More Details
CVE-2026-54507	Vvveb is a powerful and easy to use CMS with page builder to build websites, blogs or ecommerce stores. Prior to 1.0.8.5, the oEmbedProxy() handler in admin/controller/editor/editor.php accepts an attacker-controlled url parameter and passes it to getUrl(), while validateUrl() in system/functions.php checks only the hostname string and does not validate its resolved addresses. An authenticated admin-panel user with editor/* permission can invoke GET /admin/index.php?module=editor/editor&action=oEmbedProxy with a dotted hostname or normalized loopback form that resolves to a private, loopback, link-local, or reserved address, causing the server to issue an HTTP or HTTPS request and return the response body. Storefront users and anonymous visitors cannot invoke the endpoint, but no CSRF token is required because the action uses GET. This can disclose internal service responses or cloud instance metadata and associated credentials. This issue is fixed in version 1.0.8.5.	N/A	More Details
CVE-2026-14916	A JWT signature verification vulnerability affects Kong components that perform JWT validation for MCP OAuth2 or DataKit integrations inside Kong API Gateway Enterprise. The affected code does not properly validate that the JWT signing algorithm is compatible with the type of key used for verification. As a result, an unauthenticated remote attacker may be able to craft a forged JWT that is incorrectly accepted as valid, leading to authentication bypass and potential compromise of confidentiality, integrity, and availability.	N/A	More Details
CVE-2026-88255	Improper Validation of Unsafe Equivalence in Input in ZenHive mpp allows an unauthenticated remote client to pass the Tempo duplicate-submission gate twice with one signed transaction. MPP.Methods.Tempo reserves the pre-broadcast dedup slot on the caller-supplied hex in reserve_hash_atomic/2, keyed through store_key/1 on tx.raw rather than on a canonical form of the transaction. The deserializer stores the caller's hex verbatim and accepts both recovery-id encodings, so one signed transaction submitted once with v=27 and once with v=0 yields two distinct reserve keys, and both pass the reserve and reach the broadcast path. The plug-level credential replay store is deliberately carved out for tempo in lib/mpp/replay.ex, leaving this reserve as the only gate, and the post-broadcast mark writes the canonical hash key that the raw-keyed reserve never reads. What the duplicate submission yields depends on the node: a nonce-reuse rejection fails closed, while a node that answers with the canonical hash for an already-known transaction returns a second valid Payment-Receipt for a single on-chain payment. This issue affects mpp: from 0.2.0 before 0.16.2.	N/A	More Details
	In NLnet Labs Unbound up to and including 1.26.0, a vulnerability was found in that can progressively corrupt heap memory and under certain systems and compilation options could lead to remote code execution. The vulnerability starts when CNAME		

CVE-2026-82717	synthesis during an upstream response needs to enforce(rewrite) a max TTL value in the packet buffer. Coupled with a compression pointer that points to the overwritten value and invalidates the domain name, it leads to an error path that does not properly move the buffer position and allows for the heap buffer overflow. Since this is heavily reliant on heap memory layout, results are memory corruption that eventually leads to a crash and under specific systems and compilation options remote code execution.	N/A	More Details
CVE-2026-79954	NASA CryptoLib 1.5.0 contains an authentication downgrade vulnerability in the Telecommand (TC) receive path. The receiver selects the Security Association used for SDLS processing solely from the SPI field inside the incoming frame, but it does not verify that the selected SA is authorized for the frame's GVCID.	N/A	More Details
CVE-2026-92355	In affected versions of Octopus Server, a user with permission to modify non built-in external feeds could exploit a path traversal flaw to overwrite arbitrary files on the server, which in some configurations could lead to remote code execution.	N/A	More Details
CVE-2026-68493	After guessing a 62^15 complex unique identifier, a malicious logged in user was able to retrieve a list of memberships for a circle they are not a member of.	N/A	More Details
CVE-2026-77164	Circles' remote-instance signature verification fetches the attacker-supplied keyId URL before trust in the remote instance is established, and explicitly allows local/private addresses for this request, bypassing Nextcloud's core SSRF protections. The public, unauthenticated endpoints POST /apps/circles/event/ and POST /apps/circles/incoming/ reach this code path, allowing any unauthenticated user to force the server to issue a GET request to an internal address. The response body of the internal request is never returned to the requester, so this is blind SSRF: an attacker can determine whether an internal service is reachable, but cannot read its response contents through this endpoint alone.	N/A	More Details
CVE-2026-77169	A vulnerability in the team folders (formerly group folders) app when used in combination with the workspace app allowed API/REST-only delegated administrators to bypass folder-level authorization controls. The workspace app enables organizations to delegate limited administrative privileges for team folder management via API/REST only, restricting access to folders for which the admin has advanced permissions.	N/A	More Details
CVE-2026-77170	The Deck config API allows authenticated users to set board-scoped configuration keys for arbitrary board IDs without validating whether the user owns or has permission to manage the referenced board.	N/A	More Details
CVE-2026-82980	Any authenticated user can lock or unlock files they do not own by targeting absolute WebDAV paths of other users. The DAV plugin resolves files from the absolute request URI without verifying that the path segment matches the authenticated session user. This enables: Cross-user manual locks : attacker locks a victim's files, blocking writes (PUT/MOVE/DELETE, editor saves). Lock-token disclosure: the app returns the lock token to unauthorized callers, enabling them to remove token-based locks (client locks) of other users.	N/A	More Details
CVE-2026-82982	The Approval app's approve/reject endpoint is meant to require the file's current etag as a freshness check, preventing an approver from approving or rejecting a file whose contents changed after they reviewed it. The backend only enforced this check when the etag parameter was present and non-empty in the request. An attacker able to intercept and modify the approval request could omit the etag field entirely, bypassing the freshness check and approving or rejecting a file version they never reviewed.	N/A	More Details
CVE-2026-82985	The Photos app's filter-based "smart albums" build their file listing using the search configuration (photosSourceFolders) of the user viewing the album, rather than the album owner's configuration. When an album owner shares a smart album with another user, that user's own folder configuration is used to determine which of the owner's files are searched — allowing them to discover files (name, file ID, and other metadata) in folders the album owner never intended to include in the shared album. This requires the album owner to have shared a filter-based smart album with the attacker; it does not allow access to arbitrary users' files without such a share.	N/A	More Details
CVE-2026-88263	XikeStor Layer3 switches miss authentication for downloading configuration data. Unauthenticated attacker may retrieve the configuration data containing network configurations and passwords to operate the affected product improperly or to exploit the affected product as a jump host.	N/A	More Details
CVE-2026-84408	QND contains an improper access control vulnerability in a named pipe, which may allow a local attacker who is logged in to a Windows PC where the affected product's client is installed to execute arbitrary commands with SYSTEM privileges.	N/A	More Details
CVE-2026-81326	QND uses a hard-coded cryptographic key, which may allow a local attacker who is logged in to a Windows PC where the affected product's client is installed to obtain administrator credentials, including an ID and password.	N/A	More Details
CVE-2026-19248	QDomDocument XML parsing is vulnerable to a remotely-triggerable denial-of-service crash when processing untrusted input.	N/A	More Details
CVE-2024-27123	A cross-site scripting (XSS) vulnerability has been reported to affect QcalAgent. The local attackers can then exploit the vulnerability to bypass security mechanisms or read application data. We have already fixed the vulnerability in the following version: QcalAgent 1.1.9 and later	N/A	More Details
CVE-2026-61396	Rejected reason: Did not need CVE ID	N/A	More Details
CVE-2026-15640	Under certain conditions a valid SAML IdP response may be used to impersonate another Secret Server user.	N/A	More Details
	Ash field_policies are documented to protect against filter-based information disclosure: when a field the actor may not see is		

CVE-2026-86338	referenced in a filter, it is replaced with an expression that evaluates to nil, so a filter cannot be used as a yes/no oracle to read a value the actor cannot see. This nilling was applied to attributes but not to calculations or aggregates. A user-supplied filter reference to a calculation or aggregate carries an Ash.Query.Calculation / Ash.Query.Aggregate struct, which the authorizer's reference replacement did not match (it only matched the Ash.Resource.* structs), so the filter ran against the real value. As a result, an actor whose field policies forbid a calculation or aggregate can still filter by it (for example filter(secret_calc == "x") or filter(comment_count == n)) and learn the value from whether rows match — an oracle that recovers field-policy-protected values one probe at a time. Filtering is commonly exposed to lower-privileged actors (for example via AshGraphql or AshJsonApi filter arguments), which is exactly the surface field policies are meant to protect. The fix routes filter references to calculations and aggregates through the same field-policy nilling as attributes. This issue affects ash: from 2.11.0-rc.0 before 3.33.4.	N/A	More Details
CVE-2026-89186	Use of Cache Containing Sensitive Information in ZenHive mpp allows a shared HTTP cache to store a paid response and serve it to clients that never paid. MPP.Plug.verify_credential in lib/mpp/plug.ex sets payment-receipt and cache-control: private on the connection before the wrapped application runs, and registers no register_before_send/2 callback. Plug.Conn.put_resp_header/3 replaces an existing header, so a mounting application that sets its own cache-control on the paid resource (for example public, max-age=3600) silently overrides the private the library relies on, and a CDN or reverse proxy can then store the paid 200 together with its Payment-Receipt and serve both to unpaid clients. The library-level guarantee is therefore defeatable by the application it protects. For the same reason a downstream non-2xx response still carried Payment-Receipt, issuing a receipt for a response that delivered no resource. This issue affects mpp: from 0.1.0 before 0.16.2.	N/A	More Details
CVE-2026-86474	The lack of TLS certificate validation when downloading firmware updates in VEO and VEO-XS Wi-Fi monitors, in versions prior to 01.48.001, allows an attacker to perform man-in-the-middle attacks on the update channel.	N/A	More Details
CVE-2026-87886	Local privilege escalation due to insecure file permissions. The following products are affected: Acronis Backup plugin for cPanel & WHM (Linux) before build 1.9.3.1021, Acronis Backup extension for Plesk (Linux) before build 1.8.11.638, Acronis Backup plugin for DirectAdmin (Linux) before build 1.2.3.238.	N/A	More Details
CVE-2026-54608	MythicalDash is a Pterodactyl client area. In 3.5.4-aurora and earlier, GET /api/stripe/process in backend/app/Api/System/Gateways/Stripe.php creates a pending row in mythicaldash_stripe_payments before Stripe checkout succeeds and embeds the payment code in the success redirect, while GET /api/stripe/processed accepts that code without constructing a Session, checking ownership, or retrieving the Stripe Checkout Session to require payment_status to be paid and amount_total to match the expected charge. An ordinary authenticated user can request an attacker-selected coins amount, abandon or fail payment, and submit the pending code directly to the unauthenticated processed endpoint. StripeDB::isPending() then permits User::addCreditsAtomic() to grant the unpaid amount and mark the row processed even though Stripe has not confirmed payment. This permits arbitrary free virtual-currency top-ups and direct financial loss through consumption of hosting resources. No fixed version is available as of this review.	N/A	More Details
CVE-2026-86443	Cleartext storage of sensitive information in the DuoxMe application for Android, in versions prior to 4.3.4, allows an attacker with local access to the device to retrieve the credentials stored by the application and impersonate the user account.	N/A	More Details
CVE-2026-85628	Transmission of the home Wi-Fi credentials without encryption during the pairing process between the DuoxMe application and VEO and VEO-XS Wi-Fi monitors, in versions prior to 4.3.4 of the application and 01.50.001 of the monitor firmware, allows an attacker on the Wi-Fi Direct network to intercept the network password.	N/A	More Details
CVE-2026-54633	PoDoFo is a C++17 PDF manipulation library. From version 1.0.0 until 1.1.1, processing a crafted PDF with an Indexed color-space image can cause a heap out-of-bounds read in PdfColorSpaceFilterIndexed::FetchScanLine in src/podofo/main/PdfColorSpaceFilter.cpp. PODOFO_INVARIANT does not perform a runtime check, so a pixel index greater than or equal to m_MapSize can address beyond m_lookup. PdfColorSpaceFilterFactory::TryCreateFromObject also validates hival with an incorrect conjunction and no upper bound, allowing malformed Indexed color-space metadata outside the expected range. The resulting read can disclose adjacent heap data or crash the processing application. This issue is fixed in version 1.1.1.	N/A	More Details
CVE-2026-54642	CubeCart is an ecommerce software solution. Prior to 6.7.5, the reset_id download-counter action and delete_card stored-payment-card action in admin/sources/orders.index.inc.php use state-changing GET requests and are omitted from the protection map in admin/skins/default/csrf.inc.php. A remote attacker can induce an authenticated administrator to issue one of these requests without a validated session token, causing unintended resets of electronic download usage counters or deletion of stored customer payment-card tokens. This issue is fixed in version 6.7.5.	N/A	More Details
CVE-2026-76151	Out-of-bounds read (buffer over-read) in the HTTP Cache-Control response header parsing in the QtNetwork module in Qt Group Qt 6.0.0 through 6.8.8, and 6.9.0 through 6.11.1, allows remote attackers to cause a denial of service (application crash) via an excessively large Cache-Control header value returned by an untrusted or compromised HTTP server to an application using QNetworkAccessManager. Only the client side of the connection is affected and 32-bit builds are not affected; the out-of-bounds access is read-only, with no information disclosure and no code execution.	N/A	More Details
CVE-2026-14917	A SAML authentication bypass vulnerability affects the Kong SAML plugin when the validate_assertion_signature option is explicitly set to false. This option is enabled by default. When disabled, the plugin may extract the SAML identity from an unsigned assertion and authenticate the user without verifying a valid cryptographic signature. As a result, an unauthenticated remote attacker may be able to submit a crafted SAML response and impersonate arbitrary users, including administrators	N/A	More Details
CVE-2026-89790	In the Linux kernel, the following vulnerability has been resolved: ipv6: avoid divide by zero in rt6_multipath_rebalance rt6_multipath_rebalance() calculates the total eligible nexthop weight in one pass and programs upper bounds in a second pass. Since RTM_NEWROUTE is RTNL-free, a concurrent ignore_routes_with_linkdown update can make the first pass return zero while the second sees an eligible nexthop, causing rt6_upper_bound_set() to divide by zero. UBSAN: division-overflow in net/ipv6/route.c:4845:17 Oops: divide error: 0000 [#1] SMP KASAN NOPTI rt6_upper_bound_set() net/ipv6/route.c:4845 rt6_multipath_rebalance() fib6_add_rt2node() ip6_route_multipath_add() inet6_rtm_newroute() Skip upper-bound calculation when the first pass reports a zero total. This respects the lock-free performance considerations here and solves insecure scenarios.	N/A	More Details
	Authentication Bypass by Spoofing vulnerability in team-alembic ash_authentication allows an attacker who can plant a remember-me cookie in a victim's browser to replace that victim's authenticated session with one for the attacker's own		

CVE-2026-76949	account. AshAuthentication.Plug.Helpers.sign_in_using_remember_me/3 skips re-authenticating an already-signed-in visitor by checking the session for "<subject_name>_token", but store_in_session/2 writes that key only when require_token_presence_for_authentication? is enabled and otherwise writes the bare subject name. At the default setting the guard therefore reads a key that is never written, its already-signed-in branch is unreachable, and the remember-me sign-in runs on every request through the per-request browser pipeline plug. A planted remember-me cookie is consequently honoured even for a visitor holding a live authenticated session, so whatever the victim enters afterwards lands in data the attacker controls. The read path in authenticate_resource_from_session/4 selects the key correctly, so the guard and the reader disagree about which key holds the session. This issue affects ash_authentication: from 4.10.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-78668	Rejected reason: reserved but not needed	N/A	More Details
CVE-2026-86688	Session Fixation vulnerability in team-alembic ash_authentication allows an attacker who can plant a session identifier in a victim's browser to hold an authenticated session once that victim signs in. AshAuthentication.Plug.Helpers.store_in_session/2 writes the authenticated subject into the existing session with Plug.Conn.put_session/3 and never calls Plug.Conn.configure_session(renew: true), so the identifier the visitor arrived with carries into their authenticated session. Every authentication event reaches this one function: the default success/4 injected by AshAuthentication.Phoenix.Controller.__using__/1, the AuthController emitted by mix ash_authentication_phoenix.install, and remember-me auto-login. AshAuthentication.Phoenix.Plug.store_in_session/2 is a defdelegate to it. Logout does not close the window either, because clear_session/2 ends with Plug.Conn.clear_session/1, which clears session contents but leaves the identifier intact, so a planted identifier survives a logout-then-login cycle. This issue affects ash_authentication: from 0.2.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-89787	In the Linux kernel, the following vulnerability has been resolved: ext4: check dir entry fits before reading the hash trailer in ext4_search_dir() For casefolded encrypted directories ext4 stores an 8-byte hash trailer after the name (EXT4_DIRENT_HASHES()), at an offset derived from de->name_len. On the sb_no_casefold_compat_fallback() path ext4_match() reads that trailer, but ext4_search_dir()'s by-hand pre-check only tests de->name + de->name_len <= dlimit, which proves the name fits, not the rounded trailer. A crafted entry whose name ends at the block boundary passes the check while EXT4_DIRENT_HASHES(de) lands past the block end, so ext4_match() reads out of bounds on an ordinary lookup. KASAN reports it as a use-after-free when the page after the directory block holds a freed object: BUG: KASAN: use-after-free in ext4_match (fs/ext4/namei.c:1435) Read of size 4 at addr ffff888010458000 by task exploit Call Trace: ext4_match (fs/ext4/namei.c:1435) ext4_search_dir (fs/ext4/namei.c:1470) __ext4_find_entry (fs/ext4/namei.c:1268 fs/ext4/namei.c:1632) ext4_lookup (fs/ext4/namei.c:1703 fs/ext4/namei.c:1769) ... filename_lookup (fs/namei.c:2842) vfs_statx (fs/stat.c:353) __do_sys_newfstatat (fs/stat.c:538) do_syscall_64 (arch/x86/entry/syscall_64.c:94) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:121) Require, for hash-in-dirent directories, that the whole entry including the rounded trailer fits before calling ext4_match(). This is the same bound ext4_check_dir_entry() already enforces via ext4_dir_rec_len(), so no well-formed entry is rejected. The other caller, ext4_find_dest_de(), runs ext4_check_dir_entry() first and is unaffected.	N/A	More Details
CVE-2026-89785	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: fix out-of-bounds read of INDEX_ROOT in reparse/objid init ntfs_reparse_init() and ntfs_objid_init() parse the index root of the $Extend/$Reparse and $Extend/$ObjId metafiles (the INDEX_ROOT attributes named $R and $O). They read its type and rule fields through resident_data(), which does not check that the resident attribute is large enough to hold them. mi_enum_attr() accepts a resident attribute with data_off == asize and data_size == 0. For such an attribute placed last in its MFT record, resident_data() returns a pointer to the end of the record_size buffer, so reading root->type / root->rule reads past the allocation. Use resident_data_ex(attr, sizeof(struct INDEX_ROOT)) and bail out when it returns NULL, as ntfs_security_init() already does for $SDH / $SII. The attribute is only parsed while mounting a crafted image, so this needs CAP_SYS_ADMIN. BUG: KASAN: slab-out-of-bounds in ntfs_reparse_init (fs/ntfs3/fsntfs.c:2306) Read of size 4 at addr ffff88801219dc00 by task mount ntfs_reparse_init (fs/ntfs3/fsntfs.c:2306) ntfs_fill_super (fs/ntfs3/super.c:1604) get_tree_bdev_flags (fs/super.c:1703) vfs_get_tree (fs/super.c:1758) path_mount (fs/namespace.c:4131) __x64_sys_mount (fs/namespace.c:4360)	N/A	More Details
CVE-2026-65323	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89784	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: check rpc_sockaddr2uaddr() return value in rpcb_register_inet4/6 rpcb_register_inet4() and rpcb_register_inet6() store the result of rpc_sockaddr2uaddr() into map->r_addr without checking it for NULL. rpc_sockaddr2uaddr() returns NULL when its final kstrdup() fails, and the unchecked NULL is then carried into the synchronous RPCBPROC_SET encode path: rpcb_register_call() -> rpc_call_sync() -> rpcb_enc_getaddr() -> encode_rpcb_string(), whose first statement is strlen(string), dereferencing NULL and oopsing the kernel. The crash reproduces under failslab on v6.12; with KASAN the NULL dereference surfaces as a fault on the shadow of address zero: Oops: general protection fault, probably for non-canonical address 0xdfffffc0000000000 [#1] PREEMPT SMP KASAN RIP: 0010:strlen (lib/string.c:409) Call Trace: encode_rpcb_string (net/sunrpc/rpcb_clnt.c:890) rpcb_enc_getaddr (net/sunrpc/rpcb_clnt.c:910) rpcauth_wrap_req_encode (net/sunrpc/auth.c:745) call_encode (net/sunrpc/clnt.c:1966) __rpc_execute (net/sunrpc/sched.c:952) rpc_run_task (net/sunrpc/clnt.c:1243) rpc_call_sync (net/sunrpc/clnt.c:1272) rpcb_v4_register (net/sunrpc/rpcb_clnt.c:500) svc_generic_rpcbind_set nfsd_rpcbind_set svc_register svc_setup_socket svc_addsock write_ports nfsctl_transaction_write vfs_write The crash is reachable when an in-kernel RPC service (nfsd, lockd, nfs-callback) registers with the local rpcbind under enough memory pressure for the small GFP_KERNEL kstrdup() in rpc_sockaddr2uaddr() to fail. The asynchronous getport path already handles this exact failure mode by returning -ENOMEM; only the two register helpers omit the check. Mirror that handling: bail out with -ENOMEM when rpc_sockaddr2uaddr() returns NULL, before the address is fed into the encoder.	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: net: qualcomm: rmnet: restore skb->dev on deaggregated frames rmnet_map_deaggregate() allocates each sub-frame with alloc_skb() and leaves skb->dev NULL. __rmnet_map_ingress_handler() assigns skb->dev = ep->egress_dev only on the data path, but a MAP command frame is dispatched to rmnet_map_command() before that, so rmnet_map_send_ack() runs netif_tx_lock(skb->dev) on a NULL device. An unprivileged user reaches this by unsharing a user+net namespace, creating an rmnet link over a tap device with INGRESS_DEAGGREGATION and INGRESS_MAP_COMMANDS, and writing an aggregated frame carrying a flow-control command to the tap fd. Restore the assignment dropped by 378e25357ac7, so every skb leaving rmnet_map_deaggregate() has a valid device. BUG: KASAN: null-ptr-deref in _raw_spin_lock (kernel/locking/spinlock.c:158) Write of size 4 at addr 00000000000004b4	N/A	More

89780	by task exploit/144 Call Trace: _raw_spin_lock (kernel/locking/spinlock.c:158) netif_tx_lock (net/sched/sch_generic.c:497) rmnet_map_command (drivers/net/ethernet/qualcomm/rmnet/rmnet_map_command.c:67) rmnet_rx_handler (drivers/net/ethernet/qualcomm/rmnet/rmnet_handlers.c:125) __netif_receive_skb_core.constprop.0 (net/core/dev.c:6103) ... __netif_receive_skb_one_core (net/core/dev.c:6214) netif_receive_skb (net/core/dev.c:6474) tun_get_user (drivers/net/tun.c:1966) tun_chr_write_iter (drivers/net/tun.c:2012) vfs_write (fs/read_write.c:687) ksys_write (fs/read_write.c:739) do_syscall_64 (arch/x86/entry/syscall_64.c:94) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:121) Kernel panic - not syncing: Fatal exception in interrupt		Details
CVE-2026-89776	In the Linux kernel, the following vulnerability has been resolved: vxlan: vnifilter: enforce exact length of GROUP/GROUP6 attributes The VXLAN VNI filter entry policy declares the GROUP/GROUP6 address attributes as NLA_BINARY with only a maximum length, so validate_nla() accepts a payload shorter than the address. The GROUP consumer reads it with nla_get_in_addr(), an unconditional 4-byte load, so a short attribute over-reads up to 3 bytes of uninitialised slab data, which are stored into remote_ip and echoed back via RTM_GETTUNNEL, disclosing kernel memory. Switch both entries to NLA_POLICY_EXACT_LEN() so the validator rejects any GROUP/GROUP6 that is not exactly 4 / 16 bytes; a valid address is always sent at full width.	N/A	More Details
CVE-2026-89850	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Don't query firmware state while chip is down qla2x00_fw_state_show() initializes rval to QLA_FUNCTION_FAILED and jumps to the out: label when the chip is down or EEH is busy. The out: block then re-issued qla2x00_get_firmware_state() because rval != QLA_SUCCESS, defeating the chip-down/EEH-busy guards and issuing a mailbox command (outside optrom_mutex) during ISP reset or PCI error recovery, which can hang the adapter. It also turned a normal in-lock mailbox failure into a second unsynchronized mailbox attempt. Make the out: fallback only mark the firmware state as unknown. The mailbox is now issued at most once, inside optrom_mutex, and only when the chip is up and not EEH-busy.	N/A	More Details
CVE-2026-54237	Wavelog is web-based amateur radio logging software. From 1.8 until 2.4.2, Wavelog exposes /install/ajax.php and /install/includes/interface_assets/triggers.php after installation without an installation lock or permission check. Unsanitized input reaches write_config() and write_configfile() in install/includes/core/core_class.php, allowing a remote unauthenticated attacker to read or write log files and place attacker-controlled content into PHP configuration files. The resulting PHP configuration content can execute on the server. This issue is fixed in version 2.4.2.	N/A	More Details
CVE-2026-89946	In the Linux kernel, the following vulnerability has been resolved: ASoC: cs35l33: drain threaded IRQ before runtime suspend cs35l33_runtime_suspend() currently switches the codec into regcache_cache_only(true) and powers it down without first quiescing the threaded IRQ registered by devm_request_threaded_irq(). That leaves a window where cs35l33_irq_thread() can still run after suspend has closed off live register access. A running system can reach this during runtime PM while the driver still has critical fault IRQs unmasked. If the threaded handler runs in that window, it reads volatile INT_STATUS_1/2 after cache_only has been enabled, ignores the regmap_read() failures, and can still drive the AMP_SHORT_RLS, CAL_ERR_RLS, OTE_RLS, and OTW_RLS release paths. Use disable_irq() before entering cache_only/power-off so any in-flight threaded handler is drained and no new IRQ thread can run during the suspended state. Re-enable the IRQ only after runtime_resume() has restored live register access with regcache_sync(). Since probe only warns if devm_request_threaded_irq() fails, track whether the IRQ was actually installed before disabling or re-enabling it.	N/A	More Details
CVE-2026-93295	MISP contains a vulnerability in its background job dispatch mechanism that allows remote code execution as the web user. Background job arguments are passed directly as the argv of the CakePHP console process. CakePHP's ShellDispatcher::_parsePaths() scans the entire argv for path switches (-app, --app, -working, --working, -root, --root, -webroot, --webroot) and uses the following element as the application root. The events/contact endpoint passes user-controlled fields (message and person) into job arguments without validation. An attacker who can submit the contact form can set the person field to a reserved switch and the message field to a phar:// URI pointing to a malicious archive. The CakePHP bootstrap then includes Config/core.php from within that archive, executing attacker-controlled PHP code with the privileges of the web user. The vulnerability requires the ability to submit the events/contact form (or any other endpoint that forwards user input into background job arguments). No special timing or race condition is required; the attack is deterministic once the crafted parameters are accepted. The impact is full remote code execution in the context of the MISP web server process, allowing data exfiltration, persistence, and lateral movement within the host.	N/A	More Details
CVE-2026-93198	In the Linux kernel, the following vulnerability has been resolved: dm-pcache: validate the persisted dirty_tail chain at load The writeback worker follows the persisted dirty_tail chain, which is decoded from the cache device independently of the key_tail chain that cache_replay() walks and bounds. A crafted image, whose on-media fields are authenticated only by a crc32c with a fixed seed, can aim dirty_tail at a chain of last ksets that never terminates, so cache_writeback_fn() re-arms itself with no delay forever. Walk the dirty_tail chain once at load with the same hop cap cache_replay() uses and fail the table load with -EIO if it does not reach an end within n_segs hops.	N/A	More Details
CVE-2026-93199	In the Linux kernel, the following vulnerability has been resolved: i3c: master: Do not treat master device as a duplicate target i3c_master_search_i3c_dev_duplicate() searches the bus for another I3C device with the same PID as the reference device. The search can match master->this, causing the controller itself to be returned as a duplicate. Since the controller is not a target device, it cannot be a duplicate of one. Exclude master->this from matching so that the function only returns real duplicate target devices.	N/A	More Details
CVE-2026-93200	In the Linux kernel, the following vulnerability has been resolved: i3c: master: Fix use-after-free of master->this sysfs attribute callbacks for the master controller device dereference master->this. However, master->this is freed in i3c_master_detach_free_devs() before the master device itself is released. As a result, sysfs accesses can dereference a freed master->this pointer, leading to a use-after-free. Keep master->this alive until i3c_masterdev_release(), which is called after the master device and its sysfs state are being torn down. Do not free master->this as part of the normal device detach path. On the error path in i3c_master_set_info(), reset master->this and bus.cur_master to NULL before freeing the allocated device.	N/A	More Details
CVE-2026-89934	In the Linux kernel, the following vulnerability has been resolved: iio: light: ltrf216a: fix runtime PM reference leak in error path ltrf216a_get_lux() acquires a runtime PM reference by calling ltrf216a_set_power_state(data, true). However, if ltrf216a_read_data() fails, the function returns immediately without dropping the reference. This leaves the runtime PM usage count unbalanced, preventing the device from autosuspending after a failed read. Fix this by releasing the runtime PM reference before returning from the error path.	N/A	More Details

CVE-2026-93202	In the Linux kernel, the following vulnerability has been resolved: i3c: master: Fix recursive locking during device registration i3c_master_register_new_i3c_devs() registers newly discovered devices while holding i3c_bus_normaluse_lock(), a down_read(). device_register() can immediately probe the device, and probe callbacks typically invoke I3C helpers that take i3c_bus_normaluse_lock() again, leading to a recursive acquisition of the same rwsem. rwsems do not support recursive read locking and can deadlock when a writer is waiting. See the "Recursive read locks" section of Documentation/locking/lockdep-design.rst. For example, with Intel LPSS I3C, LOCKDEP generates a WARNING like: # echo intel-lpss-i3c.0 > /sys/bus/platform/drivers/mipi-i3c-hci/unbind # echo intel-lpss-i3c.0 > /sys/bus/platform/drivers/mipi-i3c-hci/bind WARNING: possible recursive locking detected kworker/5:1/94 is trying to acquire lock: ffff88811c810d78 (&i3cbus->lock){++++}-{4:4}, at: i3c_device_match_id+0x45/0x370 but task is already holding lock: ffff88811c810d78 (&i3cbus->lock){++++}-{4:4}, at: i3c_master_reg_work_fn+0x21/0x5f0 Fix this by separating device creation from device registration. Populate desc->dev under the maintenance lock, collect the devices that still need registration into a local list, then release the lock before calling device_register(). Finally retake the lock and clean up any devices that failed to register. Use the maintenance lock rather than the normal-use lock while adding device objects. A write-side maintenance lock prevents readers from observing a partially initialized desc->dev during initial device population, or desc->dev disappearing if registration fails. The local list requires a list node, so add a list node member to struct i3c_device.	N/A	More Details
CVE-2026-89933	In the Linux kernel, the following vulnerability has been resolved: iio: pressure: dps310: fix NULL pointer dereference on ACPI probe When the device is enumerated through its ACPI HID (IFX3100), i2c_client_get_device_id() returns NULL: the ACPI-derived client name does not match the driver's i2c_device_id table. dps310_probe() then dereferences that NULL pointer in "iio->name = id->name" and crashes the kernel during probe. The IIO device name is always "dps310", so set it directly and drop the now-unused device-id lookup.	N/A	More Details
CVE-2026-93204	In the Linux kernel, the following vulnerability has been resolved: batman-adv: dat: atomically update mac addresses When a MAC address is updated in batadv_dat_entry_add(), it is done using a simple copy function. A parallel reader might only see parts of this update. In worst case, the reader is transporting the half updated MAC address over the network or is creating an ARP response using it - poisoning the ARP cache. atomic64_t can be used to store the 48 bit of a mac address. A reader will then either see the old mac address or the new one - never a mixture of both.	N/A	More Details
CVE-2026-89931	In the Linux kernel, the following vulnerability has been resolved: KVM: nVMX: Ensure KVM_REQ_GET_NESTED_STATE_PAGES is cleared on VM-Exit Always check and clear KVM_REQ_GET_NESTED_STATE_PAGES when emulating a nested VM-Exit to ensure the request is cleared, even when KVM was built with CONFIG_KVM_HYPERV=n, as KVM subtly relies on the "check" to clear the flag and thus avoid double-mapping the vmcs12 pages, e.g. if KVM manages to bail from VM-Enter without processing the request, and then emulates VMLAUNCH or VMRESUME.	N/A	More Details
CVE-2026-93296	MISP contains a stored cross-site scripting (XSS) vulnerability in the Overmind theme's statistics views. The event General card and the server/feed preview card constructed donut chart legend labels by directly concatenating object name or category keys into an innerHTML string without HTML-encoding. Because MISP object names are user-controllable by any authenticated user with sufficient permissions to create or modify such objects, an attacker could craft a name containing malicious markup. When any other user viewed the affected Overmind dashboard, the injected markup would be interpreted as live HTML/JavaScript in the victim's browser, executing in the context of the MISP application origin. This could allow session hijacking, data exfiltration, or arbitrary actions performed on behalf of the victim. The vulnerability requires the attacker to have low-level authenticated access to create or rename an object whose name is rendered in the legend, and the victim to view the Overmind event or server preview page. No special browser conditions or race conditions are required.	N/A	More Details
CVE-2026-89852	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Zero mailbox struct in qla2x00_get_firmware_state() The mbx_cmd_t is allocated on the stack but left uninitialized. qla2x00_mailbox_command() has several early-return paths (PCI permanent failure, device failed, EEH busy, ISP abort pending, mailbox access timeout, purge mbox) that return without writing the input mailbox registers back into mcp->mb[]. qla2x00_get_firmware_state() then unconditionally copies mcp->mb[1..6] (and mb[12]) into the caller's states[] array regardless of the return value. On such a failure the copied values are uninitialized kernel stack memory, which is then exposed to userspace via the fw_state and mpi_fw_state sysfs handlers. Zero the mailbox struct so a failed query yields deterministic zeroed state instead of leaking stack contents.	N/A	More Details
CVE-2026-9314	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89926	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Fix length check __import_wp_info() struct kvm_hw_breakpoint::len is a __u64 that is fully controlled by user space. This is then assigned to wp_info->len, which is an int. The bounds check is done on the truncated value while the allocation uses the untruncated one: wp_info->len = bp_data->len; [...] if (wp_info->len < 0 || wp_info->len > MAX_WP_SIZE) return -EINVAL; wp_info->old_data = kmalloc(bp_data->len, GFP_KERNEL_ACCOUNT); Use the validated value for the allocation as intended. Without this fix userspace can trigger >4GB allocations which will fail and result in a WARN due to MAX_PAGE_ORDER.	N/A	More Details
CVE-2026-89925	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Fix memory leak in guest debug handling bp_data is freed only for the error case by kfree(bp_data). Every successful KVM_SET_GUEST_DEBUG will leak bp_data.	N/A	More Details
CVE-2026-52836	OpenDDS is an open source C++ implementation of the Object Management Group (OMG) Data Distribution Service (DDS). Prior to 3.34.0, a network attacker can crash a reachable OpenDDS participant by sending a malformed RTPS UDP submessage whose crafted length or sequence-number state causes dds/DCPS/transport/rtps_udp/RtpsUdpReceiveStrategy.cpp in RtpsUdpReceiveStrategy::handle_input() to advance ACE_Message_Block::rd_ptr() beyond valid data. The parser can then call dds/DCPS/transport/rtps_udp/RtpsSampleHeader.cpp in RtpsSampleHeader::init(), which dereferences the invalid read pointer without first validating it against wr_ptr() or ensuring that a complete submessage header remains. The resulting SIGSEGV occurs in the receive thread, terminates the DDS process, and destroys the DDS entities hosted by that participant. No authentication, prior protocol state, or victim interaction is required. This issue is fixed in version 3.34.0.	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Fix old_data leak in guest debug error path __import_wp_info() allocates a per-watchpoint old_data buffer to back up the original guest memory contents. If a later watchpoint of the same KVM_SET_GUEST_DEBUG request fails to import, kvm_s390_import_bp_data() jumps to the error label,	N/A	More

89924	which frees the wp_info array but not the old_data buffers of the entries that were imported successfully. Up to MAX_BP_COUNT - 1 buffers of up to MAX_WP_SIZE bytes are leaked per failed request, and the request can be repeated. Create error handling for cleaning up all created old_data memory areas.		Details
CVE-2026-89923	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Free guest debug data on vcpu destroy kvm_s390_clear_bp_data() is only called from kvm_arch_vcpu_ioctl_set_guest_debug(), i.e. when user space changes or disables debugging. A vCPU that is destroyed while hardware breakpoints are still armed - the normal case when the VMM just exits or crashes - leaks hw_bp_info, hw_wp_info and all old_data buffers, since generic KVM frees the vCPU right after kvm_arch_vcpu_destroy(). That is bounded by MAX_BP_COUNT entries, so roughly 8 KiB per vCPU, but it is unbounded over VM lifetimes. The allocations are GFP_KERNEL_ACCOUNT, so the charge also outlives the exiting process and pins dying memcgs. Fix by clearing the debug data on vCPU destruction. Calling it unconditionally is fine: struct kvm_vcpu is zero allocated, so for a vCPU that never enabled debugging the counters are 0 and the pointers NULL.	N/A	More Details
CVE-2026-89921	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: Zero initialize data structures for inject_pfault_token __kvm_inject_pfault_token() only sets .type and .u.ext.ext_params2 of the on-stack struct kvm_s390_irq but the full ext substructure is copied into the cpu local variable on inject. ext_params and pad contain stale stack values. Interrupt delivery only uses ext_params2, so nothing leaks to the guest, but a host user can use the migration ioctls to get to the data. Fix by zero-initializing the irq struct. Do the same for the inti data structure.	N/A	More Details
CVE-2026-89917	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Handle VNCR TLB invalidation race with vcpu_put() VNCR unmapping While VNCR TLB invalidation always occurs under the MMU lock, vcpu_put() doesn't, while it unmaps the VNCR page. The problem is that the invalidation evaluates vncr_tlb::cpu to decide whether an unmapping needs to take place (cpu != -1) before performing it. On the other hand, this_cpu_reset_vncr_fixmap() unconditionally unmaps if L1_VNCR_MAPPED is set. These two obviously can race, with a TOCTOU pattern on the TLBI path, and a BUG_ON() on the vcpu_put() path. And the two can end-up calling vncr_fixmap(-1), with extra lethal effects. Move the reset of vncr_tlb::cpu to -1 to a common function, and make this update atomic so that only a single thread can reset the field and perform the corresponding unmap. The vcpu_put() still need to unconditionally unmap the current VNCR to close another ugly race. Finally, the assignment of vncr_tlb::cpu is moved to be kept in sync with the actual mapping, similar to L1_VNCR_MAPPED being set.	N/A	More Details
CVE-2026-93197	In the Linux kernel, the following vulnerability has been resolved: memcg: move LRU size accounting on reparenting instead of copying it When a memory cgroup is offlined its LRU folios are reparented to the parent. lruvec_reparent_lru() splices the child's lists into the parent's and credits the parent with the child's per-zone lru_zone_size[], but never clears the child's copy, so the size is copied rather than moved. lru_gen_reparent_memcg() does the same for MGLRU. The parent is left correct, credited with exactly the folios it took over. The stale value sits on the child and nothing will correct it: folio->memcg_data now resolves to the parent, so every later update_lru_size() for those folios goes there. Dying cgroups are not freed immediately and mem_cgroup_iter() still walks them, so shrink_lruvec() keeps being called on them. get_scan_count() reads the phantom counter through lruvec_lru_size() and the scan loop then grinds through nr[] in SWAP_CLUSTER_MAX steps against an empty list, for as long as the dead cgroup lives. Under MGLRU the MGLRU scanner runs instead, but count_shadow_nodes() sums all of NR_LRU_LISTS through lruvec_lru_size() and over-budgets the shadow node limit just the same. On one 251 GiB host a sweep of every mz->lru_zone_size[] found 380 counters describing folios on no list at all: 124777314 pages, 476 GiB, 1.89x the machine's RAM, across 57 cgroups. All were on memcgs with CSS_DYING set and CSS_ONLINE clear, and parent/child pairs reported byte-identical sizes. LRU_UNEVICTABLE needs its size moved too. Its list is deliberately not spliced because lruvec_init() poisons the head - the unevictable LRU is imaginary and folios are never threaded on it - but the size is kept by lruvec_add_folio()/lruvec_del_folio() and those folios account to the parent from here on. This depends on commit bf4ade7dbd76 ("memcg: keep folio's objcg same as its node") and must not be backported ahead of it. Without that invariant a folio's objcg can belong to another node, so a folio already spliced onto the parent's list can still resolve to the child's lruvec until the objcg's node is reparented in a later iteration of memcg_reparent_objcgs(); clearing the child's counter early then lets lruvec_del_folio() underflow it and trip the WARN_ONCE()/VM_BUG_ON() in mem_cgroup_update_lru_size().	N/A	More Details
CVE-2026-89935	In the Linux kernel, the following vulnerability has been resolved: iio: light: apds9306: fix PM reference leak in apds9306_read_data() apds9306_read_data() calls pm_runtime_resume_and_get() but several error paths return directly without calling pm_runtime_put_autosuspend(), leaking the runtime PM reference and preventing the device from autosuspending. Use PM_RUNTIME_ACQUIRE_AUTOSUSPEND() and PM_RUNTIME_ACQUIRE_ERR() to automatically handle runtime PM reference release on all return paths.	N/A	More Details
CVE-2026-93195	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: synopsys: dw-dp: Support unregistering the AUX channel The DisplayPort AUX channel gets initialized and registered during dw_dp_bind(), but it is never unregistered, which may lead to resource leaks and/or use-after-free. Add the missing dw_dp_unbind() function to allow the users of the library to handle the required cleanup, i.e. unregister the AUX adapter.	N/A	More Details
CVE-2026-93194	In the Linux kernel, the following vulnerability has been resolved: drm/rockchip: dw_dp: Release core resources Core resources such as the DisplayPort AUX channel get initialized and registered during dw_dp_bind(), but are never unregistered, which may lead to memory leaks and/or use-after-free: [224.661371] BUG: KASAN: slab-use-after-free in device_is_dependent+0xe0/0x2b0 [224.662015] Read of size 8 at addr ffff00011aee8550 by task modprobe/658 [224.662612] [224.662752] CPU: 7 UID: 0 PID: 658 Comm: modprobe Not tainted 7.0.0-rc2-next-20260305 #14 PREEMPT [224.662759] Hardware name: Radxa ROCK 5B (DT) [224.662762] Call trace: [224.662764] show_stack+0x20/0x38 (C) [224.662772] dump_stack_lvl+0x6c/0x98 [224.662777] print_report+0x160/0x4b8 [224.662783] kasan_report+0xb4/0xe0 [224.662790] __asan_report_load8_noabort+0x20/0x30 [224.662796] device_is_dependent+0xe0/0x2b0 [224.662802] device_is_dependent+0x108/0x2b0 [224.662808] device_link_add+0x1f8/0x10b0 [224.662813] devm_of_phy_get_by_index+0x120/0x200 [224.662819] dw_dp_bind+0x34c/0xb10 [dw_dp] [224.662830] dw_dp_rockchip_bind+0x194/0x250 [rockchipdrm] [224.662864] component_bind_all+0x3a8/0x720 [224.662869] rockchip_drm_bind+0x120/0x390 [rockchipdrm] [224.662899] try_to_bring_up_aggregate_device+0x76c/0x838 [224.662904] component_master_add_with_match+0x1f4/0x230 [224.662909] rockchip_drm_platform_probe+0x420/0x538 [rockchipdrm] [224.662939] platform_probe+0xe8/0x168 [224.662945] really_probe+0x340/0x828 [224.662950] __driver_probe_device+0x2e0/0x350 [224.662954] driver_probe_device+0x80/0x140 [224.662959] __driver_attach+0x398/0x460 [224.662964] bus_for_each_dev+0xe0/0x198 [224.662968] driver_attach+0x50/0x68 [224.662972] bus_add_driver+0x2a0/0x4c0 [224.662977] driver_register+0x294/0x360 [224.662982] __platform_driver_register+0x7c/0x98 [224.662987] rockchip_drm_init+0xc4/0xff8 [rockchipdrm] Since a previous commit exported dw_dp_unbind() function in DW DP core library to take care of the necessary cleanup, use this in the component's	N/A	More Details

	unbind() callback, as well as in its bind() error path.		
CVE-2026-89945	In the Linux kernel, the following vulnerability has been resolved: ASoC: cs35l34: drain threaded IRQ before runtime suspend cs35l34_runtime_suspend() currently switches the codec into regcache_cache_only(true), asserts reset low, and powers the device off without first quiescing the threaded IRQ registered by devm_request_threaded_irq(). That leaves a window where cs35l34_irq_thread() can still run after suspend has removed live hardware access. A running system can reach this during runtime PM while the driver still has critical fault IRQs unmasked. If the threaded handler runs in that window, it reads volatile INT_STATUS_1..4 after cache_only has been enabled, ignores the regmap_read() failures, and can still execute the PROT_RELEASE_CTL release sequence or the BST fault power-down writes. Use disable_irq() before entering cache_only/reset-low/power-off so any in-flight threaded handler is drained and no new IRQ thread can run while the device is suspended. Re-enable the IRQ only after runtime_resume() has restored live register access with regcache_sync(). Since probe only logs request_threaded_irq() failures and keeps going, track whether the IRQ was actually installed before disabling or re-enabling it.	N/A	More Details
CVE-2026-89944	In the Linux kernel, the following vulnerability has been resolved: ASoC: hdac_hda: Fix hlink refcount leak on component registration failure hdac_hda_dev_probe() gets the HDA link with snd_hdac_ext_bus_link_get() before registering the ASoC component. If component registration fails, the function returns without dropping the link reference. Always call snd_hdac_ext_bus_link_put() after the registration attempt so the reference taken during probe is balanced on both success and failure.	N/A	More Details
CVE-2026-93179	In the Linux kernel, the following vulnerability has been resolved: drm/amd/powerplay: fix VoltageObjectInfo zero-stride loop and OOB read Reject voltage objects whose usSize is smaller than the header or would advance the cursor past the table end, preventing an infinite loop or heap OOB read when the VBIOS supplies a malformed VoltageObjectInfo table.	N/A	More Details
CVE-2026-93180	In the Linux kernel, the following vulnerability has been resolved: drm/panthor: Fix NPD issue on partial unmap of an evicted BO This commit fixes the NULL pointer dereference issue that would have happened on the split of GPU mapping due to partial unmap of an evicted BO. There is a logic to handle the partial unmap of huge pages when the GPU mapping is split. That logic was not being completely skipped for the VMA of an evicted BO and that resulted in a NPD possibility for the 'bo->backing.pages' pointer, which is set to NULL when pages of a BO are released on eviction. Following dump was seen when a partial unmap was exercised for an evicted BO. Unable to handle kernel paging request at virtual address 0000000000002000 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtable: 4k pages, 48-bit VAs, pgdp=00000008842e8000 [0000000000002000] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] SMP <snip> pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : iova_mapped_as_huge_page+0x20/0x68 [panthor] lr : panthor_gpuva_sm_step_remap+0x39c/0x498 [panthor] sp : ffff800086193920 x29: ffff800086193920 x28: ffff800086193a18 x27: ffff800086193b80 x26: 0000000000400000 x25: 0000000000810000 x24: 0000000000400000 x23: ffff000808af1800 x22: 0000000000a00000 x21: ffff800086193a00 x20: ffff000806fd3f00 x19: 0000000000410000 x18: 00000000ffffffff x17: 0000000000000000 x16: 0000000000000000 x15: ffff800083ce2d83 x14: 0000000000000000 x13: 3120646574636976 x12: 6520303030303138 x11: 2d303030303313420 x10: ffff8000836e6c80 x9 : ffff80007bfc889c x8 : 3fffffffffffefff x7 : ffff8000836e6c80 x6 : 0000000000000000 x5 : ffff00097ef19088 x4 : 0000000000000000 x3 : 0000000000000000 x2 : 0000000000010000 x1 : 0000000000000400 x0 : 0000000000000000 Call trace: iova_mapped_as_huge_page+0x20/0x68 [panthor] (P) op_remap_cb.isra.0+0x70/0xb0 __drm_gpuvm_sm_unmap+0xf8/0x1c0 drm_gpuvm_sm_unmap+0x40/0x60 panthor_vm_exec_op+0xa0/0x168 [panthor] panthor_vm_bind_exec_sync_op+0x8c/0xb8 [panthor] panthor_ioctl_vm_bind+0xbc/0x170 [panthor] drm_ioctl_kernel+0xc0/0x140 drm_ioctl+0x20c/0x500 __arm64_sys_ioctl+0xb4/0x118 invoke_syscall+0x5c/0x120 el0_svc_common.constprop.0+0x48/0xf8 do_el0_svc+0x28/0x40 el0_svc+0x38/0x128 el0t_64_sync_handler+0xa0/0xe8 el0t_64_sync+0x198/0x1a0 Code: 8b030021 cb020021 f940b800 d34cfc21 (f8617801) ---[end trace 0000000000000000]--- v2: Fix indentation	N/A	More Details
CVE-2026-93181	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/uncore: Fix uncore_box ref/unref ordering In uncore_event_cpu_online(), uncore_box_ref() was called before uncore_change_context(). uncore_box_ref() gates on box->cpu >= 0, but box->cpu is still -1 at that point because uncore_change_context() has not run yet. As a result, the box is never initialized on the first CPU to come online in a die, leaving it permanently uninitialized in the single-CPU-per-die case. Thus, box->refcnt is one count below the true value, and in the CPU offline path, the box will be torn down on the second-to-last CPU. In uncore_event_cpu_offline(), uncore_box_unref() was called after uncore_change_context(), so box->cpu is already -1 when the collector CPU goes offline, which prevents it from tearing down the box. Fix by swapping the call order in both paths so that uncore_box_{ref,unref}() runs at the point where box->cpu reflects the correct context. Move allocate_boxes() out of uncore_box_ref() to enable this reordering.	N/A	More Details
CVE-2026-93182	In the Linux kernel, the following vulnerability has been resolved: sched/fair: Fix overflow in update_tg_cfs_runnable() A divide-by-zero crash is observed when running hackbench: [14697.488452] CPU: 112 UID: 0 PID: 124791 Comm: hackbench Not tainted 7.1.0-rc2+ [14697.492627] RIP: 0010:propagate_entity_load_avg+0x35f/0x3e0 [14697.506799] <TASK> [14697.507411] __dequeue_task+0x2b4/0xc70 [14697.508677] dequeue_task_fair+0x36/0x370 [14697.509047] dequeue_task+0x101/0x2f0 [14697.509426] __schedule+0x1b1/0x1a00 [14697.510868] anon_pipe_read+0x3da/0x450 [14697.511400] vfs_read+0x361/0x390 [14697.512053] __x64_sys_read+0x19/0x30 The divide-by-zero happens here: if (scale_load_down(gcfs_rq->load.weight)) { load_sum = div_u64(gcfs_rq->avg.load_sum, scale_load_down(gcfs_rq->load.weight)); } gcfs_rq->load.weight is an insane large value and is truncated to the lower 32 bits by div_u64, which happen to be 0. Using AI for investigation, the cause is a u32 overflow in update_tg_cfs_runnable(), and flat pickup became a victim when using tg_tasks(): u32 new_sum, divider; ... new_sum = se->avg.runnable_avg * divider; <-- boom The following sequence shows how this triggers the crash: propagate_entity_load_avg() update_tg_cfs_runnable() # u32 overflow corrupts runnable_sum __update_load_avg_cfs_rq() ___update_load_avg() # computes insane runnable_avg update_tg_load_avg() # propagates to tg->runnable_avg update_cfs_group() calc_concur_shares() tg_tasks() # long-to-int truncation, negative nr reweight_entity() # corrupted se->load.weight update_load_add() # corrupted cfs_rq->load.weight propagate_entity_load_avg() update_tg_cfs_load() div_u64() # divide-by-zero Fix by widening new_sum from u32 to u64 (no need to force tg_tasks() to return unsigned long after this fix)	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: drm/lima: call drm_mm_init() with a valid allocation range lima_vm_create() is currently run before va_start and va_end are set up, meaning they are both 0. lima_vm_create() runs drm_mm_init() with them as arguments for the allocator, and if DRM_DEBUG_MM is enabled the DRM_MM_BUG_ON check in drm_mm_init then fires, as seen here on exynos4412-odroid-u2: [1.736297] ------------[cut here]------------ [1.740370] kernel		More

2026-93183	BUG at drivers/gpu/drm/drm_mm.c:931! [1.745574] Internal error: Oops - BUG: 0 [#1] SMP ARM [1.750697] Modules linked in: [1.753734] CPU: 0 UID: 0 PID: 41 Comm: kworker/u16:1 Not tainted 7.0.10-postmarketos-exynos4 #11 PREEMPT [1.763372] Hardware name: Samsung Exynos (Flattened Device Tree) [1.769446] Workqueue: events_unbound deferred_probe_work_func [1.775261] PC is at drm_mm_init+0x9c/0xa4 [1.779339] LR is at lima_vm_create+0x144/0x17c [...] Fix the issue by moving the lima_vm_create() call after va_start and va_end are set up.	N/A	Details
CVE-2026-93184	In the Linux kernel, the following vulnerability has been resolved: ASoC: fsl_audmix: rework runtime PM handling in probe After pm_runtime_enable() the AUDMIX block is powered off and stays suspended until the first runtime resume. Register writes issued between probe() and the first resume (e.g. from DAPM or ALSA control paths) target unpowered hardware and cause a system hang. Fix this by calling pm_runtime_resume_and_get() immediately after pm_runtime_enable() to power the hardware up and enable its clocks. Release the reference afterwards with pm_runtime_put() to allow the runtime PM framework to suspend the device and switch the regmap to cache-only mode when idle. When CONFIG_PM is disabled or runtime PM is not enabled, pm_runtime_* calls are stubs that do not power up the hardware. Handle this case explicitly by calling fsl_audmix_runtime_resume() directly so the hardware is always initialised and its clocks are enabled, ensuring register accesses succeed regardless of PM configuration.	N/A	More Details
CVE-2026-93185	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt700-sdw: always drain jack work on remove rt700_sdw_remove() drains jack_detect_work and jack_btn_check_work only when rt700->hw_init is true. That state bit is cleared by rt700_update_status() when the SoundWire slave becomes UNATTACHED, but a jack work item can already have been queued by rt700_interrupt_callback() or rt700_jack_init() while the device was initialized. Do not use hw_init as the remove-time guard for draining these work objects. The delayed works are initialized during rt700_init(), so remove can cancel them unconditionally and pair the object lifetime with the codec-private data lifetime instead of a mutable hardware state bit. This issue was found by our static analysis tool and then confirmed by manual review of the SoundWire status, interrupt and remove paths. The remove path should drain work based on whether the work object exists, not on a runtime hardware state bit that can change after the work was queued. A QEMU PoC queued jack_detect_work, simulated SDW_SLAVE_UNATTACHED, and then entered remove. DEBUG_OBJECTS reported an active timer/work object associated with the rt700 jack work path after remove skipped the cancel. This is sent as an RFC because the practical trigger depends on SoundWire core remove ordering after an UNATTACHED status update. If remove cannot run after hw_init has been cleared while jack work is still pending, this is a defensive lifecycle cleanup rather than a reachable race on current systems.	N/A	More Details
CVE-2026-93186	In the Linux kernel, the following vulnerability has been resolved: cxl/mbox: Clamp mailbox output allocation to the payload size CXL_MEM_SEND_COMMAND bounds the user's in.size to the mailbox payload size but leaves out.size unbounded, then cxl_mbox_cmd_ctor() calls kvzalloc(out.size). A large out.size drives a huge allocation, above INT_MAX it WARNs and taints, and with panic_on_warn=1 it panics. The transport __cxl_pci_mbox_send_cmd() already clamps the response copy to min(out.size, payload_size, device len), so the output buffer is never written beyond payload_size. Clamp the allocation to payload_size too, matching the RAW path.	N/A	More Details
CVE-2026-93187	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Return error for invalid number of formats When the number of input or output formats is zero, sof_ipc4_widget_setup_comp_src() and sof_ipc4_widget_setup_comp_asrc() print an error and jump to the cleanup label. At that point 'ret' is still 0, because the earlier sof_ipc4_get_audio_fmt() call succeeded, so the function returns success and the caller never finds out that the widget setup actually failed. Set ret to -EINVAL before the goto so the error gets reported.	N/A	More Details
CVE-2026-93188	In the Linux kernel, the following vulnerability has been resolved: HID: roccat: bound device-supplied profile index kone_keep_values_up_to_date() and kone_profile_activated() use an 8-bit, device-supplied profile value as an index into the 5-element kone->profiles[] array without a range check. A malicious USB device claiming the Roccat Kone id can send a switch-profile event (or a startup_profile read at probe) with an out-of-range value and make the driver read out of bounds; the result is exposed via the actual_dpi sysfs attribute. Reject out-of-range indices in both paths. This was found with static analysis and confirmed with the KUnit test added in the following patch (KASAN: slab-out-of-bounds).	N/A	More Details
CVE-2026-89939	In the Linux kernel, the following vulnerability has been resolved: iio: chemical: atlas-sensor: fix PM reference leak in buffer postenable atlas_buffer_postenable() acquires a runtime PM reference with pm_runtime_resume_and_get() but returns the result of atlas_set_interrupt() directly. If atlas_set_interrupt() fails, the runtime PM reference is leaked and the device can never autosuspend. Add pm_runtime_put_autosuspend() on the error path to balance the reference.	N/A	More Details
CVE-2026-89937	In the Linux kernel, the following vulnerability has been resolved: iio: chemical: sgp30: Handle IAQ thread creation failure kthread_run() can fail and return an error pointer, but sgp_probe() stores it and returns success, so the device is registered without its IAQ thread and sgp_remove() later passes the error pointer to kthread_stop(). Return the error from probe instead.	N/A	More Details
CVE-2026-93191	In the Linux kernel, the following vulnerability has been resolved: smack: fix incorrect task context in smack_msg_queue_msgrcv The smack_msg_queue_msgrcv() function incorrectly checks the permissions of the 'current' task instead of the 'target' task. In the msgsnd() syscall path, if a receiver is already waiting, the pipelined_send() optimization is used to push the message directly to the receiver task: ipc/msg.c`pipelined_send(): ` smp_store_release(&msr->r_msg, msg) In this case, the 'sender' (current) task performs the check on behalf of the 'receiver' task (msr->r_tsk, passed as the 'target' parameter): ipc/msg.c`pipelined_send(): ` security_msg_queue_msgrcv(,, target := msr->r_tsk,,) However, smack_msg_queue_msgrcv() ignores the 'target' and checks 'current': smack_msg_queue_msgrcv(...) ` smk_curacc_msq(isp, MAY_READWRITE); // current task 'current' MAY satisfy smack_msg_queue_msgrcv r/w requirement, but 'target' (the receiver task) might NOT; as a result, an unauthorized receiver gets the message, violating MAC policy. Test: 1) create a sysv message queue with label “foo” 2) echo "bar foo r" >/smack/load2 3) msgrcv(,,,0,MSG_NOERROR) in "bar"-labeled task. The task is waiting for the messages ... 4) msgsnd() from a "foo"-labeled task: "bar"-labeled task gets the message. This patch fixes the issue by checking permission on the 'target' task instead of 'current'. (2008-02-04, Casey Schaufler)	N/A	More Details
CVE-2026-89936	In the Linux kernel, the following vulnerability has been resolved: iio: dac: m62332: Fix regulator reference count imbalance m62332_set_value() enables the Vcc regulator on every write of a non-zero value and disables it on every write of zero, without tracking the channel's current state. Because the regulator is reference counted, changing a channel directly from one non-zero value to another enables it more than once, while a later write of zero disables it only once. The reference count never returns to zero and the regulator is left enabled indefinitely. Only enable the regulator on the transition from zero to non-zero, and only disable it on the transition from non-zero to zero, using the previously stored channel value to detect the edge. Balance the regulator on the I2C error path so the reference count stays consistent if the write fails.	N/A	More Details

CVE-2026-93193	In the Linux kernel, the following vulnerability has been resolved: drm/rockchip: analogix_dp: Fix OF node reference leak via auto cleanup Sashiko reported a reference leak in rockchip_dp_drm_encoder_enable(), the of_get_child_by_name() function does not call of_node_put() in a symmetrical way [1]. Fix the device node reference leak by using __free(device_node) to automatically manage of_node_put() for all device nodes.	N/A	More Details
CVE-2026-89909	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Free init resources if kvm_init() fails kvm_loongarch_init() calls kvm_loongarch_env_init() to allocate the per-CPU kvm_context (vmcs) and kvm_loongarch_ops and to register the perf callbacks, and then calls kvm_init(). If kvm_init() fails its result is returned directly, but since module_init() does not run the module_exit() stuff on failure, so kvm_loongarch_env_exit() is never called and those resources are leaked. So call kvm_loongarch_env_exit() when kvm_init() fails, matching the teardown-on-failure pattern used by riscv_kvm_init().	N/A	More Details
CVE-2026-89905	In the Linux kernel, the following vulnerability has been resolved: LoongArch: BPF: Move arena register slot below TCC context Currently, the stack layout places the optional arena register slot above the tail call counter context. When arena_vm_start is dynamically enabled, it shifts the relative offset of the tcc_ptr slot within the stack frame, causing hardcoded tracking macros to mismatch and leading to memory misalignment or corruption potentially. To fix this, move the arena register save and restore sequences below the tail call counter context slots in both build_prologue() and the epilogue. Update __build_epilogue() to insert a proper offset decrement to safely skip the unneeded tcc_ptr reading block while accurately aligning with the relocated arena slot at the very bottom. With this patch, the tcc_ptr slot is always positioned at a fixed distance directly underneath the base callee-saved registers that is independent of whether the arena features are on.	N/A	More Details
CVE-2026-89901	In the Linux kernel, the following vulnerability has been resolved: media: airspy: use vb2_video_unregister_device() on disconnect to fix NULL deref airspy_disconnect() clears s->udev under v4l2_lock, but airspy_stop_streaming() unconditionally calls airspy_ctrl_msg() and airspy_free_stream_bufs() afterwards. If a streaming user closes the device after disconnect, stop_streaming() runs and dereferences the NULL s->udev: airspy_stop_streaming() airspy_ctrl_msg(s, CMD_RECEIVER_MODE, 0, 0, NULL, 0) usb_sndctrlpipe(s->udev, 0) /* NULL deref */ airspy_free_stream_bufs(s) usb_free_coherent(s->udev, ...) /* NULL deref */ The airspy driver uses vb2_fop_release() in its file_operations, so replace video_unregister_device(&s->vdev) with vb2_video_unregister_device(&s->vdev) and move it before clearing s->udev. vb2_video_unregister_device() releases the vb2 queue, which synchronously runs airspy_stop_streaming() if streaming is active, so the URBs, coherent DMA stream buffers and the hardware stop control message all execute while s->udev is still valid. vb2_video_unregister_device() locks vdev->queue->lock (vb_queue_lock) internally, and stop_streaming() locks v4l2_lock, so the previous outer mutex_lock(&s->vb_queue_lock) / mutex_lock(&s->v4l2_lock) pair around the unregister sequence would self-deadlock and has been removed. A short v4l2_lock critical section around s->udev = NULL remains so any ioctl path that still holds the file descriptor sees coherent state. Issue identified by automated review of the INV-003 series at https://sashiko.dev/	N/A	More Details
CVE-2026-89872	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-fwnode: Fix fwnode leak in v4l2_fwnode_parse_link In v4l2_fwnode_parse_link(), the remote endpoint fwnode reference is acquired using fwnode_graph_get_remote_endpoint(). This reference is properly released in the error paths, but it is leaked on the success path. Add the missing fwnode_handle_put() before returning 0 to prevent the reference leak. [Sakari Ailus: Fix subject prefix and coding style a little.]	N/A	More Details
CVE-2026-57846	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89869	In the Linux kernel, the following vulnerability has been resolved: media: qcom: iris: use disable_irq() during power-off The IRQ is registered as a threaded IRQ. Using disable_irq_nosync() in iris_vpu_power_off() does not wait for an already queued threaded IRQ handler to complete before returning. As a result, a threaded IRQ handler may still run after the VPU has been powered down and access hardware registers after power-off. Replace disable_irq_nosync() with disable_irq() so the power-off path waits for any in-flight threaded IRQ handler to complete before returning.	N/A	More Details
CVE-2026-89868	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Add timeout while stop_streaming When stop_streaming is called, an infinite loop may occur in some cases. Add a bounded poll of the queue status: loop until the queues drain, sleeping briefly between polls, and bail out once VPU_DEC_STOP_TIMEOUT elapses.	N/A	More Details
CVE-2026-89867	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Defer job_finish() only when a DEC_PIC was queued Decoder instances sharing a VPU also share one v4l2_m2m job slot, released when the running context calls v4l2_m2m_job_finish(). While draining, device_run() defers job_finish() once EOS is sent (sent_eos), expecting a later finish_decode() (from a DEC_PIC completion IRQ) to release the slot. But the m2m core checks job_ready() only when a job is queued, not when it is dispatched. A job queued while draining can run after finish_decode() has already moved the instance to STOP and sent EOS. device_run() then runs in STOP, issues no DEC_PIC, yet still skips job_finish() - so no IRQ, no finish_decode(), and the shared slot is leaked, stalling every instance. With several v4l2h264dec instances in parallel, GStreamer hangs at EOS. Track whether the run actually queued a DEC_PIC (cmd_issued) and defer job_finish() only then. Otherwise finish the job immediately	N/A	More Details
CVE-2026-89866	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Resume device before setting EOS flag Setting the EOS flag talks to the firmware via send_firmware_command(), which accesses VPU registers. Both the STREAMOFF path (wave5_vpu_dec_job_abort()) and the V4L2_DEC_CMD_STOP path (wave5_vpu_dec_stop()) can run while the device is runtime suspended, so those register accesses hit powered-down hardware and the SoC raises an asynchronous SError, panicking the kernel: SError Interrupt on CPU3, code 0x00000000bf000000 -- SError send_firmware_command+0x2c/0x160 [wave5] wave5_vpu_dec_set_bitstream_flag+0x6c/0x80 [wave5] wave5_vpu_dec_update_bitstream_buffer+0x80/0xec [wave5] wave5_vpu_dec_job_abort+0x44/0xa0 [wave5] v4l2_m2m_cancel_job+0x110/0x19c [v4l2_mem2mem] v4l2_m2m_streamoff+0x24/0x140 [v4l2_mem2mem] Resume the device with pm_runtime_resume_and_get() around the EOS firmware command and release it with pm_runtime_put_autosuspend(), matching the runtime PM handling already done in wave5_vpu_dec_device_run().	N/A	More Details
CVE-2026-92509	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix potential use after free in counter_release() When accessing a counter via the netlink path the only synchronization mechanism for the said counter is rdma_restrack_get(). Currently, rdma_restrack_del() is invoked at the end of counter_release(), which is too late, since by that point vendor-specific resources associated with the counter might already be freed. This can leave a short window where the counter remains accessible through restrack, leading to a potential use-after-free. Fix this by moving the rdma_restrack_del() call to be before the freeing of the vendor-specific resources, ensuring that the counter is removed from restrack before its internal resources	N/A	More Details

	are released. This guarantees that no new users hold references to a counter that is in the process of destruction.		
CVE-2026-89864	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Bound i2c->length in I2C bsg handlers struct qla_i2c_access carries a 16-bit length field alongside a fixed 64-byte buffer: struct qla_i2c_access { uint16_t device, offset, option, length; uint8_t buffer[0x40]; } __packed; qla2x00_write_i2c() and qla2x00_read_i2c() use the user-supplied i2c->length without any bounds check. i2c is overlaid on a 256-byte on-stack buffer and sfp is a 256-byte DMA-pool buffer, so a length up to 65535 overruns both: - write: memcpy(sfp, i2c->buffer, i2c->length) over-reads the stack and over-writes the sfp heap buffer, and qla2x00_write_sfp() then DMAs i2c->length bytes out of the 256-byte buffer. - read: qla2x00_read_sfp() DMAs i2c->length bytes into the 256-byte sfp, then memcpy(i2c->buffer, sfp, i2c->length) overflows the 64-byte buffer inside the on-stack array. A caller holding CAP_SYS_RAWIO can use this to corrupt the heap and the kernel stack. Reject requests whose length exceeds the buffer before any copy or DMA transfer in both handlers.	N/A	More Details
CVE-2026-89862	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix BSG job leak on validate flash image error path qla28xx_validate_flash_image() returns QLA_SUCCESS (0) unconditionally, telling the FC BSG transport (fc_bsg_host_dispatch()) that the driver owns and will complete the request. But bsg_job_done() is guarded by "if (!rval)", so on the error path (rval == -EINVAL) neither the driver nor the transport completes the job. The request dangles until it times out, leaking block layer resources. Commit c2c68225b145 ("scsi: qla2xxx: Fix bsg_done() causing double free") added the "if (!rval)" guard to a batch of BSG handlers. That is correct for handlers that also return the error code (the transport then completes the job once via fail_host_msg), but this function returns QLA_SUCCESS unconditionally, so the guard turned a correct single completion into a leak. Always call bsg_job_done(): bsg_reply->result is DID_OK and the error is reported in vendor_rsp[0], and since the function returns 0 the transport will not complete the job a second time.	N/A	More Details
CVE-2025-62167	Rejected reason: ** REJECT ** DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2026-33626. Reason: This candidate is a duplicate of CVE-2026-33626. Notes: All CVE users should reference CVE-2026-33626 instead of this candidate.	N/A	More Details
CVE-2026-10594	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-11314	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89859	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Zero dport diagnostics buffer to avoid info leak qla2x00_do_dport_diagnostics() allocates the qla_dport_diag response buffer with kmalloc_obj() (non-zeroing) and, on success, copies the full sizeof(*dd) back to user space via sg_copy_from_buffer(). The inbound sg_copy_to_buffer() only fills as many bytes as the user request payload provides, and qla26xx_dport_diagnostics() zeroes only dd->buf. The options and unused[] fields are therefore copied out uninitialized, leaking kernel heap contents to user space. Allocate with kzalloc_obj(), matching qla2x00_do_dport_diagnostics_v2().	N/A	More Details
CVE-2026-48977	OpenSlide is a C library for reading whole slide image files. From 3.4.1 until 4.0.1, OpenSlide's parse_level0_xml() processing in src/openslide-vendor-ventana.c accepts nonpositive row or column tile counts from a crafted Ventana BIF file. The invalid counts produce attacker-controlled relative memory offsets and allow arbitrary values to be written at those offsets, affecting all supported platforms and configurations and resulting in a crash or potential arbitrary code execution. This issue is fixed in version 4.0.1.	N/A	More Details
CVE-2026-49137	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-89858	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Bound image count in qla2x00_update_fru_versions() qla2x00_update_fru_versions() copies the user-supplied BSG request into a fixed 256-byte stack buffer (bsg[DMA_POOL_SIZE]) and then iterates list->count times over the qla_image_version array embedded in that buffer, advancing the image pointer each iteration. count is taken directly from user input with no upper bound, while only (DMA_POOL_SIZE - sizeof(list->count)) / sizeof(struct qla_image_version) = 6 entries actually fit. A larger count walks the image pointer off the end of the stack buffer, reading adjacent kernel stack memory and sending it to the device via qla2x00_write_sfp(). Reject requests whose declared count does not fit in the buffer.	N/A	More Details
CVE-2026-89855	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Serialize flash version read in reset handler The "update cache versions without reset" sysfs reset operation (0x20261) calls get_flash_version(), which reads hardware flash registers, without holding ha->optrom_mutex. The VPD update path serializes the same call under optrom_mutex, so this reset path can interleave its flash register accesses with a concurrent VPD or optrom flash operation and corrupt the reads. Hold ha->optrom_mutex across the get_flash_version() call to match the VPD update path.	N/A	More Details
CVE-2026-89853	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix FCE trace use-after-free during firmware dump qla2x00_free_fce_trace() freed and cleared ha->fce while holding only fce_mutex. The firmware-dump consumers qla27xx_fwdt_entry_t264() and qla25xx_copy_fce() read ha->fce (NULL check followed by a copy of the buffer) under hardware_lock and never take fce_mutex. A debugfs FCE disable could therefore free the DMA buffer between a dump's NULL check and its copy, resulting in a use-after-free. Unpublish ha->fce under hardware_lock, then release the lock and free the DMA buffer (dma_free_coherent() may sleep). A concurrent dump either completes its check and copy with the buffer still valid, or observes ha->fce == NULL and skips it.	N/A	More Details
CVE-2026-89871	In the Linux kernel, the following vulnerability has been resolved: media: video-i2c: fix kthread error pointer left in kthread_vid_cap on failure kthread_run() returns an ERR_PTR on failure, not NULL. When start_streaming() fails, data->kthread_vid_cap is left holding this error pointer instead of being cleared. This causes two subsequent bugs: 1. A future call to start_streaming() sees a non-NULL kthread_vid_cap and returns 0 (success) immediately, without actually starting the capture thread. 2. A call to stop_streaming() checks 'kthread_vid_cap == NULL' which is false for an error pointer, and proceeds to call kthread_stop() on the error pointer, leading to a kernel crash. Fix this by resetting kthread_vid_cap to NULL on failure before jumping to the error path.	N/A	More Details

CVE-2026-54594	OmniBlocks is a monorepo for the OmniBlocks project. Prior to the June 6, 2026 workflow remediation, .github/workflows/disc.yml runs for the issues opened event and the issues edited event and invokes the createDiscussion mutation whenever an issue is classified as off-topic, without recording that the issue was already converted or otherwise suppressing duplicate runs. A user who creates one off-topic issue and repeatedly edits its description before conversion completes can therefore cause multiple discussions to be created for the same issue, producing discussion spam and additional moderation work. This issue is fixed with commit 627e0f0a16a7d74b09128106b57dd7e85d2545df.	N/A	More Details
CVE-2026-89900	In the Linux kernel, the following vulnerability has been resolved: media: cec: core: Fix kmemleak due to missed rc_free_device() call The commit dccc0c3ddf8f ("media: rc: fix race between unregister and urb/irq callbacks") removed the implicit call to rc_free_device() from rc_unregister_device(). However, the commit missed to remove the NULL assignment of adap->rc that is now causing rc_free_device() to never be called on an allocated rc device. kmemleak reports following after e.g. dw-hdmi unbind: unreferenced object 0xffff00010ac10000 (size 4096): comm "kworker/u16:1", pid 39, jiffies 4294897739 hex dump (first 32 bytes): 20 23 4b 0a 01 00 ff ff 08 00 c1 0a 01 00 ff ff #K..... 08 00 c1 0a 01 00 ff ff 00 00 00 00 00 00 00 00 ..... backtrace (crc e11baccc): kmemleak_alloc+0x38/0x44 __kmalloc_cache_noprof+0x4a8/0x5e0 rc_allocate_device+0x48/0x2a0 cec_allocate_adapter+0x3ac/0x800 dw_hdmi_cec_probe+0x264/0x634 platform_probe+0xc0/0x188 really_probe+0x4a4/0x8e0 __driver_probe_device+0x2f8/0x440 driver_probe_device+0x60/0x160 __device_attach_driver+0x1a0/0x2a0 bus_for_each_drv+0x100/0x1a0 __device_attach+0x174/0x350 device_initial_probe+0x90/0xb0 bus_probe_device+0x4c/0x120 device_add+0xdec/0x116c platform_device_add+0x354/0x598 Remove the assignment of adap->rc to NULL to let cec_delete_adapter() free the allocated rc device after last user of the cec device exits to fix the kmemleak.	N/A	More Details
CVE-2026-89874	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-async: avoid deleting unlinked ASC entry on link error v4l2_async_match_notify() creates ancillary media links before adding asc->asc_subdev_entry to sd->asc_list. If ancillary link creation fails, the function jumps to err_call_unbind while asc_subdev_entry has not been linked yet. Async connections are zero-allocated, so the list entry still has NULL next and prev pointers on this path. Calling list_del() on it can therefore dereference NULL instead of returning the original link creation error. Do not delete asc_subdev_entry from err_call_unbind. There is no list insertion to undo on this path; the bound callback and sub-device registration are the operations that need to be rolled back.	N/A	More Details
CVE-2026-89896	In the Linux kernel, the following vulnerability has been resolved: media: cedrus: fix memory leak in cedrus_init_ctrls() In cedrus_init_ctrls(), the V4L2 control handler is initialized before allocating memory for ctx->ctrls. If this allocation fails, the function returns -ENOMEM without freeing the previously allocated handler resources, leading to a memory leak. Fix this by calling v4l2_ctrl_handler_free() on the ctx->ctrls allocation failure path. The bug was first flagged by an experimental analysis tool we are developing for kernel memory-management bugs while analyzing v6.13-rc1. The tool is still under development and is not yet publicly available. Manual inspection confirms that the bug is still present in v7.1.1. An x86_64 allyesconfig build showed no new warnings. As we do not have an Allwinner SoC or board with a Cedrus VPU available to test with, no runtime testing was able to be performed.	N/A	More Details
CVE-2026-89895	In the Linux kernel, the following vulnerability has been resolved: media: cobalt: Avoid freeing ALSA private data twice snd_cobalt_card_create() stores cobsc in sc->private_data and installs snd_cobalt_card_private_free() as sc->private_free. From that point, snd_card_free(sc) releases cobsc through the ALSA card cleanup path. If cobalt_alsa_init() fails after snd_cobalt_card_create(), the err_exit_free path calls snd_card_free(sc) and then kfree(cobsc). That second free releases the same object again. Remove the explicit kfree(cobsc) and leave ownership with the ALSA card. This issue was found by a static analysis checker and confirmed by manual source review.	N/A	More Details
CVE-2026-89892	In the Linux kernel, the following vulnerability has been resolved: media: em28xx: defer audio-only extension registration The audio-only path registers extensions while probing the primary device. For a dual-TS board, this happens before dev_next is created. The duplicate device inherits is_audio_only and is then independently inserted into em28xx_devlist. The list is intended to contain only primary devices: extension operations reach the secondary device through dev_next. The independently linked secondary can be freed during disconnect while its list node remains reachable, resulting in a use-after-free. Defer audio-only extension registration to the module-request work item. It runs only after probing has completed construction of the optional secondary device, so only the primary is registered and extension callbacks reach the secondary through dev_next.	N/A	More Details
CVE-2026-89891	In the Linux kernel, the following vulnerability has been resolved: media: em28xx: fix use-after-free of dev_next->devlist on disconnect When a device with has_dual_ts=1 is probed and the is_audio_only path is taken, both dev and dev->dev_next are added to the global em28xx_devlist via em28xx_init_extension(). However, during disconnect, em28xx_close_extension(dev) only calls list_del(&dev->devlist), leaving dev->dev_next->devlist still linked in the global list. When dev_next is subsequently freed via kref_put(), its devlist entry becomes a dangling pointer in em28xx_devlist. The next device probe that calls em28xx_init_extension() triggers a list corruption BUG when list_add_tail detects the freed node. This bug was exposed by commit a368ecde8a50 ("USB: core: Fix duplicate endpoint bug by clearing reserved bits in the descriptor") which clears reserved bits in bEndpointAddress during endpoint parsing. This causes fuzzed endpoint addresses like 0xf3 to be normalized to 0x83, which em28xx interprets as a vendor audio endpoint, enabling the is_audio_only + has_dual_ts code path that was previously unreachable with such descriptors. Fix this by removing dev->dev_next->devlist from the global list in em28xx_close_extension() before the device is freed.	N/A	More Details
CVE-2026-54451	Elixir protobuf is a pure Elixir implementation of Google Protobuf. From 0.8.0 until 0.16.1, services that decode attacker-controlled protobuf bytes with Protobuf.Decoder can be taken offline when the schema contains a self-referential or cyclic message type. In lib/protobuf/decoder.ex, Protobuf.Decoder.value_for_field/3 handles an embedded?: true field by recursively entering the decode / build_message / handle_value / value_for_field call chain without enforcing a nesting-depth limit. Deeply nested embedded fields retain non-tail recursive frames, allowing a comparatively small request to consume substantial CPU and memory, pin a BEAM scheduler, and exhaust the node. This issue is fixed in version 0.16.1.	N/A	More Details
CVE-2026-89889	In the Linux kernel, the following vulnerability has been resolved: media: i2c: imx415: Release runtime PM reference on VBLANK error The VBLANK path returned immediately when programming VMAX failed after pm_runtime_get_if_in_use() had taken a runtime PM reference. Break out of the switch instead so the common pm_runtime_put() path is used.	N/A	More Details
CVE-2026-54524	Frappe HR is an open-source human resources management solution (HRMS). Prior to 16.7.0, an authenticated user with the HR User role can inject SQL through filters in the Salary Payments Based on Payment Mode report. In hrms/payroll/report/salary_payments_based_on_payment_mode/salary_payments_based_on_payment_mode.py, get_conditions constructs filter clauses from user-controlled values and get_data incorporates those clauses into a string-formatted SQL query, allowing extraction of arbitrary database data. This issue is fixed in 16.7.0.	N/A	More Details

CVE-2026-54571	ESPAsyncWebServer is an asynchronous HTTP and WebSocket server library for ESP32, ESP8266, RP2040 and RP2350. Prior to 3.11.1, the multipart/form-data parser in src/WebRequest.cpp stores _boundaryPosition as an 8-bit value while _parseMultipartPostByte processes the boundary. A remote request containing an exactly 256-byte multipart boundary wraps _boundaryPosition from 255 to zero, prevents the boundary parsing loop from terminating, consumes excessive CPU, and triggers a FreeRTOS watchdog reset on affected ESP32 or ESP8266 devices. This issue is fixed in version 3.11.1.	N/A	More Details
CVE-2026-54649	punchin-email is a Cloudflare Email Worker that provides two-way role aliases while relaying mail to a private inbox. Prior to 1.5.0, handleInbound delivers inbound alias mail with message.forward(), which silently drops the added Reply-To header intended to route responses through the relay. When a correspondent sends mail to an alias and the operator replies, the mail client can send directly to the correspondent from the private FORWARD_TO inbox address, exposing that address. The disclosure is limited to the operator's own email address and does not expose third-party data or provide code execution or authentication bypass. This issue is fixed in version 1.5.0.	N/A	More Details
CVE-2026-55061	uniget is a universal installer and updater for (container) tools. Prior to 0.27.6, the hooks edit command in cmd/uniget/hooks.go parses UNIGET_EDITOR or EDITOR with strings.Split(editor, " ") and passes every space-delimited suffix as an argument to the selected editor executable. An attacker who can influence the editor environment and cause hook editing can supply unexpected editor arguments, potentially causing unintended actions with the privileges of the uniget process account. Go os/exec does not evaluate shell operators in these arguments, so the advisory's wrapper demonstration establishes argument delivery but does not establish shell command interpretation. This issue is fixed in version 0.27.6.	N/A	More Details
CVE-2026-55062	uniget is a universal installer and updater for (container) tools. Prior to 0.27.6, the hooks edit command in cmd/uniget/hooks.go concatenates an unvalidated hook filename with the selected hooks directory, allowing parent-directory components to escape that directory. The resulting path is passed to the configured editor, which can access or modify files outside the hooks directory with the privileges of the uniget process account. This issue is fixed in version 0.27.6.	N/A	More Details
CVE-2026-89886	In the Linux kernel, the following vulnerability has been resolved: media: intel/ipu6: fix async notifier cleanup leak on parse error isys_notifier_init() calls v4l2_async_nf_init() and then adds fwnode remote subdevs in a loop with v4l2_async_nf_add_fwnode_remote(). If an endpoint parse or add fails partway through the loop, it jumps to err_parse and returns without calling v4l2_async_nf_cleanup(), leaking every v4l2_async_connection already added to the notifier's waiting list. The register-failure path just below already cleans up correctly, and the caller only tears the notifier down (isys_notifier_cleanup()) once isys_notifier_init() has returned success. Clean up the notifier on the parse error path too.	N/A	More Details
CVE-2026-89884	In the Linux kernel, the following vulnerability has been resolved: media: platform: mtk-mdp3: fix NULL deref on failed SCP lookup Add the missing sanity check after looking up the SCP to avoid dereferencing a NULL-pointer in case its driver has not yet been bound.	N/A	More Details
CVE-2026-89881	In the Linux kernel, the following vulnerability has been resolved: media: rtl2832_sdr: use vb2_video_unregister_device() on remove to fix DMA leak rtl2832_sdr_remove() runs on USB disconnect and clears dev->udev to NULL before any pending streaming teardown has run. When user space later closes its file descriptor, vb2 calls rtl2832_sdr_stop_streaming() which in turn calls rtl2832_sdr_free_stream_bufs(). That helper releases each coherent buffer with: usb_free_coherent(dev->udev, dev->buf_size, dev->buf_list[dev->buf_num], dev->dma_addr[dev->buf_num]); usb_free_coherent() returns immediately when its dev argument is NULL, so every DMA stream buffer that was live at disconnect is silently leaked. The URBs allocated in rtl2832_sdr_alloc_urbs() outlive the device for the same reason. The rtl2832_sdr driver uses vb2_fop_release() in its file_operations, so replace video_unregister_device(&dev->vdev) with vb2_video_unregister_device(&dev->vdev) and move it before clearing dev->udev. vb2_video_unregister_device() releases the vb2 queue, which synchronously runs rtl2832_sdr_stop_streaming() if streaming is active, so URBs and coherent DMA stream buffers are freed while dev->udev is still valid. vb2_video_unregister_device() locks vdev->queue->lock (vb_queue_lock) internally, and stop_streaming() locks v4l2_lock, so the previous outer mutex_lock(&dev->vb_queue_lock) / mutex_lock(&dev->v4l2_lock) pair around the unregister sequence would self-deadlock and has been removed. A short v4l2_lock critical section around dev->udev = NULL remains so any ioctl path that still holds the file descriptor sees coherent state. Issue identified by automated review of the INV-003 series at https://sashiko.dev/	N/A	More Details
CVE-2026-89879	In the Linux kernel, the following vulnerability has been resolved: media: s2255: bound JPEG frame size before copying into the buffer s2255_fillbuff() memcpy()s vc->jpg_size bytes of a captured JPEG/MJPEG frame into the vb2 plane. vc->jpg_size is taken verbatim from the S2255_MARKER_FRAME header the device sends (pdword[4] in save_frame()) and, unlike the frame payload length just above it, is never bounded: payload = le32_to_cpu(pdword[3]); if (payload > vc->req_image_size) /* payload is checked ... */ return -EINVAL; vc->pkt_size = payload; vc->jpg_size = le32_to_cpu(pdword[4]); /* ... jpg_size is not */ A malicious or malfunctioning device can therefore report a jpg_size larger than the destination vb2 plane, and the memcpy() writes past it. jpg_size is a signed int, so a value with the top bit set also turns into a huge length. Reject a frame whose jpg_size is negative or exceeds the plane size before copying it.	N/A	More Details
CVE-2026-89878	In the Linux kernel, the following vulnerability has been resolved: media: s2255: check firmware size before reading trailing marker s2255_probe() reads a 4-byte marker and version from the last 8 bytes of the firmware blob (fw->data[fw_size - 8] and [fw_size - 4]). If the firmware file is shorter than 8 bytes, fw_size - 8 underflows and the access reads out of bounds. Validate the firmware size before indexing.	N/A	More Details
CVE-2026-89876	In the Linux kernel, the following vulnerability has been resolved: media: tda18250: fix possible integer overflow Integer overflow may occur, when variable exp equals to zero. Result of shift 1 << (exp - 1) may then leads to undefined behavior.	N/A	More Details
CVE-2026-90028	In the Linux kernel, the following vulnerability has been resolved: usb: typec: hd3ss3220: track VBUS enable state per consumer regulator_is_enabled() reports the aggregate regulator state, not whether this consumer holds an enable reference. If another consumer enables VBUS first, the driver can skip its own regulator_enable() call and later attempt to drop a reference it never acquired, triggering an unbalanced regulator disable warning. Track successful enable and disable calls locally. Keep the state unchanged when an operation fails so a later role or ID notification retries the operation while this consumer keeps balanced references.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: media: stm32: dcmi: fix some error handling bugs in probe() There are a few issues here: 1) After we assign: chan = dma_request_chan(&pdev->dev, "tx"); Then the error paths need to clean up before returning. The first error path does a direct return. 2) The error paths check "dcmi->mdma_chan" but that is		

2026-90405	not assigned until later so it results in memory leaks. Test "mdma_chan" instead. 3) The error handling calls dma_release_channel(dcmi->dma_chan) before "dcmi->dma_chan" has been assigned which leads to a NULL pointer dereference. Use the "chan" variable instead. I also moved the call to dma_release_channel() after the call to dma_release_channel() so it mirrors the allocation code better.	N/A	More Details
CVE-2026-90029	In the Linux kernel, the following vulnerability has been resolved: usb: storage: realtek_cr: fix use-after-free on disconnect realtek_cr_destructor() calls timer_delete() before the chip containing the timer is freed. The timer callback may still be running and can rearm itself, resulting in a use-after-free. Use timer_shutdown_sync() to wait for the callback and prevent further rearming. Do this unconditionally because ss_en may be changed after the timer is armed. Move timer_setup() into init_realtek_cr() so the timer is initialized before any failure path can invoke the destructor. Found by static analysis.	N/A	More Details
CVE-2026-90157	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject negative optlen in cgroup getsockopt hook A cgroup getsockopt BPF program can shrink ctx->optlen after the kernel getsockopt handler has run. The kernel-buffer variant, used by TCP_ZEROCOPY_RECEIVE, only rejects values larger than the original length. If BPF writes a negative optlen, that value is accepted and propagated back to the TCP getsockopt code. It can then be passed to copy_to_sockptr() as a size_t and trigger the hardened usercopy bytes > INT_MAX warning. Reject negative ctx.optlen in __cgroup_bpf_run_filter_getsockopt_kern(), matching the lower-bound validation already present in the sockptr-based getsockopt hook.	N/A	More Details
CVE-2026-89082	HP has identified potential security vulnerabilities in the HP Advance software that may enable elevation of privilege, remote code execution, or arbitrary file write under certain conditions, impacting the HP Advance server hosting the software.	N/A	More Details
CVE-2026-81870	OpenTelemetry-Go is the Go implementation of OpenTelemetry. From version 1.5.0 to 1.44.0, sdk/trace.NewTracerProvider emits a TracerProvider created internal Info-level diagnostic event whose MarshalLog implementations recursively include span processor, exporter, and client configuration. Applications that call otel.SetLogger to enable OpenTelemetry internal Info logging can therefore record OTLP gRPC and HTTP collector endpoints, the OTLP HTTP Insecure flag, and complete Zipkin collector URLs. A person or system with access to those logs can learn internal collector topology and can recover credentials or tokens embedded in Zipkin URL user information or query strings. The default OpenTelemetry logger does not emit the event, and this path does not log OTLP authentication headers, TLS key material, or span payloads. This issue is fixed in version 1.45.0.	N/A	More Details
CVE-2026-90144	In the Linux kernel, the following vulnerability has been resolved: dpll: fix NULL deref in dpll_device_ops() during teardown race When the last owner of a dpll device unregisters while a foreign driver still holds a pin on it via dpll_pin_on_pin_register(), the dpll object stays alive with an empty registration list. A pin notification queued before the unregister (e.g. ice reacting to zl3073x_i2c removal) then walks pin->dpll_refs into dpll_device_ops(), which trips the WARN_ON and dereferences the missing registration. dpll_lock cannot help because the notification work was queued before the unregistering driver took the lock. Treat the empty registration list as a legitimate transient state. Make dpll_priv() and dpll_device_ops() return NULL in that case and make every pin netlink path that resolves a device from a pin skip such dplls. dpll_cmd_pin_get_one() picks a ref with a live registration and returns -ENODEV when there is none, the pin dumpit skips such a pin instead of aborting the dump, dpll_msg_add_pin_dplls() and the frequency, esync, reference sync and phase adjust set paths skip dead refs, and dpll_pin_parent_device_set() validates the parent with dpll_device_get_by_id(). dpll_pin_register() is the last caller that dereferenced the device ops without a check, so move its frequency monitor validation under dpll_lock and tolerate a missing registration there as well. The empty registration list is equivalent to a cleared DPLL_REGISTERED mark, both transitions happen under dpll_lock in dpll_device_register() and dpll_device_unregister(). A pin notification for a pin whose dplls are all gone is now dropped with -ENODEV instead of crashing, all callers in the core ignore that return value. WARNING: drivers/dpll/dpll_core.c:1092 at dpll_device_ops+0x24/0x40, CPU#83: kworker/u576:3/23471 Modules linked in: ... ice ... zl3073x_i2c(-) ... zl3073x ... Workqueue: ice_dpll_wq ice_dpll_pin_notify_work [ice] RIP: 0010:dpll_device_ops+0x24/0x40 Call Trace: <TASK> dpll_cmd_pin_get_one+0x336/0x520 dpll_pin_event_send+0x82/0x140 dpll_pin_on_pin_unregister+0xbb/0x160 ice_dpll_pin_notify_work+0x1bc/0x1f0 [ice] process_one_work+0x19e/0x370 worker_thread+0x1a6/0x310 kthread+0xe4/0x120 ret_from_fork+0x1a1/0x270 ret_from_fork_asm+0x1a/0x30 </TASK> ---[end trace 0000000000000000]-- BUG: kernel NULL pointer dereference, address: 0000000000000010 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page	N/A	More Details
CVE-2026-90147	In the Linux kernel, the following vulnerability has been resolved: clk: devres: fix cleanup in devm_clk_get_optional_enabled_with_rate() devm_clk_get_optional_enabled_with_rate() registers its cleanup action before setting the clock rate. If setting the rate fails, it attempts to disable and unprepare a clock that was never enabled. This issue was spotted while reviewing "rust: clk: add devres-managed clks" [1]. Register the cleanup action only after successfully preparing and enabling the clock. [1]: https://lore.kernel.org/rust-for-linux/20260706-clk-type-state-v5-3-67c5f326a16c@collabora.com	N/A	More Details
CVE-2026-90148	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Fix incorrect argument passed to nfs4_delete_lease() in nfs4_add_lease() When nfs4_add_lease() races with a delegation return, it calls nfs4_delete_lease() to clean up. Previously, it passed priv, which can legitimately be NULL. Passing a NULL priv eventually leads to a NULL pointer dereference in generic_setlease().	N/A	More Details
CVE-2026-90150	In the Linux kernel, the following vulnerability has been resolved: pnfs/blocklayout: Fix device leaks on parse failure bl_parse_concat() and bl_parse_stripe() allocate a child device array and then parse each child in turn. If parsing a child fails, the failed child is not counted in nr_children and the parent may be left with a children array that bl_free_device() will not release when nr_children is zero. Release the failed child and the already parsed children before returning the error. Also make bl_free_device() release the child array whenever the children pointer is set, so that partially initialised concat or stripe devices are cleaned up correctly. bl_parse_scsi() can also fail after assigning d->bdev_file and dropping the file reference. Clear the pointer after fput() so that an outer cleanup path does not put it again.	N/A	More Details
CVE-2026-90152	In the Linux kernel, the following vulnerability has been resolved: smb/server: fix session leak in ksmbd_session_register() See the procedure below: smb2_sess_setup ksmbd_smb2_session_create __session_create atomic_set(&sess->refcnt, 2) hash_add(sessions_table, &sess->hlist, sess->id) ksmbd_session_register xa_store(&conn->sessions, sess->id, sess) // fail ksmbd_user_session_put atomic_dec(&sess->refcnt) // refcnt is 1, session is not freed Remove the session from sessions_table and drop its table reference if xa_store() fails.	N/A	More Details

CVE-2026-90154	In the Linux kernel, the following vulnerability has been resolved: ksmbd: scope session state changes to bound connections ksmbd_all_conn_set_status() treats every connection whose transient binding flag is set as belonging to the target SessionId. A logoff or session replacement can consequently move an unrelated connection to NEED_RECONNECT or NEED_SETUP. Pass the target session itself and select connections using either the connection-local session xarray or the session's permanent channel list. Use the same association test while waiting for requests to drain. Serialize session-wide status changes under request_lock and do not overwrite EXITING or RELEASING. Protect the shutdown transition with the same lock so a concurrent session update cannot revive a closing connection.	N/A	More Details
CVE-2026-90155	In the Linux kernel, the following vulnerability has been resolved: ksmbd: detach blocked lock requests before freeing A file_lock retained by ksmbd for byte-range lock bookkeeping can still be part of the VFS blocked-request graph. In particular, the VFS can chain a new waiter below an already blocked request through flc_blocked_requests. The ksmbd_file reference count does not cover that graph. Both __ksmbd_close_fd() and the cross-request unlock path free these retained file_lock objects directly. If a dependent waiter is still attached, locks_release_private() hits BUG_ON(!list_empty(&flc->flc_blocked_requests)). The same lifetime mismatch can leave a freed ksmbd_lock reachable through its request-local llist. Detach the file_lock from the blocked-request graph before freeing it in the close, cross-request unlock, and rollback paths. locks_delete_block() also wakes requests chained below the object. Remove llist when a completed lock is published so a globally visible ksmbd_lock no longer points into the submitting worker's stack.	N/A	More Details
CVE-2026-90156	In the Linux kernel, the following vulnerability has been resolved: ksmbd: safely discard unregistered deferred locks When vfs_lock_file() defers a lock, smb2_lock() puts its ksmbd_lock on rollback_list before allocating and registering the asynchronous work. If either operation fails, rollback assumes that smb_lock->conn is initialized and dereferences NULL. The deferred file_lock also remains linked into the VFS blocked-lock state while it is freed. Keep the lock off rollback_list until async setup succeeds. On setup failures, explicitly unblock and wake the deferred lock before freeing it and its ksmbd wrapper.	N/A	More Details
CVE-2026-90158	In the Linux kernel, the following vulnerability has been resolved: m68k: nfcon: Do not call console_is_registered() in nfcon_device() Since 7c2af0f634f1 ("tty: tty_io: use console_list_lock for list synchronization") show_cons_active() calls the .device() method under the console_list_lock, but console_is_registered() tries to acquire console_list_lock as well, causing a deadlock. It should not be necessary to check console_is_registered() here since the function should not be called in the fist place when the console is not registered.	N/A	More Details
CVE-2026-77360	oRPC is an tool that helps build APIs that are end-to-end type-safe and adhere to OpenAPI standards. Prior to 1.14.8, the @orpc/server CORS plugin in packages/server/src/plugins/cors.ts copies a client's incoming Vary request header into the response instead of controlling Vary as a response-only header and using Origin for request-origin variation. In deployments behind a shared cache, CDN, or reverse proxy that keys responses using Vary, a client can inject arbitrary variation values, pollute cache keys, and cause inconsistent CORS enforcement for other clients. Default non-cached configurations have no established direct confidentiality, integrity, or availability impact. This issue is fixed in version 1.14.8.	N/A	More Details
CVE-2026-90159	In the Linux kernel, the following vulnerability has been resolved: bpf: Disallow bpf_{g,s}etsockopt() in cgroup UNIX getname hooks _bpf_setsockopt() and _bpf_getsockopt() call sock_owned_by_me() for full sockets, so these helpers expect the socket lock to be held. BPF_CGROUP_UNIX_GETPEERNAME and BPF_CGROUP_UNIX_GETSOCKNAME run BPF programs without acquiring the socket lock. A program attached to either hook can therefore trigger the sock_owned_by_me() warning by calling bpf_setsockopt() or bpf_getsockopt(). Disallow bpf_setsockopt() and bpf_getsockopt() for CGROUP_UNIX_GETPEERNAME and CGROUP_UNIX_GETSOCKNAME.	N/A	More Details
CVE-2026-90160	In the Linux kernel, the following vulnerability has been resolved: lwt_bpf: Restore reserved headroom after xmit program ip_finish_output2() expands an skb to LL_RESERVED_SPACE(dev) before LWT xmit. An LWT_XMIT BPF program can then modify the skb head and still return BPF_OK, so bpf_xmit() rechecks the remaining headroom before the skb continues to neighbour output. That recheck uses dst->dev->hard_header_len. This is not enough for the neighbour cached-header path: neigh_hh_output() copies the cached hardware header using the aligned hh_cache size, HH_DATA_MOD for short headers or HH_DATA_ALIGN(hh_len) otherwise. On Ethernet, hard_header_len is 14 but the cached copy needs 16 bytes. If an LWT_XMIT BPF program calls bpf_skb_change_head(skb, 1, 0), the skb can still have 15 bytes of headroom after the program. The existing check accepts that, after which neigh_hh_output() hits its headroom warning and drops the skb. Use LL_RESERVED_SPACE(dst->dev) in the post-BPF headroom check to match the reservation made before LWT xmit.	N/A	More Details
CVE-2026-90163	In the Linux kernel, the following vulnerability has been resolved: smb/server: call ksmbd_proc_cleanup() on module init failure When a later initializer fails, the unwind chain releases resources created after procfs and then jumps directly to class_unregister(). Returning an error from module_init() leaves the proc tree and its per-CPU counters allocated.	N/A	More Details
CVE-2026-90164	In the Linux kernel, the following vulnerability has been resolved: smb/server: abort initialization when proc setup fails ksmbd_server_init() calls ksmbd_proc_init() before creating the remaining proc entries and server subsystems. ksmbd_proc_init() tears down partial state on a procfs or percpu_counter allocation failure, but returns void, so ksmbd_server_init() continues as if the counters were usable. Once userspace starts the server, server_ctrl_handle_init() calls ksmbd_proc_reset(), which reaches percpu_counter_set() with a NULL per-CPU counters pointer on SMP systems. The later ksmbd_proc_create() calls also receive a NULL parent and may create entries in the /proc root; ksmbd_proc_cleanup() cannot remove those entries because ksmbd_proc_fs is NULL.	N/A	More Details
CVE-2026-90165	In the Linux kernel, the following vulnerability has been resolved: smb/server: fix invalid pointer dereference in ksmbd_stop_durable_scavenger() See the procedure below: ksmbd_launch_ksmbd_durable_scavenger durable_scavenger_running = true server_conf.dh_task = kthread_run() // fail, dh_task is an ERR_PTR() server_ctrl_handle_reset ksmbd_stop_durable_scavenger kthread_stop(server_conf.dh_task) // invalid pointer	N/A	More Details
CVE-2026-90166	In the Linux kernel, the following vulnerability has been resolved: smb/server: fix null-ptr-deref in ksmbd_ipc_tree_connect_request() See the procedure below: ksmbd_tree_conn_connect ksmbd_share_config_get share->name = kstrdup() // fail if (!test_share_config_flag(share, KSMBD_SHARE_FLAG_PIPE)) // false // do not check `share->name` ksmbd_ipc_tree_connect_request strlen(share->name) // null-ptr-deref	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: ksmbd: serialize oplock close with pending break ownership close may abort an in-flight oplock break while another breaker already holds an opinfo reference. Releasing pending_break wakes that waiter, but without serializing the close transition with bit acquisition it can become a new break owner through the test_and_set_bit() fast path. It can then overwrite OPLOCK_CLOSING with OPLOCK_ACK_WAIT and continue a break for a dying opinfo. Make OPLOCK_CLOSING terminal once the opinfo is removed from the inode list. Serialize that transition, pending_break	N/A	More Details

<table>
<tr><td>90167</td><td>acquisition, and OPLOCK_ACK_WAIT setup with an opinfo state lock. A breaker which loses the race releases its ownership and returns -ENOENT. Explicitly wake pending_break waiters during close so they can observe the terminal state. Also prevent ACK and timeout paths from replacing OPLOCK_CLOSING with OPLOCK_STATE_NONE.</td><td></td><td></td></tr>
<tr><td>CVE-2026-90168</td><td>Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.</td><td>N/A</td><td>More Details</td></tr>
<tr><td>CVE-2026-90169</td><td>In the Linux kernel, the following vulnerability has been resolved: ksmbd: free preauth sessions on connection teardown SMB3.1.1 multichannel binding preserves the preauthentication hash in a preauth_session between the NTLM negotiate and authenticate requests. The binding NTLM negotiate allocates this object and returns STATUS_MORE_PROCESSING_REQUIRED. If the client disconnects before it sends the authenticate request, neither the authenticate nor error cleanup paths free the object. Release any remaining preauthentication sessions when tearing down the connection. Initialize the list when allocating the connection so that this cleanup is safe regardless of the negotiated dialect.</td><td>N/A</td><td>More Details</td></tr>
<tr><td>CVE-2026-90170</td><td>In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate ipc response length before dereferencing its fields ipc_validate_msg() computes the expected message size by reading length fields out of the response buffer supplied by the userspace ksmbd daemon (payload_sz, session_key_len, ngroups, ...). Those fields are read before the buffer is verified to be large enough to contain the struct they belong to, so a short response makes the read land past the end of the allocation. handle_response() sizes entry->response purely from the netlink attribute length (nla_len()) and only guards the leading handle read, so the daemon can install a response as small as the kmalloc-8 object seen below. When ipc_msg_send_request() then calls ipc_validate_msg() for a KSMBD_EVENT_RPC_REQUEST, the cast to struct ksmbd_rpc_command reads resp->payload_sz at offset 8 of an 8-byte allocation: [3697.841381] == [3697.844099] BUG: KASAN: slab-out-of-bounds in ipc_msg_send_request+0x763/0x800 [3697.846604] Read of size 4 at addr ffff888105f95910 by task kworker/4:3/20682 [3697.849061] [3697.849801] CPU: 4 UID: 0 PID: 20682 Comm: kworker/4:3 Not tainted 7.2.0-rc3-next-20260717-virtme #117 PREEMPT(lazy) [3697.850077] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 [3697.850303] Workqueue: ksmbd-io handle_ksmbd_work [3697.850592] Call Trace: [3697.850794] <TASK> [3697.850952] __dump_stack+0x21/0x60 [3697.851239] dump_stack_lvl+0xc2/0x100 [3697.851528] print_address_description+0x77/0x200 [3697.851816] ? ipc_msg_send_request+0x763/0x800 [3697.852024] print_report+0x58/0x70 [3697.852316] kasan_report+0x117/0x150 [3697.852585] ? down_write+0x146/0x1f0 [3697.852809] ? ipc_msg_send_request+0x763/0x800 [3697.853082] ipc_msg_send_request+0x763/0x800 [3697.853385] ? __pfx_ipc_msg_send_request+0x10/0x10 [3697.853604] ? kasan_unpoison+0x48/0x70 [3697.853936] ? __pfx___up_read+0x10/0x10 [3697.854221] ksmbd_rpc_ioctl+0x380/0x520 [3697.854542] ? __pfx_ksmbd_rpc_ioctl+0x10/0x10 [3697.854757] ? kasan_unpoison+0x48/0x70 [3697.854962] ? copy_from_kernel_nofault+0x32c/0x4e0 [3697.855166] ? kasan_unpoison+0x48/0x70 [3697.855416] fsctl_pipe_transceive+0x139/0x7a0 [3697.855705] ? __pfx_copy_from_kernel_nofault+0x10/0x10 [3697.855937] ? __pfx_fsctl_pipe_transceive+0x10/0x10 [3697.856388] ? __sanitizer_cov_trace_switch+0x7b/0x140 [3697.856620] smb2_ioctl+0x1141/0x3420 [3697.856994] ? __pfx_smb2_ioctl+0x10/0x10 [3697.857182] ? get_smb2_cmd_val+0xe3/0x1c0 [3697.857655] handle_ksmbd_work+0x9ad/0x15e0 [3697.858034] ? __pfx_handle_ksmbd_work+0x10/0x10 [3697.858251] ? lock_release+0xf7/0x360 [3697.858466] ? process_scheduled_works+0x954/0x1600 [3697.858698] ? process_scheduled_works+0x954/0x1600 [3697.858905] process_scheduled_works+0xc22/0x1600 [3697.859368] ? __pfx_process_scheduled_works+0x10/0x10 [3697.859637] ? __pfx_assign_work+0x10/0x10 [3697.859896] ? lock_is_held_type+0x7b/0x110 [3697.860146] worker_thread+0x975/0xee0 [3697.860524] ? __pfx_do_raw_spin_lock+0x10/0x10 [3697.860830] ? __kthread_parkme+0x21e/0x260 [3697.861105] kthread+0x3a6/0x490 [3697.861423] ? __pfx_worker_thread+0x10/0x10 [3697.861643] ? __pfx_kthread+0x10/0x10 [3697.861878] ret_from_fork+0x55a/0xa20 [3697.862194] ? __pfx_ret_from_fork+0x10/0x10 [3697.862480] ? __pfx_kthread+0x10/0x10 [3697.862714] ret_from_fork_asm+0x1a/0x30 [3697.862965] </TASK> [3697.863039] [3697.938882] Allocated by task 20761: [3697.940257] kasan_save_track+0x3e/0x80 [3697.941782] __kasan_kmalloc+0x72/0x90 [3697.943228] __kvmalloc_node_noprof+0x3e9/0x6a0 [3697.944948] handle_generic_event+0x59b/0x750 [3697.946592] genl_family_rcv_msg_doit+0x3d6/0x560 [3697.946977] genl_rcv_msg+0x67c/0x900 [3697.947224] netlink_rcv_skb+0x286/0x580 [3697.947488] genl_rcv+0x2d/0x80 [3 ---truncated---</td><td>N/A</td><td>More Details</td></tr>
<tr><td>CVE-2026-90140</td><td>In the Linux kernel, the following vulnerability has been resolved: cuse: wait for pending RCU callbacks on module exit Since commit 053fc4f755ad ("fuse: fix UAF in rcu pathwalks"), fuse_conn_put() frees the fuse_conn through call_rcu() rather than synchronously. For cuse, fc->release is cuse_fc_release(), which lives in the cuse module. If the module is removed before the RCU grace period ends, the callback jumps into freed module memory: userspace / module unload | RCU softirq ~~~~~~ close(/dev/cuse) | cuse_channel_release() | fuse_dev_release() | fuse_conn_put(fch->conn) | call_rcu(delayed_release) ------+---> callback queued | rmmod cuse | cuse_exit() | cuse_channel_destroy() | ... | return | | <module text freed> | | rcu_do_batch() | delayed_release() | fc->release() | -> cuse_fc_release() | ^^^ freed text! The freed module text is unmapped by vfree(), so the jump into the stale callback triggers a page-fault Oops. If the virtual address is subsequently reused, the callback could execute unrelated code (undefined behaviour). Fix this by calling rcu_barrier() in cuse_exit() so that any pending fuse_conn release callback completes before the module is removed.</td><td>N/A</td><td>More Details</td></tr>
<tr><td>CVE-2026-90139</td><td>In the Linux kernel, the following vulnerability has been resolved: fuse: check for NULL root inode in fuse_fill_super_submount fuse_iget() can return NULL when its inode allocation fails, but fuse_fill_super_submount() passed the result straight to get_fuse_inode() and decremented fi->nlookup without checking it: root = fuse_iget(sb, parent_fi->nodeid, ...); fi = get_fuse_inode(root); fi->nlookup--; Inside fuse_iget() the inode allocation can fail and return NULL. The submount root takes the iget5_locked() path, whose alloc_inode() can fail under memory pressure (the auto-submount branch can fail the same way in new_inode() or fuse_alloc_submount_lookup()): inode = iget5_locked(sb, nodeid, fuse_inode_eq, fuse_inode_set, &nodeid); if (!inode) return NULL; A NULL root makes get_fuse_inode() a container_of() on NULL and the nlookup decrement a write to a bogus address, oopsing the mount. With CONFIG_KASAN the following null pointer dereference is reported when the root inode allocation of an auto-submount fails (e.g. under memory pressure): == BUG: KASAN: null-ptr-deref in fuse_get_tree_submount+0x656/0x8b0 Read of size 8 at addr 00000000000002b0 by task ls/942 CPU: 0 PID: 942 Comm: ls Tainted: G W 6.6 #15 Call Trace: <TASK> fuse_get_tree_submount+0x656/0x8b0 vfs_get_tree+0x48/0x140 fc_mount+0x13/0x50 fuse_dentry_automount+0x7a/0xb0 __traverse_mounts+0xca/0x330 step_into+0x339/0xac0 path_lookupat+0xc5/0x2f0 filename_lookup+0x163/0x2a0 vfs_statx+0xd5/0x200 do_statx+0x83/0xd0</td><td>N/A</td><td>More Details</td></tr>
</table>

	__x64_sys_statx+0xa0/0xc0 do_syscall_64+0x37/0x90 entry_SYSCALL_64_after_hwframe+0x78/0xe2 </TASK> == Return -ENOMEM instead; the caller tears down the partially built superblock on error, matching the other error returns in this function.		
CVE-2026-90138	In the Linux kernel, the following vulnerability has been resolved: vsock: don't check the listener's sk_err in vsock_accept() Syzbot reported an issue which can be reproduced with these steps: r0 = socket(AF_VSOCK, SOCK_STREAM, 0) bind(r0, {VMADDR_CID_ANY, PORT}) connect(r0, {VMADDR_CID_LOCAL, PORT}) -> -1, EPROTO (self-connect) listen(r0, backlog) -> 0 r1 = socket(AF_VSOCK, SOCK_STREAM, 0) connect(r1, {VMADDR_CID_LOCAL, PORT}) -> 0 accept(r0) -> -1, EPROTO (stale sk_err) Basically, it creates a socket (r0) and triggers a self-connect after binding it. This self-connect fails with EPROTO because it loops back to r0 while the socket is still in the TCP_SYN_SENT state, causing it to be incorrectly dispatched to the connecting-client path. The unexpected packet type encountered there sets sk_err to EPROTO. After that, it invokes a listen() call on the same socket. This listen() call succeeds because the kernel's listening path never inspects or clears sk_err. Then, a new socket (r1) is created as a normal client and connects to r0. However, vsock_accept() rejects this incoming connection because the listener's sk_err still holds the EPROTO error from the earlier failed self-connect. This rejection causes the child socket created for r1's connection to never be freed on virtio or hyperv transports; only the VMCI transport implements pending_work to revisit and clean up a rejected socket. For a non-blocking connect(), vsock_connect() may return -EINPROGRESS immediately, and vsock_connect_timeout() can later set sk->sk_err asynchronously. Since no vsock transport ever sets sk_err on a socket while it is in TCP_LISTEN state, checking it in vsock_accept() serves no purpose and only carries forward errors left behind by earlier, unrelated connection attempts on the same socket. Remove the checks so accept() no longer rejects valid incoming connections because of a stale error, which also avoids the resource leak described above.	N/A	More Details
CVE-2026-89083	HP has identified potential security vulnerabilities in the HP Advance software that may enable elevation of privilege, remote code execution, or arbitrary file write under certain conditions, impacting the HP Advance server hosting the software.	N/A	More Details
CVE-2026-90113	In the Linux kernel, the following vulnerability has been resolved: netdevsim: update queue NAPI association on queue reset In netdevsim, receive queues (struct nsim_rq) embed their own struct napi_struct. When queue reset is performed (e.g. via queue_reset debugfs), nsim_queue_start() swaps in a newly allocated struct nsim_rq, and nsim_queue_mem_free() later deletes and frees the old one. However, nsim_queue_start() failed to update the queue-to-NAPI mapping via netif_queue_set_napi(). As a result, dev->_rx[idx].napi continued to point to the old NAPI struct. After the old queue was freed, a subsequent queue dump via Netlink (NETDEV_CMD_QUEUE_GET) triggered a KASAN slab-use-after-free read in nla_put_napi_id() when accessing rxq->napi->napi_id. Fix this by calling netif_queue_set_napi() in nsim_queue_start() to associate the new NAPI with the RX queue, and clear the association with netif_queue_set_napi(..., NULL) in nsim_del_napi() during teardown.	N/A	More Details
CVE-2026-90114	In the Linux kernel, the following vulnerability has been resolved: net: bridge: Reject descending VLAN tunnel ranges A pair of descending VLAN and tunnel IDs can pass the tunnel range span check. The VLAN subtraction produces a negative int, which is converted to unsigned when compared with the u32 tunnel ID subtraction. It can therefore equal the wrapped tunnel ID delta. The range loop then performs no iterations. Since the batched notification handling added a post-loop error check, this leaves err uninitialized and makes the request's return value unpredictable. Reject descending VLAN ranges before comparing the spans. Valid ascending and single-entry ranges remain unchanged, while malformed descending ranges consistently return -EINVAL. This issue was found by a static analysis checker and confirmed by manual source review.	N/A	More Details
CVE-2026-90115	In the Linux kernel, the following vulnerability has been resolved: xsk: fix NULL pointer dereference in __xsk_rcv() In the __xsk_rcv() multi-buffer path, xsk_buff_alloc() is called in a loop without checking its return value. xsk_buff_can_alloc() only counts fill queue entries without validating their addresses, so it can succeed while xsk_buff_alloc() rejects all remaining entries and returns NULL. Oops: general protection fault, probably for non-canonical address 0xdfffffc0000000000 KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] RIP: 0010:__xsk_rcv+0x426/0xc20 (net/xdp/xsk.c:350) Call Trace: xsk_generic_rcv+0x26d/0x5f0 xdp_do_generic_redirect+0x3c5/0xcf0 do_xdp_generic+0x92f/0xe70 __netif_receive_skb_core.constprop.0+0xf7e/0x2b30 Fix this with a two-stage transaction. First allocate and stage all buffers required for the packet, recycling all staged buffers with xsk_buff_free() if any allocation fails. Only after this stage succeeds, copy the data, reserve the RX descriptors, and release the buffers in an error-free loop.	N/A	More Details
CVE-2026-90116	In the Linux kernel, the following vulnerability has been resolved: ALSA: mtpav: shut down output timer before card teardown snd_mtpav_output_timer() rearms chip->timer while holding chip->spinlock and accesses the card-private mtpav state. snd_mtpav_free() currently takes the same lock and calls timer_delete() when the timer is active. This only removes a pending timer; it does not wait for a callback that is already running and does not prevent the callback from rearming the timer. A callback running on another CPU can therefore continue after snd_mtpav_free() releases the lock and access the card-private state while the card is being torn down. It can also rearm the timer after timer_delete() has returned. Call timer_shutdown_sync() without holding chip->spinlock. This waits for any running callback to finish and prevents further rearming before the card-private mtpav state is released.	N/A	More Details
CVE-2026-90117	In the Linux kernel, the following vulnerability has been resolved: ntfs: validate usa_ofs before preserving the update sequence number When ntfs_mft_record_alloc() reuses a free mft record it reads the old update sequence number straight from the on-disk record: usn = *(__le16 *)((u8 *)m + le16_to_cpu(m->usa_ofs)); Here m points into the raw $MFT page-cache folio, which still holds unvalidated, MST-protected bytes: the folio is read by a plain iomap_read_folio() and neither post_read_mst_fixup() nor ntfs_mft_record_check() has run on it (both work on private copies). m->usa_ofs is therefore an untrusted u16, and a corrupted record can put it past the end of the record so the two-byte read lands outside the folio. Reading such a record while creating a file gives, under KASAN: BUG: KASAN: use-after-free in ntfs_mft_record_alloc+... Read of size 2 at addr ... ntfs_mft_record_alloc -> __ntfs_create -> ntfs_create -> path_openat Only preserve the old update sequence number when usa_ofs is even and in range, mirroring the check ntfs_mft_record_check() already applies; otherwise leave usn zero, which the existing restore below skips.	N/A	More Details
CVE-2026-90119	In the Linux kernel, the following vulnerability has been resolved: ALSA: ice1712: Fix the card leak at probe error with the auto-cleanup snd_ice1712_probe() performs multiple initialization steps after snd_card_new(), but directly returns on failures from later steps without releasing the ALSA card, causing resource leaks when probing fails. Use snd_devm_card_new() together with scope-based cleanup via __free(snd_card_unref), and clear the card pointer after successful registration to keep it alive.	N/A	More Details
CVE-	A server-side request forgery (SSRF) vulnerability exists in the UnifiedLogin service of Altium Enterprise Server. An unauthenticated network attacker can cause the server to issue outbound HTTP requests to a destination of the attacker's choosing, including internal services that are reachable only from the server itself. One such internal service exposes server		More

2026-92808	configuration and credential material without authentication, relying only on the request originating locally. Because the forged requests originate from the server process, that check is satisfied. An unauthenticated attacker can therefore retrieve stored credentials and use them to obtain an administrative session, resulting in full compromise of the server and all of its services. Altium 365 cloud deployments are not affected, as the affected endpoint is disabled in cloud mode.	N/A	Details
CVE-2026-90121	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v5: Clear per-CPU IRS data on teardown IRS affinity setup publishes an IRS pointer and IAFFID state in the per-CPU data before the remaining IRS initialization can fail. The error path then frees the IRS data without clearing that published state, leaving CPUs associated with freed memory. On initialization failure and normal IRS teardown, clear the per-CPU IRS association by removing the stale pointer to irs_data. Also invalidate the per-CPU IAFFID state for any CPUs that were tied to the IRS before it was freed.	N/A	More Details
CVE-2026-90122	In the Linux kernel, the following vulnerability has been resolved: clk: visconti: Make sure clk_init_data is fully initialized The clk_init_data structure contains several mutually-exclusive members for different methods to specify the possible parents of a clock, prompting drivers to initialize only the members they need. However, not initializing all members may cause subtle issues, which are only exposed when CONFIG_INIT_STACK_ALL_PATTERN or CONFIG_INIT_STACK_NONE is enabled. visconti_clk_register_gate() fills in init.parent_data, and assumes that init.parent_names is NULL. However, the latter in uninitialized, and thus may cause a crash. Make sure all members are fully initialized, to fix such bugs, and to avoid future breakage when converting drivers to a different method for specifying the parents.	N/A	More Details
CVE-2026-90123	In the Linux kernel, the following vulnerability has been resolved: irqchip/ast2700-intc: Avoid allocating in the irq_domain activate() callback The interrupt core calls the irq_domain_activate() callback from __setup_irq() with desc->lock held and interrupts disabled. Both aspeed_intc1_irq_domain_activate() and aspeed_intc0_resolve_route() test a compatible string with fwnode_device_is_compatible(). fwnode_device_is_compatible() invokes fwnode_property_match_string(), which allocates with GFP_KERNEL. That's obviously not possible with interrupts disabled and a raw spinlock held. Both call sites are only ever handed OF nodes, so use of_device_is_compatible() instead: it walks the property in place and does not allocate.	N/A	More Details
CVE-2026-90124	In the Linux kernel, the following vulnerability has been resolved: irqchip/renesas-rzg2l: Fix loss of interrupt rzg2l_clear_irq_int() and rzg2l_clear_tint_int() perform a read-modify-write on the ISCR/TSCR status registers to clear the bit for the interrupt just handled. Since these registers are write-0-to-clear per bit, this is racy: If another interrupt's status bit gets set between the read and the write, that bit is written back as 0 by the software-constructed value, clearing an interrupt that hasn't been serviced yet and losing it. This can be reproduced by triggering multiple interrupts at once, e.g.: gpioset -c gpiochip0 355=0 353=0 328=0 352=0 Fix this by writing back only the bit being cleared, with all other bits set to 1, instead of read-modify-writing the whole register. Since 1-bits are left unchanged by hardware, concurrently-set status bits for other interrupts are preserved.	N/A	More Details
CVE-2026-90125	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix request buffer leak in smb2_new_read_req() smb2_new_read_req() allocates the request buffer with smb2_plain_req_init() but only publishes it to the caller with *buf = req at the very end of the function. Two error returns sit in between: rc = smb2_plain_req_init(SMB2_READ, io_parms->tcon, server, (void **) &req, total_len); if (rc) return rc; if (server == NULL) return -ECONNABORTED; [...] rdata->mr = smbd_register_mr(server->smbd_conn, &rdata->subreq.io_iter, true, need_invalidate); if (!rdata->mr) return -EAGAIN; On either of them the buffer is neither released nor handed back, so it is leaked. The caller cannot clean up after it: smb2_async_readv() does 'goto out' on a non-zero return, which skips the cifs_small_buf_release(buf) at async_readv_out, and buf has not been assigned at that point in any case. The write path has never had this problem. smb2_async_writev() registers the memory region inline and jumps to its release label instead of returning: wdata->mr = smbd_register_mr(...); if (!wdata->mr) { rc = -EAGAIN; goto async_writev_out; } Commit b7972092199f ("cifs: smbd: Retry on memory registration failure") changed both sides from -ENOBUFS to -EAGAIN in a single patch, which puts the two shapes next to each other. Only the -EAGAIN return is reachable in practice, because smb2_plain_req_init() calls smb2_reconnect() first and that already fails with -EIO when server is NULL, before anything is allocated. Both returns are given the same treatment here rather than leaving one of them correct only by accident. Because -EAGAIN is a replayable error, the failure also reaches the retry block at the end of smb2_async_readv(), which marks the subrequest NETFS_SREQ_NEED_RETRY, so a failing registration can be retried rather than ending the I/O, and every attempt that reaches it leaks another buffer. smb2_should_replay() short-circuits on tcon->retry, so on a hard mount the attempt count is not bounded by the retrans setting. Only the asynchronous read path is affected. The synchronous SMB2_read() caller passes rdata == NULL and the memory registration block is guarded on rdata. The memory registration failure path was pointed out by the Sashiko AI reviewer while it was reviewing an unrelated patch to smb2_async_readv().	N/A	More Details
CVE-2026-90126	In the Linux kernel, the following vulnerability has been resolved: rtc: pcf8563: fix clock provider leak on unbind pcf8563_clkout_register_clk() registers the CLKOUT clock provider with of_clk_add_provider(), but nothing ever unwinds it: there is no of_clk_del_provider() call and the driver has no remove callback. Each of_clk_add_provider() allocates a struct of_clk_provider, takes a reference on the OF node and adds an entry to the global of_clk_providers list, none of which is released when the device is unbound. Every bind/unbind (or module reload) therefore leaks a provider structure and an of_node reference. The clock itself is already device-managed (devm_clk_register()); only the provider registration was not. Use devm_of_clk_add_hw_provider() so the provider is removed automatically on unbind. Tie it to the parent i2c device, whose OF node carries the #clock-cells and clock-output-names properties (the RTC class device has no OF node of its own).	N/A	More Details
CVE-2026-90127	In the Linux kernel, the following vulnerability has been resolved: virtio: rtc: time out alarm requests RTC class operations run with rtc_device.ops_lock held. The virtio RTC alarm requests currently wait without a timeout for the device to return their requestq buffers. On surprise removal, virtio-pci marks the virtqueues broken before unregistering the virtio device. If an alarm request is waiting when the device stops responding, viortc_remove() blocks in viortc_class_stop() while trying to acquire ops_lock. The request cannot complete and device removal hangs until the waiting task is signalled. Use the same 60-second timeout as clock read requests for alarm reads, alarm programming, and alarm interrupt enable requests. The existing message reference counting keeps a timed-out request alive until a late response or device teardown.	N/A	More Details
CVE-2026-90128	In the Linux kernel, the following vulnerability has been resolved: vdpa/mlx5: fix wrong list iterated in add_direct_chain error path In add_direct_chain(), newly allocated direct MR entries are added to the local list 'tmp', which is spliced into mr->head only on success. On the error path, the cleanup loop was incorrectly iterating over mr->head instead of tmp. Fix by iterating over 'tmp' in the err_alloc cleanup path.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: virtio_balloon: quiesce balloon work before device shutdown Commit 8bd2fa086a04 ("virtio: break and reset virtio devices on device_shutdown()") added a generic virtio bus .shutdown handler that breaks and resets every virtio device during device_shutdown(), i.e. on reboot and kexec. virtio_balloon provides		

CVE-2026-90129	no .shutdown of its own, so that generic path runs while the balloon's asynchronous work is still armed. Once the device has been broken, virtqueue_add_inbuf() in virtballoon_free_page_report() returns -EIO and trips its WARN_ON_ONCE(). On a kernel booted with panic_on_warn that turns an ordinary reboot, for example a kexec based upgrade, into a fatal panic in the middle of device_shutdown(), so the machine never reaches the new kernel. Relaxing that single WARN_ON_ONCE() would only hide the symptom: the inflate/deflate and OOM paths do not warn, they call wait_event(vb->acked, ...) and would instead block forever on a broken queue that can no longer complete. The device has to be quiesced, not just kept quiet. Add a .shutdown handler that quiesces the balloon via the shared virtballoon_quiesce() helper while the device is still alive, and only then breaks and resets it via virtio_device_shutdown(). Unlike virtballoon_remove() the balloon workqueue is not destroyed, as shutdown does not free the device and cancel_work_sync() together with stop_update already prevent any further work from being queued.	N/A	More Details
CVE-2026-90130	In the Linux kernel, the following vulnerability has been resolved: vdpa_sim: fix cleanup after worker creation failure vdpasim_create() leaves vdpasim->worker as an ERR_PTR when kthread_run_worker() fails. The error path then drops the device reference, which releases the partially initialized simulator. vdpasim_free() unconditionally passes the worker pointer to kthread_destroy_worker(), so the ERR_PTR is dereferenced and can trigger a general protection fault. Store the worker error, clear the pointer, and only clean up the worker when it was successfully initialized. Also make the release path tolerate partially initialized objects by guarding virtqueue and IOTLB cleanup, since the same release path can be reached from other initialization failures. I found this bug myself, though the patch was written with AI assistance.	N/A	More Details
CVE-2026-89084	HP has identified potential security vulnerabilities in the HP Advance software that may enable elevation of privilege, remote code execution, or arbitrary file write under certain conditions, impacting the HP Advance server hosting the software.	N/A	More Details
CVE-2026-90134	In the Linux kernel, the following vulnerability has been resolved: ntfs: fix kmap_local_page() usage in compress Several compressed I/O paths discard the address returned by kmap_local_page() and later access or unmap the page using page_address(). This is invalid for highmem pages, and local mappings must also be unmapped using the address returned by kmap_local_page(). Map each destination page in ntfs_decompress() only while producing the current sub-block. Use memcpy_from_page(), memcpy_to_page(), and memzero_page() for the other page accesses. Remove unnecessary local mappings from ntfs_write_cb(), where pages are accessed through the vmap() mapping.	N/A	More Details
CVE-2026-90135	In the Linux kernel, the following vulnerability has been resolved: net: add missing ref_tracker_dir_exit() to alloc_netdev_mqs() sashiko is reporting that trying to read /sys/kernel/debug/ref_tracker/* causes use-afer-free crash when either alloc_percpu() or dev_addr_init() in alloc_netdev_mqs() failed, for commit 4d92b95ff2f9 ("net: add net device refcount tracker infrastructure") added ref_tracker_dir_exit() to only free_netdev() path.	N/A	More Details
CVE-2026-90136	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd/hsmp: Reject negative power cap writes in hwmon hsmp_hwmon_write() takes the user-supplied hwmon value as a signed long and assigns "val / MICROWATT_PER_MILLIWATT" to msg.args[0], which is a __u32. MICROWATT_PER_MILLIWATT is an unsigned long, so a negative write to power1_cap (e.g. "echo -1 > power1_cap") is first converted to a huge unsigned value by the division and then stored into the u32 argument. As a result a nonsensical, multi-gigawatt socket power limit is sent to the SMU via HSMP_SET_SOCKET_POWER_LIMIT instead of the write being rejected. Reject negative values with -EINVAL before the conversion. Tested with HSMP enabled: CAP=$(dirname $(grep -l amd_hsmp_hwmon \ /sys/class/hwmon/hwmon*/name | head -1))/power1_cap # negative write echo -1000000 > $CAP ; echo "ret=$?" # valid positive write must still work echo 400000000 > $CAP ; echo "ret=$?" Before: # echo -1000000 > $CAP ; echo "ret=$?" ret=0 <- accepted; bogus limit sent to SMU # echo 400000000 > $CAP ; echo "ret=$?" ret=0 After: # echo -1000000 > $CAP ; echo "ret=$?" bash: echo: write error: Invalid argument ret=1 <- rejected with -EINVAL # echo 400000000 > $CAP ; echo "ret=$?" ret=0 <- valid write still works	N/A	More Details
CVE-2026-90171	In the Linux kernel, the following vulnerability has been resolved: smb: smbdirect: release pending child sockets outside the handler lock smbdirect_socket_destroy() releases the listener's pending/ready child sockets while still holding the listener's handler lock, the &id_priv->handler_mutex taken via rdma_lock_handler(), not sc->listen.lock, and before the listener's own rdma_destroy_id(). That ordering has one real consequence and one cosmetic one. The real one: smbdirect_socket_release() drops the child's last reference, which destroys the child's cm_id. Doing that before the listener's rdma_destroy_id() lets _cma_cancel_listens(), running from the listener's _destroy_id(), walk an already freed child id_priv, which KASAN catches as a slab-use-after-free during listener shutdown: [4758.909130] BUG: KASAN: slab-use-after-free in __mutex_lock+0x1469/0x1560 [4758.911450] Read of size 1 at addr ffff88821c381db4 by task ksmbd.control/1652 [4758.913262] Call Trace: [4758.913267] <TASK> [4758.913299] __mutex_lock+0x1469/0x1560 [4758.913408] _cma_cancel_listens+0x312/0x3b0 [4758.913413] _destroy_id+0x363/0xee0 [4758.913417] smbdirect_socket_destroy_sync+0x17d5/0x2440 [4758.913443] smbdirect_socket_release+0x124/0x230 [4758.913451] ksmbd_rdma_stop_listening+0x9f/0x190 [4758.913457] ksmbd_conn_transport_destroy+0x65/0x3c0 [4758.913463] kill_server_store+0x1fb/0x2b0 [4758.913501] kernfs_fop_write_iter+0x349/0x4d0 [4758.913507] vfs_write+0x5e7/0xc70 [4758.913528] ksys_write+0x12a/0x210 [4758.913541] do_syscall_64+0x135/0x460 [4758.913555] entry_SYSCALL_64_after_hwframe+0x77/0x7f The cosmetic one: releasing a child recurses into smbdirect_socket_destroy(), which takes the child's own rdma_lock_handler() lock nested under the listener's. The listener's and the child's cm_id are always different instances, so this cannot deadlock for real; the CM core itself nests a new connection id's handler_mutex under the listening id's in cma_ib_req_handler(). But lockdep only sees one lock class, reports possible recursive locking, and then disables itself, hiding real locking bugs for the rest of the run: [2424.579653] WARNING: possible recursive locking detected [2424.581180] 7.1.0-next-20260623+ #89 Not tainted [2424.582548] ---------- [2424.584500] ksmbd.control/8854 is trying to acquire lock: [2424.586817] ffff888102303c20 (&id_priv->handler_mutex){+.+.}-{4:4}, at: smbdirect_socket_destroy_sync+0xc39/0x2440 [2424.590590] [2424.590590] but task is already holding lock: [2424.591601] ffff888102046c20 (&id_priv->handler_mutex){+.+.}-{4:4}, at: smbdirect_socket_destroy_sync+0xc39/0x2440 [2424.594178] [2424.594178] other info that might help us debug this: [2424.596634] Possible unsafe locking scenario: [2424.596634] [2424.598841] CPU0 [2424.599765] ---- [2424.600695] lock(&id_priv->handler_mutex); [2424.601836] lock(&id_priv->handler_mutex); [2424.602590] [2424.602590] *** DEADLOCK *** [2424.602590] [2424.604512] May be due to missing lock nesting notation Splice the pending/ready children onto a local list under the listener's listen.lock, while the handler lock is held so a concurrent CM CONNECT_REQUEST cannot add more, but defer the actual smbdirect_socket_release() calls until after the listener's cm_id has been destroyed and its handler lock dropped. The children are independent sockets whose teardown needs neither the listener's handler lock nor its cm_id. Found with ksmbdzzer [2], a KSMBD fuzzer that drives libFuzzer with a kcov-dataflow [1] coverage vector: it folds each instrumented comparison/argument's runtime operand value together with its PC (the default arm mixes them as pc⊕val) so that a new operand value at a known site counts as new coverage. [1] https://lwn.net/Articles/1077606/ [2]	N/A	More Details

	https://github.com/yskzalloc/kcov-dataflow		
CVE-2026-75516	The RabbitMQ Java client library allows Java and JVM-based applications to connect to and interact with RabbitMQ nodes. Prior to 5.34.0, AMQConnection.start() applies Math.min(maxInboundMessageBodySize, frameMax) after Connection.Tune negotiation even though AMQP defines frameMax value zero as unlimited and ConnectionFactory.DEFAULT_FRAME_MAX is zero. When the client default and server-negotiated value are both zero, the result is passed to Utils.framePayloadLimit(int), which interprets zero as Integer.MAX_VALUE and disables the configured maxInboundMessageBodySize cap. A malicious AMQP server, or a man-in-the-middle attacker able to modify Connection.Tune and inject frames into the connection, can then send an oversized frame of any frame type, causing Frame.readFrom() to allocate a large byte array before content-level validation and potentially terminate the client process through memory exhaustion. This issue is fixed in version 5.34.0.	N/A	More Details
CVE-2026-90031	In the Linux kernel, the following vulnerability has been resolved: usb-storage: ene_ub6250: fix race between scan work and probe ene_ub6250_probe() calls usb_stor_probe2(), which starts the usb-storage infrastructure and schedules the delayed scan work. The driver then calls ene_get_card_type(), which sends an ENE command through ene_send_scsi_cmd() and the usb-storage bulk transfer helpers. Both the delayed scan work, through usb_stor_Bulk_max_lun(), and ene_get_card_type() use us->current_urb. The scan work serializes this access with us->dev_mutex, but the ENE card-type probe does not. If the scan work runs while ene_get_card_type() is still using us->current_urb, usb_submit_urb() warns that the URB is already active. Serialize ene_get_card_type() with us->dev_mutex, matching the locking used by the scan path.	N/A	More Details
CVE-2026-90220	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: Don't leak the extension cell pointer in the bounce payload The bounce_error_event() embeds the failed event in the bounce payload by pointing data.ext.ptr at it. When that event is a queued variable-length event, its own data.ext.ptr holds the address of its first extension cell, put there by snd_seq_event_dup(). The payload goes out verbatim through snd_seq_expand_var_event(), so the address reaches userspace. That is the same address commit 705dd6dcbc0e ("ALSA: seq: Clear variable event pointer on read") removed from the event header. The read path still clears it there, just above the call that expands the payload. Embed a sanitised copy instead, treated exactly as snd_seq_read() treats the header. A stack copy is enough because delivery is synchronous and snd_seq_event_dup() copies before returning. An unprivileged client reaches this by setting SNDRV_SEQ_FILTER_BOUNCE, queueing a variable-length event to a port that does not exist and reading the bounce back. Eight bytes on 64-bit, from its own pool.	N/A	More Details
CVE-2026-90208	In the Linux kernel, the following vulnerability has been resolved: clocksource/drivers/samsung_pwm: Switch to raw_spinlock_t type Samsung PWM timer might be used as a clock source on some legacy systems. When PREEMPT_RT is enabled on ARM, regular spinlock is converted to a sleeping lock (mutex-based), which must not be used in atomic context such as hard interrupt handlers. Switch the samsung_pwm_lock to the raw_spinlock, which remains a true non-sleeping spinlock even under PREEMPT_RT.	N/A	More Details
CVE-2026-90209	In the Linux kernel, the following vulnerability has been resolved: s390/debug: Fix deadlock during unregister Unregistering an s390dbf debug area while one of the associated debugfs files is being written to can cause a deadlock: $ echo >.../vmur/level $ rmmod vmur == debugfs write debugfs_file_get() debug_unregister() mutex_lock(debug_mutex) debugfs_remove() wait for debugfs_file_put() debug_file_ops.write() debug_input() mutex_lock(debug_mutex) ==> DEADLOCK Fix this by splitting debug_unregister() into an s390dbf and debugfs part, and running only the s390dbf part with debug_mutex locked.	N/A	More Details
CVE-2026-90211	In the Linux kernel, the following vulnerability has been resolved: bpf, s390: Clear fetch destination on faulting arena atomic Same missing register clear as on riscv64. A RMW atomic on an arena pointer is converted to BPF_PROBE_ATOMIC and gets an exception table entry, but bpf_jit_probe_atomic_pre() only fills in the arena base and the probe offset, leaving probe->reg at the -1 that bpf_jit_probe_init() set, which bpf_jit_probe_post() writes into the entry and ex_handler_bpf() then reads back as "there is nothing to clear". That is right for a plain BPF_{ADD,AND,OR,XOR}, which only writes memory, but an RMW carrying BPF_FETCH also reads the old value into a register: src_reg for BPF_{ADD,AND,OR,XOR} | BPF_FETCH and BPF_XCHG, and r0 for BPF_CMPXCHG. So on a fault over an unmapped arena page the program resumes at the landing pad with whatever that register held before the atomic instead of the 0 that every other BPF_PROBE_* access delivers. Fill probe->reg in from bpf_atomic_load_reg(). Unlike x86-64 and arm64, s390x does not report arena violations from its exception handler, so there is no access direction to correct here, only the missing register clear.	N/A	More Details
CVE-2026-90212	In the Linux kernel, the following vulnerability has been resolved: arm64/efi: Avoid voluntary preemption with efi_mm installed Gus reports a bad kernel memory access when using software PAN (CONFIG_ARM64_SW_TTBR0_PAN=y) on a machine with support for EFI runtime services: Unable to handle kernel access to user memory outside uaccess routines at virtual address 00000000f322ff30 Mem abort info: ESR = 0x0000000096000004 FSC = 0x04: level 0 translation fault Internal error: Oops: 0000000096000004 [#1] SMP Workqueue: efi_rts_wq efi_call_rts pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : efi_call_rts+0xd8/0x288 Call trace: efi_call_rts+0xd8/0x288 (P) process_one_work+0x178/0x4f8 worker_thread+0x194/0x328 This is because the fpsimd context management code called from __efi_fpsimd_begin() can preempt voluntarily, returning later to the EFI code with an incorrect value for TTBR0_EL1 thanks to the deferred mm switching used by the software PAN implementation. Since EFI runtime services cannot preempt voluntarily and because the fpsimd switching code does not rely on the TTBR0_EL1 mappings, simply reorder the fpsimd switch so that it occurs before we change the page-table.	N/A	More Details
CVE-2026-90213	In the Linux kernel, the following vulnerability has been resolved: firewire: core: fix memory leak in error path of build_tree() In the error path of build_tree(), node instances can remain in the local linked list when the function returns. Whenever an invalid value is detected in the self ID sequence, each allocated node instance is either an entry in the linked list or an entry in the ports array of its parent node. Therefore, the allocate node instances can be safely released by traversing the linked list from its head. Release the remaining node instances with for_each_fw_node() before returning to the caller.	N/A	More Details
CVE-2026-90214	In the Linux kernel, the following vulnerability has been resolved: ASoC: xilinx: formatter_pcm: fix stream_data leak on open error In xlnx_formatter_pcm_open(), stream_data is allocated and adata->play_stream or adata->capture_stream is assigned early. If a later step, such as snd_pcm_hw_constraint_step() or snd_pcm_hw_constraint_integer(), fails, the function returns the error immediately. ALSA does not call the close callback when open fails, so stream_data is leaked and the stream pointer is left dangling, pointing to a substream that ALSA frees. A later interrupt would then call snd_pcm_period_elapsed() on the freed substream. Free stream_data and clear the stream pointer on the error paths.	N/A	More Details
CVE-2026-	In the Linux kernel, the following vulnerability has been resolved: mtd: ubi: Release device reference on busy detach ubi_detach_mtd_dev() obtains a device reference through ubi_get_device() before checking whether the UBI device is busy. The	N/A	More

90215	busy return path drops ubi->ref_count but leaves the device reference held, so the device object cannot be released after a later detach. Drop the device reference before returning -EBUSY.		Details
CVE-2026-90216	In the Linux kernel, the following vulnerability has been resolved: ubi: Fix rollback for explicit UBI device numbers ubi_init_attach() rolls back module initialization failures by scanning ubi_devices[0..i-1], where i is the mtd= parameter index. That assumes the parameter index matches the UBI device number. That assumption is not true when mtd= specifies an explicit ubi_num. A successfully attached device can be stored at a higher ubi_devices[] slot, and a later failure can miss it during rollback. Scan the full ubi_devices[] array and detach by the actual array index, matching the way UBI devices are stored.	N/A	More Details
CVE-2026-90218	In the Linux kernel, the following vulnerability has been resolved: RDMA/cma: Fix WARNING in res_to_rt syzbot reported a WARN_ON(!res->dev) in res_to_rt() triggered via addr_handler() during asynchronous address resolution: " WARNING: drivers/infiniband/core/restrack.c:138 at res_to_rt+0x1c4/0x230 CPU#1: kworker/u8:4/59 Modules linked in: CPU: 1 UID: 0 PID: 59 Comm: kworker/u8:4 Not tainted syzkaller #0 PREEMPT(full) Hardware name: Google Compute Engine, BIOS Google 07/24/2026 Workqueue: ib_addr process_one_req RIP: 0010:res_to_rt+0x1c4/0x230 drivers/infiniband/core/restrack.c:138 RSP: 0018:ffffc9000201f850 EFLAGS: 00010293 RAX: ffffffff88d00ce5 RBX: ffff88807f0fd4f8 RCX: ffff88801e6e0000 RDX: 0000000000000000 RSI: ffffffff8fd996f0 RDI: 0000000000000003 RBP: 0000000000000000 R08: ffff88801e6e0000 R09: 000000000000000a R10: 0000000000000009 R11: 0000000000000000 R12: dfffffc0000000000 R13: 1ffff1100fe1fa9f R14: 0000000000000000 R15: 0000000000000003 FS: 0000000000000000(0000) GS:ffff888125012000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00001d559c3d2000 CR3: 0000000077c4c000 CR4: 00000000003526f0 Call Trace: <TASK> rdma_restrack_add+0x5a/0x8a0 drivers/infiniband/core/restrack.c:236 addr_handler+0x41a/0x5a0 drivers/infiniband/core/cma.c:3534 process_one_req+0x2eb/0x540 drivers/infiniband/core/addr.c:624 process_one_work kernel/workqueue.c:3375 [inline] process_scheduled_works+0xc4e/0x1630 kernel/workqueue.c:3458 worker_thread+0xa47/0xfb0 kernel/workqueue.c:3539 kthread+0x388/0x470 kernel/kthread.c:436 ret_from_fork+0x514/0xb70 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK> " In addr_handler(), cma_acquire_dev_by_src_ip() is called to populate id_priv->cma_dev and bind the associated ib_device to id_priv->id.device. If cma_acquire_dev_by_src_ip() returns an error (non-zero status), the ID remains unassociated with any RDMA device. Previously, rdma_restrack_add(&id_priv->res) was invoked unconditionally even when cma_acquire_dev_by_src_ip() failed, passing a resource with a NULL dev pointer and triggering the WARN_ON assertion in res_to_rt(). Fix this by only adding the resource to restrack when acquiring the device succeeds.	N/A	More Details
CVE-2026-90219	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Free debugfs on registration failure c4iw_alloc() creates the per-device debugfs tree (dev->debugfs_root via setup_debugfs()), but it is removed only in c4iw_remove(), not in c4iw_dealloc(). When RDMA device registration fails, the registration worker's err_dealloc_ctx path calls c4iw_dealloc() directly, bypassing c4iw_remove(), so the debugfs dentries leak and outlive the freed c4iw_dev. Move debugfs_remove_recursive() into c4iw_dealloc() so every path that frees ctx->dev also removes its debugfs tree.	N/A	More Details
CVE-2026-90221	In the Linux kernel, the following vulnerability has been resolved: nfc: nci: fix use of uninitialized memory in CORE_INIT_RSP parsing nci_core_init_rsp_packet_v1() and nci_core_init_rsp_packet_v2() parse the CORE_INIT_RSP packet without validating that the skb contains enough data. A malformed response (e.g. injected via virtual_ncidev) can declare a large num_supported_rf_interfaces while providing insufficient data, causing reads of uninitialized slab memory. This is later used in nci_init_complete_req(), triggering a KMSAN uninit-value warning. Add skb length checks before accessing packet fields: - Validate the skb has at least 1 byte for the status field. - Validate the skb can hold the fixed-size header before parsing. - In v2, bounds-check each variable-length rf_interface entry and its extension parameters within the parsing loop. - In v1, verify the skb is large enough for both the variable-length rf_interfaces array and the trailing rsp_2 structure.	N/A	More Details
CVE-2026-90175	In the Linux kernel, the following vulnerability has been resolved: smb: server: fix leak of ksmbd_ipc_login_request_ext() returned buffer Free it unconditionally after ksmbd_alloc_user() calls. kmemleak splat: unreferenced object 0xffff888103b83540 (size 192): comm "pool-0", pid 16970, jiffies 4377290937 hex dump (first 32 bytes): 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... backtrace (crc 408ccc66): __kvmalloc_node_noprof+0x730/0x920 handle_generic_event+0xec/0x1a0 [ksmbd] genl_family_rcv_msg_doit+0xe0/0x130 genl_rcv_msg+0x181/0x290 netlink_rcv_skb+0x4f/0x100 genl_rcv+0x28/0x40 netlink_unicast+0x1e6/0x2c0 netlink_sendmsg+0x20a/0x450 ____sys_sendmsg+0x2e8/0x310 ___sys_sendmsg+0x78/0xc0 __sys_sendmsg+0x63/0xc0 do_syscall_64+0xa1/0x670 entry_SYSCALL_64_after_hwframe+0x76/0x7e	N/A	More Details
CVE-2026-90222	In the Linux kernel, the following vulnerability has been resolved: nfc: pn533: hold a reference to the request skb during send_frame __pn533_send_async() publishes the command and then calls dev->phy_ops->send_frame(). Once dev->cmd is set, an incoming frame can be matched to this command: the I2C threaded IRQ runs pn533_recv_frame(), which queues cmd_complete_work, and pn533_send_async_complete() frees cmd->req with consume_skb(). On the I2C transport, pn533_i2c_send_frame() still dereferences the same skb after i2c_master_send() returns, so a completion that races the send can free the skb while the transport is still using it. The request skb is owned by the command object and may be freed by command completion at any time after dev->cmd is published, so the transport send path must not assume it stays alive. Hold a temporary reference to the request skb across the send_frame() call so the transport always sees a live skb even if completion races the send. Add a pn533_send_cmd_frame() helper and use it from all three send paths.	N/A	More Details
CVE-2026-90226	In the Linux kernel, the following vulnerability has been resolved: nfc: llcp: avoid userspace overflow on invalid optlen nfc_llcp_getsockopt() casts optval to (u32 __user *) for put_user(), so the kernel always stores 4 bytes regardless of the caller-supplied optlen. The existing min_t(u32, len, sizeof(u32)) only clamps the length reported back to userspace; it does not constrain the store. A call with optlen < 4 therefore writes past the user buffer, violating the getsockopt(2) contract for all five supported optnames. Reject any call with optlen < sizeof(u32) up front. 'len' is int, so a plain size comparison would promote a negative optlen to size_t and slip past the check; an explicit 'len < 0' test is added first to catch negative values before the size compare.	N/A	More Details
CVE-2026-92627	A heap-use-after-free vulnerability exists in H5T__conv_f_f() in src/H5Tconv.c in HDF5 before 1.14.2. When converting a compound datatype containing floating-point members during a dataset read, a temporary buffer allocated with calloc() is freed and subsequently read from within the same conversion routine. An attacker who can supply a crafted HDF5 file containing a specially constructed compound datatype can trigger the use-after-free when the file is parsed by an application that reads the affected dataset, such as h5dump. This can result in a crash and, depending on heap layout and allocator behavior, may be exploitable for further memory corruption up to remote code execution.	N/A	More Details

CVE-2026-92571	Rejected reason: CVE ID reserved in error and not assigned to a vulnerability. The correct CVE ID is CVE-2026-92574.	N/A	More Details
CVE-2026-90232	In the Linux kernel, the following vulnerability has been resolved: amt: Don't support cross-netns setup. When a lower device is unregistered, amt_device_event() tries to unregister its upper AMT device, but it has two problems. 1. amt_lookup_upper_dev() looks up an upper device in the lower device's netns only 2. amt_device_event() unregisters a single upper device only If AMT device is created on a lower device in another netns, removing the lower device triggers the splat below and gets stuck until all upper devices are removed. [0] The cross-netns setup seems unintentional considering 1. and the following points: * amt_link_setup() sets dev->netns_immutable to true * skb_scrub_packet() is not called in the fast path * iproute2 binary fails to find cross-netns lower device via link-netns: # ip -n ns1 link add amt0 link-netns ns2 type amt dev veth1 Cannot find device "veth1" Instead of supporting it properly and preparing for per-netns netdev unreg, let's forbid cross-netns setup. Note that the problem 2. needs a separate fix. [0]: WARNING: net/core/dev.c:12518 at unregister_netdevice_many_notify+0x1cce/0x2250, CPU#48: ip/2031 Modules linked in: CPU: 48 UID: 0 PID: 2031 Comm: ip Not tainted 7.2.0-rc5+ #27 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 RIP: 0010:unregister_netdevice_many_notify (net/core/dev.c:12518) Code: 89 ef e8 d5 52 ae fe e9 d0 f4 ff ff 48 8d 3d f9 3b 9c 02 48 c7 c6 c0 0b 63 84 ba ab 1f 00 00 67 48 0f b9 3a e9 65 ff ff ff 90 <0f> 0b 90 eb 81 48 8d 3d f6 3b 9c 02 48 c7 c6 c0 0b 63 84 ba e2 1f RSP: 0018:ffffc90004abf160 EFLAGS: 00010212 RAX: ffff888104d38260 RBX: ffff88800b0911b8 RCX: dfffffc0000000000 RDX: 0000000000000000 RSI: 0000000000000008 RDI: ffffffff85b9f880 RBP: ffffc90004abf2d0 R08: ffffffff85b9f887 R09: 1ffffffff0b73f10 R10: dfffffc0000000000 R11: fffffbfff0b73f11 R12: ffff88800b091d08 R13: ffff88800b091178 R14: dfffffc0000000000 R15: ffff88800b091000 FS: 00007f555b86c600(0000) GS:ffff8881942a0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000562107d489c0 CR3: 0000000109a40002 CR4: 0000000000372ef0 Call Trace: <TASK> rtnl_dellink (net/core/rtnetlink.c:3632 net/core/rtnetlink.c:3674) rtnetlink_rcv_msg (net/core/rtnetlink.c:7112) netlink_rcv_skb (net/netlink/af_netlink.c:2556) netlink_unicast (net/netlink/af_netlink.c:1319) netlink_sendmsg (net/netlink/af_netlink.c:1900) ____sys_sendmsg (net/socket.c:775) __sys_sendmsg (net/socket.c:2738) do_syscall_64 (arch/x86/entry/syscall_64.c:63) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:121) ... unregister_netdevice: waiting for veth0 to become free. Usage count = 7 ref_tracker: netdev@ffff88800d7496d8 has 3/3 users at __netdev_adjacent_dev_insert (./include/linux/netdevice.h:4525 ./include/linux/netdevice.h:4554 net/core/dev.c:8791) __netdev_upper_dev_link (net/core/dev.c:8879 net/core/dev.c:8963) netdev_upper_dev_link (net/core/dev.c:9009) amt_newlink (drivers/net/amt.c:3321)	N/A	More Details
CVE-2026-90233	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: release descriptor pools on probe failure The per-NUMA-node descriptor DMA pools are created lazily from nvme_init_hctx_common() once the admin tag set is allocated, but they are only destroyed in nvme_remove() via nvme_release_descriptor_pools(). Any probe failure after the admin tag set has been allocated unwinds through the out_disable label and nvme_pci_free_ctrl(), neither of which releases the pools, leaking the dma_pool objects. Release the descriptor pools in the out_disable error path. It must not be added to nvme_pci_free_ctrl(), as that would double-free against nvme_remove() on the normal teardown path.	N/A	More Details
CVE-2026-90236	In the Linux kernel, the following vulnerability has been resolved: NFSD: Release the export reference when reaping open stateids nfs4_put_stid() releases the svc_export tracked in nfs4_stid.sc_export, but free_ol_stateid_reaplist() frees open and lock stateids by calling ->sc_free() directly, bypassing that path. An open stateid takes an sc_export reference in nfs4_open() and a lock stateid takes its own in init_lock_stateid(); both reach free_ol_stateid_reaplist() through their normal teardown, the open stateid via release_open_stateid() and the lock stateid via nfsd4_release_lockowner(), each through put_ol_stateid_locked(). The reference is therefore never dropped, pinning the export and blocking unmount for the lifetime of the stateid. Release sc_export in free_ol_stateid_reaplist() the way nfs4_put_stid() does. ->sc_free() runs once per stateid, and a stateid reaches free_ol_stateid_reaplist() or nfs4_put_stid() but never both, so the reference is dropped exactly once. Revoked stateids reach this path with sc_export already cleared by drop_stid_export(), so they are skipped rather than double-freed. nfs4_put_stid() itself read sc_export before acquiring cl_lock. drop_stid_export() clears that field and releases the reference under cl_lock, so a concurrent revocation could drop the export in the window between the read and the final put, releasing the same reference twice. Read sc_export while cl_lock is held so the two paths serialize and the reference is released exactly once.	N/A	More Details
CVE-2026-90238	In the Linux kernel, the following vulnerability has been resolved: media: amd: isp4: fix self-deadlock in isp4sd_pwron_and_init() error path isp4sd_pwron_and_init() holds ops_mutex via guard(mutex) and, on any init failure, jumps to err_deinit and calls isp4sd_pwroff_and_deinit(). That helper takes the same ops_mutex, re-acquiring a non-recursive mutex already held by the current thread, so any init failure deadlocks. Unwind the error path in stages instead, releasing only what each failure point acquired. This also avoids the issues that an unconditional teardown would hit at the earlier failures, such as a runtime-PM underflow from pm_runtime_resume_and_get() and MMIO access while the device is unpowered.	N/A	More Details
CVE-2026-90239	In the Linux kernel, the following vulnerability has been resolved: media: amd: isp4: release partial allocations in isp4if_alloc_fw_gpumem() isp4if_alloc_fw_gpumem() allocates several GPU memory pools in sequence. If one of them fails, it jumps to error_no_memory and returns -ENOMEM without releasing the pools that were already allocated, leaking them. Release the already-allocated pools before returning. isp4if_gpu_mem_free() is a no-op on pools that were not allocated, so calling isp4if_dealloc_fw_gpumem() here safely frees exactly the pools that succeeded. isp4if_gpu_mem_free() previously logged an error for a NULL entry, which is a normal case during partial-allocation cleanup, so make it silent.	N/A	More Details
CVE-2026-90242	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix iopf_refcount leak on RID domain replacement intel_iommu_attach_device() enables IOPF for the new domain but never disables it for the old one. device_block_translation(), called at the start of the function, tears down translation but does not touch any IOPF state; blocking_domain_attach_dev() has to call iopf_for_domain_remove() explicitly before invoking it for exactly this reason. identity_domain_attach_dev() has the same problem. Its comment claims that no PRI handling is needed because the device has been put in the blocking state, but the blocking state and the IOPF reference count are independent of each other. As a result, replacing a domain that has an iopf_handler with another domain at RID level leaks a reference in info->iopf_refcount. The count never drops back to zero, so iopf_queue_remove_device() is never called and iommu_disable_pci_pri() triggers its WARN_ON(info->iopf_refcount) when the device is released. The PASID paths already handle this correctly by way of iopf_for_domain_replace(); convert the two RID paths to do the same. Using the replace helper rather than a bare remove keeps the enable before the disable, so the reference count does not transiently reach zero and evict the device from the IOPF queue.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix max_qid race between configfs and controller allocation The function nvmet_subsys_attr_qid_max_store() can race against nvmet_alloc_ctrl() when a subsystem's max_qid limit is modified. Suppose max_qid is currently 64. If nvmet_alloc_ctrl() executes: ctrl->sqs = kzalloc_objs(struct nvmet_sq *,		

CVE-2026-90206	subsys->max_qid + 1); and at this exact point, a userspace process changes max_qid to 128, nvmet_subsys_attr_qid_max_store() will set the new max_qid value. It attempts to delete active controllers to force a reconnect, but the new controller won't be deleted because it hasn't been added to the subsys->ctrls list yet. nvmet_alloc_ctrl() then proceeds and adds the new controller to the subsys->ctrls list. Later, when nvmet_install_queue() is called, it will see max_qid set to 128, but the memory allocated for sqs is only sized for 64 entries. This results in a KASAN out-of-bounds warning and potential memory corruptions. Fix this by protecting the queue allocations and list insertion in nvmet_alloc_ctrl() with down_read(&nvmet_config_sem). Because nvmet_subsys_attr_qid_max_store() acquires down_write(&nvmet_config_sem) to modify the attribute, this safely prevents the configfs writer from modifying max_qid during controller creation. Copy the max_qid from the subsystem to the controller's structure during the allocation; ctrl->max_qid never changes as long as the controller remains in LIVE state, so this will prevent similar race conditions.	N/A	More Details
CVE-2026-59823	LiteLLM is a proxy server (AI Gateway) to call LLM APIs in OpenAI (or native) format. Prior to 1.83.9, an authenticated LiteLLM Proxy caller with a valid virtual key can place api_base inside the user_config request body to bypass is_request_body_safe, which blocks top-level api_base and base_url but previously did not inspect or reject user_config. Because user_config constructs the outbound router, the nested destination redirects a server-side request to an internal or external host selected by the caller and can expose endpoints the caller cannot otherwise access. This issue is fixed in version 1.83.9.	N/A	More Details
CVE-2026-90202	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Avoid freeing unallocated PCIe SGL buffers _base_release_memory_pools() unconditionally frees every ioc->pcie_sg_lookup[] entry, including ones the setup loop never allocated after a partial failure, causing a "bad dma" warning on debug kernels or a NULL pointer dereference otherwise.	N/A	More Details
CVE-2026-90201	In the Linux kernel, the following vulnerability has been resolved: net: page_pool: fix UAF in __page_pool_release_netmem_dma on xa_cmpxchg race This bug was discovered while testing the hns3 driver under channel reconfiguration (`ethtool -L` / `ethtool -G`) with iperf3 traffic on arm64. The race is intermittently triggered when page_pool_destroy() runs page_pool_scrub() concurrently with page return via page_pool_put_netmem() on a different CPU. A WARN in page_pool_clear_pp_info() surfaced the dangling DMA index bits left by the cmpxchg loser, which led to the investigation. page_pool_scrub() iterates pool->dma_mapped via xa_for_each() with no page ref held. __page_pool_release_netmem_dma() currently reads and writes netmem fields (dma_addr, DMA index bits in pp_magic) after xa_cmpxchg() returns. The unref path calls put_page() unconditionally regardless of the cmpxchg outcome; when it loses the cmpxchg, it still frees the page before the scrub winner finishes these netmem accesses, so scrub touches a freed page -- a Use-After-Free. Fix this by splitting the DMA release into two functions: 1. __page_pool_unmap_netmem_dma() caches dma_addr before xa_cmpxchg(), does the cmpxchg to remove the DMA mapping, and calls dma_unmap on the cached address. It never touches netmem fields after the cmpxchg, making it safe for the scrub path which holds no page ref. 2. __page_pool_release_netmem_dma() wraps the above and additionally clears dma_addr and DMA index bits in netmem fields. This is safe only when the caller holds a page ref, so it is used by the return path (page_pool_return_netmem). The scrub path calls __page_pool_unmap_netmem_dma() directly; the return path calls __page_pool_release_netmem_dma().	N/A	More Details
CVE-2026-90178	In the Linux kernel, the following vulnerability has been resolved: hwmon: (coretemp) Fix core_data leak on CPUs without PTS pdata->core_data is allocated in init_temp_data() when the first core temp_data of a package is created, but it is only released from destroy_temp_data(), and only in the branch that handles the package temp_data. Package temp_data is created solely when the CPU supports X86_FEATURE_PTS. On a CPU without it, coretemp_cpu_online() never calls coretemp_add_core() with pkg_flag set, so pdata->pkg_data stays NULL. coretemp_cpu_offline() then skips the removal of the package interface, destroy_temp_data() is never called for package data, and the array is still allocated when coretemp_device_remove() frees the platform data that pointed at it. Release the array in coretemp_device_remove(). destroy_temp_data() sets pdata->core_data to NULL when it frees it, so the added kfree() is a no-op on CPUs that do have PTS. Tested on an Intel Core i5-1135G7. The driver was instrumented to log every allocation and release of pdata->core_data, and the PTS check in coretemp_cpu_online() was patched out to emulate a CPU without package thermal support. Without this change the array was allocated and never released, and coretemp_device_remove() still saw a non-NULL pointer. With it the array is released and the pointer accounting balances. On an unmodified build the release still happens via the package temp_data and the added kfree() sees NULL, with no slab warnings over repeated module load and unload cycles.	N/A	More Details
CVE-2026-90179	In the Linux kernel, the following vulnerability has been resolved: apparmor: fix deadlock in complain-mode change_hat The use of change_hat when in complain mode can cause a deadlock when the hat doesn't exist and a new learning profile is created for the missing profile. This is because change_hat() has taken the lock to search the hat list and creating the new learning profile needs to take the lock to add it to the list. From the bug report: Originally found in 7.0.0 in LTS ubuntu 26.04 with pam_apparmor + su in complain mode set to change hats. Then verified in newest available vanilla kernel I've compiled to see if still present: 7.2-rc7 vanilla -> affected checked also some other kernels: 6.18.44 vanilla -> affected 6.12.95 with debian patches -> unaffected On systems without bug (for example 6.12.95 debian) it just prints: aa_change_hat rc=0 On systems with bug, the executable always hangs, prints nothing and becomes unkillable. (And once stuck this way, it will cause any further hat changes to also cause the changing process to get stuck) Then in syslog you can find hint about cause: kernel: INFO: task hat:3409 blocked for more than 483 seconds. kernel: Not tainted 7.2.0-rc7 #1 kernel: "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. kernel: task:hat state:D stack:0 pid:3409 tgid:3409 ppid:2605 task_flags:0x400000 flags:0x00080800 kernel: Call Trace: kernel: <TASK> kernel: __schedule+0x48f/0xfe0 kernel: schedule+0x27/0xa0 kernel: schedule_preempt_disabled+0x15/0x30 kernel: __mutex_lock.constprop.0+0x569/0xa10 kernel: aa_new_learning_profile+0x15f/0x210 kernel: build_change_hat+0x19f/0x3b0 kernel: change_hat.isra.0+0x5dd/0xd60 kernel: aa_change_hat+0x2f3/0x710 kernel: aa_setprocattr_changehat+0x121/0x1f0 kernel: do_setattr+0x28c/0x340 kernel: apparmor_setselfattr+0x20/0x50 kernel: security_setselfattr+0xf6/0x110 kernel: __x64_sys_lsm_set_self_attr+0x53/0x90 kernel: do_syscall_64+0xdd/0x5e0 kernel: ? __mod_memcg_lruvec_state+0xfd/0x260 kernel: ? lruvec_stat_mod_folio+0x8d/0xd0 kernel: ? __folio_mod_stat+0x2d/0x90 kernel: ? map_anon_folio_pte_nopf+0xd1/0x1f0 kernel: ? do_anonymous_page+0x184/0xa10 kernel: ? __handle_mm_fault+0x805/0x870 kernel: ? count_memcg_events+0xef/0x230 kernel: ? handle_mm_fault+0x1f0/0x2f0 kernel: ? do_user_addr_fault+0x2bb/0x7b0 kernel: ? do_syscall_64+0x94/0x5e0 kernel: ? exc_page_fault+0x75/0x160 kernel: entry_SYSCALL_64_after_hwframe+0x76/0x7e kernel: RIP: 0033:0x7f815e134c8d kernel: RSP: 002b:00007fff6df94ea8 EFLAGS: 00000246 ORIG_RAX: 00000000000001cc kernel: RAX: ffffffffffffffda RBX: 0000556d8c81d040 RCX: 00007f815e134c8d kernel: RDX: 0000000000000046 RSI: 0000556d8c81d040 RDI: 0000000000000064 kernel: RBP: 00007fff6df94ef0 R08: 00007f815e212ac8 R09: 000000000000000c kernel: R10: 0000000000000000 R11: 0000000000000246 R12: 0000556d8c81d010 kernel: R13: 0000000000000026 R14: 0000000000000046 R15: 0000000000000064 kernel: </TASK> kernel: INFO: task hat:3409 is blocked on a mutex likely owned by task hat:3409. To fix the issue, lift the locking out of the core of aa_new_learning_profile(), introduce a wrapper function that takes the lock where needed, and have build_change_hat() call the core function that no longer takes the lock. In addition fix 4	N/A	More Details

	other issues introduced by commit 32e92764d6f8d ("apparmor: grab ns lock and refresh when looking up changehat child profiles") - aa_get_profile_rcu() was replaced-by: aa_get_profile without the accompanying rcu_dereference_protected() - an extra aa_get_label(label) was introduced at the start of change_hat() without an accompanying aa_put_label() causing a reference count leak. - a reference count leak was introduced in the label_is_stale(label) case, where the newest profile would be leaked instead of the label passed to the function. - a potential UAF when the lookup walks up the tree with new_ns != ns the new label refere ---truncated---		
CVE-2026-90180	In the Linux kernel, the following vulnerability has been resolved: block: mtip32xx: synchronize ioctls with device removal The ioctl handlers only test REMOVE_PENDING before entering mtip_hw_ioctl(). Removal can set that bit immediately afterwards and free dd->port in mtip_hw_exit() while an ioctl still dereferences it. An already open block device can reach the handlers while del_gendisk() is in progress. Serialize both native and compat ioctls with removal. Set REMOVE_PENDING before taking the mutex so new callers fail after an in-flight ioctl has drained, and hold the mutex until the port has been torn down.	N/A	More Details
CVE-2026-90181	In the Linux kernel, the following vulnerability has been resolved: ublk: avoid teardown retry loop on xarray allocation failure __ublk_shmem_remove_ranges() removes matching maple tree ranges in batches, but first stores each range into a temporary xarray so that the pages can be unpinned after dropping the maple tree lock. That temporary xarray is filled under the maple tree lock with xa_store(..., GFP_ATOMIC). If the store fails before mas_erase(), the current range is left in the tree and the helper returns false. The outer ublk_shmem_remove_ranges() loop then immediately retries the same range. While the atomic allocation keeps failing, the teardown path has no forward progress. The issue can be reproduced with radix_tree_node failslab injection after a SHMEM_ZC buffer has already been registered: # Kernel config: # CONFIG_BLK_DEV_UBLK=y # CONFIG_DEBUG_FS=y # CONFIG_FAULT_INJECTION=y # CONFIG_FAULT_INJECTION_DEBUG_FS=y # CONFIG_FAILSLAB=y echo 10 > /proc/sys/vm/nr_hugepages mkdir -p /tmp/htlb mount -t hugetlbfs none /tmp/htlb fallocate -l 4M /tmp/htlb/ublk_buf dev_id=$(kublk add -t null --shmem_zc \ --htlb /tmp/htlb/ublk_buf | awk -F '[:]' '/dev id/ {print $3}') echo 1 > /sys/kernel/slab/radix_tree_node/failslab echo Y > /sys/kernel/debug/failslab/cache-filter echo Y > /sys/kernel/debug/failslab/ignore-gfp-wait echo 1 > /sys/kernel/debug/failslab/interval echo -1 > /sys/kernel/debug/failslab/times echo 100 > /sys/kernel/debug/failslab/probability kublk del -n "$dev_id" On the unfixed kernel the delete command was still running after 3 seconds. Disabling failslab made it return. The fault-injection stack showed: should_failslab kmem_cache_alloc_lru_noprof __xas_nomem __xa_store xa_store __ublk_shmem_remove_ranges ublk_cdev_rel ublk_ctrl_del_dev Remove the allocation from the teardown loop. Keep the existing batch limit, but collect {base_pfn, nr_pages} pairs in a fixed-size stack array. Once a matching range is found, the range is erased from the maple tree before dropping the lock, so each successful scan makes progress without depending on any GFP_ATOMIC allocation. With the same failslab settings, the fixed kernel completed "kublk del -n $dev_id" successfully in about 45 ms.	N/A	More Details
CVE-2026-90182	In the Linux kernel, the following vulnerability has been resolved: blk-iocost: clear delay state when freeing policy data iocg_kick_delay() turns sufficiently large debt into an explicit block-cgroup delay with blkcg_set_delay(), setting blkg->use_delay to -1 and incrementing blkcg->congestion_count. Clearing it again depends on iocg_kick_delay() running from the period timer, the waitq timer or the issue path. ioc_pd_free() removes the iocg from active_iocgs and cancels its waitq timer, and no further bios can arrive, so once it has run nothing is left which can reduce the debt and clear the delay. The blkcg stays marked congested for the rest of its life. blk_cgroup_congested() then returns true for every task in that cgroup and its descendants: page_cache_sync_ra() cuts readahead to a single page, page_cache_async_ra() skips it altogether, and __folio_throttle_swaprate() takes swap_avail_lock and schedules a throttle on anonymous folio allocation. Clear it explicitly, after the list removal and the synchronous hrtimer_cancel() so that neither timer processing nor an I/O path can re-arm it. The free callback can also see policy data which was never attached to a blkg, hence the pd->blkg check.	N/A	More Details
CVE-2026-90183	In the Linux kernel, the following vulnerability has been resolved: blk-iolatency: clear delay state when freeing policy data io.latency can throttle a group which has no latency target of its own. When a sibling misses its target, check_scale_change() scales down its peers, and a peer that reaches queue depth one gets blkcg_use_delay() called on it on every further scale-down, even with min_lat_nsec == 0. iolatency_pd_offline() resets the target through iolatency_set_min_lat_nsec(), which clears the delay only on a nonzero to zero transition, so it never clears such a peer. Freeing the policy data then leaves blkg->use_delay set and blkcg->congestion_count elevated with nothing left that can drop it. blk_cgroup_congested() then returns true for every task in that cgroup and its descendants for as long as the cgroup lives: page_cache_sync_ra() cuts readahead to a single page, page_cache_async_ra() skips it altogether, and __folio_throttle_swaprate() takes swap_avail_lock and schedules a throttle on anonymous folio allocation. Clear the delay in iolatency_pd_free(). By then bio-held blkg references have drained, or the queue is frozen for policy deactivation, so check_scale_change() cannot re-arm it. The free callback can also see policy data which was never attached to a blkg, hence the pd->blkg check.	N/A	More Details
CVE-2026-90184	In the Linux kernel, the following vulnerability has been resolved: null_blk: serialize configfs attribute updates with device setup The attribute store methods generated with NULLB_DEVICE_ATTR() refuse to change the configuration of a live device by testing NULLB_DEV_FL_CONFIGURED, but that flag is only set by nullb_device_power_store() after null_add_dev() has returned, and the store methods take no lock at all. configfs only serializes writes to the same open file (buffer->mutex), so a write to any attribute can run concurrently with null_add_dev() and change the device configuration while it is being used. null_add_dev() reads the configuration several times, e.g. dev->zoned is read once to set up the queue limits and once to initialize the zone resources: CPU0: echo 1 > nullb0/power CPU1: echo 1 > nullb0/zoned nullb_device_power_store() mutex_lock(&lock) null_add_dev() if (dev->zoned) -> false /* no BLK_FEAT_ZONED */ nullb_device_zoned_store() test_bit(FL_CONFIGURED) -> 0 dev->zoned = true blk_mq_alloc_disk() /* queue is not zoned */ if (nullb->dev->zoned) -> true null_register_zoned_dev() blk_revalidate_disk_zones() blk_revalidate_disk_zones() is then called for a queue that does not have BLK_FEAT_ZONED set, which triggers its WARN_ON_ONCE() and fails the device setup with -EIO: WARNING: CPU: 2 PID: 322 at block/blk-zoned.c:2357 blk_revalidate_disk_zones+0x4c/0x560 Clearing dev->zoned in the same window is worse: the queue is created with BLK_FEAT_ZONED but the zone resources are never initialized, so add_disk() succeeds for a zoned disk that has no zones. And a store that lands after the last dev->zoned test leaves dev->zoned set while dev->zones is still NULL, which null_process_zoned_cmd() dereferences on the first write. Fix this by taking the global lock, which nullb_device_power_store() already holds across null_add_dev() and null_del_dev(), around both the NULLB_DEV_FL_CONFIGURED test and the update of the device configuration. The submit_queues and poll_queues apply callbacks are now called with that lock held, so remove the locking they did themselves. Since the store methods can run as soon as configfs_register_subsystem() returns, that is, before null_init() gets to mutex_init(&lock), also initialize the lock statically with DEFINE_MUTEX().	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: null_blk: serialize configfs attribute stores with the lock The NULLB_DEVICE_ATTR _store takes no lock: apply_fn attributes (submit_queues, poll_queues) get dev->NAME written again after apply_fn returns, outside its lock; APPLY=NULL attributes are entirely lockless. configfs only serializes stores per-open-file, so concurrent stores on separate fds race. For apply_fn attributes, once one store's apply_fn has reconfigured the hardware, a		

CVE-2026-90185	second (losing) store can still overwrite dev->NAME afterwards. This leaves dev->submit_queues out of sync with the live queue count, which is later caught by the WARN_ON_ONCE() in null_map_queues(). For !apply_fn attributes, power_store()'s null_add_dev() validates and builds the device under "lock" but only sets CONFIGURED afterwards. A store slipping in during this window can change a field mid-setup -- for example, zone_nr_conv can be pushed above nr_zones after it has already been clamped, leading to an out-of-bounds dev->zones[] access. Take "lock" in the macro around the apply_fn call, the CONFIGURED test and the field write, and move it out of nullb_apply_submit_queues()/ nullb_apply_poll_queues() so both paths are covered once. This serializes stores with power_store's setup and with each other.	N/A	More Details
CVE-2026-90186	In the Linux kernel, the following vulnerability has been resolved: null_blk: reject per-device queue resize for shared tag set When shared_tags is enabled, null_setup_tagset() makes the device use the global tag_set, whose driver_data stays NULL. null_map_queues() therefore falls back to the module-wide g_submit_queues/g_poll_queues instead of any per-device value. Resizing submit_queues or poll_queues via configfs on such a device calls blk_mq_update_nr_hw_queues() on the shared set, shrinking set->nr_hw_queues. __blk_mq_realloc_hw_ctxs() only grows the q->queue_hw_ctx[] allocation, so on shrink it merely exits and NULLs the now-excess hctx slots. null_map_queues(), however, keeps mapping CPUs with the unchanged g_submit_queues/g_poll_queues, so mq_map[] ends up pointing at those NULLed hctx slots. blk_mq_map_swqueue() then dereferences the NULL hctx (hctx->cpumask), crashing the kernel: [460.218374] KASAN: null-ptr-deref in range [0x0000000000000098-0x000000000000009f] [460.219003] CPU: 24 UID: 0 PID: 1492 Comm: sh Not tainted 7.2.0-rc2+ #67 PREEMPT(full) [460.219792] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-4.fc41 04/01/2014 [460.220452] RIP: 0010:blk_mq_map_swqueue+0x4db/0x1430 ..... [460.228977] Call Trace: [460.229175] <TASK> [460.229354] blk_mq_update_nr_hw_queues+0xd49/0x11c0 [460.229779] ? __pfx_blk_mq_update_nr_hw_queues+0x10/0x10 [460.230200] nullb_update_nr_hw_queues+0x1a9/0x370 [null_blk] [460.230694] nullb_device_submit_queues_store+0xd9/0x170 [null_blk] [460.231190] ? __pfx_nullb_device_submit_queues_store+0x10/0x10 [null_blk] [460.231776] ? configfs_write_iter+0x35c/0x4e0 [460.232122] configfs_write_iter+0x286/0x4e0 [460.232460] vfs_write+0x52d/0xd00 [460.232779] ? __x64_sys_openat+0x108/0x1d0 [460.233106] ? __pfx_vfs_write+0x10/0x10 [460.233413] ? fdget_pos+0x1cf/0x4c0 [460.233745] ? fput_close+0x133/0x190 [460.234038] ? __pfx_expand_files+0x10/0x10 [460.234368] ksys_write+0xfc/0x1d0 Reproducer: modprobe null_blk shared_tags=1 submit_queues=64 poll_queues=1 mkdir /sys/kernel/config/nullb/dev echo 1 > /sys/kernel/config/nullb/dev/power echo 1 > /sys/kernel/config/nullb/dev/submit_queues A per-device resize of a shared tag set is meaningless anyway, so reject it with -EINVAL in nullb_update_nr_hw_queues() when the device is bound to the global tag_set.	N/A	More Details
CVE-2026-90187	In the Linux kernel, the following vulnerability has been resolved: null_blk: free zones array on device power-off null_init_zoned_dev() allocates dev->zones when a zoned device is powered on, but null_del_dev() never frees it on power-off; dev->zones is only freed later in null_free_dev(), when the configfs directory is removed. If the device is powered off and then on again, null_init_zoned_dev() allocates a new array and overwrites the dev->zones pointer, leaking the previous allocation each power cycle. Free dev->zones in null_del_dev() via null_free_zoned_dev() to solve it. And calling null_free_zoned_dev() in null_free_dev() is no longer necessary because every caller already invokes null_del_dev() first: via nullb_group_drop_item() before nullb_device_release(), in the null_add_dev() error path of null_create_dev(), and in null_destroy_dev(). Remove the redundant call. And take &lock around zone_cond_store() in the two store wrappers to serialize dev->zones check-and-deref against its alloc/free, which already run under &lock. The reason there was no problem before is that only nullb_device_release() or null_exit() frees the dev->zones, which guarantees that subsequent users won't access the configfs interface.	N/A	More Details
CVE-2026-90188	In the Linux kernel, the following vulnerability has been resolved: null_blk: free global tag_set on init error path If shared_tags is enabled, null_setup_tagset() allocates the global tag_set via null_init_global_tag_set(). If device creation later fails, err_dev destroys the default devices and calls unregister_blkdev(), but never frees the global tag_set. Since module init failed, null_exit() is never invoked, so the global tag_set's tags and maps are permanently leaked. Free the global tag_set in err_dev, matching null_exit() which does if (tag_set.ops) blk_mq_free_tag_set(&tag_set).	N/A	More Details
CVE-2026-90189	In the Linux kernel, the following vulnerability has been resolved: null_blk: register configfs subsystem after creating default devices In null_init(), configfs_register_subsystem() currently runs before register_blkdev(), so when null_blk is built as a module, a racing mkdir() + poweron from userspace can reach null_add_dev() while null_major is still 0. __add_disk() then hits WARN_ON(disk->minors) (major=0 with minors!=0) and fails: [root@fedora ~]# [2366.521436] WARNING: block/genhd.c:476 at __add_disk+0x8a7/0xde0, [2366.523552] Modules linked in: null_blk(+) nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib [2366.529081] CPU: 26 UID: 0 PID: 1600 Comm: sh Not tainted 7.2.0-rc1+ #66 PREEMPT(full) ..... [2366.547251] Call Trace: [2366.547575] <TASK> [2366.547831] ? _raw_spin_lock+0x84/0xe0 [2366.548260] add_disk_fwnode+0x114/0x560 [2366.548739] null_add_dev+0x102d/0x1b80 [null_blk] [2366.549310] ? __pfx_null_add_dev+0x10/0x10 [null_blk] [2366.549906] ? mutex_lock+0xde/0x1c0 [2366.550361] ? __pfx_mutex_lock+0x10/0x10 [2366.550827] nullb_device_power_store+0x1e7/0x280 [null_blk] [2366.551499] ? __pfx_nullb_device_power_store+0x10/0x10 [null_blk] [2366.552177] ? __kmalloc_cache_noprof+0x1f5/0x470 [2366.552748] ? configfs_write_iter+0x35c/0x4e0 [2366.553242] configfs_write_iter+0x286/0x4e0 [2366.553787] vfs_write+0x52d/0xd00 [2366.554169] ? __pfx_vfs_write+0x10/0x10 [2366.554679] ? __pfx___css_rstat_updated+0x10/0x10 [2366.555196] ? fdget_pos+0x1cf/0x4c0 [2366.555649] ksys_write+0xfc/0x1d0 ..... Additionally, the err_dev path destroys all devices on nullb_list while configfs is still registered. If a racing mkdir() + poweron puts a user device on the list, null_destroy_dev()->null_free_dev() kfrees the user device's nullb_device but /sys/kernel/config/nullb/<name> is still reachable. Any userspace access to the item will trigger a UAF. For simplicity, move configfs_register_subsystem() to the end to solve the problems above.	N/A	More Details
CVE-2026-90190	In the Linux kernel, the following vulnerability has been resolved: null_blk: use DEFINE_MUTEX for the file-scope mutex In null_init(), mutex_init(&lock) currently happens after configfs_register_subsystem(), which exposes the nullb subsystem to userspace. A racing mkdir() into /sys/kernel/config/nullb/ can reach null_find_dev_by_name() -> mutex_lock(&lock) before the mutex is initialized, trigger warning: [123.137788] DEBUG_LOCKS_WARN_ON(lock->magic != lock) [123.137796] WARNING: kernel/locking/mutex.c:159 at mutex_lock+0x171/0x1c0, CPU#13: mkdir/1301 [123.140090] Modules linked in: null_blk(+) nft_fib_inet nft_fib_ipv4 ..... [123.154926] Call Trace: [123.155172] <TASK> [123.155419] ? __pfx_mutex_lock+0x10/0x10 [123.156181] ? __pfx__raw_spin_lock+0x10/0x10 [123.156571] nullb_group_make_group+0x20/0x100 [null_blk] [123.157011] configfs_mkdir+0x47b/0xc70 [123.157337] ? __pfx_configfs_mkdir+0x10/0x10 [123.157719] ? may_create_dentry+0x242/0x2e0 [123.158061] vfs_mkdir+0x2a9/0x6c0 [123.158352] filename_mkdirat+0x3dc/0x500 [123.158710] ? __pfx_filename_mkdirat+0x10/0x10 [123.159070] ? strncpy_from_user+0x3a/0x1d0 [123.159413] __x64_sys_mkdir+0x6b/0x90 [123.159760] do_syscall_64+0xea/0x600 Replace the runtime mutex_init(&lock) with a static DEFINE_MUTEX(lock) declaration to fix this issue.	N/A	More Details

CVE-2026-90192	In the Linux kernel, the following vulnerability has been resolved: mailbox: qcom-cpucp: handle NULL data in send_data callback mailbox_clear_channel() calls mbox_send_message() with NULL data to notify the remote side that the RX channel has been cleared. qcom_cpucp_mbox_send_data() blindly dereferenced the data pointer, causing a NULL pointer dereference kernel panic when invoked from this path under PREEMPT_RT. Add an explicit NULL check and return early without writing to the TX register, which is the correct behaviour for a channel-clear notification.	N/A	More Details
CVE-2026-90193	In the Linux kernel, the following vulnerability has been resolved: mailbox: qcom-cpucp: fix PREEMPT_RT self-deadlock in IRQ handler qcom_cpucp_mbox_irq_fn() calls mbox_chan_received_data() while holding chan->lock. Under PREEMPT_RT, spin_lock_irqsave() is converted to an rt_spinlock (rtmutex-based), which tracks ownership and can sleep. The callback chain triggered by mbox_chan_received_data() eventually reaches mailbox_clear_channel() -> mbox_send_message() -> add_to_rbuf(), which attempts to re-acquire the same chan->lock. Since rtmutex detects the re-entrant lock attempt by the same owner, the thread blocks waiting for a lock it already holds, causing a permanent deadlock. This deadlock manifests as 'irq/N-apss_cpucp_mbox' stuck in D state with the following call trace: rt_spin_lock -> mbox_send_message -> mailbox_clear_channel -> scmi_rx_callback -> mbox_chan_received_data [<- held chan->lock here] Fix by saving chan->cl locally and clearing the HW interrupt register inside the lock, then invoking mbox_chan_received_data() after releasing the lock. This preserves the mutual exclusion for chan->cl access while avoiding the lock re-entrancy that causes the PREEMPT_RT deadlock.	N/A	More Details
CVE-2026-90194	In the Linux kernel, the following vulnerability has been resolved: ACPI: scan: fix bus ID cleanup on device_add() failures When device_add() fails after acpi_device_set_name() has allocated an instance ID and a new acpi_device_bus_id has been linked into acpi_bus_id_list, the rollback path only removes wakeup_list and detaches the ACPI handle data. That leaves the bus-ID bookkeeping behind and keeps the allocated instance number consumed. Move the bus-ID cleanup and wakeup-list removal into a single helper. Use it from both the normal device teardown path and the device_add() rollback path. The wakeup list node is initialized before registration, so it can be deleted without checking whether the device is wakeup- capable like in the original teardown path. [rjw: Rename acpi_device_del_list() to acpi_device_cleanup()] [rjw: Subject and changelog edits]	N/A	More Details
CVE-2026-90195	In the Linux kernel, the following vulnerability has been resolved: riscv, bpf: Fix missing sign-ext for signed 1-byte and 2-byte kfunc args On RV64, the ABI requires sign-extension for signed 1-byte and 2-byte kfunc args. However, the RV64 JIT currently does not perform sign-extension for such kfunc args. Before commit 7ce090afbf72 ("bpf: Infer zext_dst based on static register liveness analysis"), state pruning could potentially omit zero-extension of 32-bit subregisters, which inadvertently masked the above issue by making the args appear as if they had been properly sign-extended. After that commit, the problem is exposed, causing the kfunc_call/kfunc_call_test4 selftest to fail. Fix this by extending the existing sign-extension logic to handle signed 1-byte and 2-byte kfunc args as well.	N/A	More Details
CVE-2026-90196	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: validate topology volume range before allocation SOF treats the topology mixer min and max values as non-negative indices into its volume table. It stores them in signed fields, allocates max + 1 entries through an int argument, and later indexes the table with the stored range. An inverted range is invalid, while a maximum at or above INT_MAX cannot be represented safely after the increment or in the signed fields. Validate the complete range before storing it or allocating the table.	N/A	More Details
CVE-2026-90197	In the Linux kernel, the following vulnerability has been resolved: HID: haptic: don't write an uninitialized value to unhandled usages fill_effect_buf() initializes value only for the four haptic usages handled by its switch, but writes it to field->value[] for every usage. An unhandled usage can therefore receive either an uninitialized value or one left over from the previous usage. hid_output_report() then serializes that value into the effect's report buffer. Skip unhandled usages instead. This also matches switch_mode(), which only updates fields it recognizes. Found with Clang's -Wconditional-uninitialized.	N/A	More Details
CVE-2026-90198	In the Linux kernel, the following vulnerability has been resolved: ALSA: core: Fix use-after-free in snd_card_do_free() A use-after-free was detected in snd_card_do_free() when a sound card managed by devres is unbound while a user-space application still holds an open file descriptor. For managed cards, the memory is allocated using devres_alloc(), and its release function is set to __snd_card_release(), which calls snd_card_free(). When the device is unbound, the unbind thread calls snd_card_free(), which drops a reference to the card's device. If the user thread still has an open file descriptor, the reference count does not reach zero, and the unbind thread blocks on wait_for_completion(&released). When the user thread closes the file descriptor, it drops the final reference, invoking the device release callback release_card_device(), which calls snd_card_do_free(). snd_card_do_free() performs cleanup and calls complete(card->release_completion). This wakes up the unbind thread, which returns from snd_card_free() and __snd_card_release(). The devres core then immediately frees the memory block containing the snd_card structure. Meanwhile, the user thread continues execution in snd_card_do_free() and evaluates `if (!card->managed)`. It reads the `managed` boolean from the snd_card structure that was just freed by the unbind thread, triggering a KASAN use-after-free. Fix this by caching the value of card->managed in a local variable before calling complete(). This ensures that the card pointer is not dereferenced after the unbind thread has been woken up and potentially freed the card. BUG: KASAN: use-after-free in snd_card_do_free sound/core/init.c:604 [inline] BUG: KASAN: use-after-free in release_card_device+0x1ab/0x1b0 sound/core/init.c:153 Read of size 1 at addr ffff8881912ec909 by task syz-executor130/5857 Call Trace: <TASK> dump_stack_lvl+0xe8/0x150 lib/dump_stack.c:120 print_address_description+0x55/0x1e0 mm/kasan/report.c:378 print_report+0x58/0x70 mm/kasan/report.c:482 kasan_report+0x117/0x150 mm/kasan/report.c:595 snd_card_do_free sound/core/init.c:604 [inline] release_card_device+0x1ab/0x1b0 sound/core/init.c:153 device_release+0xc4/0x1f0 drivers/base/core.c:-1 kobject_cleanup lib/kobject.c:689 [inline] kobject_release lib/kobject.c:720 [inline] kref_put include/linux/kref.h:65 [inline] kobject_put+0x222/0x550 lib/kobject.c:737 snd_card_file_remove+0x331/0x390 sound/core/init.c:1125 snd_pcm_release+0x12c/0x160 sound/core/pcm_native.c:2986 __fput+0x418/0xa50 fs/file_table.c:512 fput_close_sync+0x11f/0x240 fs/file_table.c:617 __do_sys_close fs/open.c:1511 [inline] __se_sys_close fs/open.c:1496 [inline] __x64_sys_close+0x7e/0x110 fs/open.c:1496 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x174/0x580 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK>	N/A	More Details
CVE-2026-90200	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: fix integer overflow in MFT cluster validation In ntfs_init_from_boot(), the boot sector's MFT cluster numbers are validated against the volume size with: if (mlcn * sct_per_clst >= sectors || mlcn2 * sct_per_clst >= sectors) goto out; mlcn and mlcn2 are u64 fields read directly from the boot sector. sct_per_clst is bounded above by 4096 (true_sectors_per_clst() plus the is_power_of_2() check below it), but the multiplication is done in u64 and wraps when mlcn (or mlcn2) is large enough -- e.g. mlcn near 2^62 with sct_per_clst == 4 wraps to 0, which compares below any non-zero 'sectors', so the check is bypassed and the malformed record is accepted. The accepted mlcn is then used unchanged in sbi->mft.lbo = mlcn << cluster_bits; In practice the resulting reads fail at the block layer (sb_bread() returns NULL via grow_buffers()'s check_mul_overflow() guard), so today this manifests as mount failing in odd places rather	N/A	More Details

	than as something more dangerous, but the validation step is still wrong and there is no reason for callers to rely on the block layer to catch a value that should never have been accepted in the first place. Use check_mul_overflow() to compute the two sector positions and fail the mount if either multiplication wraps; this preserves the existing semantics (mlcn * sct_per_clst >= sectors) instead of switching to division (mlcn >= sectors / sct_per_clst), which would tighten the check at edge cases where 'sectors' is not a multiple of sct_per_clst. The check_*_overflow() style is the one ntfs3 already uses for similar on-disk arithmetic in fs/ntfs3/run.c.		
CVE-2026-90112	In the Linux kernel, the following vulnerability has been resolved: net: qlcnic: validate unified ROM sections before loading The unified ROM parser reads directory, product, and data-descriptor fields from the firmware file. Existing validation forms table and data ends with unchecked additions and multiplications. Malformed values can wrap before they are compared with the firmware size. The parser also dereferences typed pointers at firmware-controlled offsets. Valid descriptor extents alone are insufficient for the consumers. The loader reads a fixed-size bootloader regardless of its declared size, the version parser assumes a 17-byte tail, and a partial final firmware word is read as a full u64. A truncated image can therefore make the driver read beyond the firmware allocation during validation or loading. Replace the pointer-returning parser with bounded range helpers. Validate table entry sizes, descriptor indices, section ranges, the fixed bootloader load length, and the version tail before exposing any section. Read all file fields with unaligned little-endian accessors and assemble a partial final word from only the bytes that remain. Apply the same range checks to the legacy image before reading its fixed fields.	N/A	More Details
CVE-2026-90109	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix 32-bit backlog wrap in gred, bfifo and plug enqueue gred_enqueue(), bfifo_enqueue() and plug_enqueue() admit a packet when the current backlog plus the packet length fits within the queue limit: sch->qstats.backlog + qdisc_pkt_len(skb) <= sch->limit (gred default VQ) gred_backlog+qdisc_pkt_len(skb) <= q->limit (gred configured VQ) sch->qstats.backlog + qdisc_pkt_len(skb) <= sch->limit (bfifo) sch->qstats.backlog + skb->len <= q->limit (plug) sch->qstats.backlog and q->backlog are u32, and qdisc_pkt_len()/skb->len are unsigned int, so all sums are computed in 32 bits and wrap at 2^32. Once the true backlog exceeds 4 GiB the wrapped sum becomes small and admission keeps succeeding, so the queue grows without bound and the kernel can be driven to OOM. Promote the sums to u64 so admission stops once the true backlog exceeds the limit. The limit is u32, so the bounded queue stays below 2^32 and the stored u32 backlog never wraps. The bug can only be reproduced as root (albeit with ridiculous setup): attach a gred (or bfifo/plug) qdisc with a limit near 4 GiB, leaving the default VQ unconfigured (for gred), and drive >4 GiB of queued traffic (e.g. via a size table / stab to inflate qdisc_pkt_len, or sustained high-rate traffic). The u32 backlog+len sum wraps at 2^32, admission keeps succeeding, and the queue grows unboundedly to OOM.	N/A	More Details
CVE-2026-90108	In the Linux kernel, the following vulnerability has been resolved: net/smc: free stashed qentry before overwrite in REQ_ADD_LINK to ADD_LINK transition When smc_llc_event_handler() transitions the local LLC flow from SMC_LLC_FLOW_REQ_ADD_LINK to SMC_LLC_FLOW_ADD_LINK on arrival of an ADD_LINK request, it calls smc_llc_flow_qentry_set() unconditionally: if (lgr->llc_flow_lcl.type == SMC_LLC_FLOW_REQ_ADD_LINK) { lgr->llc_flow_lcl.type = SMC_LLC_FLOW_ADD_LINK; smc_llc_flow_qentry_set(&lgr->llc_flow_lcl, qentry); ... } A CONFIRM_LINK or ADD_LINK_CONT arriving while flow->type is SMC_LLC_FLOW_REQ_ADD_LINK is stashed into flow->qentry via the SMC_LLC_CONFIRM_LINK / SMC_LLC_ADD_LINK_CONT handler (which stores into flow->qentry for any non-NONE flow type). When the subsequent ADD_LINK arrives, the REQ_ADD_LINK branch overwrites flow->qentry with the new pointer without first freeing the stashed allocation, leaking one kmalloc object. The stashed entry has no consumer: smc_llc_wait() is only called from llc_add_link_work, which is not yet scheduled while the flow type remains REQ_ADD_LINK. No waiter is sleeping on llc_msg_waiter at this point. It is safe to unconditionally free any stashed qentry before the overwrite. Call smc_llc_flow_qentry_del() before smc_llc_flow_qentry_set() in the REQ_ADD_LINK branch. smc_llc_flow_qentry_del() already checks flow->qentry before freeing, so the normal path where no entry is stashed is a no-op.	N/A	More Details
CVE-2026-61599	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, the djust live transport resolves the LiveView to mount from a client-supplied dotted path by calling `__import__(module_path, ...)`. The module is imported — running its top-level code (import side effects) — before the framework checks that the resolved object is a `LiveView` subclass and before any per-view authentication. The `LIVEVIEW_ALLOWED_MODULES` allowlist that should contain this is fail-open (`if allowed_modules:` — skipped when the setting is unset, the framework default) and uses loose `startswith` matching. An unauthenticated WebSocket client (the WS handshake does not require auth; per-view auth runs only after import + instantiate) can therefore send a `mount` / `live_redirect_mount` / `url_change` frame (or an SSE mount) with `view = "<any.importable.module>.AnyName"` and cause the server to import — and execute the top-level code of — any importable Python module by name. Version 1.0.7 fixes the issue with a fail-closed resolution gate (`djust._view_resolution.is_view_import_allowed`): a client view path resolves only if (a) its module is already loaded (`sys.modules` — so resolving runs no new code; URL-routed views loaded by URLconf at startup keep working with zero config) or (b) it matches `LIVEVIEW_ALLOWED_MODULES` on a module-segment boundary (explicit opt-in for lazily-imported views). The gate runs before `__import__` at all three sinks (+ defense-in-depth inside `_instantiate_view`). As a workaround, set `LIVEVIEW_ALLOWED_MODULES` to the narrow list of modules that contain your mountable LiveView classes. (Note: pre-patch the allowlist is `startswith`-matched and the import still precedes the subclass check, so this is mitigation, not a complete fix.)	N/A	More Details
CVE-2026-50607	A vulnerability has been identified in the Acer System Monitoring component included with NitroSense and PredatorSense. A WebSocket service was configured to listen on all network interfaces, which may expose the service to unintended network access.	N/A	More Details
CVE-2026-50606	A vulnerability has been identified in the Acer System Monitoring component included with NitroSense and PredatorSense. The vulnerability is caused by the use of a hard-coded AES encryption key within the software. Under certain circumstances, a local attacker may be able to use the embedded key to access protected information or perform unauthorized actions.	N/A	More Details
CVE-2026-50605	A vulnerability has been identified in the Acer Agent Service component included with NitroSense and PredatorSense. Insufficient access controls within a privileged service may allow an authenticated local user to perform unauthorized registry operations. In certain situations, this could lead to privilege escalation or compromise of the affected system.	N/A	More Details
CVE-2026-61793	Nuxt OG Image generates OG Images with Vue templates in Nuxt. From 6.0.2 until 6.7.0, nuxt-og-image exposes the unauthenticated /_og/d/** route when the documented defaults security.strict = false and security.secret = "" are used, and base64url-decodes the fonts parameter through decodeOgImageParams. Attacker-controlled fonts[].path values flow through loadDefinedFonts into the font-assets/node.js binding, which performs a server-side fetch without validating the URL scheme, origin, resolved address, or redirects. This permits blind requests to loopback, private, link-local, cloud metadata, and other internal HTTP services, while differences in the outer response status and timing can reveal service reachability. Slow targets	N/A	More Details

	can also occupy OG image render workers for the configured fetch and render timeouts. This issue is fixed in version 6.7.0.		
CVE-2026-71538	@cyclonedx/cyclonedx-npm creates CycloneDX Software Bill of Materials from npm projects. Prior to version 6.0.0, the Windows fallback path in src/npmRunner.ts, used when npm_execpath does not provide the npm CLI path, can construct a shell command containing an untrusted value from the --workspace option. When an attacker can influence that option and the fallback npm execution path is reached, shell metacharacters in the workspace value can execute arbitrary operating-system commands with the privileges of the user running the CLI, allowing data access, file modification, or service disruption. This issue is fixed in version 6.0.0.	N/A	More Details
CVE-2026-79752	CakePHP is a rapid development framework for PHP. Prior to 4.5.12, 4.6.5, 5.1.9, 5.2.14, and 5.3.7, FunctionsBuilder::cast, FunctionsBuilder::extract, FunctionsBuilder::datePart, and FunctionsBuilder::dateAdd in src/Database/FunctionsBuilder.php accept user-controlled dataType, part, or unit values and incorporate them into generated SQL as unescaped structural fragments. An application that passes untrusted input to these parameters can permit SQL injection with confidentiality, integrity, and availability impact according to the database connection's privileges. This issue is fixed in versions 4.5.12, 4.6.5, 5.1.9, 5.2.14, and 5.3.7.	N/A	More Details
CVE-2026-50604	A vulnerability has been identified in the Acer Agent Service component included with NitroSense and PredatorSense. The socket handshake process does not properly require authentication before granting access to the service. Under certain circumstances, an unauthorized connection may be established, potentially allowing access to functionality that should be restricted.	N/A	More Details
CVE-2026-88952	Improper Authentication vulnerability in team-alembic AshAuthentication allows an attacker to be signed in as another user by linking an OAuth2 identity to an account that is not theirs. AshAuthentication.Strategy.OAuth2.UserResolver.resolve/3 matches an existing account using the register action's upsert_identity keys, then gates linking the incoming provider identity to it on email_trusted?/2, which reads only the provider's email_verified boolean and never compares the provider's email value with the matched account's email. That gate assumes the account was matched by its email field, so under any other upsert_identity it is vacuous and an attacker presenting their own verified email is attached to, and issued a session for, an account matched on some other attribute. The same unguarded gate applies in OAuth2.SignInPreparation on the registration_enabled? false path, where the account is matched by the sign-in action's read filter instead. The upsert also rewrites the matched account's email to the attacker's address, so later account recovery reaches the attacker rather than the owner. This issue affects ash_authentication: from 4.14.0 before 4.15.0 and from 5.0.0-rc.10 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-50603	A vulnerability has been identified in the Acer Agent Service component included with NitroSense and PredatorSense. The vulnerability is caused by the use of a hard-coded AES encryption key within the software. Under certain circumstances, a local attacker may be able to use the embedded key to access protected information or perform unauthorized actions.	N/A	More Details
CVE-2026-85789	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-69197	Umbraco is an ASP.NET CMS. Prior to 13.15.1, 17.5.3, and 18.0.2, the Content Delivery API applies member and Public Access checks to the directly requested node but not to referenced nodes serialized through Content Picker or Multi-Node Tree Picker properties, including pickers nested in Block List, Block Grid, or Rich Text Editor blocks. When DeliveryApi:PublicAccess is enabled, an anonymous caller can retrieve a protected node's name, route, and id through an unprotected referencing node and use ?expand to retrieve full property values. When the Delivery API is instead gated by the organization-wide API key, a key holder can still bypass per-node Public Access through the same expansion path. The same RequestContextOutputExpansionStrategyV2 and ElementOnlyOutputExpansionStrategy path also bypasses allowed or disallowed content-type alias restrictions for referenced content. Direct requests for the protected node still return 401, and no integrity or availability impact is established. This issue is fixed in versions 13.15.1, 17.5.3, and 18.0.2.	N/A	More Details
CVE-2026-90107	In the Linux kernel, the following vulnerability has been resolved: net/smc: free pending qentry in smc_llc_flow_stop() before memset smc_llc_flow_stop() resets a flow struct with a blind memset: spin_lock_bh(&lgr->llc_flow_lock); memset(flow, 0, sizeof(*flow)); flow->type = SMC_LLC_FLOW_NONE; spin_unlock_bh(&lgr->llc_flow_lock); If flow->qentry is non-NULL at this point the pointer is overwritten without the allocation being freed, leaking one kmalloc object. A late-arriving duplicate CONFIRM_LINK or ADD_LINK_CONT message can set flow->qentry after the legitimate message has been consumed by the waiter via smc_llc_flow_qentry_clr() (which NULLs the pointer but leaves flow->type non-zero) but before the flow completes and smc_llc_flow_stop() runs. In that window the duplicate is stashed into flow->qentry, and then lost when smc_llc_flow_stop() zeros the struct. Call smc_llc_flow_qentry_del() inside the lock before the memset. smc_llc_flow_qentry_del() already checks flow->qentry before freeing, so the normal case where no entry is pending is a no-op.	N/A	More Details
CVE-2026-61597	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, many djust built-in component template tags (`djust.components.templatetags.*`) render a developer/user-supplied URL into an `href` / `action` attribute, HTML-escaping it with `conditional_escape` but never validating the URL scheme. HTML escaping prevents attribute breakout but does not neutralize a `javascript:` URI (which needs no escapable characters), so a URL value of `javascript:alert(document.cookie)` lands verbatim in `<a href="javascript:alert(document.cookie)">` and executes in the victim's session on click. Version 1.0.7 contains a fix. As a workaround, do not pass user-controllable URLs to the affected built-in component tags; pre-validate URL schemes in application code before binding them to component arguments.	N/A	More Details
CVE-2026-81872	OpenTelemetry-Go is the Go implementation of OpenTelemetry. Prior to version 0.21.0, the go.opentelemetry.io/otel/sdk/log BatchingProcessor can enter a tight CPU loop when attacker-driven log emission fills its asynchronous export buffer while the exporter is backpressured. NewBatchingProcessor wraps the exporter with newBufferExporter(exporter, 1), and the poll loop calls queue.TryDequeue and bufferExporter.EnqueueExport before immediately signaling pollTrigger whenever the queue remains at or above batchSize. Because a failed nonblocking EnqueueExport leaves the queue length unchanged, the processor repeatedly retries without waiting for its ticker, exhausting CPU and degrading or denying service in the embedding process. This issue is fixed in version 0.21.0.	N/A	More Details
CVE-2026-	OpenTelemetry-Go is the Go implementation of OpenTelemetry. Prior to version 0.21.0, the exporters/otlp/otlplog/otlploggrpc package loads OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE, OTEL_EXPORTER_OTLP_CERTIFICATE, and related client certificate environment variables through loadEnvTLS into cfg.tlsCfg, but newGRPCDialOptions does not apply cfg.tlsCfg when creating gRPC transport credentials. The environment-only TLS path instead uses credentials.NewTLS with system roots and no	N/A	More

81871	configured client certificate, bypassing intended private CA pinning and mutual TLS unless the application also supplies WithTLSCredentials. A network attacker able to intercept or spoof the collector connection with a system-trusted certificate can read or alter log telemetry. This issue is fixed in version 0.21.0.		Details
CVE-2026-81869	OpenTelemetry-Go is the Go implementation of OpenTelemetry. From version 1.10.0 until 1.33.0, the sdk/trace/span.go attribute truncation path can fail to enforce AttributeValueLengthLimit for string and string-slice attributes containing the valid Unicode replacement character U+FFFD. safeTruncateValidUTF8 treats the valid replacement rune as invalid UTF-8 and returns the original input, while strings.ToValidUTF8 leaves that valid rune unchanged, so a second safeTruncate attempt can also return the oversized value. An attacker who controls span attribute content can retain values longer than the configured limit, increasing per-span memory use and weakening denial-of-service protection in the instrumented process. This issue is fixed in version 1.33.0.	N/A	More Details
CVE-2026-91039	Authentication Bypass by Spoofing vulnerability in team-alembic ash_authentication allows an attacker who operates one identity-provider connection of a dynamic_oidc strategy to be signed in as a local user established through a different connection. The strategy is meant to keep each connection in its own identity namespace by writing every UserIdentity row's strategy field as "<name>/<connection_id>", but that namespacing never takes effect. __connection_id__ is populated only on the ephemeral runtime struct built per request in dynamic_oidc/plug.ex, and DynamicOidc.IdentityChange.change/3 re-fetches the strategy from the compile-time DSL through Info.strategy_for_action, yielding the persisted struct whose __connection_id__ is its defstruct default of nil. OAuth2.identity_strategy_name/1 therefore falls back to the bare strategy name for both the identity write and the reads in oauth2/user_resolver.ex and oauth2/sign_in_preparation.ex. Since the identity resource's unique key is (uid, strategy), one row exists per sub across every connection, and the identity-match branch runs before any email check. Neither strategy handles iss, so nothing else distinguishes the issuers: OpenID Connect Core section 5.7 makes sub unique only within an issuer, so two connections numbering subjects independently share one subject space. This issue affects ash_authentication: from 5.0.0-rc.10 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-54575	mport is the MidnightBSD Package Manager. Prior to 2.7.8, privileged package fetch and cache-cleaning operations used race-prone path handling across libmport/fetch.c, libmport/clean.c, libmport/util.c, libmport/bundle_read_install_pkg.c, libmport/delete_primative.c, and libexec/mport.create/mport.create.c. A local attacker with write access to a participating package cache or staging path could race path checks and replacement operations to redirect package downloads, cleanup, or install-related side effects outside the intended cache. The affected lifecycle helper paths also used shell-form invocation, increasing command-line interpretation risk during privileged helper execution. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54576	mport is the MidnightBSD Package Manager. Prior to 2.7.8, do_actual_install() in libmport/bundle_read_install_pkg.c used path-based lstat(), chown(), stat(), and chmod() operations while installing package files. A local attacker with write access to a target directory could replace a checked file with a symlink before privileged ownership or mode changes were applied, redirecting those changes to an attacker-selected path and compromising filesystem integrity or permissions. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54577	mport is the MidnightBSD Package Manager. Prior to 2.7.8, the audit command in mport/mport.c computed option-adjusted local_argv and local_argc values but passed the original argument entry to audit_package(). When an operator or automation used an option such as -r before a package name, stale optind state and the unadjusted argument could cause mport to audit the option token instead of the requested package, producing a false-negative or useless result that could leave a vulnerable package unidentified. The corrected parsing resets optind and optreset before using the adjusted local arguments. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54578	mport is the MidnightBSD Package Manager. Prior to 2.7.8, mport_verify_package() in libmport/verify.c could continue after MD5File() or SHA256_File() failed and compare an expected checksum with stale data in the hash buffer rather than a newly computed digest. An attacker able to influence an installed file or the conditions that make hashing fail could receive a misleading integrity result or hide a checksum failure. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54579	mport is the MidnightBSD Package Manager. Prior to 2.7.8, ping() in libmport/ping.c accepted ICMP replies without validating icmp_id or icmp_seq and parsed the reply using a fixed IP-header offset instead of ip_hl. A network attacker able to inject or spoof visible ICMP replies could influence mirror latency selection, while a malformed packet carrying IP options could shift the ICMP header and trigger an out-of-bounds read. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-92962	vm2 is a sandbox for running untrusted JavaScript. In vm2 versions up to and including 3.11.3, the defaultSandboxPrepareStackTrace function in lib/setup-sandbox.js builds its output array using prototype-walking index assignment (lines[lines.length] = value) rather than a prototype-bypassing define-property primitive. Because this bridge-internal array is allocated in the sandbox realm, code inside the sandbox can install an accessor on Array.prototype for the relevant index; the accessor is then invoked whenever the sandbox reads error.stack (or otherwise triggers Error.prepareStackTrace), allowing sandbox code to observe and intercept each stack-trace line written by the bridge. The same pattern is used in the error-handling (catch) branch. The values written are formatted strings only, so the practical impact is limited to an information side channel and a violation of vm2's bridge-container defense invariant rather than a sandbox escape; the vendor rates the issue Low. The issue is fixed in vm2 3.11.4, which installs each entry as an own data property via Reflect.defineProperty.	N/A	More Details
CVE-2026-15688	Incorrect Implementation of Authentication Algorithm Vulnerability in Mitsubishi Electric GX Works3 and Motion Control Setting allows a local attacker to successfully authenticate even with an invalid block password by executing the affected product and modifying part of the executable module in memory, and thereby may be able to view, tamper with, destroy, or delete control programs.	N/A	More Details
CVE-2026-50608	A vulnerability has been identified in the Acer System Monitoring component included with NitroSense and PredatorSense. The WebSocket handshake process does not properly require authentication before allowing connections to the service. Under certain circumstances, unauthorized access to service functionality may be possible.	N/A	More Details
CVE-2026-50609	A vulnerability has been identified in the Acer System Monitoring component included with NitroSense and PredatorSense. Insufficient access controls within a privileged Named Pipe service may allow an authenticated local user to perform unauthorized registry operations. In certain situations, this could lead to privilege escalation or compromise of the affected system.	N/A	More Details

CVE-2026-92932	In the MISP sachertortephp library, the Xml::build() static method in lib/Cake/Utility/Xml.php contains a logic error in the conditional that gates network-based XML fetching. The original condition was written as: $options['readFile'] && strpos($input, 'http://') === 0 || strpos($input, 'https://') === 0. Because PHP's && operator has higher precedence than ||, the expression is evaluated as ($options['readFile'] && strpos($input, 'http://') === 0) || strpos($input, 'https://') === 0. As a result, when a caller explicitly sets the readFile option to false to disable file and URL reading, an input string beginning with https:// still satisfies the condition and triggers a network fetch via HttpSocket (configured to follow up to 10 redirects). The http:// branch is correctly gated by the readFile flag, but the https:// branch is not. An attacker who can influence the $input parameter passed to Xml::build() can therefore force the application to issue an outbound HTTPS request to an attacker-controlled or internal URL, even though the caller intended to suppress all remote reads. The fetched response is parsed as XML and may be returned to the caller, enabling information disclosure from internal services or external targets. This constitutes a Server-Side Request Forgery (SSRF) weakness with an information-disclosure impact. The vulnerability requires that the code path in Xml::build() be reachable with attacker-controlled input and that the readFile option be set to false (or the caller expects it to be false).	N/A	More Details
CVE-2026-53681	Rejected reason: Red Hat Product Security has come to the conclusion that this CVE is not needed.	N/A	More Details
CVE-2026-78223	Improper Verification of Cryptographic Signature vulnerability in team-alembic AshAuthentication allows a caller of the token revocation action to neutralise a revocation or write arbitrary rows into the token resource. AshAuthentication.TokenResource.RevokeTokenChange.change/3 reads the :token argument and decodes it with AshAuthentication.Jwt.peek/1, which delegates to Joken.peek_claims/1 and performs no signature check, unlike Jwt.verify/4. The jti, exp and sub claims it returns are written straight onto the revocation record, guarded only by byte_size(token) > 0. Because expires_at derives from the attacker-chosen exp, a forged copy of a genuine token that keeps the real jti but backdates exp yields a revocation row that is already expired: expunge_expired removes it and the genuine token passes revoked? again. Arbitrary jti and sub values can be inserted the same way. This issue affects ash_authentication: from 0.2.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-53679	Rejected reason: Red Hat Product Security has come to the conclusion that this CVE is not needed.	N/A	More Details
CVE-2026-80218	Improper Authentication vulnerability in team-alembic AshAuthentication allows an attacker holding a sign-in token for one authenticated resource to be signed in as a user of a different resource. AshAuthentication.Strategy.Password.SignInWithTokenPreparation.extract_primary_keys_from_subject/2 parses the JWT sub claim (for example user?id=1) with URI.parse/1 and keeps only its query string, discarding the path segment that names the subject the token was issued for. Nothing else restores that binding: AshAuthentication.Jwt.verify/3 checks the signature, exp, nbf, jti and the library-version claims, the purpose check only requires sign_in, and the remaining comparison is over primary-key field names, which are identical across resources. The WebAuthn sign-in and remember-me preparations carry copies of the same helper and drop the path in the same way. The magic link sign-in path pins the subject name against the resource and is not affected. This issue affects ash_authentication: from 3.10.5 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-15639	An attacker can craft a malicious link that, if used by a legitimate user, may cause the user's browser to run JavaScript supplied by the attacker.	N/A	More Details
CVE-2026-11874	Rejected reason: Red Hat Product Security has come to the conclusion that this CVE is not needed.	N/A	More Details
CVE-2026-92611	In Eclipse Ankaios versions 0.6.0 to before 1.0.4, `LogRule::matches` in the agent control-interface authorizer stops at the first wildcard pattern in a single rule instead of evaluating later entries, which can cause deny `LogRule` entries to be skipped and allow unauthorized access to another workload's logs.	N/A	More Details
CVE-2026-81632	Use of HTTP Request With Sensitive Query String vulnerability in team-alembic AshAuthenticationPhoenix allows someone able to read access logs, proxy logs or browser history to recover a single-use sign-in token and authenticate as its owner. After a successful password sign-in, AshAuthentication.Phoenix.Components.Password.SignInForm builds the sign_in_with_token path with the freshly issued user.__metadata__.token as a query parameter and redirects the browser to it with a GET. The token therefore travels in the request line, where web servers, reverse proxies, request telemetry and the browser's own history record it, all of which outlive the request and are ordinarily less protected than session storage. The redirect destination is restricted to a local path, so this is not an open redirect; the exposure is the retention of a live credential. This issue affects ash_authentication_phoenix: from 1.7.0 before 2.17.4 and from 3.0.0-rc.0 before 3.0.0-rc.11; ash_authentication: from 3.10.5 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-81637	Insufficient Session Expiration vulnerability in team-alembic AshAuthentication allows an attacker who obtains a victim's OAuth2 state value to replay the callback and sign that victim into an attacker-controlled account. AshAuthentication.Strategy.OAuth2.Plug.callback/2 clears the stored session_params through a rebinding step inside its with chain, conn <- delete_session(conn, session_key). Elixir evaluates the else block in the scope enclosing the with, so every failure path (the provider returning ?error=access_denied, an invalid code, a token-exchange error, or a registration or sign-in failure) reaches store_authentication_result/2 holding the original connection and the session entry is never removed. The value the module's own comment describes as protection against a CSRF-related attack is therefore consumed only when authentication succeeds, and survives a cancelled or failed attempt until the next request phase or session expiry. This issue affects ash_authentication: from 0.6.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-82685	Authorization Bypass Through User-Controlled Key vulnerability in team-alembic AshAuthentication allows an authenticated attacker to overwrite and confirm another user's email address, and so take over that account. A confirmation token issued to one user is accepted on any other user's record. AshAuthentication.AddOn.Confirmation.ConfirmChange verifies the token's signature and its act claim, then applies the changes stored against that token to whichever record the changeset targets, never comparing the sub claim against changeset.data. An attacker who registers an account and changes their own email replays the resulting token against a victim's record id, writing in their own address with force_change_attributes/2 and stamping confirmed_at, after which an ordinary password reset yields the account. The library's own confirmation flow is	N/A	More Details

	unaffected, because AshAuthentication.AddOn.Confirmation.Actions.confirm/3 resolves sub to a user and targets that record. This issue affects ash_authentication: from 0.5.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.		
CVE-2026-82723	Insertion of Sensitive Information into Log File vulnerability in team-alembic AshAuthentication allows disclosure of user password digests to readers of the audit store. The audit_log add-on builds each entry's extra_data in AshAuthentication.AddOn.AuditLog.Auditor.build_extra_data/4, which takes :actor from the action callback context verbatim. Any audited action invoked with actor: set to a user record therefore deposits that record, including its hashed_password attribute, into the audit entry. The same module already collapses the audited identity to an opaque string via AshAuthentication.user_to_subject/1 and filters params against the strategy's configured allow-list, so the actor is the only stored value that reaches the audit store unfiltered. Marking the attribute sensitive?: true does not help, because that redacts inspect/1 output rather than JSON encoding or raw-term storage. There is no attacker-controlled trigger and no network disclosure path: entries accumulate from ordinary authenticated activity, and an attacker's own requests deposit only their own digest. Exploitation requires independent read access to the audit store, such as database credentials, an audit role, a backup, or a log shipper, at which point the digests support offline password attack against every active account. Whether the material persists depends on the data layer, since raw-term stores keep it verbatim while a SQL store raises Protocol.UndefinedError and drops the entry unless the user resource derives Jason.Encoder. This issue affects ash_authentication: from 4.12.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.2.	N/A	More Details
CVE-2026-82759	Use of a One-Way Hash with a Predictable Salt vulnerability in team-alembic AshAuthentication allows readers of the audit store to recover the client IP addresses that the audit log add-on's :hash privacy mode is meant to pseudonymise. AshAuthentication.AddOn.AuditLog.IpPrivacy.hash_ip/1 computes a single unkeyed :crypto.hash(:sha256, salt <> ip) and truncates the result to 16 hexadecimal characters. The salt is read from the :audit_log_ip_salt or :secret application config keys, and falls back to the constant "default-salt-change-in-production" published in the library source when neither is set, with nothing warning that the default is in use. The IPv4 space is only 2^32 values and SHA-256 is fast, so the whole hash table is precomputable and every stored value maps back to its source address. Truncating to 16 characters does not help, and even a configured salt leaves the hash cheap enough to enumerate once it leaks. This issue affects ash_authentication: from 4.12.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-82760	Inefficient Algorithmic Complexity vulnerability in team-alembic AshAuthentication allows an unauthenticated attacker to exhaust CPU and memory via an oversized base62 segment in a submitted API key. AshAuthentication.Base.decode62/1 in lib/ash_authentication/base.ex splits its argument into one binary per character and folds it with charval62/2, which recomputes Integer.pow(62, index) at every position instead of accumulating by Horner's method, so cost grows roughly cubically in the input length. bindecode62/1 in the same module is quadratic through Integer.undigits/2 and Integer.digits/2. Neither function caps byte_size/1, and AshAuthentication.Strategy.ApiKey.SignInPreparation passes the underscore-separated segments of the submitted key straight into both, before any key lookup and without prior authentication. The surrounding rescue clauses catch exceptions, not CPU or memory exhaustion. This issue affects ash_authentication: from 4.8.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-82761	Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in team-alembic AshAuthentication allows an attacker holding a leaked magic link to replay its single-use token and authenticate as the target subject. A magic link configured with single_use_token?, which is the default, is meant to be redeemable exactly once, but nothing serialises the token's validity check against its consumption, so concurrent redemptions of one token all succeed and each yields a full user token. Sign-in verifies the JWT with Jwt.verify/4 and revokes it only afterwards: AshAuthentication.Strategy.MagicLink.SignInPreparation revokes in a Query.after_action callback, and AshAuthentication.Strategy.MagicLink.SignInChange in an after_transaction hook that runs once the sign-in has already committed. AshAuthentication.TokenResource.Actions.revoke/3 writes the revocation as an upsert, so a concurrent duplicate revocation silently succeeds instead of conflicting and no request ever loses the race. This issue affects ash_authentication: from 3.9.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-85500	Authentication Bypass by Primary Weakness vulnerability in team-alembic AshAuthentication allows an unconfirmed user to obtain a session, defeating a mandatory email confirmation requirement. AshAuthentication.Strategy.Password.Actions.check_user/2 decides whether the attribute named by require_confirmed_with is set using a bare is_nil(Map.get(user, value)). When that attribute is not selected on the loaded record Map.get/2 returns %Ash.NotLoaded{}, and when a field policy denies it for the current actor it returns %Ash.ForbiddenField{}. Neither is nil, so the rejection branch is skipped and sign-i require_confirmed_with is enforced in two places, and neither holds in every configuration. sign_in_with_token and register are checked only inside AshAuthentication.Strategy.Password.Actions, not on the action itself, so any caller that invokes the action directly skips the check. An API layer such as AshGraphql or AshJsonApi invokes the action directly, so this applies to the default configuration. Where a check does run it compares the confirmation attribute against nil. That attribute holds %Ash.NotLoaded{} or %Ash.ForbiddenField{} when it sets select_by_default?: false, when an API layer narrows the read's select, or when a field policy hides it from the sign-in actor. Neither struct is nil, so those configurations read every user as confirmed. This issue affects ash_authentication: from 4.3.8 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-86522	Improper Output Neutralization for Logs vulnerability in team-alembic AshAuthentication allows an unauthenticated attacker to forge application log entries by submitting a password reset identity containing newlines or control characters. AshAuthentication.Strategy.Password.RequestPasswordReset.run/3 interpolates the identity argument, the email or username taken straight from the reset request, into its Logger.warning/1 heredocs without escaping, truncating or type-restricting it. The resource logged beside it is passed through inspect/1, which would have neutralized the value. A newline in the identity therefore ends the log record, and everything after it is written as a line of its own, so an attacker chooses the severity tag and the content of entries that appear to have come from the application. This issue affects ash_authentication: from 4.2.0 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14.	N/A	More Details
CVE-2026-86533	Insufficient Session Expiration vulnerability in team-alembic AshAuthentication and AshAuthentication Phoenix allows a revoked session to remain fully authenticated. A resource configured with session_identifier :jti and require_token_presence_for_authentication? disabled stores its session value as <jti>:<subject>. The jti is there so that signing out can revoke that one session. Neither reader consults it: AshAuthentication.Plug.Helpers.authenticate_resource_from_session/4 and AshAuthentication.Phoenix.LiveSession.on_mount/4 both split the value with split_identifier/2, discard the jti and pass the bare subject to AshAuthentication.subject_to_user/3, which reloads the record. The token-presence branch of each function does check its token, calling AshAuthentication.TokenResource.Actions.get_token/3 with the jti and the purpose user. Because the revocation record is never read, neither its revoked state nor its expiry constrains the session, so a session captured before sign-out keeps working. This	N/A	More Details

	issue affects ash_authentication: from 4.9.1 before 4.15.0 and from 5.0.0-rc.0 before 5.0.0-rc.14; ash_authentication_phoenix: from 2.10.0 before 2.17.4 and from 3.0.0-rc.0 before 3.0.0-rc.11.		
CVE-2026-89418	google-protobuf contains an unbounded recursion when parsing unknown protobuf group fields. An attacker can send a small crafted payload of deeply nested START_GROUP wire bytes to any Node.js service that calls the generated deserializeBinary() API, causing a RangeError: Maximum call stack size exceeded and crashing the process. No authentication or prior knowledge of the schema is required.	N/A	More Details
CVE-2026-78425	Authorised users of outside applications behind the same corporate identity provider (IdP), for example, a wiki, a ticketing system, an expenses tool, or anything they legitimately hold an account on can log into their system via SAML SSO. The IdP issues an assertion to them. If that assertion is presented to NeuVector, NeuVector accepts it because the only thing distinguishing "an assertion for NeuVector" from "an assertion for the wiki" is the element, and the `NotInAudience` warning that reports the mismatch is never read.	N/A	More Details
CVE-2026-50610	A vulnerability has been identified in the Acer System Monitoring component included with NitroSense and PredatorSense due to insufficient access controls in a privileged service. An authenticated local user may be able to access the service and perform unauthorized registry modifications, potentially resulting in local privilege escalation.	N/A	More Details
CVE-2026-54580	mport is the MidnightBSD Package Manager. Prior to 2.7.8, libmport/util.c did not make every truncated, corrupt, or failed zstd stream fatal in mport_decompress_zstd(), and libmport/fetch.c did not consistently propagate those failures to index-fetch callers. A malicious or faulty mirror could supply compressed package index data that caused ZSTD_decompressStream() or an output write to fail while leaving partial index output available for later use, resulting in package-index integrity loss or denial of service. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54581	mport is the MidnightBSD Package Manager. Prior to 2.7.8, the mport_fetch_bootstrap_index() function in libmport/fetch.c could return success when bootstrap index hash verification encountered a missing or invalid hash because the failure path did not preserve a fatal result. A network attacker or compromised mirror able to alter bootstrap index content or its transport path could therefore cause mport to proceed with an unverified or tampered bootstrap package index. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54582	mport is the MidnightBSD Package Manager. Prior to 2.7.8, package installation lacked a preflight check for incoming non-directory assets that already existed on disk. The affected logic across libmport/check_preconditions.c, libmport/install_primative.c, and libmport/mport_private.h did not apply MPORT_PRECHECK_FILE_CONFLICTS, so a crafted or conflicting package could overwrite a file owned by another package or unmanaged by mport. The check is bypassed only when the operator explicitly enables mport->force. Privileged installation without that override could compromise local filesystem integrity and package database consistency. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-90077	In the Linux kernel, the following vulnerability has been resolved: net: fix a resource leak in copy_net_ns() error handling path Currently, preinit_net() does two things: (1) call ns_common_init() which might fail (2) initialize resources which does not fail However, preinit_net() is returning early when (1) fails, and copy_net_ns() is jumping to the dec_ucounts: label. As a result, resources allocated by net_alloc() are leaking. We need to call key_remove_domain() and net_passive_dec() in order to release resources allocated by net_alloc(). We cannot simply jump to the put_userns: label when preinit_net() failed, for (2) is not yet done. But we can reorder (1) and (2), for there is no dependency between (1) and (2). Therefore, this patch decouples (1) from preinit_net() and changes preinit_net() back to a void function, and calls ns_common_init() after preinit_net() succeeded. Then, we can jump to immediately after ns_common_free() of the put_userns: label.	N/A	More Details
CVE-2026-90079	In the Linux kernel, the following vulnerability has been resolved: octeontx2-af: fix cn20k mailbox lifetime on repeated rvu_mbox_init() rvu_mbox_init() is called separately for AF-PF mailboxes during probe and for AF-VF mailboxes when SR-IOV is enabled. Each call used to allocate a new ng_rvu object, leaking the first allocation when the pointer was overwritten on the second call. Sharing one ng_rvu across both paths exposed several teardown bugs: the error path freed all cn20k mailbox DMA and kfree()d ng_rvu even when only the failing init type should be unwound, leaving live AF-PF mailbox memory in use after an AF-VF init failure. mutex_init() was also re-run on the AF-VF path while AF-PF mailbox handlers could still hold rvu->mbox_lock. Probe and SR-IOV failure paths did not release cn20k mailbox DMA either, since cleanup only happened in rvu_remove(). Allocate ng_rvu once with devm_kzalloc(), initialize mbox_lock in the same block, unwind only the mailbox memory for the failing init type, and free cn20k mailbox DMA from the probe and pci_enable_sriov() error paths.	N/A	More Details
CVE-2026-90080	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: fix NULL deref of af_xdp_zc_qidx on rep setup af_xdp_zc_qidx tracks receive queues using AF_XDP zero-copy and is allocated during PF/VF probe. Representors and other non-AF_XDP paths leave the pointer NULL, but several call sites used test_bit() on it unconditionally. Switching to devlink eswitch mode creates representors and runs otx2_init_hw_resources(), which reaches otx2_pool_aq_init() and oopses when dereferencing the NULL bitmap. Add NULL checks before every af_xdp_zc_qidx test_bit() use in the RSS, ethtool, XSK, and pool init paths.	N/A	More Details
CVE-2026-90081	In the Linux kernel, the following vulnerability has been resolved: net/rds: use wq_has_sleeper() in rds_cong_map_updated() rds_cong_map_updated() runs after a peer's congestion map has been rewritten (by rds_tcp_cong_recv() and rds_ib_cong_recv(), or the clear-all in the loopback and IB send-completion paths). It bumps rds_cong_generation and then checks waitqueue_active() on map->m_waitq and on rds_poll_waitq to decide whether anyone needs waking. atomic_inc() carries no ordering and waitqueue_active() is a plain load, so nothing orders the map and generation stores before the wait queue reads. The waiters do the mirror image: rds_cong_wait() adds itself to m_waitq and then tests the port bit, and rds_poll() registers on rds_poll_waitq and then reads the generation. That is the store-buffering pattern described above waitqueue_active() in include/linux/wait.h - the updater can observe an empty wait queue while the waiter still observes the port as congested, and no wake-up is issued. rds_cong_wait() is an interruptible sleep with no timeout, so a sender blocked on a congested port stays blocked until the next congestion update from that peer arrives or a signal is delivered. A poll() waiter misses the map-updated notification the same way. Use wq_has_sleeper(), which is waitqueue_active() preceded by the required full barrier, as rds_tcp_state_change() already does for the same pattern.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: net: mana: Cap MSI-X vectors to the device MSI-X table size mana_gd_query_max_resources() sizes gc->num_msix_usable from resp.max_msix and the CPU count, but never from the device MSI-X table. On a 1792 vCPU M-series VM that yields 1793 while the table has 1024 entries, and mana_gd_setup_remaining_irqs() then walks indices 1..1792, running off the end of the region mapped by msix_map_region():		

CVE-2026-90082	BUG: unable to handle page fault for address: ff8e347f8b99800c RIP: 0010:msix_prepare_msi_desc+0x7a/0x90 RAX: 0000000000004000 RBX: ff4330cb164ea780 RCX: ff8e347f8b998000 Call Trace: <TASK> __msi_domain_alloc_irqs+0x13a/0x440 msi_domain_alloc_irq_at+0x149/0x1b0 mana_gd_setup+0x351/0x890 mana_gd_probe+0x274/0x390 </TASK> RAX is index 1024 * PCI_MSIX_ENTRY_SIZE, one entry past the table. msi_insert_desc() does range check the index, but only against the MSI domain hwsize, which matches the table only for devices on an MSI parent domain. With a global PCI/MSI domain hwsize is MSI_XA_DOMAIN_SIZE, so nothing bounds the request. Cap num_msix_usable with pci_msix_vec_count().	N/A	More Details
CVE-2026-90083	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_ife: Only operate on Ethernet frames act_ife encapsulates/decapsulates the original Ethernet header and uses skb->dev->hard_header_len as the length of that header. That is only correct for Ethernet devices: on a device where hard_header_len does not match the L2 header that was actually pulled (PPP reports PPP_HDRLEN while nothing is stripped on ingress), the ingress skb_push()/skb_pull() use the wrong length and can hit skb_under_panic when headroom is tight. IFE is Ethernet-only by design - it builds an outer ethhdr, rewrites h_source/h_dest/h_proto, and calls eth_type_trans() on decode - so instead of trying to make the offsets work for arbitrary link types, simply drop packets that do not carry an Ethernet header. Checking skb->dev->type alone is not enough. We have to cater for a corner case where mirred can redirect an skb from a non-Ethernet device to an Ethernet one, and skb->dev then says nothing about the framing the skb actually has: an skb redirected from ppp0 reaches the target's ingress hook with mac_len 0 and no Ethernet header at all. So at ingress also require mac_len to be ETH_HLEN. On egress mac_len is not maintained, so the device type is all we have; a bogus redirect there yields a malformed frame rather than an out-of-bounds push, and it would be malformed with or without IFE. That corner case is not theoretical - redirecting from ppp0 into a veth that has an ife encode action on its ingress hook panics without this patch: skbuff: skb_under_panic: len:98 put:14 head:ffff88800e410000 data:ffff88800e40fff5 tail:0x57 end:0x640 dev:veth3 kernel BUG at net/core/skbuff.c:214! Call Trace: skb_push (net/core/skbuff.c:224 net/core/skbuff.c:2657) tcf_ife_act (net/sched/act_ife.c:829 net/sched/act_ife.c:874) tc_run (net/core/dev.c:4463) netif_receive_skb (net/core/dev.c:6463 net/core/dev.c:6522) tcf_mirred_to_dev (net/sched/act_mirred.c:248 net/sched/act_mirred.c:328) tcf_mirred_act (net/sched/act_mirred.c:489) tc_run (net/core/dev.c:4463) process_backlog (net/core/dev.c:6728) With Ethernet framing guaranteed, use ETH_HLEN instead of hard_header_len.	N/A	More Details
CVE-2026-90084	In the Linux kernel, the following vulnerability has been resolved: octeontx2-vf: fix workqueue and netdev race in probe/remove Initialize the VF workqueue before register_netdev() so ndo_set_rx_mode does not queue work on a NULL workqueue. Unregister the netdev before destroying the workqueue, and add proper probe error cleanup.	N/A	More Details
CVE-2026-90085	In the Linux kernel, the following vulnerability has been resolved: octeontx2-af: fix NULL deref in NIX TM tree debugfs read path rvu_dbg_nix_tm_tree_display() dereferences pfvf->sq_ctx without checking whether the SQ context has been allocated. Reading /sys/kernel/debug/octeontx2/nix/tm_tree for a NIX LF whose transmit queues are not set up triggers a kernel oops. Guard the read path the same way rvu_dbg_nix_tm_tree_write() already does and return -EINVAL with a seq_file message when sq_ctx is NULL.	N/A	More Details
CVE-2026-90086	In the Linux kernel, the following vulnerability has been resolved: xsk: honor XDP_TX_METADATA in zero-copy path The zero-copy path reads TX metadata whenever the UMEM has metadata space, even if the descriptor does not set XDP_TX_METADATA. Pass descriptor options through the metadata helpers and ignore metadata unless the option is set. This does not fix the existing per-WQE metadata handling for mlx5 MPWQEs. Only the descriptor that starts a session passes through xsk_tx_metadata_request() and configures offload state shared by the batch. Metadata on descriptors joining an open session is therefore not validated and does not configure its requested offloads. In addition, a non-NULL metadata pointer from such a descriptor is treated as a timestamp completion request even when XDP_TXMD_FLAGS_TIMESTAMP is not set, so its metadata union can be overwritten with an unrequested timestamp. Fixing mixed metadata states within one MPWQE requires a separate change.	N/A	More Details
CVE-2026-90087	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: do not leak an hci_conn when a second LE connect is rejected create_le_conn_complete() decides whether the failed connection is still pending by comparing it against hci_lookup_le_connect(), which returns the first LE connection in BT_CONNECT. That is the same connection only while at most one is pending. Two can be pending. Connections created on the passive scan path sit in BT_CONNECT with HCI_CONN_SCANNING set and are invisible to hci_lookup_le_connect() until hci_le_create_conn_sync() clears the flag when their command is issued, so the -EBUSY guard in hci_connect_le() does not prevent a second connection from being queued while the first is still on the scan path. Whenever two connections are in BT_CONNECT at once, the lookup may return one connection while create_le_conn_complete() is reporting the failure of the other; the early exit then drops the error and hci_conn_failed() never runs on the connection that failed. The controller also rejects a second HCI_OP_LE_CREATE_CONN issued while another connection creation is still outstanding, per Core Spec Vol 4, Part E. The spec calls for Command Disallowed there; the bcm43438 observed here answers with an LMP/LL error code instead, which bt_to_errno() maps to the -EPROTO (-71) in the log below. The leaked connection stays in BT_CONNECT forever, and because hci_connect_le() refuses to dial while hci_lookup_le_connect() finds anything, every subsequent attempt to reach any peer fails with -EBUSY and no command reaches the controller at all. Seen on a bcm43438 with two BLE peers polled on the same interval (state 5 is BT_CONNECT; both handles are UNSET ones, allocated from the ida above HCI_CONN_HANDLE_MAX): Bluetooth: hci1: Opcode 0x2013 failed: -71 # hcitool con < LE 14:9C:EF:03:68:81 handle 3840 state 5 lm CENTRAL < LE C4:D3:6A:8C:B5:38 handle 3841 state 5 lm CENTRAL A btmon capture across the next ten minutes of connect attempts contains no HCI_OP_LE_CREATE_CONN at all; outgoing LE connections do not recover until the adapter is reset. With this change the same scenario fails the rejected connection cleanly and further connects to both peers go through. Ask about the connection itself instead of about the device.	N/A	More Details
CVE-2026-90088	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: Validate MTU in rfcomm_apply_pn() to prevent infinite loop rfcomm_apply_pn() accepts the MTU value from a remote PN (Parameter Negotiation) frame without checking for zero. When the remote peer sends an MTU of zero, d->mtu is set to 0. This causes the sendmsg path to enter an infinite loop when fragmenting data, as each fragment has size == min_t(size_t, len, 0) == 0, so the remaining length never decreases. The infinite allocation of zero-length skbs exhausts all system memory. Fix by clamping d->mtu to RFCOMM_DEFAULT_MTU when the negotiated value is zero, consistent with the initial value assigned in rfcomm_dlc_alloc().	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmtksdio: Fix out-of-bounds DMA read in the TX path btmtksdio_tx_packet() rounds the transfer size up to the SDIO block size of 256 bytes, but hands the host controller the SKB buffer as is: err = sdio_writesb(bdev->func, MTK_REG_CTDR, skb->data, round_up(skb->len, MTK_SDIO_BLOCK_SIZE)); Only skb->len bytes hold packet data, so the controller reads up to 255 bytes of uninitialised memory and sends it to the device over the SDIO bus. Depending on how much tailroom slack the SKB allocation happens to carry, that read can also extend past		More

2026-90090	the end of the buffer. Compute the padded length up front, ensure the SKB has tailroom for it, and zero-fill the padding with skb_put_zero(). skb->len then covers the padding, so sdio_writesb() no longer needs to round up. byte_tx keeps counting the header and the payload only, and the error path restores the SKB so that the caller can requeue it. Writing behind skb->tail is only safe because the driver owns the buffer, which "Bluetooth: btmtksdio: Take exclusive ownership of the SKB before TX" ensures.	N/A	Details
CVE-2026-90094	In the Linux kernel, the following vulnerability has been resolved: arm64: process: Fix context switching MTE store-only tag check SCTLR_EL1.TCSO0 is set when user opt-in for MTE store-only tag check mode. However, it is not part of SCTLR_USER_MASK which imply that on context switch we never clear SCTLR_EL1.TCSO0, so we are leaking that setting into another task. Fix that by including SCTLR_EL1_TCSO0_MASK into SCTLR_USER_MASK	N/A	More Details
CVE-2026-90095	In the Linux kernel, the following vulnerability has been resolved: fuse: Fix the condition to enable over-io-uring The existing condition in fuse_uring_cmd() is there only to avoid disabling io-uring for connections that already run with it, missing was a condition to refuse any IORING_OP_URING_CMD if the connection/channel didn't get enabled because of missing FUSE_INIT reply flag FUSE_OVER_IO_URING. Without the reply flag the barrier in fuse_uring_ready() doesn't work and IO could already be going on and cause deadlock states (at a minimum one between fch->bg_lock and queue->lock). The change itself is trivial, but brings behavior change, FUSE_OVER_IO_URING has to be set in the FUSE_INIT_REPLY by fuse servers to accept any IORING_OP_URING_CMD. Libfuse does that and the only non-libfuse implementation I found (fractal-fuse) also does it. Qemu patches for fuse-io-uring are not merged yet, as far as I know. Moved up is the smp_load_acquire(&fch->initialized) check, as a fuse-server implementation might try to setup io-uring before FUSE_INIT is processed and might have gotten -EOPNOTSUPP instead of -EAGAIN. Also fixed is a stale comment that explains the handling of the FUSE_OVER_IO_URING flag in early RFC versions. If there should be a report from any library or application we probably need to revert this commit.	N/A	More Details
CVE-2026-90096	In the Linux kernel, the following vulnerability has been resolved: fuse: invalidate the correct range after O_APPEND direct write fuse_direct_write_iter() captures pos before generic_write_checks(), which moves ki_pos to EOF for O_APPEND writes: fuse_direct_write_iter() { pos = iocb->ki_pos; /* 0 (user-supplied) */ generic_write_checks(); /* ki_pos -> EOF */ fuse_direct_io(); /* writes at EOF, correct */ invalidate(pos, pos + res); /* [0, res) -- wrong */ } The post-write invalidation targets a stale range instead of the actual written range at EOF. This can cause data inconsistency when the file size is not page-aligned. The tail page straddling EOF has a valid portion before EOF that concurrent readers can fault back in during the DIO write window: Tail page (file size X not page-aligned): page_start X (EOF) page_end |--- valid data ----|-- stale --| CPU0 (O_APPEND DIO writer) CPU1 (buffered reader) ------------------------- ---------- invalidate [X, X+len) tail page evicted FUSE_WRITE in flight ... read [page_start, X) tail page re-faulted [X, page_end) = stale FUSE_WRITE completes i_size = X + len invalidate [0, len) <- WRONG tail page still cached read [X, X+len) hits stale tail page returns old data Fix by reading pos back from iocb->ki_pos after generic_write_checks(), as generic_file_direct_write() does. Also fix a typo in the comment ("may have" -> "may have competed").	N/A	More Details
CVE-2026-90097	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: vmbus: Skip VMBus module cleanup for non-nested root partition The VMBus module initialization function, hv_acpi_init(), currently does nothing when running in the root partition and root is not nested in another VM. But the initialization function reports success, so the VMBus module is indeed loaded. VMBus functionality is not actually needed, but the VMBus module must be loaded so that hv_vmbus_exists() can answer correctly. Furthermore, the mshv_root dependency on the VMBus module is needed as described in the commit message for 840b740a35bf ("mshv: Add conditional VMBus dependency"). Loading the VMBus module without actually initializing it causes failures if the module should later be unloaded. The module unload code tries to clean up things that were never initialized, resulting in memory faults and a panic. Fix this by having VMBus module exit function perform the same check for non-nested root partition, and do nothing in such a case, just like hv_acpi_init(). In the long run, the code that manages the Hyper-V provided SynIC should be refactored to better coordinate the requirements of root partition scenarios and normal VM scenarios, and to hopefully remove the hv_vmbus_exists() dependnecy between mshv_root and VMBus modules. Preventing the current unload failure scenario is an expediency until such a refactoring is done.	N/A	More Details
CVE-2026-90098	In the Linux kernel, the following vulnerability has been resolved: net: sparx5: fix sleep in atomic context in MAC table access sparx5_set_rx_mode() runs with netif_addr_lock_bh held and iterates dev->mc via __dev_mc_sync(), which per address calls sparx5_mc_sync() / sparx5_mc_unsync() -> sparx5_mact_learn() / sparx5_mact_forget(). These take sparx5->lock, a mutex, and then poll the MAC access command register with readx_poll_timeout(). A mutex may block, which is not allowed from atomic context. Convert the driver to the new .ndo_set_rx_mode_async callback introduced in commit 3554b4345d85 ("net: introduce ndo_set_rx_mode_async and netdev_rx_mode_work"). The async callback is invoked from process context, so the mutex and sleeping completion poll can remain. Observed with CONFIG_PROVE_LOCKING, CONFIG_DEBUG_SPINLOCK, CONFIG_DEBUG_MUTEXES and CONFIG_DEBUG_ATOMIC_SLEEP enabled: BUG: sleeping function called from invalid context at kernel/locking/mutex.c:591 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 217, name: ip preempt_count: 201, expected: 0 Call trace: __might_resched+0x144/0x248 __might_sleep+0x48/0x7c __mutex_lock+0x74/0x850 mutex_lock_nested+0x24/0x30 sparx5_mact_learn+0x78/0x100 sparx5_mc_sync+0x40/0x54 __hw_addr_sync_dev+0xc4/0x170 sparx5_set_rx_mode+0x4c/0x58 __dev_set_rx_mode+0x64/0xa4 __dev_open+0x1ec/0x26c	N/A	More Details
CVE-2026-90099	In the Linux kernel, the following vulnerability has been resolved: net/sched: account classifier filter allocations to memcg Allocations in the tc classifier *_change() paths (filter objects, per-CPU counters, and per-filter aux data) use plain GFP_KERNEL without __GFP_ACCOUNT, allowing unprivileged users to pin kernel memory outside memcg charging. The shared tcf_exts_init_ex() action array allocation in cls_api.c was also uncharged; this patch closes it along with the per-classifier filter-object/percpu/aux allocations that remain unaccounted. Add GFP_KERNEL_ACCOUNT to: - the shared tcf_exts_init_ex() action array (cls_api.c), common to every filter of every classifier (32 pointers, 256 bytes); - the filter-object, per-CPU-counter, and per-filter aux allocations in cls_basic, cls_bpf, cls_cgroup, cls_flow, cls_flower, cls_fw, cls_matchall, cls_route and cls_u32; - the u32_init_knode() replace-path knode allocation (cls_u32.c), which allocates the same struct tc_u_knode + sel.keys on every replace of an existing knode and was missed by the create-path-only conversion. Also fix the cls_basic error path: basic_change() inserts fnew into the IDR before allocating the per-CPU counter. If alloc_percpu() fails the errout path kfree'd fnew without idr_remove, leaving a dangling pointer in the IDR. With GFP_KERNEL_ACCOUNT the percpu alloc becomes failable on demand (memcg at memory.max), making the dead path attacker-reachable and burning the handle permanently. Add the idr_remove on the percpu failure path, matching the basic_set_parms failure-path pattern. Note: vega@nebusec.ai provided a poc for basic_cls, but it was easy to extend to the other classifiers. Conditions to recreate the bug: - CONFIG_NET_SCHED, CONFIG_NET_CLS_* (the classifier being used), CONFIG_NET_CLS_ACT, CONFIG_MEMCG, CONFIG_USER_NS, CONFIG_NET_NS. - Unprivileged user in a fresh user+network namespace (unshare -Urn), or root with CAP_NET_ADMIN. - Create a large number of tc filters (e.g. tc filter add dev lo ingress ... <classifier> ...) while watching a memcg-limited cgroup: system slab grows far	N/A	More Details

	faster than memory.current, pinning kernel memory outside memcg charging.		
CVE-2026-90100	In the Linux kernel, the following vulnerability has been resolved: ptp: netc: fix period truncation and potential divide-by-zero in PEROUT The max_period bound in net_timer_enable_perout() was computed as: max_period = (u64)NETC_TMR_DEFAULT_FIPER + integral_period; which exceeds U32_MAX when integral_period > 0 (e.g. 0x100000002 for the default 333333333 Hz clock). A period_ns that passes this check but exceeds U32_MAX is then silently truncated when stored into the u32 struct netc_pp::period field. A truncated value of zero can reach netc_timer_set_perout_alarm(), where the local u32 period variable would also be 0, causing a divide-by-zero in roundup_u64(delta, period) whenever the stime < min_time branch is taken (which always happens for a start time of {0, 0}). Additionally, netc_timer_enable_periodic_pulse() and netc_timer_enable_fiper() both compute: fiper = pp->period - integral_period; A zero pp->period results in an unsigned wraparound to 0xFFFFFFFD, mis-programming the FIPER hardware register. Fix all three issues by capping max_period at NETC_TMR_DEFAULT_FIPER (0xFFFFFFFF). This ensures that any period_ns passing the range check fits in a u32 without truncation, so the stored value is always valid and non-zero. The accepted range is reduced by integral_period ns (typically only a few nanoseconds), which is negligible in practice.	N/A	More Details
CVE-2026-90101	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix call to hardware monitoring event handler The first parameter of hwmon_notify_event() is supposed to be the hardware monitoring device. The bnxt driver calls it with the platform device as first parameter instead. This API break results in undefined behavior and may result in a crash. Pass the hardware monitoring device as parameter instead to fix the problem.	N/A	More Details
CVE-2026-90105	In the Linux kernel, the following vulnerability has been resolved: vxlan: fix reading neigh ha Currently arp/neigh_reduce read neigh ha directly which can lead to partial reads while the neigh is being updated. Use neigh_ha_snapshot to take a stable snapshot of the address similar to route_shortcircuit which already does the right thing.	N/A	More Details
CVE-2026-90106	In the Linux kernel, the following vulnerability has been resolved: net: bridge: arp/nd proxy: fix reading neigh ha Currently neigh ha address is read directly, but that can result in torn/partial reads if the neigh is being updated. Use neigh_ha_snapshot to take a stable snapshot of the address.	N/A	More Details
CVE-2026-90078	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_skbmod: fix length calculations and avoid invalid header warnings syzbot reported a warning in skb_network_header_len() triggered by tcf_skbmod_act(): !skb_transport_header_was_set(skb) WARNING: CPU: 0 PID: 14949 at include/linux/skbuff.h:3243 skb_network_header_len include/linux/skbuff.h:3243 [inline] WARNING: CPU: 0 PID: 14949 at net/sched/act_skbmod.c:55 tcf_skbmod_act+0xfe8/0x1810 net/sched/act_skbmod.c:55 There are a few issues in tcf_skbmod_act(): 1. Calling skb_network_header_len() assumes skb->transport_header is set, which is not guaranteed when tcf_skbmod_act() runs at TC ingress. 2. Unconditionally calling skb_mac_header_len() at the beginning of tcf_skbmod_act() triggers a warning on L3 devices (e.g. TUN) where the MAC header is unset, evaluating to an underflowed garbage length. 3. On TC ingress, skb->data points to the network header. Adding the MAC header length to the IP header length causes skb_ensure_writable() to request more bytes than the actual IP packet length, dropping valid short packets (e.g. 28-byte UDP/IPv4 packets). Fix these by: - Using skb_network_offset(skb) + sizeof(struct iphdr/ipv6hdr) for SKBMOD_F_ECN so that the required length is correctly calculated on both ingress (offset == 0) and egress (offset == mac_len). - Setting max_edit_len to ETH_HLEN for Ethernet header modifications after validating ARPHRD_ETHER.	N/A	More Details
CVE-2026-90076	In the Linux kernel, the following vulnerability has been resolved: net/sched: fq: add overflow bounds to quantum and initial quantum fq_init() computes quantum = 2 * psched_mtu() and initial_quantum = 10 * psched_mtu() with no overflow check. A device with a huge MTU (e.g. dummy with max_mtu == 0 accepting MTU 2147483634) makes psched_mtu() return 0x80000000; the 2 * and 10 * multiplications wrap to 0 in 32-bit arithmetic, so q->quantum == 0. Then in fq_dequeue() the credit-refill loop adds 0 to f->credit (which stays <= 0) and goto begin loops forever under the qdisc lock, creating a soft lockup. Clamp psched_mtu() to [1, 1 << 20] before multiplying so the product cannot wrap, then cap the result at 1 << 20, matching the bound already enforced on TCA_FQ_QUANTUM in fq_change(). Conditions to recreate the bug: a device whose MTU (plus hard_header_len) is large enough that 2 * psched_mtu() wraps (e.g. a dummy device with max_mtu == 0 accepting MTU 2147483634). Requires CAP_NET_ADMIN in a user namespace.	N/A	More Details
CVE-2026-54583	mport is the MidnightBSD Package Manager. Prior to 2.7.8, libmport/fetch.c did not consistently reject empty, dot, dot-dot, or slash-containing bundle filenames before composing package download and write paths. Malicious package index data could place an unsafe value in indexEntry->bundlefile, and the missing is_valid_bundle_filename() checks allowed downloaded package data to be written outside the intended cache location or to an unsafe destination name. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-90075	In the Linux kernel, the following vulnerability has been resolved: net/sched: fq_codel: clamp default quantum and mtu fq_codel_init() sets q->quantum = psched_mtu(qdisc_dev(sch)) without clamping. A device with a huge MTU (e.g. dummy with max_mtu == 0 accepting MTU 2147483634) makes psched_mtu() return 0x80000000, which overflows the signed flow->deficit to INT_MIN in fq_codel_dequeue(), causing an infinite loop and soft lockup. Emulate fq_codel_change() and constrain to [256, FQ_CODEL_QUANTUM_MAX]. The same unclamped psched_mtu() is assigned to q->cparams.mtu a bit below, and fq_codel_change() never updates it. codel_should_drop() tests "*backlog <= params->mtu"; with mtu == 0x80000000 (~2 GiB) and the default 32 MiB memory_limit, the test is always true, so CoDel is silently and completely disabled (no drops, no ECN). Declare a single clamped mtu and assign both q->quantum and q->cparams.mtu from it, which also removes the double psched_mtu() call. Conditions to recreate the bug: a device whose MTU (plus hard_header_len) wraps psched_mtu() into the sign bit (e.g. a dummy device with max_mtu == 0 accepting MTU 2147483634). Requires CAP_NET_ADMIN in a user namespace.	N/A	More Details
CVE-2026-54585	mport is the MidnightBSD Package Manager. Prior to 2.7.8, create_sample_file() in libmport/bundle_read_install_pkg.c did not constrain absolute source and destination paths from the sample-file manifest directive to mport->root. A malicious or malformed package manifest could therefore direct privileged sample-file handling to copy or write outside the configured installation root, compromising local filesystem integrity. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-54586	mport is the MidnightBSD Package Manager. Prior to 2.7.8, the mport_fetch_index(), mport_fetch_bootstrap_index(), and mport_fetch_bundle() paths in libmport/fetch.c accepted non-HTTPS repository and package mirror URLs without a url_is_https() enforcement check. When a cleartext URL was configured or returned by mirror data, a network-positioned attacker could tamper with package index or package download traffic and compromise package selection or integrity. This issue is fixed in version 2.7.8.	N/A	More Details

CVE-2026-54587	mport is the MidnightBSD Package Manager. Prior to 2.7.8, directory assets handled as ASSET_DIR or ASSET_DIR_OWNER_MODE in libmport/bundle_read_install_pkg.c used path-based mport_mkdirp(), ownership, and permission operations. A local attacker able to modify part of the target installation tree could use dot-dot traversal or substitute symlinks during privileged package installation, causing directory creation or attribute changes to affect attacker-selected paths outside the intended package directories. This issue is fixed in version 2.7.8.	N/A	More Details
CVE-2026-90050	In the Linux kernel, the following vulnerability has been resolved: net/sched: fq: clamp quantum and initial_quantum in change path The fq change path accepts TCA_FQ_QUANTUM in [1, INT_MAX] and TCA_FQ_INITIAL_QUANTUM up to INT_MAX, while fq_init() already clamps to [1, 1<<20]. A user can override the init clamp via tc qdisc change, restoring the small-quantum deficit spin that the init clamp prevents. Narrow iq_range.max to 1<<20 so TCA_FQ_INITIAL_QUANTUM is rejected at parse time. Clamp TCA_FQ_QUANTUM to [256, 1<<20] in fq_change() and fq_init() quantum to [256, 1<<20] for tiny-MTU devices. Conditions to recreate the bug: CONFIG_NET_SCH_FQ=y. Requires CAP_NET_ADMIN (namespace-local via unshare -Urn suffices). tc qdisc add dev dummy0 root fq tc qdisc change dev dummy0 root fq quantum 1 stab data 32768 size_log 15 cell_log 0	N/A	More Details
CVE-2026-90053	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_htb: limit htb_classify inner-class filter hops htb_classify() follows each filter-selected inner class by switching to cl->filter_list, but never bounds the number of hops. A filter on an inner class can point back to itself or to another inner class that points back, creating an infinite loop in the packet classification path with the qdisc lock held and BH disabled — a soft lockup / panic from a single packet. Bound the traversal with a hop counter and drop the packet with a rate-limited warning once the bound is exceeded. The counter is incremented at the point the inner filter chain is picked up, after the TC_ACT_* switch has consumed the classifier verdict, so a terminal TC_ACT_QUEUED/STOLEN/TRAP on the last permitted chain still sets *qerr to __NET_XMIT_STOLEN and the packet is not charged as a drop by this qdisc or its parent. The bound is TC_HTB_MAXDEPTH, taken from HTB's own parameters rather than from the qdisc hierarchy depth limit. Class levels run from 0 to TC_HTB_MAXDEPTH - 1, so a traversal that strictly descends in level can take at most TC_HTB_MAXDEPTH hops. That descent is what a sane configuration does, but it is assumed here rather than enforced: htb_find() resolves a classid against every class in the qdisc, so a filter may equally select a sibling or an ancestor. The normal root -> inner -> leaf path takes a single hop, so the bound does not affect legitimate classification. htb_classify() can now return NULL irrespective of CONFIG_NET_CLS_ACT, whereas previously every NULL return sat inside that ifdef. The NULL handler in htb_enqueue() therefore cannot stay conditional either, so drop the ifdef around it. This matches hfsc_enqueue(), which has always handled a NULL class unconditionally. Without it, a kernel built without actions would dereference a NULL class instead of dropping. Conditions to recreate the bug: - CONFIG_NET_SCHED, CONFIG_NET_SCH_HTB, CONFIG_NET_CLS_U32, CONFIG_LOCKUP_DETECTOR. - Create an HTB qdisc on a device (e.g. lo), add an inner class 1:1 with a leaf child 1:10, install a root u32 filter selecting 1:1, and an inner-class u32 filter on 1:1 also selecting 1:1. - Send one packet (ping). On the unfixed kernel the classify loop spins with the qdisc lock held; with softlockup_panic=1 it panics. - Reachable from unprivileged user via unshare -Urn (CAP_NET_ADMIN).	N/A	More Details
CVE-2026-90054	In the Linux kernel, the following vulnerability has been resolved: tcp: fix corruption of urgent data on multi-segment retransmit On the normal xmit path, while in urgent mode we refuse to build a multi-segment TSO packet, so every segment gets its own urg_ptr: /* tcp_write_xmit() */ limit = mss_now; if (tso_segs > 1 && !tcp_urg_mode(tp)) limit = tcp_mss_split_point(...); The retransmit path has no such guard. __tcp_retransmit_skb() builds a segs > 1 skb and hands it to the GSO layer, which only advances th->seq per segment and copies urg_ptr verbatim: /* __tcp_retransmit_skb() */ len = cur_mss * segs; /* segs > 1, no urg_mode check */ ... /* tcp_gso_segment(): bumps seq only, urg_ptr is copied */ urg_ptr is an offset from the segment's own seq, so a copied value points at a different place on each segment. The receiver rebuilds the absolute urgent seq as seg.seq + urg_ptr, so it walks a moving urgent point instead of the one OOB byte: seg1 seq 1 urg_ptr 5001 -> urgent @ 5001 (ok) seg2 seq 1001 urg_ptr 5001 -> urgent @ 6001 (wrong, +MSS) seg3 seq 2001 urg_ptr 5001 -> urgent @ 7001 (wrong, +2*MSS) The real OOB byte is never pointed at, so the receiver stops splicing it out and delivers it as normal in-band data, corrupting the stream. Guard the retransmit length like the xmit path: keep segs = 1 while in urgent mode.	N/A	More Details
CVE-2026-90055	In the Linux kernel, the following vulnerability has been resolved: usb: atm: usbatm: fix invalid ci_range initialization syzbot reported a shift-out-of-bounds in __vcc_connect(): UBSAN: shift-out-of-bounds in net/atm/common.c:382:32 shift exponent -1 is negative CPU: 0 UID: 0 PID: 5987 Comm: syz.0.18 Not tainted syzkaller #0 PREEMPT(full) Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 08/05/2026 Call Trace: <TASK> dump_stack_lvl+0xe8/0x150 lib/dump_stack.c:120 ubsan_epilogue+0xa/0x30 lib/ubsan.c:233 __ubsan_handle_shift_out_of_bounds+0x36d/0x400 lib/ubsan.c:494 __vcc_connect+0x14b4/0x19c0 net/atm/common.c:382 vcc_connect+0x328/0x8f0 net/atm/common.c:498 pvc_bind+0x272/0x380 net/atm/pvc.c:52 __sys_bind+0x2e3/0x410 net/socket.c:1976 __x64_sys_bind+0x7a/0x90 net/socket.c:1979 ... ATM device ci_range fields (vpi_bits and vci_bits) represent the number of bits supported for VPI and VCI addressing on the device. net/atm/common.c directly uses these fields as bit shift counts: vpi >> dev->ci_range.vpi_bits vci >> dev->ci_range.vci_bits 1 << vcc->dev->ci_range.vpi_bits 1 << vcc->dev->ci_range.vci_bits usbatm_atm_init() sets ci_range.vpi_bits and ci_range.vci_bits to ATM_CI_MAX (-1), which is defined in <uapi/linux/atmdev.h> as a sentinel value for userspace ATM_SETCIRANGE requests, not a valid bit count. Shifting by -1 is undefined behavior and triggers UBSAN warnings. ATM UNI cell headers allow up to 8 bits for VPI (0..255) and 16 bits for VCI (0..65535). Initialize vpi_bits to 8 and vci_bits to 16, as done by solos-pci.	N/A	More Details
CVE-2026-90056	In the Linux kernel, the following vulnerability has been resolved: net: fec: only stop PTP if it was initialized fec_ptp_init() is only called when fep->bufdesc_ex is available. However, fec_probe() unconditionally calls fec_ptp_stop() on the failed_init path, and fec_drv_remove() unconditionally calls fec_ptp_stop() during device removal. Check fep->bufdesc_ex before calling fec_ptp_stop() in both paths to avoid stopping PTP when it was not initialized.	N/A	More Details
CVE-2026-90058	In the Linux kernel, the following vulnerability has been resolved: net/sched: bound qdisc_pkt_len to prevent qdisc soft lockup qdisc_get_stab() accepts a user-supplied size table, and __qdisc_calculate_pkt_len() amplifies qdisc_pkt_len() through the overhead, the size-table data (u16), and size_log (up to STAB_SIZE_LOG_MAX). A crafted stab can therefore set qdisc_pkt_len() to ~1 GiB for an ordinary skb. Per-flow deficit schedulers such as DRR and ETS replenish one quantum per loop iteration; with a tiny quantum (1) they spin billions of times under the qdisc lock, producing a soft lockup / RCU stall as illustrated by vega@nebusec.ai. Cap the final qdisc_pkt_len() to QDISC_PKT_LEN_MAX so the size-table amplification cannot drive deficit schedulers into an unbounded loop. A legitimate size table (e.g. qfq's overhead 999999999, which is handled by dropping) is still accepted. Introduce cap QDISC_PKT_LEN_MAX (1 << 20) = 1 MiB which is well above any legitimate single-skb wire length: the largest current skb->len is GSO_MAX_SIZE (524280), and an ATM-style size table (53/48 cell tax) amplifies that to ~578 KB, both comfortably below 1 MiB. At the same time, 1 MiB bounds the deficit refill loop to ~1M iterations per packet with quantum=1, which completes in a few milliseconds well under the demonstrated softlockup threshold (~10^9 iterations).	N/A	More Details

	Conditions to recreate the bug: - CONFIG_NET_SCHED=y, CONFIG_NET_SCH_DRR=y (or CONFIG_NET_SCH_ETS=y). - Attach a DRR (or ETS) root qdisc with a crafted TCA_STAB that amplifies qdisc_pkt_len to ~1 GiB (e.g. size_log=15, data=[32768]). - Add a class with a tiny quantum of 1 and send one small packet; the deficit loop spins billions of times under the qdisc lock and trips the softlockup detector (panic with kernel.softlockup_panic=1). - Reachable as root or from an unprivileged user in a fresh user+net namespace (unshare -Urn) with namespace-local CAP_NET_ADMIN.		
CVE-2026-90060	In the Linux kernel, the following vulnerability has been resolved: ALSA: control: Don't add invalid kcontrols to LED layer The kcontrol LED state layer tries to track the all associated kcontrol elements with naive assumptions that they are readable. But one can create a write-only element that has no get callback (even a user element can do it), and this may lead to a NULL dereference at the call chain of snd_ctl_led_notify(), as found by syzkaller. For avoiding the Oops, add a sanity check of the kcontrol's info and get callbacks, and just skip the invalid kcontrols before assigning the kctl to the LED layer.	N/A	More Details
CVE-2026-90061	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: skip double clone set expressions on element insert Both the dynset and newsetelem path clone the existing set expressions when setting set element expressions if no override expressions are provided. This results in a double clone, once to clone the template set expressions then another clone on the new element. Add a flag to annotate if userspace provides a override expression (ie. expression of the same type of the set but different configuration), otherwise borrow the existing expression from the set. Add conditionals to release expression iif they represent an override. Use this new override_exprs flag to dump the dynset expression override to userspace. This simplifies the existing logic and it also fixes a bug with the connlimit expression which results in a module refcount imbalance WARNING splat when resorting on the default set expressions.	N/A	More Details
CVE-2026-90063	In the Linux kernel, the following vulnerability has been resolved: virtio-net: Ensure that TCP packets don't overflow gso_segs The user can specify any gso_size in a packet crafted with an AF_PACKET PACKET_VNET_HDR socket, even smaller than TCP_MIN_GSO_SIZE = 8. At the same time, GSO_MAX_SIZE = 8 * GSO_MAX_SEGS = 8 * 65535. When the user crafts a packet with gso_size < 8, there is a risk for partial GSO to overflow the 16-bit gso_segs field when dividing the SKB length by gso_size. Adjust gso_size of TCP packets to be at least TCP_MIN_GSO_SIZE = 8. Keep gso_size of UDP GSO packets, as gso_size=1 is valid and explicitly tested at tools/testing/selftests/net/tun.c:649.	N/A	More Details
CVE-2026-90064	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Reject page faults from non-fault-mode scratch VMs Having scratch enabled does not make a VM capable of handling recoverable page faults. Allowing scratch VMs through the ASID lookup also admits dma-fence mode VMs. If such a VM faults on an already valid VMA, the handler reports success without fixing the fault, causing the GPU to retry indefinitely. Only allow fault-mode VMs through the ASID lookup. Fault-mode VMs using scratch remain supported, while faults from 3D VMs are rejected. (cherry picked from commit bfb24a06405b652d37831f3fb66b71d33a6605de)	N/A	More Details
CVE-2026-90065	In the Linux kernel, the following vulnerability has been resolved: net/smc: release the internal TCP sock on IPPROTO_SMC socket creation failure IPPROTO_SMC sockets create an internal TCP sock ("clcsock") from the proto->init hook. When socket creation fails after proto->init has run - e.g. a cgroup BPF program attached to BPF_CGROUP_INET_SOCK_CREATE denies the socket - sk_common_release() only invokes sk_prot->destroy if it is set, but neither smc_inet_prot nor smc_inet6_prot defines it, and smc_destruct() returns early unless sk_state is SMC_CLOSED. As a result, every failing socket(AF_INET, SOCK_STREAM, IPPROTO_SMC) call leaks one tcp_sock, so an unprivileged task able to attach a deny-all BPF_CGROUP_INET_SOCK_CREATE program to its own cgroup can grow kernel memory unboundedly. Add a .destroy hook to both protos that releases the clcsock via smc_clcsock_release(). smc_sk_init() hashes the sock into the smc hashinfo before the clcsock is created, and smc_diag dumps walk that hash dereferencing smc->clcsock without taking clcsock_release_lock, while sk_common_release() calls .destroy before .unhash. Unhash the sock before releasing the clcsock, as __smc_release() does, so a concurrent dump cannot observe the release; the second unhash in sk_common_release() is a no-op.	N/A	More Details
CVE-2026-90066	In the Linux kernel, the following vulnerability has been resolved: samples/ftrace: Fix kthread_stop() on ERR_PTR in ftrace-direct-multi-modify ftrace_direct_multi_init() assigns kthread_run()'s return value to simple_tsk without an IS_ERR() check. When kthread_run() fails it returns ERR_PTR(-ENOMEM), but init still returns 0, so the module loads with simple_tsk holding an error pointer. On unload, ftrace_direct_multi_exit() then passes that ERR_PTR to kthread_stop(), leading to a null-pointer-dereference. Check the return value of kthread_run() with IS_ERR(); on failure, unregister the ftrace direct call and propagate the error code.	N/A	More Details
CVE-2026-90068	In the Linux kernel, the following vulnerability has been resolved: ASoC: dapm: Fix off-by-one check on the second enum channel The snd_soc_dapm_put_enum_double() rejects item[0] once it reaches e->items, but it lets item[1] be equal to it. Both go on to snd_soc_enum_item_to_val(), which indexes e->values with no bound of its own, so an enum with a value table reads one element past the end. The indexing arrived with the MUX consolidation, which relaxed the item[1] check in the same hunk. The value MUX handler it deleted used >= there, and the snd_soc_put_enum_double() in soc-ops.c still does. Only adav80x pairs a value table with two shifts, and its second channel looks accidental, but the control does report two values. Writing three into it reads off the end of adav80x_mux_values. The core catches that only under CONFIG_SND_CTL_INPUT_VALIDATION, which defaults off.	N/A	More Details
CVE-2026-62997	Kedro-Datasets provides data connectors for Kedro. From version 5.0.0 until 9.5.0, kedro_datasets_experimental.pytorch.PyTorchDataset in kedro-datasets loads .pt model files with torch.load without enforcing weights_only=True, and user-supplied load_args are silently dropped. On PyTorch versions earlier than 2.6, a malicious pickle-backed model from an attacker-influenced shared registry, downloaded checkpoint, or partitioned external source can execute arbitrary code when a Kedro pipeline loads it. The issue affects only the opt-in kedro_datasets_experimental component and does not affect users who load only trusted files. This issue is fixed in version 9.5.0.	N/A	More Details
CVE-2026-90070	In the Linux kernel, the following vulnerability has been resolved: tpm: st33zp24: Return zero on status read failure st33zp24_status() ignores the result of the transport read and returns data even when no byte was received. The I2C transport, for example, skips i2c_master_recv() when the register-select write is short or fails, leaving data uninitialized. The resulting stack value can be interpreted as TPM_STS flags and let status checks complete spuriously. The status callback cannot propagate a transport error. Return zero unless recv() reports exactly one byte. With no status bits set, callers retry or take their existing timeout or error path instead of acting on an invalid status value. This issue was found by a static analysis checker and confirmed by manual source review.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: net/sched: sfq: clamp quantum to avoid signed overflow soft lockup sfq_init() sets q->quantum = psched_mtu(qdisc_dev(sch)) (unsigned). A device with a huge MTU (e.g. dummy with max_mtu == 0 accepting MTU 2147483634) makes psched_mtu() return 0x80000000, so slot->allot = INT_MIN and INT_MIN +		

2026-90072	INT_MIN toggles between INT_MIN and 0 forever, spinning sfq_dequeue() under the qdisc lock. Clamp the quantum to [256, 1 << 20] so the refill loop terminates. The lower bound also covers q->quantum == 0 (psched_mtu() returning 0), which spins sfq_dequeue() identically. sfq_change() already rejects a negative quantum, so only the init path was exposed. Conditions to recreate the bug: a device whose MTU (plus hard_header_len) wraps psched_mtu() into the sign bit (e.g. a dummy device with max_mtu == 0 accepting MTU 2147483634). Requires CAP_NET_ADMIN in a user namespace.	N/A	More Details
CVE-2026-90073	In the Linux kernel, the following vulnerability has been resolved: net/sched: hhf: clamp quantum before hhf_change() to avoid overflow hhf_init() sets q->quantum = psched_mtu(qdisc_dev(sch)) with no overflow check. A device with a huge MTU (e.g. dummy with max_mtu == 0 accepting MTU 2147483634) makes weight * quantum overflow the signed deficit in hhf_dequeue(), spinning forever. Clamp q->quantum before hhf_change() so both the opt and !opt paths see a sane quantum. Without this, bare "tc qdisc add ... hhf" succeeds with a clamped quantum but "tc qdisc add ... hhf limit 1000" (any option present) fails with -EINVAL because hhf_change() re-validates the unclamped default (sch_hhf.c:559). 256 matches fq_codel's floor and is a sane minimum for a DRR quantum. Conditions to recreate the bug: a device whose MTU (plus hard_header_len) wraps psched_mtu() into the sign bit (e.g. a dummy device with max_mtu == 0 accepting MTU 2147483634). Requires CAP_NET_ADMIN in a user namespace.	N/A	More Details
CVE-2026-90074	In the Linux kernel, the following vulnerability has been resolved: net/sched: fq_pie: clamp default quantum to avoid signed overflow fq_pie_init() sets q->quantum = psched_mtu(qdisc_dev(sch)) without clamping. A device with a huge MTU (e.g. dummy with max_mtu == 0 accepting MTU 2147483634) makes psched_mtu() return 0x80000000, which overflows the signed flow->deficit to INT_MIN in fq_pie_qdisc_dequeue(), causing an infinite loop and soft lockup. Emulate fq_pie_policy which is already bounded to [1, 1 << 20]; clamp the default to [256, 1 << 20]. 256 matches fq_codel's floor and is a sane minimum for a DRR quantum. Conditions to recreate the bug: a device whose MTU (plus hard_header_len) wraps psched_mtu() into the sign bit (e.g. a dummy device with max_mtu == 0 accepting MTU 2147483634). Requires CAP_NET_ADMIN in a user namespace.	N/A	More Details
CVE-2026-82410	Pocketbase is an open source web backend written in go. Prior to 0.22.48 and 0.39.7, PocketBase's panic-recovery middleware covers regular request handling but not internal child and worker goroutines. A panic in one of these internal goroutines can escape recovery and terminate the server process, causing a denial of service. The remediation introduces routine.SafeWrap to convert recovered panics into regular errors and applies it to the affected internal worker functions. This issue is fixed in versions 0.22.48 and 0.39.7.	N/A	More Details
CVE-2026-77412	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, readField in read.go reads the length of an AMQP byte-array field with type tag x into a signed int32 and passes the value directly to make when allocating the field buffer. A malicious or compromised broker can encode a value such as 0xFFFFFFFF, which becomes -1 and causes a len out of range runtime panic. The panic escapes the network reader goroutine and terminates the client process, including during connection.start server properties or message header table parsing. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90245	In the Linux kernel, the following vulnerability has been resolved: fbdev: kyro: Validate overlay viewport coordinates The overlay viewport end coordinates are computed from the viewport origin and dimensions using 32-bit unsigned arithmetic. Large input values can cause these calculations to wrap around before the resulting coordinates are passed to SetOverlayViewPort(). SetOverlayViewPort() packs the viewport coordinates into 16-bit register fields. The X coordinates are additionally adjusted by +2 and +1 before being written. Validate the coordinate calculations for 32-bit wraparound and ensure that the adjusted coordinates fit within their 16-bit register fields before calling SetOverlayViewPort(). Found by Linux Verification Center (linuxtesting.org) with SVACE.	N/A	More Details
CVE-2026-90397	In the Linux kernel, the following vulnerability has been resolved: firmware: qcom: scm: Fix NULL dereference in IRQ handler before __scm is published In qcom_scm_probe(), devm_request_threaded_irq() is called before smp_store_release(&__scm, scm). Two paths can dereference __scm before it is published, both causing a NULL pointer dereference. The IRQ handler receives scm via its data argument but passes only wq_ctx to qcom_scm_waitq_wakeup() and qcom_scm_get_completion(), which then dereference __scm directly. Thread scm through both functions so the IRQ handler path never touches __scm. Non-atomic SMC calls made during probe (e.g. from qcom_tzmem_init via qcom_scm_shm_bridge_enable) can return WAITQ_SLEEP, causing qcom_scm_wait_for_wq_completion() to run before __scm is published and dereference it. Add platform_set_drvdata(pdev, scm) early in probe and change qcom_scm_wait_for_wq_completion() to take the device pointer and use dev_get_drvdata() to reach scm, removing any dependency on __scm.	N/A	More Details
CVE-2026-73164	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary OS commands as root via crafted request parameters.	N/A	More Details
CVE-2026-73163	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary OS commands as root via crafted request parameters.	N/A	More Details
CVE-2026-90389	In the Linux kernel, the following vulnerability has been resolved: md: scope memalloc_noio to allocation critical sections Storing a memalloc_noio_save() token in mddev->noio_flags lets one task save the token and another task restore it. With concurrent suspend sysfs writes, task A can enter PF_MEMALLOC_NOIO, return to userspace still in that scope, and later task B can restore A's saved token. Avoid tying the token lifetime to mddev. Keep mddev_suspend() and mddev_resume() only responsible for array suspension, and enter PF_MEMALLOC_NOIO only in the MD paths that allocate memory after the array has been suspended. Restore the token before resuming the array. A reproducer repeatedly writes suspend_lo and suspend_hi from concurrent workers and checks each worker's /proc/self/stat flags before and after the sysfs write.	N/A	More Details
CVE-2026-90390	In the Linux kernel, the following vulnerability has been resolved: md/bitmap: resume array on backlog_store() error path backlog_store() suspends the array before checking whether a write-mostly device exists. If no such device exists, the error path only unlocks reconfig_mutex and leaves the array suspended, blocking subsequent I/O. Use mddev_unlock_and_resume() to release both states.	N/A	More Details
CVE-2026-90391	In the Linux kernel, the following vulnerability has been resolved: lib/test_hmm: fail dmirror_fault() when the mirrored mm is gone dmirror_fault() is called from the dmirror_read() and dmirror_write() retry loops after dmirror_do_read() or dmirror_do_write() finds a missing device page table entry. If the mirrored mm has already exited, mmget_not_zero() fails. The current code returns 0 in that case, which tells the caller that faulting succeeded even though no page was faulted and no device page table entry was installed. The caller then retries the same address, hits -ENOENT again, and can loop forever	N/A	More Details

	without making progress. Return -EFAULT instead, so the ioctl fails when the mirrored mm is no longer faultable.		
CVE-2026-19535	Nozomi Networks Labs identified a CWE-352: Cross-Site Request Forgery (CSRF) vulnerability in the LuCI administrative web interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote unauthenticated attacker to perform unauthorized state-changing requests on behalf of a logged-in administrator, enabling unauthorized access to privileged management functions.	N/A	More Details
CVE-2026-90393	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix potential UAF in bpf_netns_link_update_prog In bpf_netns_link_update_prog, the checks for old_prog and prog type are currently performed locklessly before acquiring netns_bpf_mutex. This creates a race condition that can lead to a UAF issue. If two threads concurrently execute BPF_LINK_UPDATE on the same netns link, the following execution path can trigger a UAF: CPU0 CPU1 bpf_netns_link_update_prog if (old_prog && old_prog != link->prog) return -EPERM; bpf_netns_link_update_prog if (old_prog && old_prog != link->prog) ... old_prog = xchg(&link->prog, new_prog); bpf_prog_put(old_prog); if (new_prog->type != link->prog->type) <-- trigger UAF Fix this by moving the old_prog and prog->type checks inside the netns_bpf_mutex critical section. Meanwhile, use guard() to simplify lock management and avoid all the goto jumping.	N/A	More Details
CVE-2026-90394	In the Linux kernel, the following vulnerability has been resolved: power: supply: sc2731_charger: cancel work on remove The USB notifier and initial charger detection can schedule info->work. The remove path unregisters the notifier, but does not cancel queued or running work before the devm-allocated driver data is released. Set the platform drvdata used by remove, then cancel the work after unregistering the notifier. This issue was found by a static analysis tool.	N/A	More Details
CVE-2026-90395	In the Linux kernel, the following vulnerability has been resolved: power: supply: isp1704_charger: cancel work on remove The USB notifier and initial VBUS detection can schedule isp->work. The remove path unregisters the notifier and power supply, but does not wait for queued or running work before tearing down the power supply state. Cancel the work after unregistering the notifier. Do this before unregistering the power supply. This issue was found by a static analysis tool.	N/A	More Details
CVE-2026-90396	In the Linux kernel, the following vulnerability has been resolved: block: fix dio leak on metadata mapping error A failed integrity mapping holds a dio reference, so we need to go through the full bio ending in case there were previously submitted bio's in the sequence.	N/A	More Details
CVE-2026-90400	In the Linux kernel, the following vulnerability has been resolved: md: recheck spare changes before starting sync remove_spares() and remove_and_add_spares() modify the array's rdev configuration. These operations are only safe after the array has been suspended. md_start_sync() checks whether spare configuration changes are needed before taking reconfig_mutex. However, the rdev state can change before the mutex is acquired, so the initial check can become stale. In that case, md_choose_sync_action() may remove or replace rdevs while normal I/O is still accessing them. The race can occur as follows: raid10d Worker Normal IO _____ _____ _____ raid10_write_request() wait_blocked_dev() set Blocked set Faulty Skip Faulty rdev rrdev->nr_pending++ .repl_bio = bio removeable_rdev = false . array not suspended . lock mddev goto err_handle lock mddev (wait) . update sb . clear Blocked . . unlock mddev . lock mddev (acquires) remove_spares() removeable_rdev = true raid10_remove_disk() rdev = replacement replacement = NULL rdev_dec_pending(NULL) unlock mddev (NULL)->nr_pending-- In this case, rdev_dec_pending() is called with a NULL pointer, resulting in a NULL pointer dereference when attempting to decrement nr_pending. Fix this by suspending the array when spare configuration changes are needed, including for non-read-write arrays, and checking again after taking reconfig_mutex. If the array was not already suspended and a change is now needed, release the mutex, suspend the array, and reacquire the mutex before continuing.	N/A	More Details
CVE-2026-90360	In the Linux kernel, the following vulnerability has been resolved: regulator: core: use system_freezable_wq for init complete work schedule_delayed_work() uses system_wq, which is non-freezable, allowing regulator_init_complete_work to run concurrently with system suspend. This work fires ~30s after boot to disable unused regulators via I2C. When it races with PM suspend, the I2C adapter may already be suspended, triggering a -ESHUTDOWN warning in __i2c_transfer(): WARNING: ... at __i2c_transfer+0x36c/0x3c8 Call trace: __i2c_transfer i2c_transfer regmap_i2c_write _regmap_update_bits regulator_disable_regmap _regulator_do_disable regulator_late_cleanup regulator_init_complete_work_function process_one_work Switch to system_freezable_wq so the work is frozen before any device is suspended, eliminating the race.	N/A	More Details
CVE-2026-58147	WNC T-Mobile 5G Box IDU router contains an OS command injection vulnerability in the portal.cgi component's password change functionality. The application improperly neutralizes special elements in the http_passwd_hidden and http_passwdConfirm_hidden parameters, allowing an authenticated attacker to execute arbitrary commands on the underlying operating system with root privileges.This issue has been fixed in firmware version 1.1.0.651412	N/A	More Details
CVE-2026-58146	WNC T-Mobile 5G Box IDU router is vulnerable to OS command injection vulnerability. The vulnerability exists within the /cgi-bin/portal.cgi endpoint, specifically through the cli_cookie POST parameter. The cli_cookie parameter value is directly concatenated into a find command string without proper sanitization. This allows a remote, unauthenticated attacker to inject and execute arbitrary shell commands as root on the underlying operating system. This issue has been fixed in firmware version 1.1.0.651412	N/A	More Details
CVE-2026-90404	In the Linux kernel, the following vulnerability has been resolved: platform/chrome: cros_ec_debugfs: Unregister panic notifier cros_ec_debugfs_probe() registers notifier_panic with the EC panic notifier chain. The remove path tears down debugfs and the console log, but leaves the notifier registered. A later panic notification can call back into the removed instance and queue work that accesses released data. Unregister the panic notifier before tearing down the debugfs and console log state. This issue was found by a static analysis tool.	N/A	More Details
CVE-2026-14850	The password reset funcionality is vulnerable to unauthorized account modification due to improper validation of the user_id parameter. An attacker can manipulate this predictable numeric identifier to reset passwords for arbitrary users without proving account ownership.	N/A	More Details
CVE-2026-90406	In the Linux kernel, the following vulnerability has been resolved: media: qcom: iris: handle runtime PM resume failure in core deinit Check the return value of pm_runtime_resume_and_get() in iris_core_deinit(). If runtime PM resume fails, skip hardware power-off operations but still perform software teardown and state transition. Also skip the corresponding pm_runtime_put_sync() call to avoid unbalanced runtime PM references.	N/A	More Details
CVE-	WNC T-Mobile 5G Box IDU router contains a cross-site request forgery (CSRF) vulnerability in the portal.cgi component. The anti-CSRF mechanism fails to validate the csrf_token_value parameter, accepting any arbitrary value as valid. This allows a		More

2026-40857	remote attacker to perform unauthorized actions on the device by tricking an authenticated user into visiting a malicious website.This issue has been fixed in firmware version 1.1.0.651412	N/A	Details
CVE-2026-40856	WNC T-Mobile 5G Box IDU router is vulnerable to improper access control. The vulnerability exists in the wnc_maccheck.cgi endpoint, which is accessible without authentication. It allows a remote attacker to retrieve sensitive configuration data, including the administrator web password, WiFi passphrase, and technical device information.This issue has been fixed in firmware version 1.1.0.651412	N/A	More Details
CVE-2026-90409	In the Linux kernel, the following vulnerability has been resolved: drm/panthor: Add vm_bind region with kbo range overlap check When a VM is created, caller has to specify the range of the address space carve-out set aside for mapping kernel BO's. That means vm_bind mappings of UM-exposed BO's should not intersect with that region, but at the moment we're not checking this. At first, I thought of giving these values to drm_gpuvm_init() through its reserve_{offset, range} arguments, but it turns out that is meant for VM address spans that are not managed through the usual drm_gpuvm split/merge circuit, so storing the end of the user VA range at VM creation time and doing a quick check in the vm_bind ioctl path was the simplest workaround. The new check also makes sure vm_bind range doesn't overflow the size of a 64-bit unsigned integer. That was already being done further down the call stack inside drm_gpuvm_sm_map -> drm_gpuvm_range_valid, but it's best to fail early in the driver before GPUVM functions are invoked so that we won't waste time allocating vm_bind context resources.	N/A	More Details
CVE-2026-90410	In the Linux kernel, the following vulnerability has been resolved: spi: davinci: switch to managed controller allocation The controller is allocated with the non-managed spi_alloc_host() while the interrupt is registered with devm_request_threaded_irq(). During removal, spi_bitbang_stop() only unregisters the controller; the subsequent spi_controller_put() then frees the controller together with its embedded davinci_spi devdata, which is the IRQ handler's dev_id. The devm_request_threaded_irq() release action (free_irq()), which drains the handler, does not run until after .remove() returns. A late or latched interrupt can therefore reach davinci_spi_irq() and dereference already-freed memory. Switch to devm_spi_alloc_host() so that the devres LIFO order releases the controller only after free_irq() has drained the handler, and drop the now-redundant spi_controller_put() from .remove(). The probe error path is simplified to direct returns. The clock is acquired with devm_clk_get_enabled(), which is registered after the IRQ and thus released before it by the devres LIFO order. Drain the interrupt explicitly with devm_free_irq() before disabling the controller so that a late interrupt cannot access the registers of a clock-gated controller. This issue was found by an in-house static analysis tool.	N/A	More Details
CVE-2026-90411	In the Linux kernel, the following vulnerability has been resolved: nvme-fc: unmap cmd_iu DMA on rsp_iu mapping failure in init_request __nvme_fc_init_request() maps cmd_iu and then rsp_iu for DMA. If the rsp_iu mapping fails, the original code only recorded the error and fell through: it left the already-mapped cmd_iu unmapped and still marked the op as FCPOP_STATE_IDLE before returning. Since blk-mq does not call .exit_request() when .init_request() fails, the cmd_iu mapping is leaked for every op whose rsp_iu mapping fails. Jump to an error path on rsp_iu mapping failure that unmaps cmd_iu and returns the error without marking the op idle, so it stays in the FCPOP_STATE_UNINIT state set by the initial memset().	N/A	More Details
CVE-2026-90386	In the Linux kernel, the following vulnerability has been resolved: i3c: dw: avoid shift-out-of-bounds when DAA assigns no devices On an empty bus ENTDAA assigns nothing, so cmd->rx_len (the count of addresses left unassigned) equals master->maxdevs. The GENMASK() index master->maxdevs - cmd->rx_len - 1 then becomes -1, which trips up UBSAN. This happens every time on boot on a Gigabyte/AMD server: UBSAN: shift-out-of-bounds in drivers/i3c/master/dw-i3c-master.c:905:12 shift exponent 64 is too large for 64-bit type 'long unsigned int' CPU: 7 UID: 0 PID: 963 Comm: (udev-worker) Not tainted 7.0.11-200.fc44.x86_64 #1 PREEMPT(lazy) Hardware name: Giga Computing E163-Z34-AAH1-000/MZ33-DC1-000, BIOS R32_F45 04/01/2026 Call Trace: <TASK> dump_stack_lvl+0x5d/0x80 ubsan_epilogue+0x5/0x2b __ubsan_handle_shift_out_of_bounds.cold+0xd7/0x1ab dw_i3c_master_daa.cold+0x1b/0x96 [dw_i3c_master] i3c_master_do_daa_ext.part.0+0x3e/0xf0 [i3c] Skip the mask when no new device was assigned.	N/A	More Details
CVE-2026-90385	In the Linux kernel, the following vulnerability has been resolved: md/raid1: create serial pool adding rdev to array with serialize_policy=1 The following bug has been observed with kernel 7.1.3 after adding a new rdev to an existing RAID1 array with serialize_policy enabled: Oops: 0002 [#1] CPU: 0 UID: 0 PID: 19639 Comm: ext4lazyinit Not tainted 7.1.3-1-default RIP: _raw_spin_lock_irqsave+0x27/0x50 CR2: 0000000000004960 Call Trace: wait_for_serialization+0xb9/0x260 [raid1] raid1_make_request+0x762/0xaff [raid1] md_handle_request+0x1c9/0x2e0 [md_mod] The raid1.c code calls wait_for_serialization() if the MD_SERIALIZE_POLICY is set, and wait_for_serialization assumes that rdev->serial is initialized. Normally this will be the case for arrays that have the serialize_policy sysfs attribute set to 1. But when a new rdev is added to an existing array in bind_rdev_to_array(), the condition at mddev_create_serial_pool() causes creation of rdev->serial to be skipped. Fix it.	N/A	More Details
CVE-2026-90384	In the Linux kernel, the following vulnerability has been resolved: iomap: release the folio batch on iomap callback failures A sashiko review of an unrelated patch points out that the folio batch mechanism used for iomap zero range fails to release the batch in a couple error scenarios. If either calls to ->iomap_end() or ->iomap_begin() fail, the direct return paths bypass the batch cleanup. The ->iomap_end() case is not a practical issue at the moment because there is no user of the mechanism that returns an error from this path. The ->iomap_begin() case is theoretically possible because XFS can invoke the fill helper and error out at various points thereafter. This subtly complicates things because XFS does not transfer iomap_flags to the iomap data structure in the error path. To deal with both of these issues, first make sure to invoke the cleanup helper in the error path for either fs callback. Second, update the helper to clear the flag unconditionally and release the batch so long as it is populated. This more clearly delineates the purpose of the flag to control the I/O path and not necessarily the status of the fbatch, so add a comment around this as well.	N/A	More Details
CVE-2026-73165	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary OS commands as root via crafted request parameters.	N/A	More Details
CVE-2026-90362	In the Linux kernel, the following vulnerability has been resolved: drm/msm/dsi: Drop dev_pm_opp_set_rate(0) dev_pm_opp_set_rate(0) removes the vote specified in required-opps but does not actually park the clock, making it run without the necessary power backing. Drop the explicit call to it. Every call site of ops->link_clk_disable() is followed by pm_runtime_put(), so the power vote will be rescinded if deemed safe. Patchwork: https://patchwork.freedesktop.org/patch/742783/	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: drm/msm: don't tear down KMS twice when KMS init fails When priv->kms_init() (mdp4_kms_init() / mdp5_kms_init()) fails partway through, both display drivers already tear their KMS		

CVE-2026-90363	state down via mdp4_destroy() / mdp5_kms_destroy() before returning the error. The common error path in msm_drm_init() then runs msm_drm_uninit() -> msm_drm_kms_uninit(), which tries to destroy the very same KMS a second time, which causes a use-after-free crash. Bring MDP4/MDP5 in line with the DPU driver whose dpu_kms_init() doesn't perform error cleanup on the failure. Let the common path own the cleanup, instead of freeing the KMS from their error paths. The crash trace for the reference: __lock_acquire from lock_acquire (kernel/locking/lockdep.c:5906 kernel/locking/lockdep.c:5863) lock_acquire from touch_wq_lockdep_map (kernel/workqueue.c:4094 (discriminator 1)) touch_wq_lockdep_map from __flush_workqueue (kernel/workqueue.c:4136) __flush_workqueue from msm_drm_kms_uninit (drivers/gpu/drm/msm/msm_kms.c:243 (discriminator 33)) msm_drm_kms_uninit from msm_drm_uninit (drivers/gpu/drm/msm/msm_drv.c:93) msm_drm_uninit from msm_drm_init (drivers/gpu/drm/msm/msm_drv.c:184) msm_drm_init from try_to_bring_up_aggregate_device (drivers/base/component.c:249 drivers/base/component.c:227) try_to_bring_up_aggregate_device from __component_add (drivers/base/component.c:269 drivers/base/component.c:748) __component_add from dsi_host_attach (drivers/gpu/drm/msm/dsi/dsi_host.c:1739) dsi_host_attach from mipi_dsi_attach (drivers/gpu/drm/drm_mipi_dsi.c:383) mipi_dsi_attach from sharp_nt_panel_probe (drivers/gpu/drm/panel/panel-sharp-ls043t1le01.c:247) Patchwork: https://patchwork.freedesktop.org/patch/742068/	N/A	More Details
CVE-2026-90364	In the Linux kernel, the following vulnerability has been resolved: ACPI: processor: Unregister cpufreq notifier on init failure acpi_processor_driver_init() registers the cpufreq policy notifier before registering the ACPI processor driver and setting up CPU hotplug state. If driver_register() or cpuhp_setup_state() fails, the error path only unregisters the ACPI processor driver and the idle driver. The cpufreq notifier remains registered even though initialization failed. Mirror the module exit path on the init failure path and unregister the cpufreq notifier when it has been registered.	N/A	More Details
CVE-2026-90365	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: cancel reset and rc work on device unregister Both drivers cancelled dump_work on unregister but left reset_work and rc_work to be flushed only by destroy_workqueue() in mt76_free_device(), which runs after the hw is unregistered and the hardware stopped. A reset_work that fires in that window calls ieee80211_restart_hw() and re-arms mac_work on an unregistered hw, and rc_work touches station state being torn down. Cancel both up front, alongside dump_work.	N/A	More Details
CVE-2026-90366	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: reserve space for the CSA-abort countdown TLV When a CSA countdown is active, mt7996_mcu_beacon_cntdwn() emits two bss_bcn_cntdwn_tlv entries (the CSA countdown and the CCA-abort BCC), but MT7996_BEACON_UPDATE_SIZE only reserved one. With MBSSID enabled and a near-maximum beacon template the extra 8 bytes could push the offload command past MT7996_MAX_BSS_OFFLOAD_SIZE and trigger skb_over_panic(). Reserve room for both countdown TLVs.	N/A	More Details
CVE-2026-73172	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the edgserver management service of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a remote unauthenticated attacker to execute arbitrary OS commands as root via crafted requests to TCP port 5058.	N/A	More Details
CVE-2026-90368	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: unwind state on add_interface failure When mt76_wcid_alloc() fails, mt7915_add_interface() returned without clearing the vif_mask/omac_mask bits it had already set, without removing the firmware dev info added earlier, and without clearing a monitor_vif pointer to the vif mac80211 is about to free. mac80211 does not call remove_interface() for a failed add, so the indices and firmware dev entry leaked permanently and testmode could dereference the stale monitor_vif. Add a proper error unwind.	N/A	More Details
CVE-2026-90369	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: fix out-of-bounds access in mmio copy helpers mt76_mmio_write_copy() and mt76_mmio_read_copy() iterate up to ALIGN(len, 4), so a length that is not a multiple of four reads past the source buffer (write_copy) or writes past the destination (read_copy). Copy the aligned body in the loop and handle the remaining tail through a 4-byte bounce buffer, keeping the register access width unchanged.	N/A	More Details
CVE-2026-90370	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: bound TLV walk in mt7996_mcu_get_chip_config The response TLV loop advanced by tlv->len without a minimum, so a theoretical firmware response containing a zero-length TLV could spin forever, hanging the CPU during device probe. The u32 payload was also read without bounds checking. Reject a short fixed field, stop on a TLV whose length underruns the header or overruns the skb.	N/A	More Details
CVE-2026-73171	Nozomi Networks Labs identified a CWE-73: External Control of File Name or Path vulnerability in the backup-restore workflow of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a remote authenticated attacker to overwrite arbitrary files on the device filesystem by uploading a crafted backup archive through the web management interface.	N/A	More Details
CVE-2026-73170	Nozomi Networks Labs identified a CWE-94: Improper Control of Generation of Code ('Code Injection') vulnerability in the Modbus CSV import workflow of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary Lua code on the device via a crafted imported file.	N/A	More Details
CVE-2026-90373	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: clear wcid mask under mutex after RCU pointer clear mt7915_remove_interface() cleared the wcid mask bit with no lock held and before clearing the RCU wcid pointer. The mask is a non-atomic RMW shared with the allocators, which all run under dev->mt76.mutex; on DBDC the two wiphys share one mt76_dev, so this raced add_interface/sta_add on the other band and could leak or double-hand-out a wcid. Clearing the bit before the RCU pointer also let a concurrent allocation reuse the index and publish its wcid, which the subsequent NULL assignment then wiped. Move the clear into the existing mutex section, after the RCU pointer is cleared.	N/A	More Details
CVE-2026-90374	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: validate RX band_idx before dereferencing phys[] band_idx comes from a 2-bit descriptor field (0-3) and was used directly to index dev->mt76.phys[] (size __MT_MAX_BAND == 3) and dereference the result. A corrupt or reserved descriptor value could index out of bounds or hit a NULL phy on parts with fewer bands. Reject invalid band indices, mirroring mt7996_rx_get_wcid().	N/A	More Details
CVE-2026-90375	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: fix non-AQL packet accounting for MLO stations __mt76_tx_queue_skb() overrides the wcid passed by the driver with sta->drv_priv, so the wcid might incorrectly be changed after TX, causing wcid->non_aql_packets to be counted on the wrong wcid. For example, on the AP side, if a station's setup link is the 5G link and the station uses 2G to transmit a frame, the value of non_aql_packets is increased on the 5G wcid but decreased on the 2G wcid. Once the inflated counter exceeds MT_MAX_NON_AQL_PKT, the TX scheduler permanently refuses to service the station. Drop the reassignment and account on the wcid used for transmission. This also records the actual wcid in the queue entry.	N/A	More Details

CVE-2026-90376	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: fix MLD ID in MAC TXD and HIF TXP Problem: MCU command timeout while the firmware state is normal, and the firmware keeps showing the error log "ERROR!! NO PAUSE...". Root cause: If the MLD_ID field in the TXD is neither the primary link id nor the secondary link id, it may lead to a firmware busy loop when the third link is in power saving mode. Remap frames directed to a third link to the primary link wcid. Since TX status events and txfree completions carry the wcid the firmware saw, use the remapped wcid for packet id tracking and non-AQL packet accounting as well, while the frame keeps its original link context for addressing, band and OMAC selection.	N/A	More Details
CVE-2026-90377	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: fix RX data queuing of RRO 3.0 For RRO 3.0, RX data released from a RRO data queue should be put to the indicator queue. The frames are processed and completed in the context of the indicator queue NAPI, which only polls skbs queued on the MT_RXQ_RRO_IND list; frames queued under the data queue id are left sitting on that list until the data queue NAPI happens to run, stalling and reordering RX data.	N/A	More Details
CVE-2026-90378	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt792x: Fix memory leak in SDIO TX path When tx_prepare_skb() returns an error in the SDIO TX path, the skb is not freed, leading to a memory leak. This can occur when zero-length frames (such as WNM NULL frames) are dropped to prevent potential hardware TX hangs. Fix this by properly releasing the skb with ieee80211_tx_status_ext() when tx_prepare_skb() fails.	N/A	More Details
CVE-2026-73169	Nozomi Networks Labs identified a CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') vulnerability in the Modbus transaction management interface of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a remote authenticated attacker to store malicious script content that executes in the browser of any administrator who later opens an affected management page.	N/A	More Details
CVE-2026-73167	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary OS commands as root via crafted request parameters.	N/A	More Details
CVE-2026-73166	Nozomi Networks Labs identified a CWE-94: Improper Control of Generation of Code ('Code Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary code on the device, including OS commands as root.	N/A	More Details
CVE-2026-90382	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt76x02: do not WARN on invalid rx descriptor length The MPDU length in the rx descriptor comes from the hardware. In monitor mode with the fcsfail filter enabled, the hardware passes up corrupted frames, and a corrupted frame can report a length larger than the received buffer. The bounds check correctly discards such frames, but its WARN_ON_ONCE wrapper means any over-the-air garbage frame taints the kernel, and panics it on the first such frame when panic_on_warn is set. Drop the WARN and discard the frame silently, matching what commit c2d4c8723dbf ("mt76x2: remove some harmless WARN_ONs in tx status and rx path") did for the neighboring rx and tx status paths. Observed immediately on rx with an MT7612U in fcsfail monitor mode on a busy channel.	N/A	More Details
CVE-2026-90412	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix return status of RMI log page on allocation failure nvmet_execute_get_log_page_rmi() leaves 'status' holding NVME_SC_SUCCESS (set by the successful nvmet_req_find_ns() call) when the kzalloc() for the log buffer fails. It then jumps to the out label and completes the request with a success status, so the host is told the command succeeded while no data was transferred. Initialize 'status' to NVME_SC_INTERNAL, matching the smart log handler, so an allocation failure is reported as an internal error.	N/A	More Details
CVE-2026-90415	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: free STAG index when TPT entry write fails write_tpt_entry() allocates a new STAG index with c4iw_get_resource() and bumps stats.stag.cur before programming the entry. When write_adapter_mem() fails, it returns the error without releasing the index or reversing the statistic. No MR is inserted into rhp->mrs, so deregistration never reclaims it, leaking the index until device teardown. Record whether this call allocated the index and, on a failed write, return it to tpt_table and decrement stats.stag.cur. Key the rollback on both the write error and that flag, not the error alone: a non-reset update carries a caller-owned STAG that this call did not allocate and must not free.	N/A	More Details
CVE-2026-90416	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix stack out-of-bounds read in cc_params debugfs get_param() reads a congestion parameter as a u32 but formats it with the signed "%d" into an 11-byte stack buffer. A value with bit 31 set, such as 0x80000000, renders as "-2147483648\n" whose full length is 12. snprintf() stores only 11 bytes yet returns 12, so simple_read_from_buffer() treats 12 bytes as valid and reads one byte past lbuf[]. Size the buffer for the widest unsigned decimal, format with "%u" to match the u32, and use scnprintf() so the length passed to simple_read_from_buffer() reflects the bytes actually stored.	N/A	More Details
CVE-2026-92483	In the Linux kernel, the following vulnerability has been resolved: liveupdate: Remember FLB retrieve() status LUO keeps track of successful retrieve attempts on an FLB. It does so to avoid multiple retrievals of the same FLB. Multiple retrievals cause problems because once the FLB is retrieved, the serialized data structures are likely freed and the FLB is likely in a very different state from what the code expects. All this works well when retrieve succeeds. When it fails, luo_flb_retrieve_one() returns the error immediately, without ever storing anywhere that a retrieve was attempted or what its error code was. If the user attempts to retrieve another file registered with the same FLB, LUO will attempt to call the FLB's retrieve() callback again. The retry is problematic for much of the same reasons listed above. The FLB is likely in a very different state than what the retrieve logic normally expects (e.g. some KHO pages may have already been restored and freed). There is no sane way of attempting the retrieve again. Remember the error retrieve returned and directly return it on a retry. This is done by changing the retrieved bool to a retrieve_status integer. A value of 0 means retrieve was never attempted, a positive value means it succeeded, and a negative value means it failed and the error code is the value. This is similar to commit f85b1c6af5bc ("liveupdate: luo_file: remember retrieve() status") which did the same for LUO files.	N/A	More Details
CVE-2026-90035	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix division by zero in get_estimated_bw() get_estimated_bw() divides by link->dpia_bw_alloc_config.bw_granularity, which is zeroed by reset_bw_alloc_struct() and only populated once DP_TUNNELING_BW_ALLOC_CAP_CHANGED has been handled. link_dp_dpia_handle_bw_alloc_status(), the DPCD interrupt handler, calls get_estimated_bw() whenever DP_TUNNELING_ESTIMATED_BW_CHANGED is set, independently of whether DP_TUNNELING_BW_ALLOC_CAP_CHANGED has ever fired for that link. A connected USB4/DPIA tunneling device that reports an estimated-bandwidth change before ever reporting a capability change drives a division by zero in this IRQ path. link_dpia_send_bw_alloc_request() already guards the same bw_granularity division; add the identical guard here rather than introducing a new pattern. (cherry picked from commit f2a961457c33dc34223aad5c9e8971de34a4eed3)	N/A	More Details

CVE-2026-92486	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix CFI mismatch in task work callback BPF subprograms use the bpf_callback_t ABI, but task work invokes the callback through a three-argument function pointer. This trips kCFI. Store and invoke the callback as bpf_callback_t.	N/A	More Details
CVE-2026-92487	In the Linux kernel, the following vulnerability has been resolved: exfat: fix valid_size extension over a shared writable mapping When a shared writable mapping has its valid_size extended by a buffered write or a page fault, exfat zeroes the page-cache gap below the new valid_size. A store through the mapping can race with this zeroing and be overwritten. Fix this by zeroing the gap lazily. Drop ->map_pages so that every first write fault goes through exfat_page_mkwrite(), which advances valid_size to cover the faulting page. With fault-around enabled, a store could install a writable PTE, skip ->page_mkwrite(), and land past valid_size without advancing it. Extending valid_size one faulting page at a time also leaves never-written pages in a large mapping alone. The gap is filled with block granularity, zeroing only the not-uptodate blocks and preserving blocks that may hold data stored through the mapping. On the buffered-write path the invalidate lock is held and the gap is unmapped before zeroing, so a racing store re-faults and, under the inode lock, completes only after the gap has been zeroed and valid_size covers it.	N/A	More Details
CVE-2026-90034	In the Linux kernel, the following vulnerability has been resolved: usb: image: mdc800: change kmalloc() to kzalloc() Change the kmalloc() calls in usb_mdc800_init() for irq_urb_buffer and download_urb_buffer to kzalloc(), avoiding potential stack leaks if a shorter message is received in mdc800_usb_irq() and mdc800_usb_download_notify()	N/A	More Details
CVE-2026-92490	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Unrequest devices if driver registration fails scmi_driver_register() requests protocol devices before registering the driver. If driver_register() fails, those requests remain in the global IDR and retain pointers to the module's ID table. Once the failed module load releases that storage, later request matching or SCMI device creation can dereference the stale pointers. Unrequest the complete protocol table before returning the registration failure. At this point table registration succeeded, so every entry is owned by the current registration attempt.	N/A	More Details
CVE-2026-92491	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Roll back partial protocol table registration scmi_protocol_table_register() can leave earlier requests registered when a later entry in the same ID table fails. Each request retains a pointer to the driver's ID table, so a failed module load can leave a dangling pointer after the module storage is released. Unrequest only the successfully registered prefix, in reverse order, before returning the failure. Leave the failed entry and the remaining entries untouched because matching requests can be owned by another driver.	N/A	More Details
CVE-2026-92492	In the Linux kernel, the following vulnerability has been resolved: cpufreq/amd-pstate: handle missing policy in dynamic EPP callbacks cpufreq_cpu_get() returns NULL when no cpufreq policy is associated with the requested CPU, for example because the CPU is offline or the policy has already been torn down. Both amd_pstate_power_supply_notifier() and amd_pstate_profile_set() acquire a policy via cpufreq_cpu_get() and then pass that pointer to amd_pstate_get_balanced_epp() and amd_pstate_set_epp(), which dereference it unconditionally. A racing CPU hotplug or driver teardown can therefore lead to a NULL pointer dereference on either of these dynamic EPP paths. The third cpufreq_cpu_get() caller in this file, amd_pstate_verify(), already handles the NULL case. Bring the two new callers in line with that pattern: return NOTIFY_OK from the power-supply notifier (matching the other "nothing to do" exits) and -ENODEV from amd_pstate_profile_set() (the usual cpufreq error for a missing CPU policy). Found by code inspection; not tested on hardware.	N/A	More Details
CVE-2026-92493	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate-ut: Skip tests when amd-pstate driver is not active The crash issue may occur when modprobe amd_pstate_ut on intel platform. amd_pstate_ut: 1 amd_pstate_ut_acpi_cpc_valid success! amd_pstate_ut: 2 amd_pstate_ut_check_enabled success! BUG: kernel NULL pointer dereference, address: 0000000000000080 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] SMP NOPTI CPU: 0 PID: 20300 Comm: modprobe Kdump: loaded Tainted: G O 6.6.0-0010.rc1.ctl4.x86_64 #1 Hardware name: FiberHome R2200 V5/Xeon Boards, BIOS 3.1a 02/24/2020 RIP: 0010:amd_pstate_ut_check_perf+0x141/0x280 [amd_pstate_ut] Call Trace: <TASK> amd_pstate_ut_init+0x1b/0xff0 [amd_pstate_ut] ? __pfx_amd_pstate_ut_init+0x10/0x10 [amd_pstate_ut] do_one_initcall+0x42/0x2e0 ? kmalloc_trace+0x26/0x90 do_init_module+0x60/0x240 __se_sys_init_module+0x185/0x1c0 do_syscall_64+0x62/0x190 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> Add state detection to amd pstate driver to prevent amd_pstate_ut driver from testing on non-AMD platforms. (ML: adjust title)	N/A	More Details
CVE-2026-92494	In the Linux kernel, the following vulnerability has been resolved: ext4: fix buffer_head leak in ext4_init_orphan_info ext4_init_orphan_info() reads orphan file blocks with ext4_bread() and stores the returned buffer_head in oi->of_binfo[i].ob_bh. If ext4_bread() succeeds but the orphan block magic or checksum validation fails, the function jumps to out_free. However, the old out_free loop starts releasing buffers from i - 1, so the current buffer_head at index i is skipped. This leaks the buffer_head reference obtained by ext4_bread() on the bad magic and bad checksum error paths. Fix this by tracking the number of successfully read buffer_heads and releasing exactly those buffer_heads on the error path.	N/A	More Details
CVE-2026-92495	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Clear VM_MAYWRITE on DBR/toggle page mmap bnxt_re_mmap() rejects VM_WRITE for the DBR_PAGE and TOGGLE_PAGE mmap flags, but a read-only mapping can still retain VM_MAYWRITE. nd later be upgraded with mprotect(PROT_WRITE). This can bypass the write check that only runs at mmap time. Clear VM_MAYWRITE before vm_insert_page() in the shared DBR/toggle-page branch, matching the existing policy that userspace writes are not expected for these pages.	N/A	More Details
CVE-2026-92496	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: Avoid buffer overread in ath11k_wmi_tlv_op_rx() Currently, in ath11k_wmi_tlv_op_rx(), the firmware buffer is read without first verifying that the buffer has enough data to hold a header. This could result in a buffer overread. Add an upfront length check before dereferencing skb->data as a wmi_cmd_hdr. The check is placed before the trace_ath11k_wmi_event() call to preserve the existing trace semantics (tracing the full raw WMI event including the header), unlike the analogous ath12k fix which could use skb_pull_data() directly. Compile tested only.	N/A	More Details
CVE-2026-92497	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Avoid buffer overread in ath12k_wmi_op_rx() Currently, in ath12k_wmi_op_rx(), the firmware buffer is read without first verifying that the buffer has enough data to hold a header. This could result in a buffer overread. Update the logic to verify the buffer contains at least enough data to hold a wmi_cmd_hdr before reading from the buffer. Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.1.c7-00108-QCAHMTSWPL_V1.0_V2.0_SILICONZ_UPSTREAM-3	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: wifi: ath6kl: avoid buffer overreads in WMI event handlers The		

CVE-2026-92498	following WMI event handlers currently read from the event buffer without first verifying that the message was large enough to hold the expected event: ath6kl_wmi_scan_complete_rx() ath6kl_wmi_addba_req_event_rx() ath6kl_wmi_delba_req_event_rx() Add length checks to prevent overread.	N/A	More Details
CVE-2026-92499	In the Linux kernel, the following vulnerability has been resolved: ext4: validate readdir offset before accessing dirent A corrupted directory can trigger the following KASAN report when ext4_readdir() resumes from an invalid position: BUG: KASAN: use-after-free in __ext4_check_dir_entry+0x5ef/0x820 Read of size 2 at addr ffff88810a646000 by task repro_linear/509 Call Trace: <TASK> dump_stack_lvl+0x53/0x70 print_report+0xd0/0x630 kasan_report+0xce/0x100 __ext4_check_dir_entry+0x5ef/0x820 ext4_readdir+0xcde/0x2b70 iterate_dir+0x1a1/0x520 __x64_sys_getdents64+0x12b/0x220 do_syscall_64+0xf9/0x540 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> KASAN reports use-after-free because the out-of-bounds access lands in an adjacent freed page. The directory buffer itself is still referenced. ext4_dir_llseek() invalidates the directory cookie so that ext4_readdir() rescans directory entries from the start of the block. The rescan checks only the lower bound of rec_len before advancing. A corrupted rec_len can therefore place the offset where the block has insufficient space for a complete directory entry. The rescan itself may dereference that truncated entry, or the main loop may pass it to __ext4_check_dir_entry(). The latter reads de->rec_len before validating the range. For example: block offset 0 4092 4096 |---- de1.rec_len = 4092 -----|----| de2.inode	de2.rec_len ^ OOB, reported as UAF de2 starts at offset 4092 in this 4 KiB block. Its four-byte inode fits in the block, but its rec_len starts at offset 4096 and crosses the boundary. The minimum safe length is inode-dependent. Encrypted and casefolded directory entries need eight additional hash bytes, while a valid metadata checksum tail is only 12 bytes. Cache the metadata checksum feature state and derive the minimum directory entry length from the on-disk format. Use it to bound both the rescan and the offset passed to the main loop. Report an offset in a truncated block tail and skip the remainder of the block, while continuing to accept an offset exactly at the block boundary.	N/A	More Details
CVE-2026-92500	In the Linux kernel, the following vulnerability has been resolved: ext4: use fsdata to track inline data write state and fix race Instead of checking the live inode state (ext4_has_inline_data(inode) and ext4_test_inode_state(inode, EXT4_STATE_MAY_INLINE_DATA)) in the write_end handlers, use the fsdata parameter of the address space operations to explicitly pass down the state in which write_begin prepared the write. A concurrent thread (such as ext4_page_mkwrite()) can convert the inline data to an extent between write_begin and write_end. If this happens, the write_end handlers would previously miss the inline write_end path and fall through to extent-based write_end logic. However, since block buffers were never allocated in write_begin, this resulted in NULL pointer dereferences or data loss because folio_buffers(folio) was NULL. Define EXT4_WRITE_DATA_INLINE (4) as a bit flag (Bit 2), treating fsdata as bitwise flags rather than mutually exclusive enums to keep states of the write path independent. Communicate this state via fsdata: 1) ext4_write_begin() and ext4_da_write_begin() set the EXT4_WRITE_DATA_INLINE bit in *fsdata via bitwise OR when an inline write is successfully prepared. 2) On entry, ext4_write_begin() clears the EXT4_WRITE_DATA_INLINE bit to safely handle VFS retries (where generic_perform_write() bypasses the fsdata initialization on its retry jump). 3) The write_end handlers perform a bitwise AND to check if the EXT4_WRITE_DATA_INLINE bit is set and invoke the inline write_end helper accordingly. Furthermore, during a buffered write, ext4_write_inline_data_end() acquires the xattr lock after preparing the write. If a concurrent page fault (ext4_page_mkwrite()) converts the inline data to an extent after the write_end handlers check the state but before ext4_write_inline_data_end() acquires the xattr write lock, the subsequent check will trigger a kernel panic via BUG_ON(!ext4_has_inline_data(inode)). To keep git history working and bisectability clean, replace the BUG_ON check in ext4_write_inline_data_end() with a graceful error- handling retry path in this same commit. If the inline data is cleared after locking the xattr, we safely release all resources (releasing iloc.bh, unlocking/putting the folio, stopping the active journal transaction handle) and return 0 (VFS retry) to let the generic write path retry the operation safely.	N/A	More Details
CVE-2026-92501	In the Linux kernel, the following vulnerability has been resolved: ext4: drain in-flight DIO before buffered write fallback generic/746 started failing intermittently on ext3 (no-extent inodes). The test triggers 'Page cache invalidation failure on direct I/O' warnings and subsequent fsync returns -EIO. Adding a 50ms delay between ext4_buffered_write_iter() and filemap_write_and_wait_range() in ext4_dio_write_iter() makes the race almost always reproducible. On no-extent inodes, DIO writes to holes cannot use unwritten extents, so ext4_iomap_alloc() leaves m_flags=0 and ext4_map_blocks() returns 0. The iomap layer then returns -ENOTBLK, causing fallback to buffered I/O. The fallback path in ext4_dio_write_iter() calls ext4_buffered_write_iter() which dirties pages, then does flush and invalidate. However, there's an unprotected window between ext4_buffered_write_iter() returning (with inode lock released) and the subsequent flush+invalidate. Concurrent async DIO completions from other threads can run kiocb_invalidate_post_direct_write() during this window. If pages have been re-dirtied, post-invalidation finds dirty pages and triggers the warning, setting -EIO in the error sequence. Consider a file with two 4k extents: [hole][written]. Thread A does DIO to the written extent, while thread B does DIO spanning both: kworker A (4k DIO, allocated block) kworker B (8k DIO, fallback) ---------- ---------- inode_lock_shared() inode_lock_shared() iomap_dio_rw(): iomap_dio_rw(): kiocb_invalidate_pages -> clean iomap_begin -> -ENOTBLK submit_bio (async) dio->size = 0 inode_unlock_shared() inode_unlock_shared() [bio pending in block layer] /* fallback: lock released */ ext4_buffered_write_iter() inode_lock(exclusive) generic_perform_write() -> dirty pages [0, 8k] inode_unlock(exclusive) /* pages dirty, no lock */ [bio completes] filemap_write_and_wait_range() iomap_dio_complete() -> flush dirty pages kiocb_invalidate_post_direct_write() invalidate_mapping_pages() invalidate_inode_pages2_range() -> finds dirty page! -> dio_warn_stale_pagecache() -> errseq_set(-EIO) This issue can be triggered through normal I/O paths, not just intentionally overlapping DIO writes from userspace. For example, generic/746 uses a loop device where multiple kworkers issue concurrent I/O to the backing file. Additionally, when block_size < folio_size, non-overlapping DIO writes that share a large folio can also trigger the race. Add inode_dio_wait() in ext4_buffered_write_iter() before ext4_write_checks() to drain all in-flight DIO. This ensures that all DIO clears existing pages before submitting IO (via kiocb_invalidate_pages()), all BIO waits for all DIO to complete (via inode_dio_wait()), and ext4_write_checks() observes the inode size after all completed DIO so that ext4_block_zero_eof() does not race with in-flight DIO, thus eliminating the race.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: ext4: clear stale xarray tags on folios skipped during writeback In data=journal mode, the writeback thread can hit the WARN_ON_ONCE(sb_rdonly(sb)) in ext4_journal_check_start() while the superblock is being remounted read-only during reboot: Workqueue: writeback wb_workfn (flush-253:0) RIP: 0010:ext4_journal_check_start+0x8b/0xd0 Call Trace: __ext4_journal_start_sb+0x3c/0x1e0 mpage_prepare_extent_to_map+0x4af/0x580 ext4_do_writepages+0x3c0/0x1080 ext4_writepages+0xc8/0x1a0 do_writepages+0xc4/0x180 __writeback_single_inode+0x45/0x2f0 writeback_sb_inodes+0x26b/0x5d0 __writeback_inodes_wb+0x54/0x100 wb_writeback+0x1ac/0x320 wb_workfn+0x394/0x470 And followed by the warning: EXT4-fs warning (device vda1): ext4_evict_inode:195: inode #6263: comm (sd-umount): data will be lost This issue is not reproduced every time, but frequently. The reproduction step is to create a VM with 8 CPUs, 16G memory and setup data=journal: sudo		

CVE-2026-92502	tune2fs -o journal_data /dev/vda1 Run fio: rm -f fiotest fio --name=fiotest --rw=randwrite --bs=4k --runtime=6 --ioengine=libaio --iodepth=256 --numjobs=8 --filename=fiotest --filesize=30G --group_reporting Reboot the VM, and check the console output from: virsh console testvm But there is no dirty inode, folio_clear_dirty_for_io clears PG_dirty but leaves tags PAGECACHE_TAG_DIRTY and PAGECACHE_TAG_TOWRITE set which are only cleared by __folio_start_writeback. In data=journal mode, jbd2 checkpoints the journalled data to its final location and clears its own dirty flag without touching folio PG_dirty or xarray dirty flags. The commit f4a2b42e7891 ("ext4: fix stale xarray tags after writeback") fixes when PG_dirty is still set but there is no dirty page. Another case is PG_dirty is cleared, but PAGECACHE_TAG_DIRTY and PAGECACHE_TAG_TOWRITE is still set. In this case, writeback thread checks clean folio and skips it in mpage_prepare_extent_to_map: if (!folio_test_dirty(folio) || ... folio_unlcok(folio); continue And never reaches ext4_bio_write_folio where the commit f4a2b42e7891 clears the stale xarray tags. Print debug logs after the filesystem is remounted read-only: writepages RDONLY nrpages=2048 dirtytag=1 wbtag=0 towrite=1 sync=0 And all folios are actually clean: folio idx=3 dirty=0 wb=0 checked=0 dirtybuf=0 jbddirty=0 mapped=1 ... We need to clear the xarray stale tags for such clean folios by cycling them through writeback in the skip path, the same way f4a2b42e7891 does in ext4_bio_write_folio.	N/A	More Details
CVE-2026-92503	In the Linux kernel, the following vulnerability has been resolved: ext4: fix ABBA deadlock in ext4_xattr_inode_cache_find() Syzbot/stress-ng reported an ABBA deadlock in ext4 when exercising concurrent xattr workloads (using the ea_inode mount/format option). The deadlock occurs between the running transaction and the eviction thread: - Task 1 (stress-ng): Holds a reference to a shared mbcache_entry (ce) and calls ext4_xattr_inode_cache_find() -> ext4_iget() to retrieve the corresponding EA inode. Since the EA inode is currently being evicted, ext4_iget() blocks in __wait_on_freeing_inode() waiting for eviction to complete. - Task 2 (eviction thread): Currently evicting the same EA inode in ext4_evict_ea_inode(). It calls mb_cache_entry_wait_unused(oe) which blocks waiting for Task 1 to release the reference to the mbcache_entry. To break this deadlock, implement a new ext4_iget() configuration flag named EXT4_IGET_NOWAIT. When set, perform a non-blocking lookup of the inode via VFS's find_inode_nowait() API. If the inode is currently being evicted (marked with I_FREEING or I_WILL_FREE) or created (I_CREATING), or if it is not present in the VFS inode cache (cache miss), simply skip it (returning -ENOENT) rather than waiting for eviction/creation to complete, breaking the ABBA cycle. Since we return -ENOENT immediately on a cache miss, we never attempt to allocate a new inode or call iget_locked(), completely eliminating any TOCTOU race window. If the returned inode is I_NEW, wait for its initialization to clear via wait_on_new_inode(). If initialization fails and the inode is unhashed during wait_on_new_inode() waking up (e.g., due to an I/O read error in another thread), safely drop the reference and return -ENOENT. This unhashed check is executed unconditionally on all cache-hit pathways to properly handle concurrent initialization failures. Finally, standard validation checks (including is_bad_inode, EXT4_EA_INODE_FL, file_acl, and xattr flags) are executed as normal inside check_igot_inode() to fully guarantee VFS-layer safety. In ext4_xattr_inode_cache_find(), invoke ext4_iget() with the new EXT4_IGET_NOWAIT flag to perform the non-blocking cache search.	N/A	More Details
CVE-2026-90033	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: fix OOB write in snd_usbmidi_us122l_output() The snd_usbmidi_us122l_output() picks a count of 2 on anything slower than high speed and never relates it to ep->max_transfer. The URB buffer holds exactly max_transfer bytes, so a device declaring a one byte bulk endpoint takes two bytes from snd_rawmidi_transmit(), and the memset that pads the rest computes 1 - 2 in int and wraps to SIZE_MAX. Only 0x800e and 0x800f are pinned to nine bytes. The US-122MKII at 0x0644:0x8021 falls to the default and takes usb_maxpacket(), which the USB core only clamps downward. The akai and novation output ops in this file were given the same guard recently. Do the same here.	N/A	More Details
CVE-2026-92505	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix undefined behavior in devid_write debugfs function When for_each_pci_segment() loop completes without finding a matching segment, the pci_seg pointer is not NULL but points to an invalid memory location (the list head). Accessing pci_seg->id after the loop causes undefined behavior. Fix this by handling the successful case inside the loop and returning -EINVAL after the loop if no matching segment is found.	N/A	More Details
CVE-2026-92506	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix requested device removal race scmi_protocol_device_unrequest() drops scmi_requested_devices_mtx while notifying listeners but continues to retain the per-protocol list head. When two SCMI drivers for the same protocol unregister concurrently, one thread can remove the final request and free the list head while the other is running its notifier. The latter then dereferences the freed list head after reacquiring the mutex and can free it a second time. Complete the list and IDR updates, including freeing an empty list head, before dropping the mutex. Keep the blocking notifier outside the critical section and retain only the detached request across the callback.	N/A	More Details
CVE-2026-92484	In the Linux kernel, the following vulnerability has been resolved: cxl/region: Fix use-after-free in find_pos_and_ways() error path The error path releases its reference to a switch decoder before logging an error that includes the decoder name. If the released reference is the last one, the decoder can be freed before the error message accesses its name. Drop the reference after the error is reported.	N/A	More Details
CVE-2026-92482	In the Linux kernel, the following vulnerability has been resolved: pinctrl: mediatek: use devm_gpiochip_add_data() for GPIO chip The gpio_chip is allocated with device-managed memory but registered with the non-managed gpiochip_add_data(). This was harmless while the drivers were built-in, but once they can be built as modules and unbound/rmmod'd, devm frees the gpio_chip's memory while it is still registered, causing a use-after-free. Register it with devm_gpiochip_add_data() so it shares the same device-managed lifecycle, which also lets the manual gpiochip_remove() error paths go away.	N/A	More Details
CVE-2026-90417	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Fix dereg_skb leak and double free in write_tpt_entry() When the device is in the fatal error state, write_tpt_entry() returns -EIO before handing the caller's preallocated skb to the transmit path; its allocation-failure returns do the same. c4iw_dereg_mr() ignores the error and frees mhp, leaking mhp->dereg_skb. c4iw_get_dma_mr() instead frees the skb a second time after dereg_mem() already consumed it, a double free. Make write_tpt_entry() the sole owner of a non-NULL skb, freeing it on every return preceding handoff to c4iw_ofld_send(): fatal error, tpt and stag allocation failure. c4iw_ofld_send() consumes the skb on success and error alike, so drop the redundant kfree_skb() in c4iw_get_dma_mr() after dereg_mem().	N/A	More Details
CVE-2026-92481	In the Linux kernel, the following vulnerability has been resolved: pinctrl: mediatek: free EINT resources on unbind mtk_eint_do_init() creates an IRQ domain, populates it with a mapping for every EINT line and installs a chained handler on the parent interrupt, but none of these are ever released. This was harmless while the drivers were built-in, but now that they can be built as modules and unbound/rmmod'd it leaves behind a dangling IRQ domain, interrupt mappings whose chip data points at freed memory, and a chained handler that keeps firing into that freed data. The plain allocations in mtk_eint_do_init() already use the device-managed devm_*() helpers, so tear the remaining resources down the same way: register a devm action that detaches the chained handler, waits for any in-flight handler to finish, disposes of the per-line mappings and removes the IRQ	N/A	More Details

	domain. This mirrors the device-managed lifecycle adopted for the GPIO chip and keeps the whole EINT setup self-cleaning on unbind.		
CVE-2026-90418	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix BUG in nilfs_copy_dirty_pages() on dirty state mismatch Syzbot reported a kernel BUG triggered within nilfs_copy_dirty_pages(), which copies dirty DAT file folios/pages to its shadow page cache. The BUG occurs when a retrieved dirty folio/page unexpectedly loses its 'dirty' status. This issue arises because, since the commit referenced below, the 'dirty' flag of a folio/page can be cleared asynchronously after the filesystem detects metadata corruption and transitions to read-only mode. Resolve the issue by returning an -EROFS error if the filesystem has transitioned to read-only mode. Also change the behavior to issue a kernel warning only once instead of triggering a kernel BUG when this unexpected 'dirty' state is detected while the filesystem is not in read-only mode.	N/A	More Details
CVE-2026-40855	WNC T-Mobile 5G Box IDU router is vulnerable to a command injection. The vulnerability exists in the ping functionality within the /cgi-bin/portal.cgi endpoint, specifically affecting the ping_ip, ping_size, and ping_times POST parameters. The root cause is the failure to verify and sanitize user-supplied input before incorporating it into a system command. This allows an authenticated attacker to execute arbitrary commands on the shell and gain root access to the system.This issue has been fixed in firmware version 1.1.0.651412	N/A	More Details
CVE-2026-90420	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix infinite loop in nilfs_clean_segments() syzbot reported a hung task in nilfs_transaction_begin(). This occurs because the cleaner ioctl falls into an infinite loop if nilfs_segctor_construct() repeatedly returns -EROFS (e.g. the device is remounted as read-only after an I/O error). Currently in nilfs_clean_segments(), if err is non-zero, it logs the error and sleeps but doesn't abort when it encounters a terminal error like -EROFS. This causes the thread to loop forever. Fix this by breaking out of the loop if nilfs_segctor_construct() returns -EROFS. This matches the behaviour in nilfs_segctor_write_out(), which also handles -EROFS.	N/A	More Details
CVE-2026-90421	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix UAF when probe runs concurrent to dyn ID removal Dynamic IDs are only guaranteed to be valid when dynids.lock is held, as remove_id_store() can free the node. Thus, make a copy in pci_match_device(). Also, clarify that the id parameter is only valid during probe.	N/A	More Details
CVE-2026-90422	In the Linux kernel, the following vulnerability has been resolved: clk: mediatek: pllfh: Fix IO remapping leak in register_pllfhs error path When mtk_clk_register_pllfhs function fails to register a PLL, it unregisters all PLLs and cleans up itself in its error path before returning, so the function callers don't need to do it. But contrary to mtk_clk_unregister_pllfhs function, that does almost the same sequence, it does not free the IO memory mapped on fhctl node, leading to a leak. Fix this leak by factorizing the cleanup sequence in a new private function and use it both mtk_clk_register_pllfhs and mtk_clk_unregister_pllfhs functions. Also, change the loop index start value to avoid the -1 operation on index at each loop.	N/A	More Details
CVE-2026-40854	WNC T-Mobile 5G Box IDU router contains an authentication bypass vulnerability in the portal.cgi component. The session verification mechanism improperly validates the sessionid cookie by checking for the existence of a corresponding file in /tmp/login_user. An attacker can bypass authentication by using directory entries such as "." or ".." in the cookie, allowing unauthorized access to the administration panel.This issue has been fixed in firmware version 1.1.0.651412	N/A	More Details
CVE-2026-90424	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Fix VINTF0 leak on the init-failure path tegra241_cmdqv_init_structures() allocates VINTF0 with kzalloc_obj(), inits it, and preallocates its logical VCMDQs. Two of its error paths leak. When tegra241_cmdqv_init_vintf() fails it returns before VINTF0 reaches the cmdqv->vintfs[] array, so the devres unwind on probe failure cannot reach it; free it directly there. A later VCMDQ preallocation failure instead leaves VINTF0 published, and so this time the unwind does reach tegra241_cmdqv_remove_vintf(), which then frees it from vintf->hyp_own. But tegra241_vintf_hw_init() sets that flag only afterward, from a HW read-back, so the still-uninited VINTF0 reads as guest-owned and leaks, with mutex_destroy() and ida_destroy() run on fields it never set up. Decide ownership from vintf->idx instead, the index assigned when its id is allocated: idx 0 is the kernel-owned VINTF0, while idx >= 1 marks a guest VINTF. So the in-kernel free decision in tegra241_cmdqv_remove_vintf() and tegra241_vintf_free_lvcmdq() now keys on idx too, and hyp_own stays a pure HW-readback state.	N/A	More Details
CVE-2026-90426	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Free the error IRQ before tearing down VINTFs tegra241_cmdqv_remove() tears each VINTF down first, then calls free_irq(). Tearing a VINTF down frees vintf0 and clears cmdqv->vintfs[0]. An error in that window makes tegra241_cmdqv_isr() read the stale slot and hand it to tegra241_vintf0_handle_error(), which dereferences a NULL or freed pointer. Free the IRQ before tearing the VINTFs down. free_irq() waits for in-flight handlers to finish and blocks new ones, so no ISR can observe a VINTF as it is torn down. Note: a user-owned VINTF (viommu) could outlive this teardown, which unmaps cmdqv->base and frees cmdqv->vintfs, so a later viommu close then touches freed memory. This is neither introduced nor fixed here: a physical IOMMU is not a pluggable device, so iommufd by design holds no reference on the one behind a viommu, and this teardown is not expected while that viommu is still alive.	N/A	More Details
CVE-2026-90428	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Don't run the error ISR before probe sets up vintfs __tegra241_cmdqv_probe() requests the error IRQ before it has allocated the cmdqv->vintfs array and set cmdqv->num_vintfs. A CMDQV left enabled with a latched error across a kexec fires the IRQ as soon as it is requested, and tegra241_cmdqv_isr() then walks the uninitialized cmdqv->vintfs array. Request the IRQ only after cmdqv->vintfs is allocated and zeroed, so that a latched interrupt firing early runs the ISR against a valid array of NULL slots that it safely skips.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: KVM: SEV: Forcefully invalidate SNP VMSA if its backing gmem page is zapped Wire up a gmem_invalidate_range() call for SNP VMs, and use it to force vCPUs to reload/recheck their guest-provided VMSA if the backing gmem page is being invalidated, e.g. is being PUNCH_HOLE'd. Use the same core logic to handle invalidations as VMX does for the APIC-access page, as the two concepts are nearly identical: shove the physical address of a page into the vCPU's control structure: 1. Snapshot the invalidation sequence counter 2. Grab the pfn (from guest_memfd in this case) 3. Acquire mmu_lock for read 4. Re-request reload if retry is needed, otherwise commit the change. Note, the re-request action in #4 is necessary as KVM's retry logic is fuzzy, i.e. can get false positives. If the guest_memfd page has been dropped, at some point a subsequent reload will fail to get a PFN from guest_memfd, and KVM will fail KVM_RUN. If the retry was due to a false positive, KVM will retry until there are no relevant MMU notifier events (and will retry in the "outer" loop, i.e. will drop locks and resched as needed). Note #2! Take care to invalidate the VMSA when a relevant memslot is DELETED or MOVED, as invalidations in response to PUNCH_HOLE are predicated on memslot bindings (KVM doesn't know what GFN range(s) to invalidate without a binding). And more importantly, the VMSA mapping requires a memslot, i.e. must be invalidated if its memslots disappears, regardless of the state of the underlying guest_memfd inode. Failure to invalidate the vCPU's control.vmsa_pa (which is checked by pre_sev_run()) can prevent KVM from properly freeing the page as firmware will		

CVE-2026-90040	reject the RMPUPDATE to reclaim the page with FAIL_INUSE if the vCPU is actively running, i.e. if VMSA page is in-use. That in turn leads to an RMP #PF on the next use, as the page will still be assigned to the SNP VM. SEV-SNP: RMPUPDATE failed for PFN 78d198, pg_level: 1, ret: 3 SEV-SNP: PFN 0x78d198, RMP entry: [0xfff0000000144001 - 0x000000000000000f] CPU: 3 UID: 0 PID: 31345 Comm: sev_snp_vmsa_pu Tainted: G U O Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 34.86.0-102 01/25/2026 Call Trace: <TASK> dump_stack_lvl+0x54/0x70 rmpupdate+0x12c/0x140 rmp_make_shared+0x3b/0x60 sev_gmem_invalidate+0xe0/0x170 [kvm_amd] delete_from_page_cache_batch+0x1d8/0x220 truncate_inode_pages_range+0x120/0x3d0 kvm_gmem_fallocate+0x19a/0x270 [kvm] vfs_fallocate+0x1bc/0x1f0 __x64_sys_fallocate+0x48/0x70 do_syscall_64+0x10a/0x480 entry_SYSCALL_64_after_hwframe+0x4b/0x53 RIP: 0033:0x496c7e </TASK> ----------[cut here]---------- SEV: Failed to update RMP entry for PFN 0x78d198 error -14 WARNING: arch/x86/kvm/svm/sev.c:5160 at sev_gmem_invalidate+0x126/0x170 [kvm_amd], CPU#3: sev_snp_vmsa_pu/31345 CPU: 3 UID: 0 PID: 31345 Comm: sev_snp_vmsa_pu Tainted: G U O Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 34.86.0-102 01/25/2026 RIP: 0010:sev_gmem_invalidate+0x12b/0x170 [kvm_amd] Call Trace: <TASK> delete_from_page_cache_batch+0x1d8/0x220 truncate_inode_pages_range+0x120/0x3d0 kvm_gmem_fallocate+0x19a/0x270 [kvm] vfs_fallocate+0x1bc/0x1f0 __x64_sys_fallocate+0x48/0x70 do_syscall_64+0x10a/0x480 entry_SYSCALL_64_after_hwframe+0x4b/0x53 RIP: 0033:0x496c7e </TASK> irq event stamp: 20689 hardirqs last enabled at (20699): [<ffffffff8e76092c>] __console_unlock+0x5c/0x60 hardirqs last disabled at (20708): [<ffffffff8e760911>] __console_unlock+0x41/0x60 softirqs last enabled at (20722): [<ffffffff8e6cd74e>] __irq_exit_rcu+0x7e/0x140 softirqs last disabled at (20717): [<ffffffff8e6cd74e>] __irq_exit_rcu+0x7e/0x140 ---[end trace 0000000000000000]--- BUG: unable to handle page fault for address: ffff99 ---truncated---	N/A	More Details
CVE-2026-90430	In the Linux kernel, the following vulnerability has been resolved: iommu/tegra241-cmdqv: Publish an LVCMDQ only after it is fully initialized tegra241_vintf_init_lvcmdq() stores the freshly allocated vcmdq pointer to the vintf->lvcmdqs[] array, before tegra241_vcmdq_alloc_smmu_cmdq() builds the vcmdq->cmdq. The error ISR dereferences that cmdq, so a latched LVCMDQ error (e.g. one inherited across a kexec) firing in this window would make tegra241_vintf0_handle_error() pass the still-zeroed arm_smmu_cmdq down to __arm_smmu_cmdq_skip_err(), dereferencing NULL queue register pointers. Drop the store from tegra241_vintf_init_lvcmdq() and publish the vcmdq at the end of the allocation instead, with an smp_store_release() that pairs with an smp_load_acquire() in the ISR, which can see a fully built LVCMDQ or NULL. The user-owned LVCMDQ allocation moves accordingly, publishing the vcmdq once tegra241_vcmdq_hw_init_user() succeeds, using a plain store since a user VINTF's lvcmdqs[] has no lockless reader -- the error ISR only walks the VINTF0 array.	N/A	More Details
CVE-2026-90431	In the Linux kernel, the following vulnerability has been resolved: remoteproc: Prevent crash handling to race with rproc_del() There's no synchronization between rproc_crash_handler_work() and rproc_del(), as such it's possible for a driver to be removed while crash-handler work is scheduled, or even executing - resulting in use-after-free issues. To avoid this the scheduled work need to be cancelled and synchronized against before the removal proceeds. In order to ensure that this doesn't race with the reporting, and thereby scheduling new work, a "deleting" flag is introduced. This is similar to the RPROC_DELETE state that was introduced to ensure that "start" didn't race with rproc_del(), but the existing mechanism can not be used as it's valid to call rproc_report_crash() in atomic context - and the "state" is protected by a mutex. In the event that work is cancelled the pm_stay_awake() is left unbalanced and need to be unrolled. The blocking and cancelling of crash-handler work prior to the actual rproc_shutdown() call does have the explicit side-effect that crashes resulting from the shutdown process will not enter the crash-handling path, and as such will not generate devcoredumps etc. Due to the existing mutual exclusion between these code paths there's no concrete reduction in functionality, but further work would be needed to handle this case.	N/A	More Details
CVE-2026-90432	In the Linux kernel, the following vulnerability has been resolved: sched_ext: Abort directly from the hardlockup handler scx_hardlockup() defers the abort to an irq_work because exit claiming used to take scx_sched_lock and couldn't run from NMI. The deferral is now unnecessary - claiming is NMI-safe and asserting ->aborting is exactly what breaks the live-locks that hard-lock CPUs. Call handle_lockup() directly and drop the irq_work. This also makes the self-detected case recoverable: the perf watchdog fires on the hard-locked CPU itself, where a queued irq_work never runs with IRQs off. Also fix the return value: %true used to be returned whenever sched_ext was loaded, suppressing the kernel's hardlockup report even when the abort was refused. Return %true only when this call initiated the abort.	N/A	More Details
CVE-2026-90433	In the Linux kernel, the following vulnerability has been resolved: spi: oc-tiny: switch to managed controller allocation The controller is allocated with the non-managed spi_alloc_host() while the interrupt is registered with devm_request_irq(). During removal, spi_bitbang_stop() only unregisters the controller; the subsequent spi_controller_put() then frees the controller together with its embedded driver-private devdata, which is the IRQ handler's dev_id. The devm_request_irq() release action (free_irq()), which drains the handler, does not run until after .remove() returns. A late or latched interrupt can therefore reach tiny_spi_irq() and dereference already-freed memory (e.g. hw->base). Switch to devm_spi_alloc_host() so that the devres LIFO order releases the controller only after free_irq() has drained the handler, and drop the now-redundant spi_controller_put() from .remove(). The probe error path is simplified to direct returns. This issue was found by an in-house static analysis tool.	N/A	More Details
CVE-2026-90434	In the Linux kernel, the following vulnerability has been resolved: isofs: release zisofs block pointer buffer head zisofs_fill_pages() reads the compressed block pointer table. The error paths release the current buffer_head, the loop also releases the old buffer_head when it advances. However, the success path leaves the last buffer_head referenced. Release it before returning success.	N/A	More Details
CVE-2026-90039	In the Linux kernel, the following vulnerability has been resolved: NFSD: Guard admin state-revocation walks with NFSD_NET_UP Writing to /proc/fs/nfsd/unlock_filesystem, or sending the NFSD_CMD_UNLOCK_FILESYSTEM or NFSD_CMD_UNLOCK_EXPORT netlink command, walks the NFSv4 client hash tables to revoke open state and cancel async COPY operations. All three handlers gate that walk on nn->nfsd_serv, but a listener added via portlist or netlink listener_set sets nn->nfsd_serv before any nfsd thread starts. nfsd_startup_net() has not yet allocated nn->conf_id_hashtbl, so the walkers dereference a NULL table. A local administrator with CAP_SYS_ADMIN can crash the kernel this way without ever starting the server. nn->nfsd_serv is set when the service is created, which precedes table allocation. NFSD_NET_UP instead brackets the window where the tables are live: set at the end of nfsd_startup_net() and cleared in nfsd_shutdown_net() after they are freed, both under nfsd_mutex. Gating the three unlock paths on NFSD_NET_UP fixes the startup-time NULL dereference while preserving the earlier post-shutdown use-after-free fix.	N/A	More Details
CVE-2026-92476	In the Linux kernel, the following vulnerability has been resolved: crypto: keembay - Initialize completion before requesting IRQ kmb_ocs_aes_probe() requests the device IRQ before initializing irq_completion. Once the handler is registered it can run immediately, and ocs_aes_irq_handler() unconditionally calls complete(). An interrupt in this window would therefore use an uninitialized completion. Initialize the completion before requesting the IRQ, as the sibling OCS HCU and ECC drivers already do.	N/A	More Details

CVE-2026-92477	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: debugfs: Reserve space for a string terminator ufs_saved_err_write() copies user input into a zero-initialized stack buffer and passes it to kstrtoint(). A write that fills the entire buffer overwrites its only terminator. Reject an input whose length leaves no room for the trailing NUL.	N/A	More Details
CVE-2026-92478	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Validate connected lane counts The connected lane count is used by TX equalization code to index arrays sized by UFS_MAX_LANES. Reject zero and out-of-range RX or TX lane counts before they can be propagated.	N/A	More Details
CVE-2026-92479	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: Avoid NULL CQE dereference when reporting invalid tags The single-doorbell completion path can call ufshcd_compl_one_cqe() with a NULL CQE. If no command is associated with the completion tag, the warning message dereferences the CQE while reporting the error. Avoid that dereference and include the invalid tag in the warning.	N/A	More Details
CVE-2026-92480	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Validate string descriptors The string descriptor length includes a two-byte header while the UTF-16 payload starts after it. utf16s_to_utf8s() expects a count of UTF-16 code units, not bytes. Passing the payload byte count can make it read beyond the descriptor buffer. Validate that the payload has an even byte count, pass a code-unit count to the converter, and allocate sufficient UTF-8 output space. The raw string buffer starts after the descriptor header but its size is bLength. Copying bLength bytes from that pointer can read beyond the response buffer. Allocate a zeroed bLength-sized buffer and copy only the UTF-16 payload. This preserves the raw buffer size consumed by the RPMB device-ID ABI while avoiding the overread.	N/A	More Details
CVE-2026-90361	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix leak in ath11k_service_ready_ext_event() Currently, during ath11k_service_ready_ext_event() processing, svc_rdy_ext.mac_phy_caps can be allocated during TLV parsing. This is a temporary allocation that is freed on the success path, but not on the error path. If parsing succeeds far enough to allocate mac_phy_caps and then fails on a later TLV, the allocation leaks. So free the allocation on the error path. Compile tested only.	N/A	More Details
CVE-2026-90359	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject >8 byte return values on return-reading trampoline paths btf_distill_func_proto() builds the function model used for the fentry/fexit/fmod_ret/fsession trampolines and struct_ops. It has accepted a 16-byte __int128 return value since the trampoline was introduced: __get_type_size() returns the integer's type size, and the return-type check only rejected ret < 0. But the BPF trampoline preserves only 8 bytes of the return value (RAX on x86, i.e. R0). For an attach type that reads the target's return value the second half (RDX / R3) is neither saved nor restored, so a program attached to a function returning a 16-byte value corrupts the value seen by the real caller and itself observes only half of it. struct_ops trampolines have the same limitation. This affects the attach types that read the target's return value: fexit, fmod_ret and fsession (plus the _multi variants of fexit and fsession), and struct_ops. fentry/fentry_multi run before the target returns and are unaffected. Reject a >8 byte return value for these attach types in bpf_check_attach_target() and bpf_check_attach_btf_id_multi(), and for struct_ops in bpf_struct_ops_desc_init().	N/A	More Details
CVE-2026-77411	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, readLongstr in read.go returns an empty string and a nil error when a declared AMQP longstr length exceeds 0x7FFFFFFF instead of returning ErrSyntax. The function leaves the declared field bytes unread, while readTable treats the operation as successful and continues parsing from the wrong offset. A malicious or compromised broker can provide an oversized longstr in a table field and desynchronize subsequent AMQP parsing, causing attacker-controlled trailing bytes to be interpreted as later fields or frames and disrupting connection integrity and availability. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90282	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb-legacy: Fix possible NULL-deref on early runtime suspend There is a small window where the runtime suspend callback may run after pm_runtime_enable() and before pm_runtime_forbid(). In this case, a crash occurs because runtime suspend/resume dereferences qmp->phy pointer, which is not yet initialized: `if (!qmp->phy->init_count) {` This can also happen if user re-enables runtime-pm via the sysfs attribute before qmp phy is initialized. Similarly to other qcom phy drivers, introduce a qmp->phy_initialized variable that can be used to avoid relying on the possibly uninitialized phy pointer.	N/A	More Details
CVE-2026-90272	In the Linux kernel, the following vulnerability has been resolved: perf: arm_pmuv3: Zero initialize hw_id branch stack field PERF_SAMPLE_BRANCH_HW_INDEX is supported by BRBE so hw_id is passed to userspace, but it's never set by the BRBE driver. Zero initialize it as it should be according to the docs: * For the architectures whose raw branch records are * already stored in age order, the hw_idx should be 0. It's probably too risky to remove PERF_SAMPLE_BRANCH_HW_INDEX from BRBE now in case anyone is setting it and reading the value, but zero initializing the whole struct also protects against the same issue with new fields that are added in the future.	N/A	More Details
CVE-2026-90273	In the Linux kernel, the following vulnerability has been resolved: coresight: etm4x: missing cscfg_csdev_disable_active_config() in perf enable In the perf enable path, there are missing cases where cscfg_csdev_disable_active_config() is not called: - Branch broadcast is selected but not supported by the hardware - etm4_enable_hw() fails This can lead to a leak of config_desc->active_cnt. Fix this by properly calling cscfg_csdev_disable_active_config() in these error paths.	N/A	More Details
CVE-2026-90274	In the Linux kernel, the following vulnerability has been resolved: coresight: etm4x: fix underflow for usage of (nrseqstate - 1) According to IHI006H Embedded Trace Macrocell Architecture Specification[0], TRCSEQEVR<n> is implemented only when TRCIDR5.NUMSEQSTATE is 0b100, in which case n ranges from 0 to 2; otherwise, TRCIDR5.NUMSEQSTATE is 0b000. IOW, the number of usage in the initialisation or setting TRCSEQEVR<n> with drvdata->nrseqstate - 1 in the loop could make underflow issue when TRCIDR5.NUMSEQSTATE is 0b000. Therefore, introduce nr_seq_ctrls field and untie it from nrseqstate. As part of this introduce ETM_MAX_SEQ_TRANSITIONS macro and apply nr_seq_ctrls and above macro to TRCSEQEVR<n> relevant fields setup.	N/A	More Details
CVE-2026-90275	In the Linux kernel, the following vulnerability has been resolved: md/raid1: don't set array_frozen in raid1_takeover() raid1_takeover() sets conf->array_frozen = 1 on the newly-allocated r1conf and nothing ever clears it, so every I/O to the array stalls permanently once _wait_barrier() sees it stuck at 1. This used to be harmless: level_store() called mddev_resume() right after pers->run(), which called raid1_quiesce(mddev, 0) and cleared array_frozen back to 0 regardless of what raid1_takeover() set. Commit b39f35ebe86d ("md: don't quiesce in mddev_suspend()") removed that quiesce(mddev, 0) call, so the pre-set now sticks. setup_conf() already zero-initializes the new r1conf via kzalloc, so just don't set array_frozen here. Same class of bug as commit 892da88d1cd9 ("md/raid10: fix a 'conf->barrier' leakage in raid10_takeover()"), also triggered by b39f35ebe86d.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: md/md-llbitmap: stop daemon timer rearm on destroy		

CVE-2026-90276	llbitmap_destroy() deletes pending_timer before flushing md_llbitmap_io_wq. However, daemon_work can still be queued or running after the timer has been deleted, and the daemon path can arm pending_timer again when it finds dirty chunks that are not ready to flush yet. If that happens during teardown, pending_timer can remain armed after llbitmap is freed and later dereference freed memory. Add a BITMAP_SHUTDOWN bit to llbitmap->flags, set it before deleting the timer, and make the timer and daemon paths stop queueing or rearming work once teardown starts. Cancel daemon_work before flushing the shared workqueue so no already queued daemon instance can race with the free. Use timer_shutdown_sync() so a daemon instance that passed the shutdown check before teardown cannot rearm the timer afterward. BITMAP_SHUTDOWN is a runtime-only state. Mask it out when reading and updating the llbitmap superblock so the shutdown state is never loaded from disk or persisted to disk.	N/A	More Details
CVE-2026-90277	In the Linux kernel, the following vulnerability has been resolved: md/md-llbitmap: prevent create failure bitmap UAF llbitmap_create() publishes mddev->bitmap before reading the bitmap superblock. This is needed because llbitmap_read_sb() can initialize a new bitmap and flush it through helpers that use mddev->bitmap. If llbitmap_read_sb() fails, the old cleanup dropped bitmap_info.mutex and freed llbitmap before clearing mddev->bitmap. Readers such as /proc/mdstat rely on bitmap_info.mutex to keep the bitmap pointer stable while collecting bitmap stats, so they could observe the stale pointer after the failed create path released the mutex. Clear mddev->bitmap while still holding bitmap_info.mutex, then free the failed llbitmap after dropping the mutex. This makes mutex-protected readers see either a live bitmap or no bitmap.	N/A	More Details
CVE-2026-90278	In the Linux kernel, the following vulnerability has been resolved: md: wait for behind writes before destroying bitmap __md_stop() destroyed the bitmap before calling mddev_detach(). That made mddev_detach() skip bitmap_ops->wait_behind_writes(), because the bitmap was already disconnected from mddev. This was still safe for the legacy bitmap because bitmap_destroy() waits for behind writes itself. llbitmap keeps that wait in its ->wait_behind_writes() operation instead, while ->destroy() tears down the llbitmap storage. With the old ordering, RAID1 behind-write completions could still run after llbitmap storage had been freed. Call mddev_detach() before md_bitmap_destroy() so the common detach path can wait for behind writes while the bitmap is still alive. Only destroy the bitmap after those users are gone.	N/A	More Details
CVE-2026-90279	In the Linux kernel, the following vulnerability has been resolved: md/raid5: round bitmap stripes with sector division raid5_bitmap_sector_map() aligns the array range to full RAID5 stripe widths before converting it to component sectors. That width is chunk_sectors multiplied by the number of data disks, and it is not always a power of two. Reproduce with a 4-disk RAID5, 1024-sector chunks, and three data disks. The full-stripe width is 3072 sectors. For a one-sector write at array sector 3072, correct rounding gives array range [3072, 6144), which maps to component range [1024, 2048). The old round_down()/round_up() logic instead gives [1024, 4096), which maps to [0, 1024). Use sector_div() based arithmetic so the rounded range is aligned to the actual RAID5 stripe width. The deterministic mapper test now reports the fixed component range as [1024, 2048), while the old mask-based range was [0, 1024).	N/A	More Details
CVE-2026-90280	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb: Fix possible NULL-deref on early runtime suspend There is a small window where the runtime suspend callback may run after pm_runtime_enable() and before pm_runtime_forbid(). In this case, a crash occurs because runtime suspend/resume dereferences qmp->phy pointer, which is not yet initialized: `if (!qmp->phy->init_count) {` This can also happen if user re-enables runtime-pm via the sysfs attribute before qmp phy is initialized. Similarly to other qcom phy drivers, introduce a qmp->phy_initialized variable that can be used to avoid relying on the possibly uninitialized phy pointer.	N/A	More Details
CVE-2026-90281	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: snps-femto-v2: Fix possible NULL-deref on early runtime suspend Runtime PM must be enabled before creating the PHY, since phy_create() only enables runtime PM on the PHY device if it is already enabled on this parent device. However, the runtime PM callbacks dereference the hsphy instance, which is not yet ready, leaving a window where a suspend callback may trigger a NULL pointer dereference. Take a runtime PM usage reference with pm_runtime_get_noresume() before enabling runtime PM and release it once the PHY has been created, so that no runtime suspend can run before the PHY is ready. This also prevents a short window where an unnecessary runtime suspend can occur. Use the devres-managed version to ensure PM runtime is symmetrically disabled during driver removal for proper cleanup.	N/A	More Details
CVE-2026-90283	In the Linux kernel, the following vulnerability has been resolved: hugetlbfs: release subpool on fill_super failure hugetlbfs_fill_super() allocates a hugepage subpool when size or min_size mount options are specified. hugepage_new_subpool() may also reserve huge pages for min_size. If root dentry creation fails after the subpool is created, the failure path frees the subpool with kfree(). This bypasses hugepage_put_subpool() and can leave min_size reservations charged. Use hugepage_put_subpool() on the failure path, matching the normal put_super path.	N/A	More Details
CVE-2026-73173	Nozomi Networks Labs identified a CWE-306: Missing Authentication for Critical Function vulnerability in the edgserver management protocol of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a remote unauthenticated attacker to invoke critical device-management functions, including network reconfiguration, reboot, reset, and firmware upgrade, by sending crafted requests to TCP port 5058.	N/A	More Details
CVE-2026-90284	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: do not queue completed sysfs fallback requests fw_load_sysfs_fallback() calls device_add() before adding the fw_priv to pending_fw_head. device_add() publishes the fallback loading interface, so a userspace helper which discovers the device by scanning sysfs can write 0 to the loading attribute and complete the request before it is queued as pending. In that interleaving firmware_loading_store() calls fw_state_done() while pending_list still points to itself, so it cannot remove an entry from pending_fw_head. The subsequent unconditional list_add() then queues an already-completed fw_priv. Once the request is released, pending_fw_head can retain a pointer to freed memory and the next fallback request can fault while validating the list. Only in-flight fallback requests need suspend or reboot abort handling. If the request is already DONE after device_add(), return success from the fallback path without sending another uevent, waiting again, or queueing it as pending. This preserves the invariant that pending_fw_head contains only active fallback requests.	N/A	More Details
CVE-2026-90285	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Remove redundant VPD flash read in sysfs read path qla2x00_sysfs_read_vpd() called ha->isp_ops->read_optrom() a second time after releasing optrom_mutex. The repeated read is redundant and, unlike the first, runs without optrom_mutex held, exposing flash access to concurrent optrom operations. Drop the duplicate call.	N/A	More Details
CVE-2026-	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, Channel.Qos in channel.go accepts negative prefetchCount and prefetchSize integers and casts them directly to uint16 and uint32 fields in the basic.qos method because validateQos is absent. Values such as -1 therefore wrap to 65535 or 4294967295 instead of being rejected. An application that permits	N/A	More

77406	untrusted configuration of these Qos values can unintentionally request extremely large prefetch limits, allowing a broker to deliver enough queued messages to exhaust client memory and disrupt processing. This issue is fixed in version 1.13.0.		Details
CVE-2026-90287	In the Linux kernel, the following vulnerability has been resolved: phy: sunplus: fix error handling in sp_uphy_init() Fix the error paths of sp_uphy_init() to undo exactly what each stage did: return directly if clk_prepare_enable() fails, release only the clock if reset_control_deassert() fails, and jump to err_reset if update_disc_vol() fails so the clock and reset are not leaked.	N/A	More Details
CVE-2026-77405	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, tlsConfigFromURI in uri.go creates tls.Config values without setting MinVersion to tls.VersionTLS12. Builds using a Go runtime whose default permits TLS 1.0 or TLS 1.1 can therefore negotiate an obsolete protocol version when connecting through an amqps URI. A network attacker able to influence TLS negotiation with such a legacy build may weaken transport protection for AMQP messages and credentials. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-77404	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, URI.String in uri.go concatenates CertFile, KeyFile, CACertFile, and ServerName values directly into an AMQPS query string instead of encoding them as URL query parameters with url.Values. If an application accepts a TLS asset path containing ampersand or equals delimiters and later reparses the serialized URI with ParseURI, the embedded delimiters can create or overwrite connection options, including paths to TLS certificate, key, or CA files. This can corrupt connection configuration or select unintended local cryptographic assets. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90290	In the Linux kernel, the following vulnerability has been resolved: arm64: hibernate: Restore DAIF state on error Sashiko AI has reported that if swsusp_mte_save_tags() for some reason fails we return from swsusp_arch_suspend() with DAIF being masked - that is not what we'd expect. Restore the saved DAIF state before returning from the error path.	N/A	More Details
CVE-2026-77403	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, Connection.openTune in connection.go accepts a server-advertised FrameMax below the AMQP frameMinSize value of 4096 bytes because the connection negotiation loop does not enforce the protocol minimum. A malicious or compromised AMQP broker can therefore advertise an extremely small FrameMax, causing later client publications to be fragmented into excessive numbers of frames and write operations. This can consume CPU and stall the client or its host. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90295	In the Linux kernel, the following vulnerability has been resolved: cpufreq: imx6q: fix out-of-bounds write when probed more than once imx6_soc_volt is allocated fresh on every probe, sized to the number of ARM OPPs: imx6_soc_volt = devm_kcalloc(cpu_dev, num, sizeof(*imx6_soc_volt), GFP_KERNEL); but it is filled through soc_opp_count, which has static storage and is never reset. A second bind after an unbind keeps indexing from where the first one stopped, and writes past the end of the new array. Unbinding and rebinding the driver on qemu's mcimx6ul-evk, under KASAN: BUG: KASAN: slab-out-of-bounds in imx6q_cpufreq_probe+0x3b0/0xa34 Write of size 4 at addr c5e90480 by task binder/73 imx6q_cpufreq_probe from platform_probe+0x88/0xe4 platform_probe from really_probe+0x108/0x384 bind_store from kernfs_fop_write_iter+0x1b4/0x28c The write lands one u32 past the end of the allocation. soc_opp_count is only read a few lines below the loop that fills it, so it never needed static storage. Make it a local.	N/A	More Details
CVE-2026-90296	In the Linux kernel, the following vulnerability has been resolved: cpufreq: imx6q: fix devres accumulation across driver rebind imx6_soc_volt is allocated with devm_kcalloc(cpu_dev, ...), where cpu_dev is the CPU device from get_cpu_device(0). That device is never unbound, so its devres list is never released, and imx6q_cpufreq_remove() does not free the array either. Every probe therefore adds an allocation that stays for the lifetime of the system. Allocate against the platform device instead. Its devres is released when the driver is unbound, which is exactly the lifetime the array wants: imx6q_set_target() reads it, and nothing may reach that after cpufreq_unregister_driver(). That makes the array actually go away on unbind, so also clear the file-scope pointer in remove and on the failed-probe path, rather than leave it pointing at memory devres is about to release. Tested by rebinding the driver on qemu's mcimx6ul-evk.	N/A	More Details
CVE-2026-90271	In the Linux kernel, the following vulnerability has been resolved: arm_mpam: Fix a NULL pointer dereference on unbinding after an error interrupt If a user unbinds an MSC after mpam_disable() has been run in response to an error interrupt then a dereference of a NULL pointer occurs as mpam_disable() sets the drvdata to NULL. Add an early return to the driver remove callback to avoid this.	N/A	More Details
CVE-2026-90270	In the Linux kernel, the following vulnerability has been resolved: arm_mpam: Disable driver unbind to avoid UAF When a user unbinds an MSC and that MSC is the only MSC left for a component then the corresponding mpam_component will be freed. If the user then goes on to read the schemata file in the resctrl filesystem then the mpam_component will be accessed from resctrl_arch_get_config() leading to a use after free. As the MPAM driver is not a module the unbind sysfs interface is the only way to trigger the remove. Instead of dealing with the complexity of allowing some unused MSC to unbind just remove the unbind sysfs interface.	N/A	More Details
CVE-2026-90269	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject load-acquire from pointers requiring fault protection A BPF_LOAD_ACQ is not rewritten to a BPF_PROBE_MEM load by the verifier, unlike a regular BPF_LDX, so the JIT emits a plain load with no exception table entry and a fault panics the kernel instead of being handled. Reject the source pointer types that a BPF_LDX would have had that fault protection applied to, i.e. the ones bpf_convert_ctx_accesses() turns into BPF_PROBE_MEM: a bare PTR_TO_BTF_ID, PTR_TO_BTF_ID | PTR_UNTRUSTED, PTR_TO_BTF_ID | MEM_ALLOC | PTR_UNTRUSTED and PTR_TO_MEM | MEM_RDONLY | PTR_UNTRUSTED. This is reachable e.g. by loading ->mm out of a trusted task_struct yields an untrusted pointer to mm_struct, and it is NULL for a kernel thread: [...] SEC("tp_btf/sched_switch") int BPF_PROG(demo, bool preempt, struct task_struct *prev, struct task_struct *next) { struct mm_struct *mm = next->mm; /* untrusted */ out_ldx = (__u64)mm->pgd; /* BPF_LDX */ out_acq = load_acquire(&mm->pgd); /* BPF_LOAD_ACQ */ return 0; } [...] Both dereference the same pointer, but only the BPF_LDX is protected (x86-64 JIT, jump targets shown prog-relative): [...] ; out_ldx = (__u64)mm->pgd; 17: movq $-10485760, %r10 1e: movq %rsi, %r11 21: addq $184, %r11 28: subq %r10, %r11 2b: movabsq $140737498841088, %r10 35: cmpq %r10, %r11 38: ja 0x3e <-- kernel addr? 3a: xorl %edi, %edi <-- no: dst = 0, skip the load 3c: jmp 0x45 3e: movq 184(%rsi), %rdi <-- yes: load + extable entry [...] ; load_acquire(&mm->pgd) 53: movq %rsi, %rdi 56: movq 184(%rdi), %rax <-- no check, no extable entry [...] Note that BPF_PROBE_MEM is not visible in a bpftool xlated dump, as bpf_insn_prepare_dump() rewrites it back to BPF_MEM. A PTR_TRUSTED pointer is deliberately not on the list. Such a load is not converted either, but it does not need to be, since the pointer is guaranteed live, so load-acquire from it stays allowed. The check is gated on BPF_LOAD_ACQ so that atomic RMW and store-release error messages are unchanged; writes (RMW / store-release) to such pointers are already rejected elsewhere, so only load-acquire needs this.	N/A	More Details

CVE-2026-77407	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, PlainAuth values defined in auth.go retain passwords as exported plaintext fields in Connection.Config.SASL after a successful PLAIN authentication handshake. The Connection.openComplete method in connection.go does not clear those values. Code with access to the Connection object, including reflective loggers, application performance monitoring agents, debugging utilities, and panic handlers, can traverse the configuration and expose the credentials to logs or state captures. The credential remains available for the lifetime of the connection instead of being cleared after authentication. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90247	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix mmap_lock leak in irq_work path stack_map_get_build_id_offset() introduced a per-CPU irq_work to defer mmap_read_unlock() from NMI context, and bpf_find_vma() later reused the same mmap_unlock_work. Both callers only check whether the work is busy before taking mmap_lock, so a nested caller can reuse the slot before the first caller queues it. Two read locks may then be acquired while only one deferred unlock runs, leaking a read lock and blocking exit_mmap(). Reserve the per-CPU slot before mmap_read_trylock(). Use the same wrapper in stackmap and bpf_find_vma() so both callers release the reservation on trylock failure. Keep rejecting the slot while the irq_work remains busy. Release it after the irq_work callback unlocks the mm.	N/A	More Details
CVE-2026-90248	In the Linux kernel, the following vulnerability has been resolved: net/sched: cls_api: fix teardown of an adopted proto on insert-race loss In tc_new_tfilter() the create branch sets tp_created = 1 before calling tcf_chain_tp_insert_unique(). When the caller loses the race (another request inserted a proto at the same chain/prio first), insert_unique() destroys the caller's own tp_new and returns the winner's proto with an extra reference. tp_created was never cleared, so the loser's errout path treated the winner's live proto as its own and called tcf_chain_tp_delete_empty() on it, silently unlinking an active classifier that the winning request already advertised via RTM_NEWTFILTER. Track the outcome of the insert step in a single tri-state variable so each errout path reacts correctly: - TP_NOT_CREATED: no proto created; pursue the old path. - TP_CREATED: proto inserted successfully; same code path as before. - TP_NOT_OWNED: New - lost the insert race; tp is another request's proto (chain ref already released by tp_new's destroy) Both errout reactions are single expressions derived from the state. This fix is motivated by the Sashiko's automated review of Patch (net/sched: cls_api: Always acquire rtnl_lock when destroying locked classifiers) [1] [2]. The review identified the silent-unlink behaviour of an adopted proto's teardown when a request loses the tcf_chain_tp_insert_unique() race. [1] https://sashiko.dev/#/patchset/20260801125632.360365-1-jhs%40mojatatu.com [2] https://netdev-ai.bots.linux.dev/sashiko/#/patchset/20260801125632.360365-1-jhs%40mojatatu.com	N/A	More Details
CVE-2026-90249	In the Linux kernel, the following vulnerability has been resolved: iio: light: gp2ap002: Fix unbalanced runtime PM on repeated event writes The IIO core does not filter duplicate writes to the event enable attribute, so writing the same value twice invokes write_event_config() twice. Enabling twice leaks a runtime PM reference, preventing the device from ever suspending again; disabling twice underflows the usage count and triggers a "Runtime PM usage count underflow" warning. Bail out early when the requested state matches the current state. While at it, switch to pm_runtime_resume_and_get() so a failed resume is propagated to userspace instead of silently marking the event enabled.	N/A	More Details
CVE-2026-90250	In the Linux kernel, the following vulnerability has been resolved: bpf, cgroup: Fix storage null-ptr-deref after replacing prog Syzkaller reported a storage null-ptr-deref issue after replacing prog. This occurs in the following scenario: 1. prog A, an empty prog, is attached to a cgrp. 2. prog B uses BPF_MAP_TYPE_PERCPU_CGROUP_STORAGE and calls the bpf_get_local_storage helper. 3. link_update is called to replace prog A with prog B. The reason is that __cgroup_bpf_replace fails to alloc and assign the required cgrp storage for the incoming replacement prog. Consequently, the new prog inherits an uninit storage, leading to null-ptr-deref panic when kick the new prog. Fix this by rejecting a link update if new_prog's cgroup storage is incompatible with link->prog.	N/A	More Details
CVE-2026-90251	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MSFT: validate evt_prefix_len against the response length read_supported_features() only checks that the response covers the fixed part of struct msft_rp_read_supported_features, which is 11 bytes: if (skb->len < sizeof(*rp)) { bt_dev_err(hdev, "MSFT supported features length mismatch"); goto failed; } evt_prefix[] is a flexible array member and rp->evt_prefix_len is an unvalidated u8 taken straight out of that response, so msft->evt_prefix = kmemdup(rp->evt_prefix, rp->evt_prefix_len, GFP_KERNEL); copies up to 255 bytes from a reply that may have carried none of them. What is copied is data the controller never sent, and it is then used to match incoming vendor events in msft_vendor_evt(). This is not an out-of-bounds access. An skb data allocation always has at least SKB_DATA_ALIGN(sizeof(struct skb_shared_info)) bytes past the payload, which is more than the 255 byte maximum, so the read stays inside the allocation and KASAN does not report it. It is still a read of bytes the host was never given, with the length fully controlled by the controller. Reject a response that is too short for the prefix it declares. Verified with an emulated controller over /dev/vhci on a KASAN kernel, with vhci made to advertise an MSFT opcode the way btintel, btqca, btmtk and btrtl do unconditionally. A reply of exactly 11 bytes declaring evt_prefix_len = 255 reaches kmemdup and copies 255 bytes ("skb->len=11 evt_prefix_len=255", with the copied buffer dumped); since the reply ends at the fixed part, all 255 come from past the end of the response. No KASAN report is produced, as expected from the allocation slack described above. With this patch the response is rejected with "MSFT event prefix length mismatch" and msft->evt_prefix is left unset.	N/A	More Details
CVE-2026-90252	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: free the HCI command when it is cancelled mgmt_hci_cmd_sync() queues the pending command with a NULL destroy callback, so it is only freed if send_hci_cmd_sync() runs. A cancelled entry is leaked, as _hci_cmd_sync_cancel_entry() does not release entry->data when there is no destroy callback, and hci_cmd_sync_clear() cancels every pending entry when the controller is unregistered. Nothing else reclaims it either: mgmt_pending_new() does not put the command on hdev->mgmt_pending. The leak also pins the socket reference taken by mgmt_pending_new(), so the mgmt socket is never released. Free the command from a destroy callback. The now-empty done label is replaced by a direct return.	N/A	More Details
CVE-2026-90253	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: free the mesh send cancel command when it is cancelled mesh_send_cancel() queues the pending command with a NULL destroy callback, so it is only freed if send_cancel() runs. A cancelled entry is leaked, as _hci_cmd_sync_cancel_entry() does not release entry->data when there is no destroy callback, and hci_cmd_sync_clear() cancels every pending entry when the controller is unregistered. Nothing else reclaims it either: mgmt_pending_new() does not put the command on hdev->mgmt_pending. The leak also pins the socket reference taken by mgmt_pending_new(), so the mgmt socket is never released. Free the command from a destroy callback.	N/A	More Details
CVE-2026-90254	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: free the advertising instance on the failure and cancel paths adv_timeout_expire() hands a kmalloc()ed instance byte to hci_cmd_sync_queue() with a NULL destroy callback, and only adv_timeout_expire_sync() frees it. That leaks on two paths: - the return value is not checked, and hci_cmd_sync_queue() does not take ownership when it fails (-ENETDOWN, -ENODEV, -ENOMEM); - a cancelled entry is not released, as _hci_cmd_sync_cancel_entry() does not free entry->data when there is no destroy callback. hci_cmd_sync_clear()	N/A	More Details

	cancels every pending entry when the controller is unregistered. Free the buffer from a destroy callback, and in the caller when the entry could not be queued at all.		
CVE-2026-77410	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, Channel.recvContent in channel.go preallocates the message body slice with the uint64 ch.header.Size value supplied by an AMQP content header without capping the allocation to the negotiated Connection.Config.FrameSize value. A malicious or compromised broker can send an extreme declared body size and cause the Go runtime to attempt a correspondingly large allocation before body data is received. The allocation can exhaust memory and terminate the client process. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-77409	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, Channel.dispatch in channel.go, confirms.confirm in confirms.go, and Connection.dispatch0 in connection.go synchronously send publisher confirmations, flow-control events, consumer cancellations, returned messages, including NotifyConfirm events and connection block notifications, to application-provided channels. If a listener channel is unbuffered, full, or not drained promptly, the sole reader goroutine blocks and stops processing frames, acknowledgments, deliveries, and heartbeats. Broker-driven event bursts can therefore cause connection stalls, missed heartbeats, deadlocks, and disconnection. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90257	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: virtio_bt: avoid OOB read of build info string The virtbt_setup_zephyr() sends the Zephyr vendor command 0xfc08 (Read Build Information) and hands the response to bt_dev_info() and hci_set_fw_info() as a "%s" string starting at skb->data + 1, without checking the length. A backend that answers with status only leaves that pointer past the end of the received data, so the walk reads adjacent slab memory until it meets a NUL. Those bytes reach the kernel log and the firmware-info debugfs file. To fix this, print the string with a bounded "%.*s" limited to skb->len - 1. A short or unterminated response then prints as much as arrived instead of failing setup. This mirrors commit dd068ef04412 ("Bluetooth: bpa10x: avoid OOB read of revision string in bpa10x_setup()"), which fixed the identical pattern.	N/A	More Details
CVE-2026-90258	In the Linux kernel, the following vulnerability has been resolved: pinctrl: airoha: add missed IRQ resource helpers Without hooking .irq_request_resources, gpiolib cannot set GPIOD_FLAG_USED_AS_IRQ. This breaks pin direction locking and can allow userspace or another driver to reconfigure an active IRQ pin as an output	N/A	More Details
CVE-2026-90259	In the Linux kernel, the following vulnerability has been resolved: btrfs: qgroup: fix a wrong length calculation in qgroup_free_reserved_data() In that function, we round down the start position and round up the ending position. But during the calculation of @len, we use "round_up(start + len, sectorsize)", which is the rounded up end position, not the rounded up length. Which results a much larger length, and later we are still using "start + len", which is completely incorrect. Fix it by declaring a local @aligned_start and @aligned_len and use them instead.	N/A	More Details
CVE-2026-77408	RabbitMQ amqp091-go is a Go AMQP 0.9.1 client. Prior to 1.13.0, the writeShortstr function in write.go casts the byte length of AMQP shortstr property values to uint8 without first rejecting values longer than 255 bytes. An application that accepts an oversized CorrelationId, ReplyTo, MessageId, Expiration, UserId, AppId, ContentType, ContentEncoding, or Type value can therefore serialize a wrapped length and only a truncated prefix, while reporting no error. The resulting silent metadata corruption can break request and reply correlation, routing, tracing, and downstream message processing. This issue is fixed in version 1.13.0.	N/A	More Details
CVE-2026-90261	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: flush active metadata block group at btree_writepages() start btree_writepages() writes the btree inode's dirty metadata in ascending logical address order. On a zoned filesystem only one metadata and one system block group is active for writing at a time, and check_bg_is_active() (via btrfs_check_meta_write_pointer()) pivots the active block group as writeback moves from one block group to the next. If the active block group sits at a higher logical address than another block group that also holds dirty metadata, the ascending walk reaches the lower one first and, to write it, has to finish the active block group and activate the lower one. It cannot finish a block group that still has unsent IO, and during WB_SYNC_ALL && !for_sync (commit) writeback it deliberately refuses to wait for that IO under fs_info->zoned_meta_io_lock, as that can deadlock. The pivot thus cannot issue the submission itself either, so it gives up: btrfs_check_meta_write_pointer() returns -EAGAIN, which btrfs_write_and_wait_transaction() treats as fatal and aborts the transaction, forcing the filesystem read-only. This happens intermittently under metadata-heavy relocation (e.g. fstests btrfs/187). Flush the active metadata and system block groups at the start of btree_writepages(), under the fs_info->zoned_meta_io_lock it already holds, so they have no unsent IO left and the later pivot can finish them and make forward progress.	N/A	More Details
CVE-2026-90262	In the Linux kernel, the following vulnerability has been resolved: btrfs: retry verity reads for not-uptodate Merkle folios btrfs_read_merkle_tree_page() can find a folio in the mapping that is not uptodate. After taking the folio lock, the current code treats that state as a read error and returns -EIO. That can make a previous transient read failure sticky. If the failed read left a not-uptodate folio in the mapping, later callers find that folio and fail instead of retrying the read. Keep the existing page-cache insertion and locking order, but retry the Merkle item read when a not-uptodate folio is found in the mapping. Also unlock the folio when read_key_bytes() fails so that a later caller can lock it and retry the read.	N/A	More Details
CVE-2026-90263	In the Linux kernel, the following vulnerability has been resolved: btrfs: check if root is readonly when setting posix acl For a filesystem which has btrfs read-only property set to true, all write operations including acl and xattr should be denied. However, acl can still be set even if btrfs ro property is true. This happens because no function on the set_acl code path checks the root is readonly or not. It was checked in btrfs_setxattr_trans() but got removed in commit 353c2ea735e4 ("btrfs: remove redundant readonly root check in btrfs_setxattr_trans") That commit didn't check if all the callers properly check the root's read-only flag. A previous fix is commit b51111271b03 ("btrfs: check if root is readonly while setting security xattr"). Always check if the root is read-only before performing the set acl operation.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: btrfs: always wait for ordered extents to avoid OE races [BUG] Syzbot reported a bug that there can be conflicting OEs for the same range: BTRFS critical (device loop4): panic in insert_ordered_extent:264: overlapping ordered extents, existing oe file_offset 16384 num_bytes 430080 flags 0x1089, new oe file_offset 16384 num_bytes 430080 flags 0x80 (errno=-17 Object alrea[179.162726][T6897] BTRFS critical (device loop4): panic in insert_ordered_extent:264: overlapping ordered extents, existing oe file_offset 16384 num_bytes 430080 flags 0x1089, new oe file_offset 16384 num_bytes 430080 flags 0x80 (errno=-17 Object already exists) ------------[cut here]------------ kernel BUG at fs/btrfs/ordered-data.c:264! Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/09/2026 RIP: 0010:btrfs_alloc_ordered_extent+0x943/0xad0 Call Trace: <TASK> cow_file_range+0x744/0x12a0 fallback_to_cow+0x5ea/0xa00 run_delalloc_nocow+0x110c/0x17a0 btrfs_run_delalloc_range+0xbe4/0x1c20		

CVE-2026-90264	writepage_delalloc+0x104d/0x1ba0 btrfs_writepages+0x1667/0x28b0 do_writepages+0x338/0x560 filemap_fdatawrite_range+0x1f2/0x300 btrfs_fdatawrite_range+0x54/0xf0 btrfs_direct_write+0x6a0/0xc30 btrfs_do_write_iter+0x329/0x790 do_iter_readv_writev+0x624/0x8d0 vfs_writev+0x34c/0x990 __se_sys_pwritev2+0x17a/0x2a0 do_syscall_64+0x174/0x580 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> ---[end trace 0000000000000000]--- [CAUSE] Since commit ff66fe666233 ("btrfs: fix incorrect buffered IO fallback for append direct writes"), if the direct IO finished short, we will revert the isize back to the original one, so that append writes can be respected during the buffered fallback. Normally we rely on lock_and_cleanup_extent_if_need() function during buffered writeback to wait for any existing ordered extents. But that ordered extent waiting only happens if the start_pos is inside the isize. Since we have reverted the isize during failed direct IO, we will not wait for any ordered extents. This means we can have a race where the direct IO OE is still in the tree, finished but not yet removed, then we're inserting the OE for the buffered write, causing the above crash. [FIX] Make the OE wait to be unconditional, to handle the reverted isize situation. And since lock_and_cleanup_extent_if_need() now either lock the extents or return -EAGAIN, also remove the branches that handles no-extent-locked cases, and rename it to remove the "_if_need" suffix. The following micro benchmark shows the runtime difference for btrfs_buffered_write(), doing `xfs_io -f -c "pwrite 0 1m"` workload, all values are the average runtime in nano seconds. function runtime | before | after ----------+------------+------------ lock_and_cleanup_extent_if_need() | 58.2 | 183.0 btrfs_buffered_write() | 2115.6 | 2973.3 The overall runtime of btrfs_buffered_write() is still pretty tiny (still less than 3 micro seconds), I'd say the extra cost is still acceptable. An alternative to fix this problem is to wait ordered extents during iomap_end() where the isize revert is done. But that solution will break nowait requirement, as if a nowait direct IO finished short, we have to wait for the OEs unconditionally or the next append buffered IO can still hit the same problem. So here we have to move the wait cost to buffered write, but at least the code is slightly more streamline.	N/A	More Details
CVE-2026-90265	In the Linux kernel, the following vulnerability has been resolved: btrfs: defrag: fix deadlock between defrag and delalloc space reservation While running fsstress with autodefrag and flushoncommit, hit a deadlock due to the fact that defrag reserves delalloc space while it's holding dirty and locked folios, besides the extent range lock. The stack traces are the following: [958.624] task:kworker/u50:3 state:D stack:0 pid:20365 tgid:20365 ppid:2 task_flags:0x4208060 flags:0x00080000 [958.626] Workqueue: events_unbound btrfs_async_reclaim_metadata_space [btrfs] [958.627] Call Trace: [958.628] <TASK> [958.628] __schedule+0x4be/0x10f0 [958.629] ? preempt_count_add+0x69/0xa0 [958.630] schedule+0x26/0xd0 [958.631] wait_current_trans+0x102/0x160 [btrfs] [958.632] ? __pfx_autoremove_wake_function+0x10/0x10 [958.633] start_transaction+0x374/0x900 [btrfs] [958.634] btrfs_commit_current_transaction+0x1d/0x70 [btrfs] [958.635] flush_space+0xca/0x5e0 [btrfs] [958.636] ? _raw_spin_unlock+0x15/0x30 [958.637] ? btrfs_reduce_alloc_profile+0x8c/0x190 [btrfs] [958.639] ? _raw_spin_unlock+0x15/0x30 [958.640] ? calc_available_free_space.isra.0+0x6f/0x110 [btrfs] [958.641] do_async_reclaim_metadata_space+0x84/0x190 [btrfs] [958.642] btrfs_async_reclaim_metadata_space+0x64/0x80 [btrfs] [958.644] process_one_work+0x19d/0x3a0 [958.644] worker_thread+0x1c4/0x330 [958.645] ? __pfx_worker_thread+0x10/0x10 [958.646] kthread+0xfc/0x130 [958.647] ? __pfx_kthread+0x10/0x10 [958.648] ret_from_fork+0x1f7/0x2c0 [958.648] ? __pfx_kthread+0x10/0x10 [958.649] ret_from_fork_asm+0x1a/0x30 [958.650] </TASK> [958.651] task:kworker/u49:7 state:D stack:0 pid:52990 tgid:52990 ppid:2 task_flags:0x4208060 flags:0x00080000 [958.653] Workqueue: writeback wb_workfn (flush-btrfs-334) [958.655] Call Trace: [958.655] <TASK> [958.656] __schedule+0x4be/0x10f0 [958.657] ? __blk_flush_plug+0xe9/0x140 [958.658] schedule+0x26/0xd0 [958.658] io_schedule+0x42/0x70 [958.659] folio_wait_bit_common+0x12b/0x330 [958.660] ? folio_wait_bit_common+0x100/0x330 [958.662] ? __pfx_wake_page_function+0x10/0x10 [958.663] extent_write_cache_pages+0x599/0x830 [btrfs] [958.664] ? acpi_fwnode_get_reference_args+0x1fa/0x270 [958.665] btrfs_writepages+0x77/0x130 [btrfs] [958.666] ? __pfx_end_bbio_data_write+0x10/0x10 [btrfs] [958.667] do_writepages+0xc6/0x160 [958.668] __writeback_single_inode+0x42/0x310 [958.669] writeback_sb_inodes+0x231/0x570 [958.670] wb_writeback+0x8a/0x340 [958.671] wb_workfn+0xbf/0x450 [958.672] ? finish_task_switch.isra.0+0xc1/0x350 [958.673] process_one_work+0x19d/0x3a0 [958.673] worker_thread+0x1c4/0x330 [958.674] ? __pfx_worker_thread+0x10/0x10 [958.675] kthread+0xfc/0x130 [958.676] ? __pfx_kthread+0x10/0x10 [958.676] ret_from_fork+0x1f7/0x2c0 [958.677] ? __pfx_kthread+0x10/0x10 [958.678] ret_from_fork_asm+0x1a/0x30 [958.679] </TASK> [958.679] task:btrfs-cleaner state:D stack:0 pid:296750 tgid:296750 ppid:2 task_flags:0x208040 flags:0x00080000 [958.681] Call Trace: [958.682] <TASK> [958.682] __schedule+0x4be/0x10f0 [958.683] schedule+0x26/0xd0 [958.684] handle_reserve_ticket+0x1b9/0x2c0 [btrfs] [958.685] ? __pfx_autoremove_wake_function+0x10/0x10 [958.686] reserve_bytes+0x283/0x4c0 [btrfs] [958.687] btrfs_reserve_metadata_bytes+0x18/0xb0 [btrfs] [958.688] btrfs_delalloc_reserve_metadata+0x121/0x320 [btrfs] [958.690] btrfs_delalloc_reserve_space+0x46/0xb0 [btrfs] [958.691] btrfs_defrag_file+0x903/0x1110 [btrfs] [958.692] btrfs_run_defrag_inodes+0x334/0x430 [btrfs] [958.694] cleaner_kthread+0x97/0x1c0 [btrfs] [958.694] ? __pfx_cleaner_kthread+0x10/0x10 [btrfs] [958.696] kthread+0xfc/0x130 [958.696] ? __pfx_kthread+0x10/0x10 [958.697] ret_ ---truncated---	N/A	More Details
CVE-2026-90266	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: don't force read-only on transient -EAGAIN from reloc merge On a zoned FS, btrfs_delayed_refs_rsv_refill() returns -EAGAIN whenever the over-committed metadata plus the zone_unusable bytes exceeds the usable size in a metadata block-group to avoid heavy over-commit of metadata and early ENOSPC in one transaction. If this happens while doing reclaim, the transaction is getting aborted. Treat -EAGAIN as a soft, retryable condition in case of block-group reclaim.	N/A	More Details
CVE-2026-90267	In the Linux kernel, the following vulnerability has been resolved: scsi: sd: Fix special_vec mempool leak when scsi_alloc_sgtables() fails sd_set_special_bvec() allocates a special payload page for UNMAP and WRITE SAME commands. If scsi_alloc_sgtables() fails afterward in sd_setup_unmap_cmnd() or sd_setup_write_same{10,16}_cmnd(), the SCSI midlayer does not call uninit_command() because RQF_DONTPREP is not set yet, leaking the page. Call sd_uninit_command() on error, and clear RQF_SPECIAL_PAYLOAD after freeing the page.	N/A	More Details
CVE-2026-90297	In the Linux kernel, the following vulnerability has been resolved: drm/sun4i: crtc: Propagate layer initialization error sun4i_crtc_init() returns plain NULL when layer initialization fails, while all its other error paths return an error pointer. The only caller, sun4i_tcon_bind(), checks the result with IS_ERR() and happily continues with tcon->crtc set to NULL. sun4i_rgb_init() and sun4i_lvds_init() then dereference it in drm_crtc_mask(), which oopses. Return the error pointer instead.	N/A	More Details
CVE-2026-90298	In the Linux kernel, the following vulnerability has been resolved: drm/sun4i: tcon: Drop TCON TOP device reference of_find_device_by_node() takes a device reference. Drop it after mux configuration succeeds.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix sleepable check for tracing/lsm prog When CONFIG_FUNCTION_ERROR_INJECTION is disabled, a sleepable tracing prog is allowed to attach to '__x64_'-alike prefix symbols. It is because the verifier does not verify whether the symbol is a kernel function or a bpf prog. That said, a sleepable tracing		

2026-90299	prog is allowed to attach to a bpf prog target whose name has '__x64_'-alike prefix. For example, a sleepable fentry prog attaches to a '__x64_sys_nop' XDP prog, and copies buffer from a user pointer with bpf_copy_from_user() helper. After attaching the XDP prog to lo interface, the kernel BUG could be triggered by 'ping -c 1 -W 1 127.0.0.1': [3.460756] BUG: sleeping function called from invalid context at kernel/bpf/trampoline.c:1324 Fix it by disallowing sleepable prog always when its target btf is not a kernel's btf.	N/A	More Details
CVE-2026-90334	In the Linux kernel, the following vulnerability has been resolved: tty: clear cdev pointer after cdev_add() failure tty_cdev_add() drops the cdev reference when cdev_add() fails, but leaves driver->cdevs[index] pointing to freed memory. tty_unregister_device() later passes that stale pointer to cdev_del(), causing a use-after-free. Clear the slot after dropping the reference.	N/A	More Details
CVE-2026-90336	In the Linux kernel, the following vulnerability has been resolved: serial: core: clear freed pointers on uart_register_driver() failure uart_register_driver() leaves drv->state pointing to freed memory when tty_alloc_driver() fails. If tty_register_driver() fails, drv->tty_driver also retains a pointer after its reference is dropped. Drivers that use drv->state as an "already registered" flag can then skip registration on the next probe and pass the freed state to uart_add_one_port(). This issue was found with failslab on QEMU's raspi1ap board by failing registration and binding the PL011 port again. Clear both pointers on their failure paths, as uart_unregister_driver() already does.	N/A	More Details
CVE-2026-90337	In the Linux kernel, the following vulnerability has been resolved: serial: core: do fallible allocations before the console can be registered serial_core_add_one_port() allocates uport->tty_groups after uart_configure_port(), which may register the console. If the allocation fails, the driver unwinds the port while its console remains registered. The earlier uport->name allocation has a related failure path that leaves state->uart_port linked to a port being freed. Failslab reproduced a NULL dereference in PL011 console output and a KASAN use-after-free in i.MX console output after failed binds. Allocate the name and tty_groups before linking the port and configuring it. Reserve space for the optional driver attribute group because config_port() may populate uport->attr_group during configuration.	N/A	More Details
CVE-2026-90338	In the Linux kernel, the following vulnerability has been resolved: serial: amba-pl011: keep console clock enabled for atomic writes pl011_console_write_atomic() runs from nbcon atomic context, where sleeping is not allowed. It calls clk_enable(), which takes the common-clk enable_lock. Under PREEMPT_RT that is a sleeping lock: clk_enable_lock() first tries spin_trylock_irqsave(), but on contention falls back to spin_lock_irqsave(). Therefore, an atomic-context printk on an RT kernel with a clk-backed pl011 can trip: BUG: sleeping function called from invalid context at spinlock_rt.c:48 __might_resched from rt_spin_lock rt_spin_lock from clk_enable_lock clk_enable_lock from clk_enable clk_enable from pl011_console_write_atomic ... from vprintk_emit This was found and reproduced on PREEMPT_RT. Arm32 and arm64 DT SoCs are affected; arm64 SBSA/ACPI has no clk, so clk_enable(NULL) short-circuits before the lock. In addition, write_atomic() may be invoked from NMI context and is documented to avoid locking. Removing clk_enable() from the callback also avoids a potentially unsafe NMI acquisition of the common-clock enable_lock. An nbcon atomic-capable console must be printable from any context, so the clock cannot be gated between writes. Enable the clock while the console is available for output: use clk_prepare_enable() in pl011_console_setup(), release it via clk_disable_unprepare() in the console .exit() callback, and drop the per-write clk_enable()/clk_disable() pairs from write_atomic() and write_thread(). When printk suspends consoles, drop the reference after uart_suspend_port() stops console access and restore it before uart_resume_port() -- but only if suspend actually marked the port suspended (a wake-capable tty stays running and must keep its clock), and keep it when console_suspend_enabled is false so no_console_suspend works. The active power cost of keeping the clock enabled is platform-dependent: none where the UART clock is a fixed always-on oscillator, real where it is a gateable clock branch, which then cannot be gated (nor possibly can its parent clocks) while the console is available for output. When serial core actually suspends the port, the reference is released so the clock provider can gate the clock tree.	N/A	More Details
CVE-2026-90339	In the Linux kernel, the following vulnerability has been resolved: powerpc/syscall: Fix syscall skip handling for seccomp and ptrace After enabling GENERIC_ENTRY on PowerPC, syscall_enter_from_user_mode() returns -1 as a sentinel to signal that seccomp or ptrace has intercepted the syscall and already set a return value via syscall_set_return_value(). system_call_exception() was not handling this sentinel, and since -1UL is >= NR_syscalls, the code fell into the out-of-range path and returned -ENOSYS, overwriting the errno already placed in regs->gpr[3]. The naive fix of checking r0 == -1L before the NR_syscalls bounds check is ambiguous: a user legitimately calling syscall(-1) also produces r0 == -1L, and a tracer intercepting such a call would have its injected return value silently discarded. Fix this by introducing a thread flag that is set whenever syscall_set_return_value() explicitly updates the return value. In system_call_exception(), check and clear this flag before dispatching the syscall, and return the preset value directly when it is present. This ensures that an explicitly supplied return value always suppresses syscall execution, regardless of the syscall number. This handles all seccomp actions correctly: - SECCOMP_RET_ERRNO, SECCOMP_RET_TRACE (no tracer), SECCOMP_RET_USER_NOTIF: all call syscall_set_return_value(), flag is set, injected value returned. - SECCOMP_RET_TRAP, SECCOMP_RET_KILL: call syscall_rollback() and deliver a signal; flag is not set, but the process is dying so the return value is irrelevant. The fix covers both ppc32 and ppc64 with no #ifdefs.	N/A	More Details
CVE-2026-90340	In the Linux kernel, the following vulnerability has been resolved: pinctrl: generic: free maps on pinctrl_generic_to_map() failure pinctrl_generic_to_map() parses DT configuration and allocates pinctrl maps via pinctrl_utils_reserve_map(). If subsequent steps (such as pinctrl_utils_add_map_mux(), pinctrl_generic_add_group(), pinconf_generic_parse_dt_config(), or pinctrl_utils_add_map_configs()) return an error, *maps may contain partially allocated map entries. Returning the error directly without freeing *maps leaks the allocated mapping memory across all drivers that rely on pinctrl_generic_to_map(). Fix this by calling pinctrl_utils_free_map() and resetting *maps, *num_maps, and *num_reserved_maps in the error path of pinctrl_generic_to_map().	N/A	More Details
CVE-2026-90342	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix mmap_lock deadlock on arena lock failure Reported by the Sashiko AI review. arena_vm_fault() returns VM_FAULT_RETRY when it can't take arena->spinlock, but it never took mmap_lock. The fault path assumes a VM_FAULT_RETRY handler already dropped mmap_lock and re-takes it on the retry, so mmap_lock gets taken twice and can deadlock: do_user_addr_fault() { fault = handle_mm_fault(...); // calls arena_vm_fault() if (fault & VM_FAULT_RETRY) goto retry; // re-locks mmap_lock mmap_read_unlock(mm); } Return VM_FAULT_SIGBUS instead, for two reasons: 1. We could keep VM_FAULT_RETRY, but then we'd have to drop the fault lock first and cap the retry ourselves, the way __folio_lock_or_retry() does. 2. A failed raw_res_spin_lock_irqsave() already means a possible deadlock was detected, so retrying just hits the same lock again. So returning VM_FAULT_RETRY here is overkill.	N/A	More Details
CVE-2026-	An authenticated, non-guest user of Curiosity Workspace could enroll themselves as an administrator and member of an existing access group without an invitation or approval. It did not grant application-wide administrator privileges, and the	N/A	More Details

88817	vulnerability could not be used to obtain root access to the application or its underlying host.		
CVE-2026-90344	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: disconnect on CSA to channel 0 The refactor for the CSA parsing erroneously equates channel zero and no information present, leading it to ignore a CSA on an AP that advertises a switch to that (invalid) channel. This leads to not disconnecting, which we should. For Intel devices, this can lead to a firmware crash. Fix this by using an int type for the channel number as well as the opclass, and using a (negative) value that cannot be encoded in the element to indicate it's not present.	N/A	More Details
CVE-2026-90345	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: fix P2P action frame handling without device vif Some P2P action frame paths assume the P2P device vif is always available. That is not true when userspace sends non-P2P public action frames through the primary interface, or when action-frame abort runs after the P2P device vif has not been created. Fall back to the primary vif when aborting an action frame without a P2P device vif, and guard P2P device saved IE access before using it for peer channel search.	N/A	More Details
CVE-2026-90346	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: clean up color-change beacon data on errors nl80211_color_change() calls nl80211_parse_beacon() for the beacon_next template, which can allocate params.beacon_next.mbssid_ies and .rnr_ies. A parsing failure returned directly instead of using the out: cleanup, leaking any allocations completed before the error. Allocate the nested attribute table before parsing beacon_next. Its allocation failure can then return before beacon data exists, while a later parsing failure uses out: to release the parsed data.	N/A	More Details
CVE-2026-73176	Nozomi Networks Labs identified a CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability in the web management interface of Advantech EKI-1242IEIMS in firmware version V1.06.01 that allows a remote authenticated attacker to execute arbitrary OS commands as root via crafted request parameters.	N/A	More Details
CVE-2026-90348	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: snoc: use memcpy_fromio() for MSA ramdump On WCN3990/SNOC the MSA region is mapped with devm_memremap(MEMREMAP_WT). On arm64 such a mapping is not Normal-cacheable, so unaligned accesses to it are not permitted. ath10k_msa_dump_memory() copies the region with a plain memcpy(), whose optimized __pi_memcpy_generic implementation issues wide/unaligned loads. This triggers an alignment fault (FSC=0x21) Oops in ath10k_snoc_fw_crashed_dump() while collecting the devcoredump: Unable to handle kernel paging request ... FSC=0x21: alignment fault pc : __pi_memcpy_generic lr : ath10k_snoc_fw_crashed_dump [ath10k_snoc] The Oops both leaves the firmware RAM dump buffer zeroed (no dump is captured) and crashes the kernel, which in turn breaks modem SSR recovery. Use memcpy_fromio(), which only performs accesses that are valid for such a device-memory mapping. The generic memcpy_fromio() implementation aligns the source before issuing word-sized reads and stores the destination with put_unaligned(), so it is also safe for the coherent DMA allocation used on the non-reserved-memory path. ath11k and ath12k use the same pattern when copying target memory into crash dumps, so call it unconditionally here too. The MEMREMAP_WT pointer is a plain void *, so an explicit __iomem cast is needed; use __force to keep sparse happy. Tested-on: WCN3990 hw1.0 SNOC WLAN.HL.3.3.7.c5-00107-QCAHLSWMTPL-1	N/A	More Details
CVE-2026-90349	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: fix out-of-bounds link array access in mt7996_tx() When mac80211 leaves the link unspecified, mt7996_tx() substitutes the primary link id of the station or vif. That value is IEEE80211_LINK_UNSPECIFIED (0xf) until the first link has been added, and it is then used unchecked to index vif->link_conf[], mvif->mt76.link[] and sta->link[], all of which hold IEEE80211_MLD_MAX_NUM_LINKS (15) entries. Clamp the primary link id to the default link before using it, and use the clamped value for the link_sta fallback as well.	N/A	More Details
CVE-2026-90350	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: reject out-of-range link ids in mt76_vif_link() mt76_vif_link() indexes mvif->link[] without validating link_id, but callers pass mvif->deflink_id / msta->deflink_id, which hold IEEE80211_LINK_UNSPECIFIED (0xf) until the first link has been added. Since IEEE80211_MLD_MAX_NUM_LINKS is 15, that reads one element past the end of the array, aliasing mt76_vif_data.offchannel_link. Reachable via mt7996_set_tsf()/mt7996_offset_tsf() and mt7996_net_fill_forward_path(). Bounds check link_id and return NULL, matching mt7996_sta_link() and mt7996_sta_link_protected().	N/A	More Details
CVE-2026-90351	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: do not attach hif2 WED when the main WED attach failed If the WED attach for the primary PCIe function fails, the probe path still attached wed_hif2 for the secondary function, leaving the device in an inconsistent half-WED configuration that crashes later. The hif2 call also re-enabled hwrro_mode, which the failed primary attach had just turned off. Skip the hif2 WED setup when the primary WED device is not active.	N/A	More Details
CVE-2026-90352	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: release hif2 reference on probe IRQ failure The hif2 reference obtained by mt7915_pci_init_hif2() is only released on error paths that key off dev->hif2, which is not assigned until after the IRQ setup. If pci_alloc_irq_vectors() or the primary devm_request_irq() fails, the reference leaks. Drop it explicitly on those paths via mt7915_put_hif2().	N/A	More Details
CVE-2026-73175	Nozomi Networks Labs identified a CWE-400: Uncontrolled Resource Consumption vulnerability in the OPC UA gateway component of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows an adjacent unauthenticated attacker to exhaust the server session pool and cause a complete denial of service to all legitimate OPC UA clients by opening multiple anonymous sessions.	N/A	More Details
CVE-2026-90354	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: fix double hif2 init on the non-WED path mt7915_pci_init_hif2() was called unconditionally and again inside the WED-inactive branch. The helper increments the global hif_idx, writes the PCIe RECOG_ID register and takes a get_device() reference via mt7915_pci_get_hif2(), while removal only drops one reference. On non-WED dual-hif hardware this double-incremented hif_idx, wrote RECOG_ID twice and leaked a device reference. Only the call inside the WED-inactive branch is correct; drop the unconditional one. hif2 is already initialised to NULL.	N/A	More Details
CVE-2026-90355	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: clear stale link state on full reset After a full chip reset, mac80211 reconfig replays interface, link and channel context setup. mt7996_vif_link_add() short-circuits when the link_id is still marked in mvif->valid_links, a state introduced for postponing link teardown to interface removal. The reset path frees the link structures without clearing those bits, so the replayed setup never re-creates dev_info/bss_info/STA records in the restarted firmware and never re-registers the link wcid, leaving the device inoperative. The reset path also leaks every allocated MLD index: per-link indices and the per-vif group/remap indices are re-allocated from scratch during reconfig, but the old bits stay set in the masks, so repeated full resets exhaust the index space. Clear valid_links in the reset vif iterator and	N/A	More Details

	reset the MLD index masks alongside the existing omac_mask clearing.		
CVE-2026-90356	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: free vif links after clearing wcid entries on full reset mt7996_mac_reset_vif_iter() queues non-default vif links for kfree_rcu while dev->wcid[] still holds pointers to the wcid embedded in each freed link; mt76_reset_device() then dereferences those entries and runs mt76_wcid_cleanup() on them. If a grace period elapses in between, the cleanup operates on freed memory. Run mt76_reset_device() first, so the wcid entries are cleaned up and cleared while the links are still valid.	N/A	More Details
CVE-2026-73174	Nozomi Networks Labs identified a CWE-319: Cleartext Transmission of Sensitive Information vulnerability in the edgserver management protocol of Advantech EKI-1242EIMS in firmware version V1.06.01 that allows a network-adjacent passive observer to intercept management traffic and recover sensitive device identity and network metadata in cleartext.	N/A	More Details
CVE-2026-90335	In the Linux kernel, the following vulnerability has been resolved: tty: skip cdev_del() when no cdev is registered TTY device registration can fail before a cdev is allocated. Serial core keeps the port so setserial can still use it, and later removal passes the NULL cdev slot to cdev_del(), causing a NULL-pointer dereference. Only delete the cdev when the slot is not NULL.	N/A	More Details
CVE-2026-90333	In the Linux kernel, the following vulnerability has been resolved: dm-integrity: replace forgeable discard filler with a keyed sector marker The discard-block check in dm_integrity_rw_tag() treats a stored tag of all 0xf6 bytes (DISCARD_FILLER) as proof a block was discarded and skips HMAC verification. allow_discards is only accepted in dm-integrity's standalone mode. An attacker with raw write access to the backing device, but without the integrity key, can stamp any block with an all-0xf6 tag and have it served as authentic. Add a new "allow_discards_keyed" target argument that marks discarded blocks with a keyed checksum of (salt || sector) instead, computed by integrity_discard_checksum().	N/A	More Details
CVE-2026-90300	In the Linux kernel, the following vulnerability has been resolved: bpf: Clear buf on error in __bpf_get_task_stack Both bpf_get_task_stack and bpf_get_task_stack_sleepable helpers that use __bpf_get_task_stack have buf defined as ARG_PTR_TO_UNINIT_MEM argument and we should initialize the buf on every return path. Adding missing buf memset for __bpf_get_task_stack fail paths. This provides deterministic buffer contents, which is useful when the buffer is used directly as a map key.	N/A	More Details
CVE-2026-90331	In the Linux kernel, the following vulnerability has been resolved: HID: asus: refactor the two workqueues and init sequence Multiple issues have been found within the hid-asus driver: - unchecked size in asus_raw_event() - unclean teardown of asus_probe on failure - possible use-after-free in asus_probe - multiple workqueue used for jobs where one was enough - sleeping calls in atomic context - packets of incorrect size being sent to the keyboard controller Join the two workqueues into one reusing the stopping mechanism of the brightness workqueue, use the joined workqueue to also move the asus_wmi_send_event() sleeping call away from atomic context and add a size check in asus_raw_event().	N/A	More Details
CVE-2026-90302	In the Linux kernel, the following vulnerability has been resolved: ocfs2: synchronize heartbeat callbacks with o2net teardown Patch series "ocfs2: harden heartbeat teardown races". This series fixes two OCFS2 heartbeat/o2net teardown races found by KASAN. This patch (of 2): Heartbeat callbacks stay registered while configfs local-node teardown enters o2net_stop_listening(). A node-down event can still run through o2net_disconnect_node() and o2net_set_nn_state() while teardown is destroying o2net_wq, so the later queue/flush operations can hit a dead workqueue. KASAN has caught this as a slab-use-after-free in __queue_work() with the call chain: KASAN slab-use-after-free in __queue_work+0x56/0xa90 Read of size 4 Call trace: dump_stack_lvl+0x66/0xa0 print_report+0xce/0x630 __queue_work+0x56/0xa90 srso_alias_return_thunk+0x5/0xfbef5 __virt_addr_valid+0x19f/0x330 kasan_report+0xe0/0x110 __queue_delayed_work+0x58/0x1e0 queue_delayed_work_on+0xb4/0xc0 o2net_set_nn_state+0x467/0x840 o2net_disconnect_node+0x7b/0xe0 o2net_hb_node_down_cb+0x54/0x60 o2hb_run_event_list+0x236/0x2d0 o2hb_check_slot+0xad4/0xbc0 lock_release+0xc8/0x290 o2hb_check_slot+0x9ea/0xbc0 trace_hardirqs_on+0x18/0x130 o2hb_do_disk_heartbeat+0x646/0xb30 (fs/ocfs2/cluster/heartbeat.c:1079) __lock_acquire+0x466/0x2260 lockdep_hardirqs_on_prepare+0xea/0x1a0 ktime_get_with_offset+0xe9/0x230 o2hb_thread+0x14e/0x770 kthread+0x1ad/0x1f0 ret_from_fork+0x3c9/0x540 __switch_to+0x2e9/0x730 ret_from_fork_asm+0x1a/0x30 Allocated by task stack: kasan_save_stack+0x33/0x60 kasan_save_track+0x14/0x30 __kasan_kmalloc+0xaa/0xb0 __kmalloc_noprof+0x292/0x760 __alloc_workqueue+0x736/0xc60 alloc_workqueue_noprof+0xb1/0x110 o2net_start_listening+0xe5/0x430 o2nm_node_local_store+0x184/0x310 configfs_write_iter+0x18a/0x210 vfs_write+0x469/0x810 ksys_write+0xd2/0x170 do_syscall_64+0x115/0x6a0 (arch/x86/entry/syscall_64.c:87) entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task stack: kasan_save_stack+0x33/0x60 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x5f/0x80 kfree+0x313/0x590 rcu_core+0x4f4/0x1320 handle_softirqs+0x156/0x660 queue_delayed_work_on o2net_set_nn_state o2net_disconnect_node o2net_hb_node_down_cb o2hb_run_event_list Keep heartbeat callbacks registered so quorum state still tracks node state, but stop them from driving o2net reconnect/disconnect work once local teardown starts. Mark the transport offline before destroying o2net_wq, wait for any in-flight heartbeat callback to finish, and delay bring-up replay until the new local node is published through o2nm_this_node(). The replay also has to stay serialized with heartbeat callback delivery. Otherwise a live-node snapshot can be copied, a real hb_down callback can install -ENOTCONN for a peer, and the stale replay can call o2net_hb_node_up() for that same peer and queue reconnect work even though heartbeat is already down. The buggy scenario involves two paths, with each column showing the order within that path: local-node teardown: heartbeat node-down callback: 1. configfs local-off enters 1. o2hb_run_event_list() invokes o2net_stop_listening(). o2net_hb_node_down_cb(). 2. teardown heads for 2. the callback reaches destroy_workqueue(o2net_wq). o2net_disconnect_node() and o2net_set_nn_state(). 3. teardown destroys and NULLs 3. the callback flushes or queues o2net_wq. work through o2net_wq.	N/A	More Details
CVE-2026-90303	In the Linux kernel, the following vulnerability has been resolved: ARM: 9485/1: mm: acquire mmap write lock around show_pte() for user faults When CONFIG_DEBUG_USER=y, and cmdline "user_debug=31" is set, a user fault may trigger show_pte() without any lock. If another thread in the same process concurrently calls munmap(), the page table pages may be freed while show_pte() is still traversing them, causing a use-after-free in show_pte(). If CONFIG_ARM_LPAE=y, this may cause a kernel panic if the pages table of PMD are freed when show_pte() is running. Acquire mmap_write_lock() around show_pte() for user faults to fix the contention. For user faults, additionally restrict that show_pte() is called only when the addr is a user-space address (addr < TASK_SIZE). This is because the lock of tsk->mm only protects the virtual memory of user address space, furthermore, dumping the page tables of a kernel-space address for user faults is unnecessary and may have security implications. Keep everything unchanged for kernel faults, because the kernel is already in the "oops" state, acquiring a lock may risk a deadlock.	N/A	More Details
	In the Linux kernel, the following vulnerability has been resolved: ARM: 9484/1: enable interrupts when unhandled user faults are triggered PREEMPT_RT requires interrupts to be enabled when sending signals. When do_DataAbort()/do_PrefetchAbort()		

CVE-2026-90304	triggers unhandled user faults, that is `inf->fn()` return a non-zero value, and the interrupts are not enabled within the hook function, force_sig_fault() will be called with interrupts disabled. This can be triggered by user programs executing the bkpt instruction, with kernel config CONFIG_PERF_EVENTS=n. Enable interrupts in do_DataAbort()/do_PrefetchAbort() when unhandled user faults are triggered to fix the issue.	N/A	More Details
CVE-2026-90305	In the Linux kernel, the following vulnerability has been resolved: ARM: 9483/1: select HAVE_POSIX_CPU_TIMERS_TASK_WORK Commit c6e61c06d606 ("ARM: 9463/1: Allow to enable RT") enabled PREEMPT_RT on ARM but did not select HAVE_POSIX_CPU_TIMERS_TASK_WORK. This leaves CONFIG_POSIX_CPU_TIMERS_TASK_WORK disabled, so CPU timers expire in hard IRQ context. On PREEMPT_RT this makes run_posix_cpu_timers() take the sleeping sighand->siglock: BUG: sleeping function called from invalid context at spinlock_rt.c:48 rt_spin_lock from lock_task_sighand lock_task_sighand from run_posix_cpu_timers run_posix_cpu_timers from update_process_times ARM handles TIF_NOTIFY_RESUME on all return-to-user paths, including v7-M. ARM32 KVM host support was removed by commit 541ad0150ca4 ("arm: Remove 32bit KVM host support"), so the select need not be conditional on KVM. Select it to defer POSIX CPU timer expiry to task context. Reproduced with setrlimit(RLIMIT_CPU, ...) and a busy loop. The same path is used by setitimer(ITIMER_PROF or ITIMER_VIRTUAL) and POSIX CPU timers created with timer_create().	N/A	More Details
CVE-2026-90306	In the Linux kernel, the following vulnerability has been resolved: ARM: 9481/2: breakpoint: CFI breakpoints only on demand This removes the stub hw_breakpoint_cfi_handler() from ARM, making it not steal breakpoint type 0x03 (ARM_ENTRY_CFI_BREAKPOINT) unless CFI is actively used in the kernel. When not instrumenting with CFI, or when a breakpoint is issued in userspace, we fall through to return 1 from hw_breakpoint_pending() "unhandled fault" so userspace can make use of this breakpoint. Tested with LKDTM and this command line: echo CFI_FORWARD_PROTO > /sys/kernel/debug/provoke-crash/DIRECT still works as expected.	N/A	More Details
CVE-2026-90307	In the Linux kernel, the following vulnerability has been resolved: RDMA/srp: fix heap information leak on a truncated SRP_CRED_REQ srp_recv_done() passes wc->byte_len to srp_process_rsp(). It passes nothing to srp_process_cred_req() and srp_process_aer_req(), which read fixed-size fields from the receive buffer without checking that those fields were received. The buffer size is max_ti_iu_len, which comes from the login response and is not validated. A target that advertises 8 and then sends an 8-byte SRP_CRED_REQ makes the initiator read req->tag from beyond the end of the buffer. req->tag is copied into the SRP_CRED_RSP and sent back, so those bytes reach the target. SRP_AER_REQ behaves the same way and also reads req->lun. The leak is 8 bytes per response. max_ti_iu_len also decides which slab cache the buffer comes from. With 8 the buffer is a kmalloc-8 object and the read is entirely outside it: BUG: KASAN: slab-out-of-bounds in srp_recv_done+0x172b/0x1aa0 Read of size 8 at addr ffff888104714da8 by task kworker/u8:3/50 which belongs to the cache kmalloc-8 of size 8 The buggy address is located 0 bytes to the right of allocated 8-byte region [ffff888104714da0, ffff888104714da8) Without KASAN the returned bytes are whatever is next in the slab. One run returned ".strtab". rsp->data[3] in srp_process_rsp() has the same problem: only resp_data_len is checked before it is read. Drop a request that is shorter than the structure being parsed, and check byte_len before the tsk_mgmt read.	N/A	More Details
CVE-2026-90310	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.	N/A	More Details
CVE-2026-90311	In the Linux kernel, the following vulnerability has been resolved: thermal: hwmon: Remove hwmon class device along with its parent The current code creates one hwmon device per thermal zone type and that device is registered under the first thermal zone of the given type. That turns out to be problematic when the thermal zone holding the hwmon device is removed. For example, say that there are two ACPI thermal zones on a system /sys/devices/virtual/thermal/thermal_zone0/ /sys/devices/virtual/thermal/thermal_zone1/ The current code registers a hwmon class device for thermal_zone0 only: /sys/devices/virtual/thermal/thermal_zone0/hwmon0/ because the type is "acpitz" for both of them, but it adds a sysfs attribute that belongs to thermal_zone1 under it: /sys/devices/virtual/thermal/thermal_zone0/hwmon0/temp2_input There is also /sys/devices/virtual/thermal/thermal_zone0/hwmon0/temp1_input which belongs to thermal_zone0. When thermal_zone0 is removed, say because the ACPI thermal driver is unbound from the underlying platform device, thermal_remove_hwmon_sysfs() skips the removal of hwmon0 because of the temp2_input attribute belonging to thermal_zone1 which effectively prevents thermal_zone0 removal from making progress. Address this by making thermal_remove_hwmon_sysfs() remove the entire hwmon class device interface for the given thermal zone type when the thermal zone device holding it is removed. To prevent races with thermal_add_hwmon_sysfs() that may interfere with this, carry out the entire addition and removal of hwmon sysfs interfaces for thermal zones under thermal_hwmon_list_lock. Also adjust the layout of the labels in thermal_add_hwmon_sysfs() to the current kernel coding style to align with the new "unlock" label.	N/A	More Details
CVE-2026-90313	In the Linux kernel, the following vulnerability has been resolved: bpf, cgroup: Fix invalid storage access after __cgroup_bpf_attach failed A potential invalid storage access issue can occur after replacing a cgroup bpf prog. This occurs in the following scenario: 1. prog1 with storage is attached to a cgroup in multi-attach mode. 2. prog1 is replaced with prog2 using BPF_F_REPLACE in multi-attach mode, but fails midway (e.g. in bpf_trampoline_link_cgroup_shim or update_effective_progs). 3. A new prog3 is attached to the cgroup in multi-attach mode. The reason is that __cgroup_bpf_attach overwrites pl->storage with the new storage prior to attachment completion. When attachment fails midway, the cleanup path calls bpf_cgroup_storages_free(new_storage) to free the newly allocated storage, but fails to restore pl->storage back to old_storage. Consequently, the still-active prog1 holds invalid or dangling storage pointers, leading to an invalid memory access when prog1 executes and calls bpf_get_local_storage. Additionally, original pl->flags and cgrp->bpf.flags[atype] are left unrestored. Fix this by saving old_pl_flags, old_storage, and old_flags prior to the update, and properly restoring all of them in the cleanup path on error.	N/A	More Details
CVE-2026-90314	In the Linux kernel, the following vulnerability has been resolved: remoteproc: fix OOB read via signed offset in rsc_table_for_each_entry() table->offset[i] is a u32 from firmware, but was stored into a signed int. A crafted offset like 0xFFFFFFF0 becomes -16, placing hdr 16 bytes before the table buffer. The subsequent avail check was bypassed because the negative int was promoted to a large size_t in the expression "table_sz - offset - sizeof(*hdr)", yielding a large positive avail and letting the out-of-bounds hdr->type read proceed undetected. Store the offset as u32 and validate it with unsigned comparisons before any pointer arithmetic.	N/A	More Details
CVE-	In the Linux kernel, the following vulnerability has been resolved: PCI/sysfs: Add lockdown checks to legacy I/O and memory handlers Currently, the legacy I/O and memory sysfs handlers do not check security_locked_down(LOCKDOWN_PCI_ACCESS), leaving the legacy_io and legacy_mem files unprotected when the kernel is locked down. Commit eb627e17727e ("PCI: Lock down BAR access when the kernel is locked down") added the check to pci_write_config(), pci_mmap_resource(), and		

2026-90315	pci_write_resource_io() to prevent userspace from programming DMA-capable hardware that could be used to modify kernel code, but did not cover the legacy handlers. As a result, root can still write arbitrary I/O ports and map the legacy I/O and memory spaces while the kernel is locked down, which is the same capability the lockdown is meant to remove. Add the same check to pci_write_legacy_io(), pci_mmap_legacy_mem(), and pci_mmap_legacy_io(). These generic handlers cover both architectures that define HAVE_PCI_LEGACY (such as Alpha and PowerPC). [bhelgaas: add Link]	N/A	More Details
CVE-2026-90318	In the Linux kernel, the following vulnerability has been resolved: fat: release buffer head after rebuilding parent fat_scan_logstart() leaves the matching directory entry's buffer head in sinfo.bh for the caller to release, just like fat_scan(). fat_rebuild_parent() uses the directory entry to rebuild the parent inode for the nostale_ro NFS export path, but does not release sinfo.bh after a successful scan. Release it once fat_build_inode() has consumed the directory entry data.	N/A	More Details
CVE-2026-90319	In the Linux kernel, the following vulnerability has been resolved: rapidio: clear mport->net when rio_add_net() fails rio_alloc_net() stores the newly allocated rio_net in mport->net before rio_scan_alloc_net() registers the device. If rio_add_net() fails, rio_scan_alloc_net() drops the device reference with put_device(), which releases the rio_net through the device release callback. However, mport->net is left pointing at the freed object. A later mport unregister path can then dereference the dangling mport->net pointer and may try to free the same rio_net again. Clear mport->net in the rio_add_net() failure path, matching the cleanup done for the destID table allocation failure path.	N/A	More Details
CVE-2026-90322	In the Linux kernel, the following vulnerability has been resolved: ocfs2/cluster: keep heartbeat local node stable o2nm_node_local_store() handles local=0 by stopping o2net and setting cl_local_node to O2NM_INVALID_NODE_NUM, but it leaves cl_has_local set. That stale state makes o2nm_this_node() return 255, blocks a later local=1 attempt with -EBUSY, and can feed 255 to heartbeat users that call o2nm_this_node() dynamically. Clearing cl_has_local is required when the local node is reset. But heartbeat threads can still be running at that point. They pin the local node config item at startup, yet o2hb_do_disk_heartbeat() and thread teardown re-read o2nm_this_node() for the local slot and for o2nm_undepend_this_node(). Once local=0 has cleared the live local-node state, those dynamic reads return O2NM_MAX_NODES, which is also the invalid node number 255. Store the local node number in the heartbeat region when the region starts. Use that stable node for heartbeat slot writes/checks, negotiation messages, and the final configfs undepend. Stop the heartbeat loop when the current local node no longer matches the stored node, and clear cl_has_local together with cl_local_node in the local=0 path so nodemanager state matches node removal. Validation reproduced this kernel report: KASAN slab-out-of-bounds in o2hb_do_disk_heartbeat+0x372/0xb30 RIP: 0010:memset+0xf/0x20 Read of size 8 Call trace: dump_stack_lvl+0x66/0xa0 print_report+0xd0/0x630 o2hb_do_disk_heartbeat+0x372/0xb30 (fs/ocfs2/cluster/heartbeat.c:1079) srso_alias_return_thunk+0x5/0xfbef5 __virt_addr_valid+0x188/0x2f0 kasan_report+0xe4/0x120 o2hb_do_disk_heartbeat+0x5/0xb30 (fs/ocfs2/cluster/heartbeat.c:1079) o2hb_thread+0x14e/0x770 kthread_affine_node+0x139/0x180 lockdep_hardirqs_on_prepare+0xda/0x190 trace_hardirqs_on+0x18/0x130 kthread+0x19d/0x1e0 ret_from_fork+0x37a/0x4d0 __switch_to+0x2d5/0x6f0 ret_from_fork_asm+0x1a/0x30	N/A	More Details
CVE-2026-90323	In the Linux kernel, the following vulnerability has been resolved: ublk: validate auto buf reg before taking uring_cmd With UBLK_F_AUTO_BUF_REG, invalid sqe->addr can fail after ublk_fill_io_cmd() has set UBLK_IO_FLAG_ACTIVE. The uring_cmd is completed while the tag stays active, which can hang teardown. Split validation from buffer apply so the check has no side effects, then take the uring_cmd and store the already-validated buffer. Apply the same order in FETCH so io->buf is not written before __ublk_fetch() state checks.	N/A	More Details
CVE-2026-85104	In Sooma 2GEN brain stimulator, an attacker within Bluetooth range can make unauthenticated changes to brain stimulation parameters.	N/A	More Details
CVE-2026-73177	Nozomi Networks Labs identified a CWE-345: Insufficient Verification of Data Authenticity vulnerability in the firmware upgrade mechanism of the Advantech EKI-1242EIMS in firmware version V1.06.01. The device accepts firmware images through the authenticated web management interface without performing any cryptographic signature or certificate verification. An authenticated administrator-level attacker can install arbitrary modified firmware on the device, enabling full persistent compromise of the platform.	N/A	More Details
CVE-2026-90327	In the Linux kernel, the following vulnerability has been resolved: phonet: pep: do not write beyond optlen in getsockopt pep_getsockopt() clamps the reported length to the caller's buffer with min_t(), but then stores the value with put_user(val, (int __user *) optval), which always writes sizeof(int) bytes. A getsockopt() call with an optlen smaller than sizeof(int) thus reports the clamped length yet writes a full int, one to three bytes past the user buffer. Write the value with copy_to_user() bounded by len, so at most optlen bytes are copied, matching the length reported back to userspace.	N/A	More Details
CVE-2026-90328	In the Linux kernel, the following vulnerability has been resolved: HID: steam: Reject short reads Steam Controller FEATURE reports encode the size of the message in the message itself. Previously we were trusting that the size reported matched the size we actually read, leading to a potential issue with short reads. Instead, we should actually verify the length of the read.	N/A	More Details
CVE-2026-61598	djust provides Phoenix LiveView-style reactive server-side rendering for Django with Rust-powered performance. Prior to version 1.0.7, `djust.mixins.model_binding.ModelBindingMixin` provides a default `update_model` event handler and is part of the LiveView base MRO, so every LiveView exposes it. It `setattr`s a view attribute whose name is client-supplied (`field`), gated only by: reject `_`-prefixed names; reject a 14-entry denylist of framework internals (`FORBIDDEN_MODEL_FIELDS`); optional `allowed_model_fields` which defaults to None = allow all; and `hasattr` existence. As a result, a client can set any public, existing view attribute — not just the fields actually bound with `dj-model=` in the rendered template. The denylist covers framework plumbing but nothing about developer business/authz state, and the allowlist is opt-in (off by default). A developer who binds one `dj-model="search"` input and also keeps `self.account_id` / `self.is_admin` / `self.total_price` as view state does not realize a client can set ALL of them via `{type:event, event:"update_model", params:{field, value}}` over the WebSocket. Type coercion matches the target attribute's type (so `"true"` -> bool True), aiding the attacker. This issue is fixed in djust 1.0.7. As a workaround, set `allowed_model_fields` explicitly on every view using dj-model (or subclassing LiveView) to the minimal list of bindable fields; do not keep authorization/ownership state in public view attributes that share the view with dj-model bindings.	N/A	More Details
CVE-2026-90330	In the Linux kernel, the following vulnerability has been resolved: HID: logitech-hidpp: Fix FF device cleanup on init failure hidpp_ff_init() creates the input force-feedback device with input_ff_create(), then allocates the HID++ FF private data, effect ID array, and workqueue. If any of those allocations fail after input_ff_create() succeeds, the function returns an error without destroying the FF device. Add an unwind path that frees the private allocations made by hidpp_ff_init() and calls	N/A	More Details

	input_ff_destroy() for failures after input_ff_create() succeeds.		
CVE-2026-15638	An unauthenticated user with access to Secret Server could leverage a padding oracle to decrypt or encrypt data using one of the server's cryptographic keys. The key itself is not exposed.	N/A	More Details