الهيئة الوطنية للأمن السيبراني
National Cybersecurity Authority

تمت مشاركة هذه المعلومة بإشارة مشاركة ***أبيض*** حيث يسمح بتبادلها أو نشرها من خلال القنوات العامة.

في ضوء دور الهيئة الوطنية للأمن السيبراني للمساعدة في حماية الفضاء السيبراني الوطني، تود الهيئة مشاركتكم النشرة الأسبوعية للثغرات المسجلة من قبل the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) للأسبوع من 26 أبريل إلى 2 مايو. علماً أنه يتم تصنيف هذه الثغرات باستخدام معيار Common Vulnerability Scoring System (CVSS) حيث يتم تصنيف الثغرات بناء على التالي:

- عالي جدًا: النتيجة الأساسية لـCVSS 9.0-10.0
- عالي: النتيجة الأساسية لـCVSS 7.0-8.9
- متوسط: النتيجة الأساسية لـCVSS 4.0-6.9
- منخفض: النتيجة الأساسية لـ CVSS 0.0-3.9

Please note that this notification/advisory has been tagged as TLP ***WHITE*** where information can be shared or published on any public forums.

As part of NCA duties to help securing the cyberspace and protecting national interests, NCA provides the weekly summary of published vulnerabilities by the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) for the week from 26th of April to 2nd of May. Vulnerabilities are scored using the Common Vulnerability Scoring System (CVSS) standard as per the following severity:

- Critical: CVSS base score of 9.0-10.0
- High: CVSS base score of 7.0-8.9
- Medium: CVSS base score 4.0-6.9
- Low: CVSS base score 0.0-3.9

CVE ID & Source	Vendor - Product	Description	Publish Date	CVSS Score
CVE-2026-33453	apache - multiple products	Improperly Controlled Modification of Dynamically-Determined Object Attributes vulnerability in Apache Camel Camel-Coap component. Apache Camel's camel-coap component is vulnerable to Camel message header injection, leading to remote code execution when routes forward CoAP requests to header-sensitive producers (e.g. camel-exec) The camel-coap component maps incoming CoAP request URI query parameters directly into Camel Exchange In message headers without applying any HeaderFilterStrategy. Specifically, CamelCoapResource.handleRequest() iterates over OptionSet.getUriQuery() and calls camelExchange.getIn().setHeader(...) for every query parameter. CoAPEndpoint extends DefaultEndpoint rather than DefaultHeaderFilterStrategyEndpoint, and CoAPComponent does not implement HeaderFilterStrategyComponent; the component contains no references to HeaderFilterStrategy at all. As a result, an unauthenticated attacker who can send a single CoAP UDP packet to a Camel route consuming from coap:// can inject arbitrary Camel internal headers (those prefixed with Camel*) into the Exchange. When the route delivers the message to a header-sensitive producer such as camel-exec, camel-sql, camel-bean, camel-file, or template components (camel-freemarker, camel-velocity), the injected headers can alter the producer's behavior. In the case of camel-exec, the CamelExecCommandExecutable and CamelExecCommandArgs headers override the executable and arguments configured on the endpoint, resulting in arbitrary OS command execution under the privileges of the Camel process. The producer's output is written back to the Exchange body and returned in the CoAP response payload by CamelCoapResource, giving the attacker an interactive RCE channel without any need for out-of-band exfiltration. Exploitation prerequisites are minimal: a single unauthenticated UDP datagram to the CoAP port (default 5683). CoAP (RFC 7252) has no built-in authentication, and DTLS is optional and disabled by default. Because the protocol is UDP-based, HTTP-layer WAF/IDS controls do not apply. This issue affects Apache Camel: from 4.14.0 through 4.14.5, from 4.18.0 before 4.18.1, 4.19.0. Users are recommended to upgrade to version 4.18.1 or 4.19.0, fixing the issue.	2026-04-27	10
CVE-2026-40453	apache - multiple products	The fix for CVE-2025-27636 added setLowerCase(true) to HttpHeaderFilterStrategy so that case-variant header names such as 'CAmelExecCommandExecutable' are filtered out alongside 'CamelExecCommandExecutable'. The same setLowerCase(true) call was not applied to five non-HTTP HeaderFilterStrategy implementations: JmsHeaderFilterStrategy and ClassicJmsHeaderFilterStrategy in camel-jms, SjmsHeaderFilterStrategy in camel-sjms, CoAPHeaderFilterStrategy in camel-coap, and GooglePubsubHeaderFilterStrategy in camel-google-pubsub. Because those strategies use case-sensitive String.startsWith('Camel'/'camel') filtering while the Camel Exchange stores headers in a case-insensitive map, an attacker with JMS (or equivalent) producer access to the broker consumed by a Camel route can inject case-variant Camel internal headers, which are then resolved by downstream components such as camel-exec and camel-file using their canonical casing. This enables remote code execution and arbitrary file write on routes that forward JMS messages to header-driven components.	2026-04-27	9.9

		This issue affects Apache Camel: from 3.0.0 before 4.14.6, from 4.15.0 before 4.18.2, from 4.19.0 before 4.20.0. Users are recommended to upgrade to version 4.20.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.6. If users are on the 4.18.x releases stream, then they are suggested to upgrade to 4.18.2.		
CVE-2026-40860	apache - multiple products	JmsBinding.extractBodyFromJms() in camel-jms, and the equivalent JmsBinding class in camel-sjms, deserialized the payload of incoming JMS ObjectMessage values via javax.jms.ObjectMessage.getObject() without applying any ObjectInputFilter, class allowlist or class denylist. Because this code path is reached whenever the mapJmsMessage option is enabled (the default) and Camel acts as a JMS consumer, an attacker able to publish a crafted ObjectMessage to a queue or topic consumed by a Camel application could achieve remote code execution when a deserialization gadget chain was present on the classpath. The same handling was reached transitively through camel-sjms2 (whose Sjms2Endpoint extends SjmsEndpoint) and through camel-amqp (whose AMQPJmsBinding extends JmsBinding), and by other JMS-family components built on JmsComponent such as camel-activemq and camel-activemq6. This issue affects Apache Camel: from 3.0.0 before 4.14.7, from 4.15.0 before 4.18.2, from 4.19.0 before 4.20.0. Users are recommended to upgrade to version 4.20.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.7. If users are on the 4.18.x releases stream, then they are suggested to upgrade to 4.18.2.	2026-04-27	9.8
CVE-2026-41635	apache - multiple products	Apache MINA's AbstractIoBuffer.resolveClass() contains two branches, one of them (for static classes or primitive types) does not check the class at all, bypassing the classname allowlist and allowing arbitrary code to be executed. The fix checks if the class is present in the accepted class filter before calling Class.forName(). Affected versions are Apache MINA 2.0.0 <= 2.0.27, 2.1.0 <= 2.1.10, and 2.2.0 <= 2.2.5. The problem is resolved in Apache MINA 2.0.28, 2.1.11, and 2.2.6 by applying the classname allowlist earlier. Affected are applications using Apache MINA that call IoBuffer.getObject(). Applications using Apache MINA are advised to upgrade.	2026-04-27	9.8
CVE-2026-41409	apache - multiple products	The fix for CVE-2024-52046 in Apache MINA AbstractIoBuffer.getObject() was incomplete. The classname allowlist of classes allowed to be deserialized was applied too late after a static initializer in a class to be read might already have been executed. Affected versions are Apache MINA 2.0.0 <= 2.0.27, 2.1.0 <= 2.1.10, and 2.2.0 <= 2.2.5. The problem is resolved in Apache MINA 2.0.28, 2.1.11, and 2.2.6 by applying the classname allowlist earlier. Affected are applications using Apache MINA that call IoBuffer.getObject().	2026-04-27	9.8

		Applications using Apache MINA are advised to upgrade		
CVE-2026-41873	apache - pony_mail	** UNSUPPORTED WHEN ASSIGNED ** Inconsistent Interpretation of HTTP Requests ('HTTP Request/Response Smuggling') vulnerability in Pony Mail leading to admin account takeover. This issue affects all versions of the Lua implementation of Pony Mail. There is a Python implementation under development under the name "Pony Mail Foal" that is not affected by this issue, but hasn't been released yet. As the Lua implementation of this project is retired, we do not plan to release a version that fixes this issue. Users are recommended to find an alternative or restrict access to the instance to trusted users. NOTE: This vulnerability only affects products that are no longer supported by the maintainer.	2026-04-28	9.8
CVE-2026-42778	apache - multiple products	The fix for CVE-2026-41409 was not applied to the 2.1.X and 2.2.X branches. Here was the original issue description: The fix for CVE-2024-52046 in Apache MINA AbstractIoBuffer.getObject() was incomplete. The classname allowlist of classes allowed to be deserialized was applied too late after a static initializer in a class to be read might already have been executed. Affected versions are Apache MINA 2.1.0 <= 2.1.11, and 2.2.0 <= 2.2.6. The problem is resolved in Apache MINA 2.1.12, and 2.2.7 by applying the classname allowlist earlier. Affected are applications using Apache MINA that call IoBuffer.getObject(). Applications using Apache MINA are advised to upgrade The fix for CVE-2024-52046 in Apache MINA AbstractIoBuffer.getObject() was incomplete. The classname allowlist of classes allowed to be deserialized was applied too late after a static initializer in a class to be read might already have been executed. Affected versions are Apache MINA 2.1.0 <= 2.1.110, and 2.2.0 <= 2.2.6. The problem is resolved in Apache MINA 2.1.12, and 2.2.7 by applying the classname allowlist earlier. Affected are applications using Apache MINA that call IoBuffer.getObject(). Applications using Apache MINA are advised to upgrade	2026-05-01	9.8
CVE-2026-42779	apache - multiple products	The fix for CVE-2026-41635 was not applied to the 2.1.X and 2.2.X branches. Here was the original issue description:	2026-05-01	9.8

		Apache MINA's AbstractIoBuffer.resolveClass() contains two branches, one of them (for static classes or primitive types) does not check the class at all, bypassing the classname allowlist and allowing arbitrary code to be executed. The fix checks if the class is present in the accepted class filter before calling Class.forName(). Affected versions are Apache MINA 2.1.0 <= 2.1.11, and 2.2.0 <= 2.2.6. The problem is resolved in Apache MINA 2.1.12, and 2.2.7 by applying the classname allowlist earlier. Affected are applications using Apache MINA that call IoBuffer.getObject(). Applications using Apache MINA are advised to upgrade.		
CVE-2026-31705	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix out-of-bounds write in smb2_get_ea() EA alignment smb2_get_ea() applies 4-byte alignment padding via memset() after writing each EA entry. The bounds check on buf_free_len is performed before the value memcpy, but the alignment memset fires unconditionally afterward with no check on remaining space. When the EA value exactly fills the remaining buffer (buf_free_len == 0 after value subtraction), the alignment memset writes 1-3 NUL bytes past the buf_free_len boundary. In compound requests where the response buffer is shared across commands, the first command (e.g., READ) can consume most of the buffer, leaving a tight remainder for the QUERY_INFO EA response. The alignment memset then overwrites past the physical kvmalloc allocation into adjacent kernel heap memory. Add a bounds check before the alignment memset to ensure buf_free_len can accommodate the padding bytes. This is the same bug pattern fixed by commit beef2634f81f ("ksmbd: fix potencial OOB in get_file_all_info() for compound requests") and commit fda9522ed6af ("ksmbd: fix OOB write in QUERY_INFO for compound requests"), both of which added bounds checks before unconditional writes in QUERY_INFO response handlers.	2026-05-01	9.8
CVE-2026-31718	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in __ksmbd_close_fd() via durable scavenger When a durable file handle survives session disconnect (TCP close without SMB2_LOGOFF), session_fd_check() sets fp->conn = NULL to preserve the handle for later reconnection. However, it did not clean up the byte-range locks on fp->lock_list. Later, when the durable scavenger thread times out and calls __ksmbd_close_fd(NULL, fp), the lock cleanup loop did:	2026-05-01	9.8

		spin_lock(&fp->conn->llist_lock); This caused a slab use-after-free because fp->conn was NULL and the original connection object had already been freed by ksmbd_tcp_disconnect(). The root cause is asymmetric cleanup: lock entries (smb_lock->clist) were left dangling on the freed conn->lock_list while fp->conn was nulled out. To fix this issue properly, we need to handle the lifetime of smb_lock->clist across three paths: - Safely skip clist deletion when list is empty and fp->conn is NULL. - Remove the lock from the old connection's lock_list in session_fd_check() - Re-add the lock to the new connection's lock_list in ksmbd_reopen_durable_fd().		

CVE-2026-43011	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net/x25: Fix potential double free of skb
		
		When alloc_skb fails in x25_queue_rx_frame it calls kfree_skb(skb) at line 48 and returns 1 (error).
		This error propagates back through the call chain:
		
		x25_queue_rx_frame returns 1
		|
		v
		x25_state3_machine receives the return value 1 and takes the else branch at line 278, setting queued=0 and returning 0
		|
		v
		x25_process_rx_frame returns queued=0
		|
		v
		x25_backlog_rcv at line 452 sees queued=0 and calls kfree_skb(skb) again
		
		This would free the same skb twice. Looking at x25_backlog_rcv:
		
		net/x25/x25_in.c:x25_backlog_rcv() {
		...
		queued = x25_process_rx_frame(sk, skb);
		...
		if (!queued)
		kfree_skb(skb);
		}
			2026-05-01	9.8

CVE-2026-43037	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ip6_tunnel: clear skb2->cb[] in ip4ip6_err() Oskar Kjos reported the following problem. ip4ip6_err() calls icmp_send() on a cloned skb whose cb[] was written by the IPv6 receive path as struct inet6_skb_parm. icmp_send() passes IPCB(skb2) to __ip_options_echo(), which interprets that cb[] region as struct inet_skb_parm (IPv4). The layouts differ: inet6_skb_parm.nhoff at offset 14 overlaps inet_skb_parm.opt.rr, producing a non-zero rr value. __ip_options_echo() then reads optlen from attacker-controlled packet data at sptr[rr+1] and copies that many bytes into dopt->__data, a fixed 40-byte stack buffer (IP_OPTIONS_DATA_FIXED_SIZE). To fix this we clear skb2->cb[], as suggested by Oskar Kjos. Also add minimal IPv4 header validation (version == 4, ihl >= 5).	2026-05-01	9.8
CVE-2026-43038	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: icmp: clear skb2->cb[] in ip6_err_gen_icmpv6_unreach() Sashiko AI-review observed: In ip6_err_gen_icmpv6_unreach(), the skb is an outer IPv4 ICMP error packet where its cb contains an IPv4 inet_skb_parm. When skb is cloned into skb2 and passed to icmp6_send(), it uses IP6CB(skb2). IP6CB interprets the IPv4 inet_skb_parm as an inet6_skb_parm. The cipso offset in inet_skb_parm.opt directly overlaps with dsthao in inet6_skb_parm at offset 18. If an attacker sends a forged ICMPv4 error with a CIPSO IP option, dsthao	2026-05-01	9.8

		would be a non-zero offset. Inside icmp6_send(), mip6_addr_swap() is called and uses ipv6_find_tlv(skb, opt->dsthao, IPV6_TLV_HAO). This would scan the inner, attacker-controlled IPv6 packet starting at that offset, potentially returning a fake TLV without checking if the remaining packet length can hold the full 18-byte struct ipv6_destopt_hao. Could mip6_addr_swap() then perform a 16-byte swap that extends past the end of the packet data into skb_shared_info? Should the cb array also be cleared in ip6_err_gen_icmpv6_unreach() and ip6ip6_err() to prevent this? This patch implements the first suggestion. I am not sure if ip6ip6_err() needs to be changed. A separate patch would be better anyway.		
CVE-2026-43039	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ti: icssg-prueth: fix missing data copy and wrong recycle in ZC RX dispatch emac_dispatch_skb_zc() allocates a new skb via napi_alloc_skb() but never copies the packet data from the XDP buffer into it. The skb is passed up the stack containing uninitialized heap memory instead of the actual received packet, leaking kernel heap contents to userspace. Copy the received packet data from the XDP buffer into the skb using skb_copy_to_linear_data(). Additionally, remove the skb_mark_for_recycle() call since the skb is backed by the NAPI page frag allocator, not page_pool. Marking a non-page_pool skb for recycle causes the free path to return pages to a page_pool that does not own them, corrupting page_pool state. The non-ZC path (emac_rx_packet) does not have these issues because it uses napi_build_skb() to wrap the existing page_pool page directly, requiring no copy, and correctly marks for recycle since the page comes from page_pool_dev_alloc_pages().	2026-05-01	9.8
CVE-2026-7321	mozilla - multiple products	Sandbox escape due to incorrect boundary conditions in the WebRTC: Networking component. This vulnerability was fixed in Firefox 150, Thunderbird 150, Firefox ESR 140.10.1, and Thunderbird 140.10.1.	2026-04-28	9.6
CVE-2026-7333	google - chrome	Use after free in GPU in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	9.6
CVE-2026-33454	apache - multiple products	The Camel-Mail component is vulnerable to Camel message header injection. The custom header filter strategy used by the component (MailHeaderFilterStrategy) only filters the 'out' direction via setOutFilterStartsWith, while it does not configure the 'in' direction via setInFilterStartsWith. As a result, when a Camel application consumes mail through camel-mail (for example via from(\"imap://...\") or from(\"pop3://...\")) the inbound filter check is skipped and Camel-prefixed MIME headers are mapped unfiltered into the Exchange. An attacker who can deliver an email to a mailbox monitored by such a consumer can inject Camel-specific headers that, for some Camel components downstream of the mail consumer (such as camel-bean, camel-exec, or camel-sql), can alter the behaviour of the route. This is the same pattern that was previously addressed in camel-undertow (CVE-2025-30177) and the broader incoming-header filter (CVE-2025-27636 and CVE-2025-29891). This issue affects Apache Camel: from 3.0.0 before 4.14.6, from 4.15.0 before 4.18.1. Users are recommended to upgrade to version 4.19.0, which fixes the issue. If users are on the 4.18.x LTS releases stream, then they are suggested to upgrade to 4.18.1. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.6.	2026-04-27	9.4
CVE-2026-40976	vmware - spring_boot	In certain circumstances, Spring Boot's default web security is ineffective allowing unauthorized access to all endpoints. For an application to be vulnerable, it must: be a servlet-based web application; have no Spring Security configuration of its own and rely on the default web security filter chain; depend on spring-boot-actuator-autoconfigure; not depend on spring-boot-health. If any of the above does not apply, the application is not vulnerable. Affected: Spring Boot 4.0.0–4.0.5; upgrade to 4.0.6 or later per vendor advisory.	2026-04-28	9.1
CVE-2026-42523	jenkins - github	Jenkins GitHub Plugin 1.46.0 and earlier improperly processes the current job URL as part of JavaScript implementing validation of the feature "GitHub hook trigger for GITScm polling", resulting in a stored cross-site scripting (XSS) vulnerability exploitable by non-anonymous attackers with Overall/Read permission.	2026-04-29	9
CVE-2026-40473	apache - multiple products	The camel-mina component's MinaConverter.toObjectInput(IoBuffer) type converter wraps an IoBuffer in a java.io.ObjectInputStream without applying any ObjectInputFilter or class-loading restrictions. When a Camel route uses camel-mina as a TCP or UDP consumer and requests conversion to ObjectInput (for example via getBody(ObjectInput.class) or @Body ObjectInput), an attacker sending a crafted serialized Java object over the network to the MINA consumer port can trigger arbitrary code execution in the context of the application during readObject(). This issue affects Apache Camel: from 3.0.0 before 4.14.6, from 4.15.0 before 4.18.2, from 4.19.0 before 4.20.0.	2026-04-27	8.8

		Users are recommended to upgrade to version 4.20.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.6. If users are on the 4.18.x releases stream, then they are suggested to upgrade to 4.18.2.		
CVE-2026-40858	apache - multiple products	The camel-infinispan component's ProtoStream-based remote aggregation repository deserializes data read from a remote Infinispan cache using java.io.ObjectInputStream without applying any ObjectInputFilter. An attacker who can write to the Infinispan cache used by a Camel application can inject a crafted serialized Java object that, when read during normal aggregation repository operations such as get or recover, results in arbitrary code execution in the context of the application. This issue affects Apache Camel: from 4.0.0 before 4.14.7, from 4.15.0 before 4.18.2, from 4.19.0 before 4.20.0. Users are recommended to upgrade to version 4.20.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.7. If users are on the 4.18.x releases stream, then they are suggested to upgrade to 4.18.2. The JIRA ticket: https://issues.apache.org/jira/browse/CAMEL-23322 refers to the various commits that resolved the issue, and have more details. This issue follows the same class of vulnerability previously addressed in CVE-2024-22369, CVE-2024-23114 and CVE-2026-25747.	2026-04-27	8.8
CVE-2026-27172	apache - multiple products	The ConsulRegistry in the camel-consul component (class org.apache.camel.component.consul.ConsulRegistry and its inner ConsulRegistryUtils.deserialize method) read Java-serialized values from the Consul KV store and passed them to ObjectInputStream.readObject() without configuring an ObjectInputFilter. An attacker who can write to the Consul KV store backing a Camel ConsulRegistry instance could inject a malicious serialized Java object that is deserialized the next time Camel performs a lookup against that registry, leading to arbitrary code execution in the Camel process. The issue mirrors the class of vulnerability already addressed for other Camel components in CVE-2024-22369, CVE-2024-23114 and CVE-2026-25747, and was overlooked during the original remediation of those CVEs. This issue affects Apache Camel: from 3.0.0 before 4.14.6, from 4.15.0 before 4.18.1. Users are recommended to upgrade to version 4.19.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, then they are suggested to upgrade to 4.14.6. If users are on the 4.18.x releases stream, then they are suggested to upgrade to 4.18.1.	2026-04-27	8.8
CVE-2026-40978	vmware - multiple products	SQL injection vulnerability in Spring AI's `CosmosDBVectorStore` allows attackers to execute arbitrary SQL queries via crafted document IDs. Affected versions: Spring AI: 1.0.0 - 1.0.5 (fixed in 1.0.6), 1.1.0 - 1.1.4 (fixed in 1.1.5)	2026-04-28	8.8
CVE-2026-7334	google - chrome	Use after free in Views in Google Chrome on Mac prior to 147.0.7727.138 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7335	google - chrome	Use after free in media in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7336	google - chrome	Use after free in WebRTC in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7337	google - chrome	Type Confusion in V8 in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7339	google - chrome	Heap buffer overflow in WebRTC in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: Medium)	2026-04-28	8.8
CVE-2026-7341	google - chrome	Use after free in WebRTC in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7342	google - chrome	Use after free in WebView in Google Chrome on Android prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7344	google - chrome	Use after free in Accessibility in Google Chrome on Windows prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Critical)	2026-04-28	8.8
CVE-2026-7348	google - chrome	Use after free in Codecs in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7354	google - chrome	Out of bounds read and write in Angle in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7355	google - chrome	Use after free in Media in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-04-28	8.8
CVE-2026-7356	google - chrome	Use after free in Navigation in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7358	google - chrome	Use after free in Animation in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8

CVE-2026-7359	google - chrome	Use after free in ANGLE in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.8
CVE-2026-7361	google - chrome	Use after free in iOS in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: Critical)	2026-04-28	8.8
CVE-2026-7363	google - chrome	Use after free in Canvas in Google Chrome on Linux, ChromeOS prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Critical)	2026-04-28	8.8
CVE-2026-6389	ibm - turbonomic_prometurbo_agent	IBM Turbonomic prometurbo agent 8.16.0 through 8.17.6 IBM Turbonomic Application Resource Management grants excessive cluster-wide permissions, including unrestricted read access to all secrets. An attacker that compromises the operator or its service account can exfiltrate sensitive credentials, escalate privileges, and potentially achieve full cluster compromise.	2026-04-30	8.8
CVE-2026-6543	ibm - Langflow Desktop	IBM Langflow Desktop 1.0.0 through 1.8.4 Langflow allows an attacker to execute arbitrary commands with the privileges of the process running Langflow. This allows reading sensitive environment variables (API keys, DB credentials), modifying files, or launching further attacks on the internal network.	2026-04-30	8.8
CVE-2026-31706	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate num_aces and harden ACE walk in smb_inherit_dacl() smb_inherit_dacl() trusts the on-disk num_aces value from the parent directory's DACL xattr and uses it to size a heap allocation: aces_base = kmalloc(sizeof(struct smb_ace) * num_aces * 2, ...); num_aces is a u16 read from le16_to_cpu(parent_pdacl->num_aces) without checking that it is consistent with the declared pdacl_size. An authenticated client whose parent directory's security.NTACL is tampered (e.g. via offline xattr corruption or a concurrent path that bypasses parse_dacl()) can present num_aces = 65535 with minimal actual ACE data. This causes a ~8 MB allocation (not kzalloc, so uninitialized) that the subsequent loop only partially populates, and may also overflow the three-way size_t multiply on 32-bit kernels. Additionally, the ACE walk loop uses the weaker offsetof(struct smb_ace, access_req) minimum size check rather than the minimum valid on-wire ACE size, and does not reject ACEs whose declared size is below the minimum. Reproduced on UML + KASAN + LOCKDEP against the real ksmbd code path. A legitimate mount.cifs client creates a parent directory over SMB (ksmbd writes a valid security.NTACL xattr), then the NTACL blob on the backing filesystem is rewritten to set num_aces = 0xFFFF while keeping the posix_acl_hash bytes intact so ksmbd_vfs_get_sd_xattr()'s hash check still passes. A subsequent SMB2 CREATE of a child under that parent drives smb2_open() into smb_inherit_dacl() (share has "vfs objects = acl_xattr" set), which fails the page allocator: WARNING: mm/page_alloc.c:5226 at __alloc_frozen_pages_noprof+0x46c/0x9c0 Workqueue: ksmbd-io handle_ksmbd_work __alloc_frozen_pages_noprof+0x46c/0x9c0 ___kmalloc_large_node+0x68/0x130 __kmalloc_large_node_noprof+0x24/0x70 __kmalloc_noprof+0x4c9/0x690 smb_inherit_dacl+0x394/0x2430 smb2_open+0x595d/0xabe0 handle_ksmbd_work+0x3d3/0x1140 With the patch applied the added guard rejects the tampered value with -EINVAL before any large allocation runs, smb2_open() falls back to smb2_create_sd_buffer(), and the child is created with a default SD. No warning, no splat. Fix by: 1. Validating num_aces against pdacl_size using the same formula applied in parse_dacl(). 2. Replacing the raw kmalloc(sizeof * num_aces * 2) with kmalloc_array(num_aces * 2, sizeof(...)) for overflow-safe allocation. 3. Tightening the per-ACE loop guard to require the minimum valid ACE size (offsetof(smb_ace, sid) + CIFS_SID_BASE_SIZE) and rejecting under-sized ACEs, matching the hardening in smb_check_perm_dacl() and parse_dacl(). v1 -> v2: - Replace the synthetic test-module splat in the changelog with a real-path UML + KASAN reproduction driven through mount.cifs and	2026-05-01	8.8

		SMB2 CREATE; Namjae flagged the kcifs3_test_inherit_dacl_old name in v1 since it does not exist in ksmbd. - Drop the commit-hash citation from the code comment per Namjae's review; keep the parse_dacl() pointer.		
CVE-2026-31709	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: smb: client: validate the whole DACL before rewriting it in cifsacl build_sec_desc() and id_mode_to_cifs_acl() derive a DACL pointer from a server-supplied dacloffset and then use the incoming ACL to rebuild the chmod/chown security descriptor. The original fix only checked that the struct smb_acl header fits before reading dacl_ptr->size or dacl_ptr->num_aces. That avoids the immediate header-field OOB read, but the rewrite helpers still walk ACEs based on pdacl->num_aces with no structural validation of the incoming DACL body. A malicious server can return a truncated DACL that still contains a header, claims one or more ACEs, and then drive replace_sids_and_copy_aces() or set_chmod_dacl() past the validated extent while they compare or copy attacker-controlled ACEs. Factor the DACL structural checks into validate_dacl(), extend them to validate each ACE against the DACL bounds, and use the shared validator before the chmod/chown rebuild paths. parse_dacl() reuses the same validator so the read-side parser and write-side rewrite paths agree on what constitutes a well-formed incoming DACL.	2026-05-01	8.8
CVE-2026-31717	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate owner of durable handle on reconnect Currently, ksmbd does not verify if the user attempting to reconnect to a durable handle is the same user who originally opened the file. This allows any authenticated user to hijack an orphaned durable handle by predicting or brute-forcing the persistent ID. According to MS-SMB2, the server MUST verify that the SecurityContext of the reconnect request matches the SecurityContext associated with the existing open. Add a durable_owner structure to ksmbd_file to store the original opener's UID, GID, and account name. and catpure the owner information when a file handle becomes orphaned. and implementing ksmbd_vfs_compare_durable_owner() to validate the identity of the requester during SMB2_CREATE (DHnC).	2026-05-01	8.8
CVE-2026-31735	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommupt: Fix short gather if the unmap goes into a large mapping unmap has the odd behavior that it can unmap more than requested if the ending point lands within the middle of a large or contiguous IOPTE. In this case the gather should flush everything unmapped which can be larger than what was requested to be unmapped. The gather was only flushing the range requested to be unmapped, not extending to the extra range, resulting in a short invalidation if the caller hits this special condition. This was found by the new invalidation/gather test I am adding in preparation for ARMv8. Claude deduced the root cause. As far as I remember nothing relies on unmapping a large entry, so this is likely not a triggerable bug.	2026-05-01	8.8
CVE-2026-31739	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: tegra - Add missing CRYPTO_ALG_ASYNC The tegra crypto driver failed to set the CRYPTO_ALG_ASYNC on its asynchronous algorithms, causing the crypto API to select them for users that request only synchronous algorithms. This causes crashes (at least). Fix this by adding the flag like what the other drivers do. Also remove the unnecessary CRYPTO_ALG_TYPE_* flags, since those just get ignored and overridden by the registration function anyway.	2026-05-01	8.8
CVE-2026-31773	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SMP: derive legacy responder STK authentication from MITM state The legacy responder path in smp_random() currently labels the stored STK as authenticated whenever pending_sec_level is BT_SECURITY_HIGH. That reflects what the local service requested, not what the pairing flow actually achieved.	2026-05-01	8.8

		For Just Works/Confirm legacy pairing, SMP_FLAG_MITM_AUTH stays clear and the resulting STK should remain unauthenticated even if the local side requested HIGH security. Use the established MITM state when storing the responder STK so the key metadata matches the pairing result. This also keeps the legacy path aligned with the Secure Connections code, which already treats JUST_WORKS/JUST_CFM as unauthenticated.		
CVE-2026-43018	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: fix potential UAF in hci_le_remote_conn_param_req_evt hci_conn lookup and field access must be covered by hdev lock in hci_le_remote_conn_param_req_evt, otherwise it's possible it is freed concurrently. Extend the hci_dev_lock critical section to cover all conn usage.	2026-05-01	8.8
CVE-2026-43048	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: core: Mitigate potential OOB by removing bogus memset() The memset() in hid_report_raw_event() has the good intention of clearing out bogus data by zeroing the area from the end of the incoming data string to the assumed end of the buffer. However, as we have previously seen, doing so can easily result in OOB reads and writes in the subsequent thread of execution. The current suggestion from one of the HID maintainers is to remove the memset() and simply return if the incoming event buffer size is not large enough to fill the associated report. Suggested-by Benjamin Tissoires <bentiss@kernel.org> [bentiss: changed the return value]	2026-05-01	8.8
CVE-2026-41636	apache - thrift	Uncontrolled Recursion vulnerability in Apache Thrift Node.js bindings This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	8.7
CVE-2026-7607	trendnet - tew-821dap_firmware	A security vulnerability has been detected in TRENDnet TEW-821DAP 1.12B01. Impacted is the function auto_update_firmware of the component Firmware Udpate. The manipulation of the argument str leads to buffer overflow. The attack may be initiated remotely. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	8.7
CVE-2026-40967	vmware - multiple products	In Spring AI, various FilterExpressionConverter implementations accept a filter expression object and translate them to specific vector store query languages. In several cases, keys and values are not properly escaped, leading to the ability to alter the query. Affected versions: Spring AI: 1.0.0 - 1.0.5 (fixed in 1.0.6), 1.1.0 - 1.1.4 (fixed in 1.1.5)	2026-04-28	8.6
CVE-2026-7345	google - chrome	Insufficient validation of untrusted input in Feedback in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.3
CVE-2026-7350	google - chrome	Use after free in WebMIDI in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.3
CVE-2026-7352	google - chrome	Use after free in Media in Google Chrome on Android prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.3
CVE-2026-7353	google - chrome	Heap buffer overflow in Skia in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.3

CVE-2026-31712	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ksmbd: require minimum ACE size in smb_check_perm_dacl()
		
		Both ACE-walk loops in smb_check_perm_dacl() only guard against an under-sized remaining buffer, not against an ACE whose declared `ace->size` is smaller than the struct it claims to describe:
		
		if (offsetof(struct smb_ace, access_req) > aces_size)
		break;
		ace_size = le16_to_cpu(ace->size);
		if (ace_size > aces_size)
		break;
		
		The first check only requires the 4-byte ACE header to be in bounds; it does not require access_req (4 bytes at offset 4) to be readable. An attacker who has set a crafted DACL on a file they own can declare
			2026-05-01	8.3

		ace->size == 4 with aces_size == 4, pass both checks, and then granted |= le32_to_cpu(ace->access_req); /* upper loop */ compare_sids(&sid, &ace->sid); /* lower loop */ reads access_req at offset 4 (OOB by up to 4 bytes) and ace->sid at offset 8 (OOB by up to CIFS_SID_BASE_SIZE + SID_MAX_SUB_AUTHORITIES * 4 bytes). Tighten both loops to require ace_size >= offsetof(struct smb_ace, sid) + CIFS_SID_BASE_SIZE which is the smallest valid on-wire ACE layout (4-byte header + 4-byte access_req + 8-byte sid base with zero sub-auths). Also reject ACEs whose sid.num_subauth exceeds SID_MAX_SUB_AUTHORITIES before letting compare_sids() dereference sub_auth[] entries. parse_sec_desc() already enforces an equivalent check (lines 441-448); smb_check_perm_dacl() simply grew weaker validation over time. Reachability: authenticated SMB client with permission to set an ACL on a file. On a subsequent CREATE against that file, the kernel walks the stored DACL via smb_check_perm_dacl() and triggers the OOB read. Not pre-auth, and the OOB read is not reflected to the attacker, but KASAN reports and kernel state corruption are possible.		
CVE-2026-40022	apache - multiple products	When authentication is enabled on the Apache Camel embedded HTTP server or embedded management server (camel-platform-http-main) and a non-root context path such as /api or /admin is configured via camel.server.path or camel.management.path, the BasicAuthenticationConfigurer and JWTAuthenticationConfigurer classes derive the authentication path from properties.getPath() when camel.server.authenticationPath / camel.management.authenticationPath is not explicitly set. Combined with the Vert.x sub-router mounting model - the sub-router is mounted at _path_* and the authentication handler is registered inside the sub-router at the resolved path - this causes the authentication handler to match only the exact configured context path, not its subpaths. Unauthenticated requests to subpaths such as /api/_route_ or /admin/observe/info therefore reach protected business routes and management endpoints without being challenged for credentials. The /observe/info endpoint can disclose runtime metadata such as the user, working directory, home directory, process ID, JVM and operating system information. This issue affects Apache Camel: from 4.14.1 before 4.14.6, from 4.18.0 before 4.18.2. Users are recommended to upgrade to version 4.20.0, which fixes the issue. If users are on the 4.14.x LTS releases stream, they are suggested to upgrade to 4.14.6. If users are on the 4.18.x LTS releases stream, they are suggested to upgrade to 4.18.2.	2026-04-27	8.2
CVE-2026-41604	apache - thrift	Out-of-bounds Read vulnerability in Apache Thrift. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	8.2
CVE-2026-6785	mozilla - multiple products	Memory safety bugs present in Firefox ESR 115.34, Firefox ESR 140.9, Thunderbird ESR 140.9, Firefox 149 and Thunderbird 149. Some of these bugs showed evidence of memory corruption and we presume that with enough effort some of these could have been exploited to run arbitrary code. This vulnerability was fixed in Firefox 150, Firefox ESR 115.35, Firefox ESR 140.10, Thunderbird 150, and Thunderbird 140.10.	2026-04-26	8.1
CVE-2026-6786	mozilla - multiple products	Memory safety bugs present in Firefox ESR 140.9, Thunderbird ESR 140.9, Firefox 149 and Thunderbird 149. Some of these bugs showed evidence of memory corruption and we presume that with enough effort some of these could have been exploited to run arbitrary code. This vulnerability was fixed in Firefox 150, Firefox ESR 140.10, Thunderbird 150, and Thunderbird 140.10.	2026-04-26	8.1
CVE-2026-7346	google - chrome	Inappropriate implementation in Tint in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to perform out of bounds memory access via a crafted HTML page. (Chromium security severity: High)	2026-04-28	8.1
CVE-2026-7347	google - chrome	Use after free in Chromoting in Google Chrome prior to 147.0.7727.138 allowed a remote attacker to execute arbitrary code via malicious network traffic. (Chromium security severity: High)	2026-04-28	8.1
CVE-2026-31708	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix OOB read in smb2_ioctl_query_info QUERY_INFO path smb2_ioctl_query_info() has two response-copy branches: PASSTHRU_FSCTL and the default QUERY_INFO path. The QUERY_INFO branch clamps qi.input_buffer_length to the server-reported OutputBufferLength and then copies qi.input_buffer_length bytes from qi_rsp->Buffer to userspace, but it never verifies that the flexible-array payload actually fits within rsp_iov[1].iov_len. A malicious server can return OutputBufferLength larger than the actual QUERY_INFO response, causing copy_to_user() to walk past the response buffer and expose adjacent kernel heap to userspace. Guard the QUERY_INFO copy with a bounds check on the actual Buffer	2026-05-01	8.1

		payload. Use struct_size(qi_rsp, Buffer, qi.input_buffer_length) rather than an open-coded addition so the guard cannot overflow on 32-bit builds.		
CVE-2026-31771	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: move wake reason storage into validated event handlers hci_store_wake_reason() is called from hci_event_packet() immediately after stripping the HCI event header but before hci_event_func() enforces the per-event minimum payload length from hci_ev_table. This means a short HCI event frame can reach bacpy() before any bounds check runs. Rather than duplicating skb parsing and per-event length checks inside hci_store_wake_reason(), move wake-address storage into the individual event handlers after their existing event-length validation has succeeded. Convert hci_store_wake_reason() into a small helper that only stores an already-validated bdaddr while the caller holds hci_dev_lock(). Use the same helper after hci_event_func() with a NULL address to preserve the existing unexpected-wake fallback semantics when no validated event handler records a wake address. Annotate the helper with __must_hold(&hdev->lock) and add lockdep_assert_held(&hdev->lock) so future call paths keep the lock contract explicit. Call the helper from hci_conn_request_evt(), hci_conn_complete_evt(), hci_sync_conn_complete_evt(), le_conn_complete_evt(), hci_le_adv_report_evt(), hci_le_ext_adv_report_evt(), hci_le_direct_adv_report_evt(), hci_le_pa_sync_established_evt(), and hci_le_past_received_evt().	2026-05-01	8.1
CVE-2026-31779	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: fix potential out-of-bounds read in iwl_mvm_nd_match_info_handler() The memcpy function assumes the dynamic array notif->matches is at least as large as the number of bytes to copy. Otherwise, results->matches may contain unwanted data. To guarantee safety, extend the validation in one of the checks to ensure sufficient packet length. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2026-05-01	8.1
CVE-2026-43051	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: fix out-of-bounds read in wacom_intuos_bt_irq The wacom_intuos_bt_irq() function processes Bluetooth HID reports without sufficient bounds checking. A maliciously crafted short report can trigger an out-of-bounds read when copying data into the wacom structure. Specifically, report 0x03 requires at least 22 bytes to safely read the processed data and battery status, while report 0x04 (which falls through to 0x03) requires 32 bytes. Add explicit length checks for these report IDs and log a warning if a short report is received.	2026-05-01	8.1
CVE-2026-42524	jenkins - html_publisher	Jenkins HTML Publisher Plugin 427 and earlier does not escape job name and URL in the legacy wrapper file, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers with Item/Configure permission.	2026-04-29	8
CVE-2026-0204	sonicwall - sonicos	A vulnerability in the access control mechanism of SonicOS may allow certain management interface functions to be accessible under specific conditions.	2026-04-29	8
CVE-2026-40048	apache - multiple products	The Camel-PQC FileBasedKeyLifecycleManager class deserializes the contents of `<keyId>.key` files in the configured key directory using java.io.ObjectInputStream without applying any ObjectInputFilter or class-loading restrictions. The cast to `java.security.KeyPair` is evaluated only after `readObject()` has already returned, so any `readObject()` side effects in the deserialized object run before the type check. An attacker who can write to the key directory used by a Camel application — for example through a path traversal into the directory, misconfigured filesystem permissions on the volume where keys are stored, a compromised key provisioning pipeline, or a symlink attack — can place a crafted serialized Java object that, when deserialized during normal key lifecycle operations, results in arbitrary code execution in the context of the application. This issue affects Apache Camel: from 4.19.0 before 4.20.0, from 4.18.0 before 4.18.2. Users are recommended to upgrade to version 4.20.0, which fixes the issue by replacing java.io.ObjectInputStream-based key and metadata storage with standard PKCS#8 (private key) / X.509 SubjectPublicKeyInfo (public key) Base64 JSON encoding. For users on the 4.18.x LTS releases stream, upgrade to 4.18.2.	2026-04-27	7.8
CVE-2026-31686	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/kasan: fix double free for kasan pXds	2026-04-27	7.8

		kasan_free_pxd() assumes the page table is always struct page aligned. But that's not always the case for all architectures. E.g. In case of powerpc with 64K pagesize, PUD table (of size 4096) comes from slab cache named pgtable-2^9. Hence instead of page_to_virt(pxd_page()) let's just directly pass the start of the pxd table which is passed as the 1st argument. This fixes the below double free kasan issue seen with PMEM: radix-mmu: Mapped 0x0000047d10000000-0x0000047f90000000 with 2.00 MiB pages ========== BUG: KASAN: double-free in kasan_remove_zero_shadow+0x9c4/0xa20 Free of addr c0000003c38e0000 by task ndctl/2164 CPU: 34 UID: 0 PID: 2164 Comm: ndctl Not tainted 6.19.0-rc1-00048-gea1013c15392 #157 VOLUNTARY Hardware name: IBM,9080-HEX POWER10 (architected) 0x800200 0xf000006 of:IBM,FW1060.00 (NH1060_012) hv:phyp pSeries Call Trace: dump_stack_lvl+0x88/0xc4 (unreliable) print_report+0x214/0x63c kasan_report_invalid_free+0xe4/0x110 check_slab_allocation+0x100/0x150 kmem_cache_free+0x128/0x6e0 kasan_remove_zero_shadow+0x9c4/0xa20 memunmap_pages+0x2b8/0x5c0 devm_action_release+0x54/0x70 release_nodes+0xc8/0x1a0 devres_release_all+0xe0/0x140 device_unbind_cleanup+0x30/0x120 device_release_driver_internal+0x3e4/0x450 unbind_store+0xfc/0x110 drv_attr_store+0x78/0xb0 sysfs_kf_write+0x114/0x140 kernfs_fop_write_iter+0x264/0x3f0 vfs_write+0x3bc/0x7d0 ksys_write+0xa4/0x190 system_call_exception+0x190/0x480 system_call_vectored_common+0x15c/0x2ec ---- interrupt: 3000 at 0x7fff93b3d3f4 NIP: 00007fff93b3d3f4 LR: 00007fff93b3d3f4 CTR: 0000000000000000 REGS: c0000003f1b07e80 TRAP: 3000 Not tainted (6.19.0-rc1-00048-gea1013c15392) MSR: 800000000280f033 <SF,VEC,VSX,EE,PR,FP,ME,IR,DR,RI,LE> CR: 48888208 XER: 00000000 <...> NIP [00007fff93b3d3f4] 0x7fff93b3d3f4 LR [00007fff93b3d3f4] 0x7fff93b3d3f4 ---- interrupt: 3000 The buggy address belongs to the object at c0000003c38e0000 which belongs to the cache pgtable-2^9 of size 4096 The buggy address is located 0 bytes inside of 4096-byte region [c0000003c38e0000, c0000003c38e1000) The buggy address belongs to the physical page: page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x3c38c head: order:2 mapcount:0 entire_mapcount:0 nr_pages_mapped:0 pincount:0 memcg:c0000003bfd63e01 flags: 0x63ffff800000040(head|node=6|zone=0|lastcpupid=0x7ffff) page_type: f5(slab) raw: 063ffff800000040 c000000140058980 5deadbeef0000122 0000000000000000 raw: 0000000000000000 0000000080200020 00000000f5000000 c0000003bfd63e01 head: 063ffff800000040 c000000140058980 5deadbeef0000122 0000000000000000 head: 0000000000000000 0000000080200020 00000000f5000000 c0000003bfd63e01 head: 063ffff800000002 c00c000000f0e301 00000000ffffffff 00000000ffffffff head: ffffffffffffffff 0000000000000000 00000000ffffffff 0000000000000004 page dumped because: kasan: bad access detected [138.953636] [T2164] Memory state around the buggy address: [138.953643] [T2164] c0000003c38dff00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [138.953652] [T2164] c0000003c38dff80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [138.953661] [T2164] >c0000003c38e0000: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [138.953669] [T2164] ^ [138.953675] [T2164] c0000003c38e0080: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [138.953684] [T2164] c0000003c38e0100: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [138.953692] [T2164] ========== [138.953701] [T2164] Disabling lock debugging due to kernel taint		
CVE-2026-31688	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved:	2026-04-27	7.8

		driver core: enforce device_lock for driver_match_device() Currently, driver_match_device() is called from three sites. One site (__device_attach_driver) holds device_lock(dev), but the other two (bind_store and __driver_attach) do not. This inconsistency means that bus match() callbacks are not guaranteed to be called with the lock held. Fix this by introducing driver_match_device_locked(), which guarantees holding the device lock using a scoped guard. Replace the unlocked calls in bind_store() and __driver_attach() with this new helper. Also add a lock assertion to driver_match_device() to enforce this guarantee. This consistency also fixes a known race condition. The driver_override implementation relies on the device_lock, so the missing lock led to the use-after-free (UAF) reported in Bugzilla for buses using this field. Stress testing the two newly locked paths for 24 hours with CONFIG_PROVE_LOCKING and CONFIG_LOCKDEP enabled showed no UAF recurrence and no lockdep warnings.		
CVE-2026-31690	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: firmware: thead: Fix buffer overflow and use standard endian macros Addresses two issues in the TH1520 AON firmware protocol driver: 1. Fix a potential buffer overflow where the code used unsafe pointer arithmetic to access the 'mode' field through the 'resource' pointer with an offset. This was flagged by Smatch static checker as: "buffer overflow 'data' 2 <= 3" 2. Replace custom RPC_SET_BE* and RPC_GET_BE* macros with standard kernel endianness conversion macros (cpu_to_be16, etc.) for better portability and maintainability. The functionality was re-tested with the GPU power-up sequence, confirming the GPU powers up correctly and the driver probes successfully. [12.702370] powervr ffef400000.gpu: [drm] loaded firmware powervr/rogue_36.52.104.182_v1.fw [12.711043] powervr ffef400000.gpu: [drm] FW version v1.0 (build 6645434 OS) [12.719787] [drm] Initialized powervr 1.0.0 for ffef400000.gpu on minor 0	2026-04-27	7.8
CVE-2026-31786	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Buffer overflow in drivers/xen/sys-hypervisor.c The build id returned by HYPERVISOR_xen_version(XENVER_build_id) is neither NUL terminated nor a string. The first causes a buffer overflow as sprintf in buildid_show will read and copy till it finds a NUL. 00000000 f4 91 51 f4 dd 38 9e 9d 65 47 52 eb 10 71 db 50 |..Q..8..eGR..q.P| 00000010 b9 a8 01 42 6f 2e 32 |...Bo.2| 00000017 So use a memcpy instead of sprintf to have the correct value: 00000000 f4 91 51 f4 dd 00 9e 9d 65 47 52 eb 10 71 db 50 |..Q.....eGR..q.P| 00000010 b9 a8 01 42 |...B| 00000014 (the above have a hack to embed a zero inside and check it's returned correctly). This is XSA-485 / CVE-2026-31786	2026-04-30	7.8
CVE-2026-31787	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xen/privcmd: fix double free via VMA splitting privcmd_vm_ops defines .close (privcmd_close), but neither .may_split nor .open. When userspace does a partial munmap() on a privcmd mapping, the kernel splits the VMA via __split_vma(). Since may_split is NULL, the split is allowed. vm_area_dup() copies vm_private_data (a pages array allocated in alloc_empty_pages()) into the new VMA without any fixup, because there is no .open callback.	2026-04-30	7.8

		Both VMAs now point to the same pages array. When the unmapped portion is closed, privcmd_close() calls: - xen_unmap_domain_gfn_range() - xen_free_unpopulated_pages() - kvfree(pages) The surviving VMA still holds the dangling pointer. When it is later destroyed, the same sequence runs again, which leads to a double free. Fix this issue by adding a .may_split callback denying the VMA split. This is XSA-487 / CVE-2026-31787		
CVE-2026-31693	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cifs: some missing initializations on replay In several places in the code, we have a label to signify the start of the code where a request can be replayed if necessary. However, some of these places were missing the necessary reinitializations of certain local variables before replay. This change makes sure that these variables get initialized after the label.	2026-04-30	7.8
CVE-2026-31694	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fuse: reject oversized dirents in page cache fuse_add_dirent_to_cache() computes a serialized dirent size from the server-controlled namelen field and copies the dirent into a single page-cache page. The existing logic only checks whether the dirent fits in the remaining space of the current page and advances to a fresh page if not. It never checks whether the dirent itself exceeds PAGE_SIZE. As a result, a malicious FUSE server can return a dirent with namelen=4095, producing a serialized record size of 4120 bytes. On 4 KiB page systems this causes memcpy() to overflow the cache page by 24 bytes into the following kernel page. Reject dirents that cannot fit in a single page before copying them into the readdir cache.	2026-05-01	7.8
CVE-2026-31695	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: virt_wifi: remove SET_NETDEV_DEV to avoid use-after-free Currently we execute `SET_NETDEV_DEV(dev, &priv->lowerdev->dev)` for the virt_wifi net devices. However, unregistering a virt_wifi device in netdev_run_todo() can happen together with the device referenced by SET_NETDEV_DEV(). It can result in use-after-free during the ethtool operations performed on a virt_wifi device that is currently being unregistered. Such a net device can have the `dev.parent` field pointing to the freed memory, but ethnl_ops_begin() calls `pm_runtime_get_sync(dev->dev.parent)`. Let's remove SET_NETDEV_DEV for virt_wifi to avoid bugs like this: ========== BUG: KASAN: slab-use-after-free in __pm_runtime_resume+0xe2/0xf0 Read of size 2 at addr ffff88810cfc46f8 by task pm/606 Call Trace: <TASK> dump_stack_lvl+0x4d/0x70 print_report+0x170/0x4f3 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 kasan_report+0xda/0x110 ? __pm_runtime_resume+0xe2/0xf0 ? __pm_runtime_resume+0xe2/0xf0 __pm_runtime_resume+0xe2/0xf0 ethnl_ops_begin+0x49/0x270 ethnl_set_features+0x23c/0xab0 ? __pfx_ethnl_set_features+0x10/0x10 ? kvm_sched_clock_read+0x11/0x20 ? local_clock_noinstr+0xf/0xf0 ? local_clock+0x10/0x30 ? kasan_save_track+0x25/0x60 ? __kasan_kmalloc+0x7f/0x90 ? genl_family_rcv_msg_attrs_parse.isra.0+0x150/0x2c0 genl_family_rcv_msg_doit+0x1e7/0x2c0	2026-05-01	7.8

		? __pfx_genl_family_rcv_msg_doit+0x10/0x10 ? __pfx_cred_has_capability.isra.0+0x10/0x10 ? stack_trace_save+0x8e/0xc0 genl_rcv_msg+0x411/0x660 ? __pfx_genl_rcv_msg+0x10/0x10 ? __pfx_ethnl_set_features+0x10/0x10 netlink_rcv_skb+0x121/0x380 ? __pfx_genl_rcv_msg+0x10/0x10 ? __pfx_netlink_rcv_skb+0x10/0x10 ? __pfx_down_read+0x10/0x10 genl_rcv+0x23/0x30 netlink_unicast+0x60f/0x830 ? __pfx_netlink_unicast+0x10/0x10 ? __pfx___alloc_skb+0x10/0x10 netlink_sendmsg+0x6ea/0xbc0 ? __pfx_netlink_sendmsg+0x10/0x10 ? __futex_queue+0x10b/0x1f0 ____sys_sendmsg+0x7a2/0x950 ? copy_msghdr_from_user+0x26b/0x430 ? __pfx_____sys_sendmsg+0x10/0x10 ? __pfx_copy_msghdr_from_user+0x10/0x10 ___sys_sendmsg+0xf8/0x180 ? __pfx____sys_sendmsg+0x10/0x10 ? __pfx_futex_wait+0x10/0x10 ? fdget+0x2e4/0x4a0 __sys_sendmsg+0x11f/0x1c0 ? __pfx___sys_sendmsg+0x10/0x10 do_syscall_64+0xe2/0x570 ? exc_page_fault+0x66/0xb0 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> This fix may be combined with another one in the ethtool subsystem: https://lore.kernel.org/all/20260322075917.254874-1-alex.popov@linux.com/T/#u		
CVE-2026-31696	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix missing validation of ticket length in non-XDR key preparsing In rxrpc_preparse(), there are two paths for parsing key payloads: the XDR path (for large payloads) and the non-XDR path (for payloads <= 28 bytes). While the XDR path (rxrpc_preparse_xdr_rxkad()) correctly validates the ticket length against AFSTOKEN_RK_TIX_MAX, the non-XDR path fails to do so. This allows an unprivileged user to provide a very large ticket length. When this key is later read via rxrpc_read(), the total token size (toksize) calculation results in a value that exceeds AFSTOKEN_LENGTH_MAX, triggering a WARN_ON(). [2001.302904] WARNING: CPU: 2 PID: 2108 at net/rxrpc/key.c:778 rxrpc_read+0x109/0x5c0 [rxrpc] Fix this by adding a check in the non-XDR parsing path of rxrpc_preparse() to ensure the ticket length does not exceed AFSTOKEN_RK_TIX_MAX, bringing it into parity with the XDR parsing logic.	2026-05-01	7.8
CVE-2026-31700	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/packet: fix TOCTOU race on mmap'd vnet_hdr in tpacket_snd() In tpacket_snd(), when PACKET_VNET_HDR is enabled, vnet_hdr points directly into the mmap'd TX ring buffer shared with userspace. The kernel validates the header via __packet_snd_vnet_parse() but then re-reads all fields later in virtio_net_hdr_to_skb(). A concurrent userspace thread can modify the vnet_hdr fields between validation and use, bypassing all safety checks. The non-TPACKET path (packet_snd()) already correctly copies vnet_hdr to a stack-local variable. All other vnet_hdr consumers in the kernel (tun.c, tap.c, virtio_net.c) also use stack copies. The TPACKET TX path is the only caller of virtio_net_hdr_to_skb() that reads directly from user-controlled shared memory. Fix this by copying vnet_hdr from the mmap'd ring buffer to a stack-local variable before validation and use, consistent with the approach used in packet_snd() and all other callers.	2026-05-01	7.8
CVE-2026-31702	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix use-after-free of sbi in f2fs_compress_write_end_io() In f2fs_compress_write_end_io(), dec_page_count(sbi, type) can bring the F2FS_WB_CP_DATA counter to zero, unblocking	2026-05-01	7.8

		f2fs_wait_on_all_pages() in f2fs_put_super() on a concurrent unmount CPU. The unmount path then proceeds to call f2fs_destroy_page_array_cache(sbi), which destroys sbi->page_array_slab via kmem_cache_destroy(), and eventually kfree(sbi). Meanwhile, the bio completion callback is still executing: when it reaches page_array_free(sbi, ...), it dereferences sbi->page_array_slab — a destroyed slab cache — to call kmem_cache_free(), causing a use-after-free. This is the same class of bug as CVE-2026-23234 (which fixed the equivalent race in f2fs_write_end_io() in data.c), but in the compressed writeback completion path that was not covered by that fix. Fix this by moving dec_page_count() to after page_array_free(), so that all sbi accesses complete before the counter decrement that can unblock unmount. For non-last folios (where atomic_dec_return on cic->pending_pages is nonzero), dec_page_count is called immediately before returning — page_array_free is not reached on this path, so there is no post-decrement sbi access. For the last folio, page_array_free runs while the F2FS_WB_CP_DATA counter is still nonzero (this folio has not yet decremented it), keeping sbi alive, and dec_page_count runs as the final operation.		

CVE-2026-31703	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		writeback: Fix use after free in inode_switch_wbs_work_fn()
		
		inode_switch_wbs_work_fn() has a loop like:
		
		wb_get(new_wb);
		while (1) {
		list = llist_del_all(&new_wb->switch_wbs_ctxs);
		/* Nothing to do? */
		if (!list)
		break;
		... process the items ...
		}
		
		Now adding of items to the list looks like:
		
		wb_queue_isw()
		if (llist_add(&isw->list, &wb->switch_wbs_ctxs))
		queue_work(isw_wq, &wb->switch_work);
		
		Because inode_switch_wbs_work_fn() loops when processing isw items, it can happen that wb->switch_work is pending while wb->switch_wbs_ctxs is empty. This is a problem because in that case wb can get freed (no isw items -> no wb reference) while the work is still pending causing use-after-free issues.
		
		We cannot just fix this by cancelling work when freeing wb because that could still trigger problematic 0 -> 1 transitions on wb refcount due to wb_get() in inode_switch_wbs_work_fn(). It could be all handled with more careful code but that seems unnecessarily complex so let's avoid that until it is proven that the looping actually brings practical benefit. Just remove the loop from inode_switch_wbs_work_fn() instead. That way when wb_queue_isw() queues work, we are guaranteed we have added the first item to wb->switch_wbs_ctxs and nobody is going to remove it (and drop the wb reference it holds) until the queued work runs.
			2026-05-01	7.8

CVE-2026-31715	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix UAF caused by decrementing sbi->nr_pages[] in f2fs_write_end_io() The xfstests case "generic/107" and syzbot have both reported a NULL pointer dereference. The concurrent scenario that triggers the panic is as follows: F2FS_WB_CP_DATA write callback umount - f2fs_write_checkpoint - f2fs_wait_on_all_pages(sbi, F2FS_WB_CP_DATA) - blk_mq_end_request - bio_endio - f2fs_write_end_io : dec_page_count(sbi, F2FS_WB_CP_DATA) : wake_up(&sbi->cp_wait) - kill_f2fs_super - kill_block_super - f2fs_put_super : iput(sbi->node_inode)	2026-05-01	7.8

		: sbi->node_inode = NULL : f2fs_in_warm_node_list - is_node_folio // sbi->node_inode is NULL and panic The root cause is that f2fs_put_super() calls iput(sbi->node_inode) and sets sbi->node_inode to NULL after sbi->nr_pages[F2FS_WB_CP_DATA] is decremented to zero. As a result, f2fs_in_warm_node_list() may dereference a NULL node_inode when checking whether a folio belongs to the node inode, leading to a panic. This patch fixes the issue by calling f2fs_in_warm_node_list() before decrementing sbi->nr_pages[F2FS_WB_CP_DATA], thus preventing the use-after-free condition.		
CVE-2026-31716	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: validate rec->used in journal-replay file record check check_file_record() validates rec->total against the record size but never validates rec->used. The do_action() journal-replay handlers read rec->used from disk and use it to compute memmove lengths: DeleteAttribute: memmove(attr, ..., used - asize - roff) CreateAttribute: memmove(..., attr, used - roff) change_attr_size: memmove(..., used - PtrOffset(rec, next)) When rec->used is smaller than the offset of a validated attribute, or larger than the record size, these subtractions can underflow allowing us to copy huge amounts of memory in to a 4kb buffer, generally considered a bad idea overall. This requires a corrupted filesystem, which isn't a threat model the kernel really needs to worry about, but checking for such an obvious out-of-bounds value is good to keep things robust, especially on journal replay Fix this up by bounding rec->used correctly. This is much like commit b2bc7c44ed17 ("fs/ntfs3: Fix slab-out-of-bounds read in DeleteIndexEntryRoot") which checked different values in this same switch statement.	2026-05-01	7.8

CVE-2026-31720	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		usb: gadget: f_uac1_legacy: validate control request size
		
		f_audio_complete() copies req->length bytes into a 4-byte stack variable:
		
		u32 data = 0;
		memcpy(&data, req->buf, req->length);
		
		req->length is derived from the host-controlled USB request path, which can lead to a stack out-of-bounds write.
		
		Validate req->actual against the expected payload size for the supported control selectors and decode only the expected amount of data.
		
		This avoids copying a host-influenced length into a fixed-size stack object.
			2026-05-01	7.8

CVE-2026-31729	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: validate connector number in ucsi_notify_common() The connector number extracted from CCI via UCSI_CCI_CONNECTOR() is a 7-bit field (0-127) that is used to index into the connector array in ucsi_connector_change(). However, the array is only allocated for the number of connectors reported by the device (typically 2-4 entries). A malicious or malfunctioning device could report an out-of-range connector number in the CCI, causing an out-of-bounds array access in ucsi_connector_change(). Add a bounds check in ucsi_notify_common(), the central point where CCI is parsed after arriving from hardware, so that bogus connector numbers are rejected before they propagate further.	2026-05-01	7.8
CVE-2026-31730	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: possible double-free of cctx->remote_heap fastrpc_init_create_static_process() may free cctx->remote_heap on the	2026-05-01	7.8

		err_map path but does not clear the pointer. Later, fastrpc_rpmsg_remove() frees cctx->remote_heap again if it is non-NULL, which can lead to a double-free if the INIT_CREATE_STATIC ioctl hits the error path and the rpmsg device is subsequently removed/unbound. Clear cctx->remote_heap after freeing it in the error path to prevent the later cleanup from freeing it again. This issue was found by an in-house analysis workflow that extracts AST-based information and runs static checks, with LLM assistance for triage, and was confirmed by manual code review. No hardware testing was performed.		
CVE-2026-31731	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Address thermal zone removal races with resume Since thermal_zone_pm_complete() and thermal_zone_device_resume() re-initialize the poll_queue delayed work for the given thermal zone, the cancel_delayed_work_sync() in thermal_zone_device_unregister() may miss some already running work items and the thermal zone may be freed prematurely [1]. There are two failing scenarios that both start with running thermal_pm_notify_complete() right before invoking thermal_zone_device_unregister() for one of the thermal zones. In the first scenario, there is a work item already running for the given thermal zone when thermal_pm_notify_complete() calls thermal_zone_pm_complete() for that thermal zone and it continues to run when thermal_zone_device_unregister() starts. Since the poll_queue delayed work has been re-initialized by thermal_pm_notify_complete(), the running work item will be missed by the cancel_delayed_work_sync() in thermal_zone_device_unregister() and if it continues to run past the freeing of the thermal zone object, a use-after-free will occur. In the second scenario, thermal_zone_device_resume() queued up by thermal_pm_notify_complete() runs right after the thermal_zone_exit() called by thermal_zone_device_unregister() has returned. The poll_queue delayed work is re-initialized by it before cancel_delayed_work_sync() is called by thermal_zone_device_unregister(), so it may continue to run after the freeing of the thermal zone object, which also leads to a use-after-free. Address the first failing scenario by ensuring that no thermal work items will be running when thermal_pm_notify_complete() is called. For this purpose, first move the cancel_delayed_work() call from thermal_zone_pm_complete() to thermal_zone_pm_prepare() to prevent new work from entering the workqueue going forward. Next, switch over to using a dedicated workqueue for thermal events and update the code in thermal_pm_notify() to flush that workqueue after thermal_pm_notify_prepare() has returned which will take care of all leftover thermal work already on the workqueue (that leftover work would do nothing useful anyway because all of the thermal zones have been flagged as suspended). The second failing scenario is addressed by adding a tz->state check to thermal_zone_device_resume() to prevent it from re-initializing the poll_queue delayed work if the thermal zone is going away. Note that the above changes will also facilitate relocating the suspend and resume of thermal zones closer to the suspend and resume of devices, respectively.	2026-05-01	7.8
CVE-2026-31742	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vt: discard stale unicode buffer on alt screen exit after resize When enter_alt_screen() saves vc_uni_lines into vc_saved_uni_lines and sets vc_uni_lines to NULL, a subsequent console resize via vc_do_resize() skips reallocating the unicode buffer because vc_uni_lines is NULL. However, vc_saved_uni_lines still points to the old buffer allocated for the original dimensions. When leave_alt_screen() later restores vc_saved_uni_lines, the buffer dimensions no longer match vc_rows/vc_cols. Any operation that iterates over the unicode buffer using the current dimensions (e.g. csi_J clearing the screen) will access memory out of bounds, causing a kernel oops: BUG: unable to handle page fault for address: 0x0000002000000020 RIP: 0010:csi_J+0x133/0x2d0 The faulting address 0x0000002000000020 is two adjacent u32 space	2026-05-01	7.8

		characters (0x20) interpreted as a pointer, read from the row data area past the end of the 25-entry pointer array in a buffer allocated for 80x25 but accessed with 240x67 dimensions. Fix this by checking whether the console dimensions changed while in the alternate screen. If they did, free the stale saved buffer instead of restoring it. The unicode screen will be lazily rebuilt via vc_uniscr_check() when next needed.		
CVE-2026-31743	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nvmem: zynqmp_nvmem: Fix buffer size in DMA and memcpy Buffer size used in dma allocation and memcpy is wrong. It can lead to undersized DMA buffer access and possible memory corruption. use correct buffer size in dma_alloc_coherent and memcpy.	2026-05-01	7.8
CVE-2026-31745	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: reset: gpio: fix double free in reset_add_gpio_aux_device() error path When __auxiliary_device_add() fails, reset_add_gpio_aux_device() calls auxiliary_device_uninit(adev). The device release callback reset_gpio_aux_device_release() frees adev, but the current error path then calls kfree(adev) again, causing a double free. Keep kfree(adev) for the auxiliary_device_init() failure path, but avoid freeing adev after auxiliary_device_uninit().	2026-05-01	7.8
CVE-2026-31747	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: me4000: Fix potential overrun of firmware buffer `me4000_xilinx_download()` loads the firmware that was requested by `request_firmware()`. It is possible for it to overrun the source buffer because it blindly trusts the file format. It reads a data stream length from the first 4 bytes into variable `file_length` and reads the data stream contents of length `file_length` from offset 16 onwards. Add a test to ensure that the supplied firmware is long enough to contain the header and the data stream. On failure, log an error and return `-EINVAL`. Note: The firmware loading was totally broken before commit ac584af59945 ("staging: comedi: me4000: fix firmware downloading"), but that is the most sensible target for this fix.	2026-05-01	7.8
CVE-2026-31748	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: me_daq: Fix potential overrun of firmware buffer `me2600_xilinx_download()` loads the firmware that was requested by `request_firmware()`. It is possible for it to overrun the source buffer because it blindly trusts the file format. It reads a data stream length from the first 4 bytes into variable `file_length` and reads the data stream contents of length `file_length` from offset 16 onwards. Although it checks that the supplied firmware is at least 16 bytes long, it does not check that it is long enough to contain the data stream. Add a test to ensure that the supplied firmware is long enough to contain the header and the data stream. On failure, log an error and return `-EINVAL`.	2026-05-01	7.8
CVE-2026-31758	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: usbtmc: Flush anchored URBs in usbtmc_release When calling usbtmc_release, pending anchored URBs must be flushed or killed to prevent use-after-free errors (e.g. in the HCD giveback path). Call usbtmc_draw_down() to allow anchored URBs to be completed.	2026-05-01	7.8
CVE-2026-31759	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: ulpi: fix double free in ulpi_register_interface() error path When device_register() fails, ulpi_register() calls put_device() on ulpi->dev. The device release callback ulpi_dev_release() drops the OF node reference and frees ulpi, but the current error path in ulpi_register_interface() then calls kfree(ulpi) again, causing a	2026-05-01	7.8

		double free. Let put_device() handle the cleanup through ulpi_dev_release() and avoid freeing ulpi again in ulpi_register_interface().		
CVE-2026-31761	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: gyro: mpu3050: Move iio_device_register() to correct location iio_device_register() should be at the end of the probe function to prevent race conditions. Place iio_device_register() at the end of the probe function and place iio_device_unregister() accordingly.	2026-05-01	7.8
CVE-2026-31764	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: imu: st_lsm6dsx: Set buffer sampling frequency for accelerometer only The st_lsm6dsx_hwfifo_odr_store() function, which is called when userspace writes the buffer sampling frequency sysfs attribute, calls st_lsm6dsx_check_odr(), which accesses the odr_table array at index `sensor->id`; since this array is only 2 entries long, an access for any sensor type other than accelerometer or gyroscope is an out-of-bounds access. The motivation for being able to set a buffer frequency different from the sensor sampling frequency is to support use cases that need accurate event detection (which requires a high sampling frequency) while retrieving sensor data at low frequency. Since all the supported event types are generated from acceleration data only, do not create the buffer sampling frequency attribute for sensor types other than the accelerometer.	2026-05-01	7.8
CVE-2026-31768	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ti-adc161s626: use DMA-safe memory for spi_read() Add a DMA-safe buffer and use it for spi_read() instead of a stack memory. All SPI buffers must be DMA-safe. Since we only need up to 3 bytes, we just use a u8[] instead of __be16 and __be32 and change the conversion functions appropriately.	2026-05-01	7.8
CVE-2026-31769	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpib: fix use-after-free in IO ioctl handlers The IBRD, IBWRT, IBCMD, and IBWAIT ioctl handlers use a gpib_descriptor pointer after board->big_gpib_mutex has been released. A concurrent IBCLOSEDEV ioctl can free the descriptor via close_dev_ioctl() during this window, causing a use-after-free. The IO handlers (read_ioctl, write_ioctl, command_ioctl) explicitly release big_gpib_mutex before calling their handler. wait_ioctl() is called with big_gpib_mutex held, but ibwait() releases it internally when wait_mask is non-zero. In all four cases, the descriptor pointer obtained from handle_to_descriptor() becomes unprotected. Fix this by introducing a kernel-only descriptor_busy reference count in struct gpib_descriptor. Each handler atomically increments descriptor_busy under file_priv->descriptors_mutex before releasing the lock, and decrements it when done. close_dev_ioctl() checks descriptor_busy under the same lock and rejects the close with -EBUSY if the count is non-zero. A reference count rather than a simple flag is necessary because multiple handlers can operate on the same descriptor concurrently (e.g. IBRD and IBWAIT on the same handle from different threads). A separate counter is needed because io_in_progress can be cleared from unprivileged userspace via the IBWAIT ioctl (through general_ibstatus() with set_mask containing CMPL), which would allow an attacker to bypass a check based solely on io_in_progress. The new descriptor_busy counter is only modified by the kernel IO paths. The lock ordering is consistent (big_gpib_mutex -> descriptors_mutex) and the handlers only hold descriptors_mutex briefly during the lookup, so there is no deadlock risk and no impact on IO throughput.	2026-05-01	7.8
CVE-2026-31772	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: fix stack buffer overflow in hci_le_big_create_sync hci_le_big_create_sync() uses DEFINE_FLEX to allocate a struct hci_cp_le_big_create_sync on the stack with room for 0x11 (17)	2026-05-01	7.8

		BIS entries. However, conn->num_bis can hold up to HCI_MAX_ISO_BIS (31) entries — validated against ISO_MAX_NUM_BIS (0x1f) in the caller hci_conn_big_create_sync(). When conn->num_bis is between 18 and 31, the memcpy that copies conn->bis into cp->bis writes up to 14 bytes past the stack buffer, corrupting adjacent stack memory. This is trivially reproducible: binding an ISO socket with bc_num_bis = ISO_MAX_NUM_BIS (31) and calling listen() will eventually trigger hci_le_big_create_sync() from the HCI command sync worker, causing a KASAN-detectable stack-out-of-bounds write: BUG: KASAN: stack-out-of-bounds in hci_le_big_create_sync+0x256/0x3b0 Write of size 31 at addr ffffc90000487b48 by task kworker/u9:0/71 Fix this by changing the DEFINE_FLEX count from the incorrect 0x11 to HCI_MAX_ISO_BIS, which matches the maximum number of BIS entries that conn->bis can actually carry.		
CVE-2026-31776	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: ctxfi: Fix missing SPDIFI1 index handling SPDIF1 DAIO type isn't properly handled in daio_device_index() for hw20k2, and it returned -EINVAL, which ended up with the out-of-bounds array access. Follow the hw20k1 pattern and return the proper index for this type, too.	2026-05-01	7.8
CVE-2026-31780	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: wilc1000: fix u8 overflow in SSID scan buffer size calculation The variable valuesize is declared as u8 but accumulates the total length of all SSIDs to scan. Each SSID contributes up to 33 bytes (IEEE80211_MAX_SSID_LEN + 1), and with WILC_MAX_NUM_PROBED_SSID (10) SSIDs the total can reach 330, which wraps around to 74 when stored in a u8. This causes kmalloc to allocate only 75 bytes while the subsequent memcpy writes up to 331 bytes into the buffer, resulting in a 256-byte heap buffer overflow. Widen valuesize from u8 to u32 to accommodate the full range.	2026-05-01	7.8
CVE-2026-31782	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: perf/x86: Fix potential bad container_of in intel_pmu_hw_config Auto counter reload may have a group of events with software events present within it. The software event PMU isn't the x86_hybrid_pmu and a container_of operation in intel_pmu_set_acr_caused_constr (via the hybrid helper) could cause out of bound memory reads. Avoid this by guarding the call to intel_pmu_set_acr_caused_constr with an is_x86_event check.	2026-05-01	7.8
CVE-2026-43007	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/qaic: Handle DBC deactivation if the owner went away When a DBC is released, the device sends a QAIC_TRANS_DEACTIVATE_FROM_DEV transaction to the host over the QAIC_CONTROL MHI channel. QAIC handles this by calling decode_deactivate() to release the resources allocated for that DBC. Since that handling is done in the qaic_manage_ioctl() context, if the user goes away before receiving and handling the deactivation, the host will be out-of-sync with the DBCs available for use, and the DBC resources will not be freed unless the device is removed. If another user loads and requests to activate a network, then the device assigns the same DBC to that network, QAIC will "indefinitely" wait for dbc->in_use = false, leading the user process to hang. As a solution to this, handle QAIC_TRANS_DEACTIVATE_FROM_DEV transactions that are received after the user has gone away.	2026-05-01	7.8
CVE-2026-43009	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix incorrect pruning due to atomic fetch precision tracking When backtrack_insn encounters a BPF_STX instruction with BPF_ATOMIC and BPF_FETCH, the src register (or r0 for BPF_CMPXCHG) also acts as a destination, thus receiving the old value from the memory location. The current backtracking logic does not account for this. It treats atomic fetch operations the same as regular stores where the src register is only an input. This leads the backtrack_insn to fail to propagate precision to the stack location, which is then not marked as precise!	2026-05-01	7.8

		Later, the verifier's path pruning can incorrectly consider two states equivalent when they differ in terms of stack state. Meaning, two branches can be treated as equivalent and thus get pruned when they should not be seen as such. Fix it as follows: Extend the BPF_LDX handling in backtrack_insn to also cover atomic fetch operations via is_atomic_fetch_insn() helper. When the fetch dst register is being tracked for precision, clear it, and propagate precision over to the stack slot. For non-stack memory, the precision walk stops at the atomic instruction, same as regular BPF_LDX. This covers all fetch variants. Before: 0: (b7) r1 = 8 ; R1=8 1: (7b) *(u64 *)(r10 -8) = r1 ; R1=8 R10=fp0 fp-8=8 2: (b7) r2 = 0 ; R2=0 3: (db) r2 = atomic64_fetch_add((u64 *)(r10 -8), r2) ; R2=8 R10=fp0 fp-8=mmmmmmmm 4: (bf) r3 = r10 ; R3=fp0 R10=fp0 5: (0f) r3 += r2 mark_precise: frame0: last_idx 5 first_idx 0 subseq_idx -1 mark_precise: frame0: regs=r2 stack= before 4: (bf) r3 = r10 mark_precise: frame0: regs=r2 stack= before 3: (db) r2 = atomic64_fetch_add((u64 *)(r10 -8), r2) mark_precise: frame0: regs=r2 stack= before 2: (b7) r2 = 0 6: R2=8 R3=fp8 6: (b7) r0 = 0 ; R0=0 7: (95) exit After: 0: (b7) r1 = 8 ; R1=8 1: (7b) *(u64 *)(r10 -8) = r1 ; R1=8 R10=fp0 fp-8=8 2: (b7) r2 = 0 ; R2=0 3: (db) r2 = atomic64_fetch_add((u64 *)(r10 -8), r2) ; R2=8 R10=fp0 fp-8=mmmmmmmm 4: (bf) r3 = r10 ; R3=fp0 R10=fp0 5: (0f) r3 += r2 mark_precise: frame0: last_idx 5 first_idx 0 subseq_idx -1 mark_precise: frame0: regs=r2 stack= before 4: (bf) r3 = r10 mark_precise: frame0: regs=r2 stack= before 3: (db) r2 = atomic64_fetch_add((u64 *)(r10 -8), r2) mark_precise: frame0: regs= stack=-8 before 2: (b7) r2 = 0 mark_precise: frame0: regs= stack=-8 before 1: (7b) *(u64 *)(r10 -8) = r1 mark_precise: frame0: regs=r1 stack= before 0: (b7) r1 = 8 6: R2=8 R3=fp8 6: (b7) r0 = 0 ; R0=0 7: (95) exit		
[CVE-2026-43015](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: macb: fix clk handling on PCI glue driver removal platform_device_unregister() may still want to use the registered clks during runtime resume callback. Note that there is a commit d82d5303c4c5 ("net: macb: fix use after free on rmmod") that addressed the similar problem of clk vs platform device unregistration but just moved the bug to another place. Save the pointers to clks into local variables for reuse after platform device is unregistered. BUG: KASAN: use-after-free in clk_prepare+0x5a/0x60 Read of size 8 at addr ffff888104f85e00 by task modprobe/597 CPU: 2 PID: 597 Comm: modprobe Not tainted 6.1.164+ #114 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.1-0-g3208b098f51a-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x8d/0xba print_report+0x17f/0x496 kasan_report+0xd9/0x180 clk_prepare+0x5a/0x60 macb_runtime_resume+0x13d/0x410 [macb] pm_generic_runtime_resume+0x97/0xd0 __rpm_callback+0xc8/0x4d0 rpm_callback+0xf6/0x230 rpm_resume+0xeeb/0x1a70 __pm_runtime_resume+0xb4/0x170 bus_remove_device+0x2e3/0x4b0 device_del+0x5b3/0xdc0	2026-05-01	7.8

		platform_device_del+0x4e/0x280 platform_device_unregister+0x11/0x50 pci_device_remove+0xae/0x210 device_remove+0xcb/0x180 device_release_driver_internal+0x529/0x770 driver_detach+0xd4/0x1a0 bus_remove_driver+0x135/0x260 driver_unregister+0x72/0xb0 pci_unregister_driver+0x26/0x220 __do_sys_delete_module+0x32e/0x550 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x6e/0xd8 </TASK> Allocated by task 519: kasan_save_stack+0x2c/0x50 kasan_set_track+0x21/0x30 __kasan_kmalloc+0x8e/0x90 __clk_register+0x458/0x2890 clk_hw_register+0x1a/0x60 __clk_hw_register_fixed_rate+0x255/0x410 clk_register_fixed_rate+0x3c/0xa0 macb_probe+0x1d8/0x42e [macb_pci] local_pci_probe+0xd7/0x190 pci_device_probe+0x252/0x600 really_probe+0x255/0x7f0 __driver_probe_device+0x1ee/0x330 driver_probe_device+0x4c/0x1f0 __driver_attach+0x1df/0x4e0 bus_for_each_dev+0x15d/0x1f0 bus_add_driver+0x486/0x5e0 driver_register+0x23a/0x3d0 do_one_initcall+0xfd/0x4d0 do_init_module+0x18b/0x5a0 load_module+0x5663/0x7950 __do_sys_finit_module+0x101/0x180 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x6e/0xd8 Freed by task 597: kasan_save_stack+0x2c/0x50 kasan_set_track+0x21/0x30 kasan_save_free_info+0x2a/0x50 __kasan_slab_free+0x106/0x180 __kmem_cache_free+0xbc/0x320 clk_unregister+0x6de/0x8d0 macb_remove+0x73/0xc0 [macb_pci] pci_device_remove+0xae/0x210 device_remove+0xcb/0x180 device_release_driver_internal+0x529/0x770 driver_detach+0xd4/0x1a0 bus_remove_driver+0x135/0x260 driver_unregister+0x72/0xb0 pci_unregister_driver+0x26/0x220 __do_sys_delete_module+0x32e/0x550 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x6e/0xd8		
CVE-2026-43016	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: sockmap: Fix use-after-free of sk->sk_socket in sk_psock_verdict_data_ready(). syzbot reported use-after-free of AF_UNIX socket's sk->sk_socket in sk_psock_verdict_data_ready(). [0] In unix_stream_sendmsg(), the peer socket's ->sk_data_ready() is called after dropping its unix_state_lock(). Although the sender socket holds the peer's refcount, it does not prevent the peer's sock_orphan(), and the peer's sk_socket might be freed after one RCU grace period. Let's fetch the peer's sk->sk_socket and sk->sk_socket->ops under RCU in sk_psock_verdict_data_ready(). [0]: BUG: KASAN: slab-use-after-free in sk_psock_verdict_data_ready+0xec/0x590 net/core/skmsg.c:1278 Read of size 8 at addr ffff8880594da860 by task syz.4.1842/11013 CPU: 1 UID: 0 PID: 11013 Comm: syz.4.1842 Not tainted syzkaller #0 PREEMPT(full)	2026-05-01	7.8

		Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2026 Call Trace: <TASK> dump_stack_lvl+0xe8/0x150 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xba/0x230 mm/kasan/report.c:482 kasan_report+0x117/0x150 mm/kasan/report.c:595 sk_psock_verdict_data_ready+0xec/0x590 net/core/skmsg.c:1278 unix_stream_sendmsg+0x8a3/0xe80 net/unix/af_unix.c:2482 sock_sendmsg_nosec net/socket.c:721 [inline] __sock_sendmsg net/socket.c:736 [inline] ____sys_sendmsg+0x972/0x9f0 net/socket.c:2585 ___sys_sendmsg+0x2a5/0x360 net/socket.c:2639 __sys_sendmsg net/socket.c:2671 [inline] __do_sys_sendmsg net/socket.c:2676 [inline] __se_sys_sendmsg net/socket.c:2674 [inline] __x64_sys_sendmsg+0x1bd/0x2a0 net/socket.c:2674 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x14d/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7facf899c819 Code: ff c3 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 e8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007facf9827028 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffffffffda RBX: 00007facf8c15fa0 RCX: 00007facf899c819 RDX: 0000000000000000 RSI: 0000200000000500 RDI: 0000000000000004 RBP: 00007facf8a32c91 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007facf8c16038 R14: 00007facf8c15fa0 R15: 00007ffd41b01c78 </TASK> Allocated by task 11013: kasan_save_stack mm/kasan/common.c:57 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:78 unpoison_slab_object mm/kasan/common.c:340 [inline] __kasan_slab_alloc+0x6c/0x80 mm/kasan/common.c:366 kasan_slab_alloc include/linux/kasan.h:253 [inline] slab_post_alloc_hook mm/slub.c:4538 [inline] slab_alloc_node mm/slub.c:4866 [inline] kmem_cache_alloc_lru_noprof+0x2b8/0x640 mm/slub.c:4885 sock_alloc_inode+0x28/0xc0 net/socket.c:316 alloc_inode+0x6a/0x1b0 fs/inode.c:347 new_inode_pseudo include/linux/fs.h:3003 [inline] sock_alloc net/socket.c:631 [inline] __sock_create+0x12d/0x9d0 net/socket.c:1562 sock_create net/socket.c:1656 [inline] __sys_socketpair+0x1c4/0x560 net/socket.c:1803 __do_sys_socketpair net/socket.c:1856 [inline] __se_sys_socketpair net/socket.c:1853 [inline] __x64_sys_socketpair+0x9b/0xb0 net/socket.c:1853 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x14d/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 15: kasan_save_stack mm/kasan/common.c:57 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:78 kasan_save_free_info+0x46/0x50 mm/kasan/generic.c:584 poison_slab_object mm/kasan/common.c:253 [inline] __kasan_slab_free+0x5c/0x80 mm/kasan/common.c:285 kasan_slab_free include/linux/kasan.h:235 [inline] slab_free_hook mm/slub.c:2685 [inline] slab_free mm/slub.c:6165 [inline] kmem_cache_free+0x187/0x630 mm/slub.c:6295 rcu_do_batch kernel/rcu/tree.c: ---truncated---		
CVE-2026-43019	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: fix potential UAF in set_cig_params_sync hci_conn lookup and field access must be covered by hdev lock in set_cig_params_sync, otherwise it's possible it is freed concurrently. Take hdev lock to prevent hci_conn from being deleted or modified concurrently. Just RCU lock is not suitable here, as we also want to avoid "tearing" in the configuration.	2026-05-01	7.8
CVE-2026-43020	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: validate LTK enc_size on load	2026-05-01	7.8

		Load Long Term Keys stores the user-provided enc_size and later uses it to size fixed-size stack operations when replying to LE LTK requests. An enc_size larger than the 16-byte key buffer can therefore overflow the reply stack buffer. Reject oversized enc_size values while validating the management LTK record so invalid keys never reach the stored key state.		
CVE-2026-43023	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: fix race conditions in sco_sock_connect() sco_sock_connect() checks sk_state and sk_type without holding the socket lock. Two concurrent connect() syscalls on the same socket can both pass the check and enter sco_connect(), leading to use-after-free. The buggy scenario involves three participants and was confirmed with additional logging instrumentation: Thread A (connect): HCI disconnect: Thread B (connect): sco_sock_connect(sk) sk_state==BT_OPEN (pass, no lock) sco_connect(sk): hci_dev_lock hci_connect_sco -> hcon1 sco_conn_add->conn1 lock_sock(sk) sco_chan_add: conn1->sk = sk sk->conn = conn1 sk_state=BT_CONNECT release_sock hci_dev_unlock sco_sock_connect(sk) sk_state==BT_OPEN (pass, no lock) sco_connect(sk): hci_dev_lock <- blocked hci_dev_lock sco_conn_del: lock_sock(sk) sco_chan_del: sk->conn=NULL conn1->sk=NULL sk_state= BT_CLOSED SOCK_ZAPPED release_sock hci_dev_unlock (unblocked) hci_connect_sco -> hcon2 sco_conn_add -> conn2 lock_sock(sk) sco_chan_add: sk->conn=conn2 sk_state= BT_CONNECT // zombie sk! release_sock hci_dev_unlock Thread B revives a BT_CLOSED + SOCK_ZAPPED socket back to BT_CONNECT. Subsequent cleanup triggers double sock_put() and use-after-free. Meanwhile conn1 is leaked as it was orphaned when sco_conn_del() cleared the association. Fix this by: - Moving lock_sock() before the sk_state/sk_type checks in sco_sock_connect() to serialize concurrent connect attempts - Fixing the sk_type != SOCK_SEQPACKET check to actually return the error instead of just assigning it - Adding a state re-check in sco_connect() after lock_sock() to catch state changes during the window between the locks - Adding sco_pi(sk)->conn check in sco_chan_add() to prevent double-attach of a socket to multiple connections - Adding hci_conn_drop() on sco_chan_add failure to prevent HCI connection leaks	2026-05-01	7.8
CVE-2026-43027	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_conntrack_helper: pass helper to expect cleanup	2026-05-01	7.8

		nf_conntrack_helper_unregister() calls nf_ct_expect_iterate_destroy() to remove expectations belonging to the helper being unregistered. However, it passes NULL instead of the helper pointer as the data argument, so expect_iter_me() never matches any expectation and all of them survive the cleanup. After unregister returns, nfnl_cthelper_del() frees the helper object immediately. Subsequent expectation dumps or packet-driven init_conntrack() calls then dereference the freed exp->helper, causing a use-after-free. Pass the actual helper pointer so expectations referencing it are properly destroyed before the helper object is freed. BUG: KASAN: slab-use-after-free in string+0x38f/0x430 Read of size 1 at addr ffff888003b14d20 by task poc/103 Call Trace: string+0x38f/0x430 vsnprintf+0x3cc/0x1170 seq_printf+0x17a/0x240 exp_seq_show+0x2e5/0x560 seq_read_iter+0x419/0x1280 proc_reg_read+0x1ac/0x270 vfs_read+0x179/0x930 ksys_read+0xef/0x1c0 Freed by task 103: The buggy address is located 32 bytes inside of freed 192-byte region [ffff888003b14d00, ffff888003b14dc0)		
CVE-2026-43030	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix regsafe() for pointers to packet In case rold->reg->range == BEYOND_PKT_END && rcur->reg->range == N regsafe() may return true which may lead to current state with valid packet range not being explored. Fix the bug.	2026-05-01	7.8
CVE-2026-43033	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: authencesn - Do not place hiseq at end of dst for out-of-place decryption When decrypting data that is not in-place (src != dst), there is no need to save the high-order sequence bits in dst as it could simply be re-copied from the source. However, the data to be hashed need to be rearranged accordingly. Thanks,	2026-05-01	7.8
CVE-2026-43044	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: caam - fix DMA corruption on long hmac keys When a key longer than block size is supplied, it is copied and then hashed into the real key. The memory allocated for the copy needs to be rounded to DMA cache alignment, as otherwise the hashed key may corrupt neighbouring memory. The rounding was performed, but never actually used for the allocation. Fix this by replacing kmemdup with kmalloc for a larger buffer, followed by memcpy.	2026-05-01	7.8
CVE-2026-43047	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: multitouch: Check to ensure report responses match the request It is possible for a malicious (or clumsy) device to respond to a specific report's feature request using a completely different report ID. This can cause confusion in the HID core resulting in nasty side-effects such as OOB writes. Add a check to ensure that the report ID in the response, matches the one that was requested. If it doesn't, omit reporting the raw event and return early.	2026-05-01	7.8
CVE-2026-43049	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: logitech-hidpp: Prevent use-after-free on force feedback initialisation failure Presently, if the force feedback initialisation fails when probing the Logitech G920 Driving Force Racing Wheel for Xbox One, an error number will be returned and propagated before the userspace infrastructure (sysfs and /dev/input) has been torn down. If userspace ignores the	2026-05-01	7.8

		errors and continues to use its references to these dangling entities, a UAF will promptly follow. We have 2 options; continue to return the error, but ensure that all of the infrastructure is torn down accordingly or continue to treat this condition as a warning by emitting the message but returning success. It is thought that the original author's intention was to emit the warning but keep the device functional, less the force feedback feature, so let's go with that.		
CVE-2026-43056	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mana: fix use-after-free in add_adev() error path If auxiliary_device_add() fails, add_adev() jumps to add_fail and calls auxiliary_device_uninit(adev). The auxiliary device has its release callback set to adev_release(), which frees the containing struct mana_adev. Since adev is embedded in struct mana_adev, the subsequent fall-through to init_fail and access to adev->id may result in a use-after-free. Fix this by saving the allocated auxiliary device id in a local variable before calling auxiliary_device_add(), and use that saved id in the cleanup path after auxiliary_device_uninit().	2026-05-01	7.8
CVE-2026-40972	vmware - multiple products	An attacker on the same network as the remote application may be able to utilize a timing attack to discover information about the remote secret. In extreme circumstances this could result in the attacker determining the secret and uploading changed classes, thereby achieving remote code execution in the remote application. Affected: Spring Boot 4.0.0–4.0.5 (fix 4.0.6), 3.5.0–3.5.13 (fix 3.5.14), 3.4.0–3.4.15 (fix 3.4.16), 3.3.0–3.3.18 (fix 3.3.19), 2.7.0–2.7.32 (fix 2.7.33); DevTools remote secret comparison. Versions that are no longer supported are also affected per vendor advisory.	2026-04-28	7.5
CVE-2025-48431	apache - thrift	Mismatched Memory Management Routines vulnerability in Apache Thrift c_glib language bindings. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue. Description: Specially crafted requests can crash an c_glib-based Thrift server with a clean but fatal "free(): invalid pointer" error message.	2026-04-28	7.5
CVE-2026-41602	apache - thrift	Integer Overflow or Wraparound vulnerability in Apache Thrift TFramedTransport Go language implementation This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	7.5
CVE-2026-7320	mozilla - multiple products	Information disclosure due to incorrect boundary conditions in the Audio/Video component. This vulnerability was fixed in Firefox 150.0.1, Firefox ESR 140.10.1, Firefox ESR 115.35.1, Thunderbird 150.0.1, and Thunderbird 140.10.1.	2026-04-28	7.5
CVE-2026-7338	google - chrome	Use after free in Cast in Google Chrome prior to 147.0.7727.138 allowed an attacker on the local network segment to potentially exploit heap corruption via malicious network traffic. (Chromium security severity: High)	2026-04-28	7.5
CVE-2026-7343	google - chrome	Use after free in Views in Google Chrome on Windows prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Critical)	2026-04-28	7.5
CVE-2026-7349	google - chrome	Use after free in Cast in Google Chrome prior to 147.0.7727.138 allowed an attacker on the local network segment to execute arbitrary code inside a sandbox via malicious network traffic. (Chromium security severity: High)	2026-04-28	7.5
CVE-2026-7357	google - chrome	Use after free in GPU in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2026-04-28	7.5
CVE-2026-42520	jenkins - credentials_binding	Jenkins Credentials Binding Plugin 719.v80e905ef14eb_ and earlier does not sanitize file names for file and zip file credentials, allowing attackers able to provide credentials to a job to write files to arbitrary locations on the node filesystem, which can lead to remote code execution if Jenkins is configured to allow a low-privileged user to configure file or zip file credentials used for a job running on the built-in node.	2026-04-29	7.5
CVE-2026-33845	gnu - multiple products	A flaw in GnuTLS DTLS handshake parsing allows malformed fragments with zero length and non-zero offset, leading to an integer underflow during reassembly and resulting in an out-of-bounds read. This issue is remotely exploitable and may cause information disclosure or denial of service.	2026-04-30	7.5
CVE-2026-4503	ibm - Langflow Desktop	IBM Langflow Desktop 1.0.0 through 1.8.4 Langflow could allow an unauthenticated user to view other users' images due to an indirect object reference through a user-controlled key.	2026-04-30	7.5
CVE-2026-42402	apache - neethi	Apache Neethi is vulnerable to a Denial of Service attack through algorithmic complexity in policy normalization. Specially crafted WS-Policy documents can trigger an exponential Cartesian cross-product expansion during the normalization process, causing unbounded memory allocation that exhausts the JVM heap. This occurs when the normalization process generates an excessive number of policy alternatives without bounds, leading to runtime memory exhaustion. Users should upgrade to 3.2.2 which limits the maximum number of normalized policy alternatives.	2026-05-01	7.5

CVE-2026-42403	apache - neethi	Apache Neethi does not properly detect circular references in policy definitions. When a WS-Policy document contains circular policy references (where Policy A references Policy B which references Policy A), the policy normalization process can enter an infinite loop or cause excessive recursion, leading to a stack overflow or application hang. An attacker can craft malicious policy documents with circular references to cause a Denial of Service condition Users are recommended to upgrade to version 3.2.2, which fixes this issue.	2026-05-01	7.5
CVE-2026-31711	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: server: fix active_num_conn leak on transport allocation failure Commit 77ffbcac4e56 ("smb: server: fix leak of active_num_conn in ksmbd_tcp_new_connection()") addressed the kthread_run() failure path. The earlier alloc_transport() == NULL path in the same function has the same leak, is reachable pre-authentication via any TCP connect to port 445, and was empirically reproduced on UML (ARCH=um, v7.0-rc7): a small number of forced allocation failures were sufficient to put ksmbd into a state where every subsequent connection attempt was rejected for the remainder of the boot. ksmbd_kthread_fn() increments active_num_conn before calling ksmbd_tcp_new_connection() and discards the return value, so when alloc_transport() returns NULL the socket is released and -ENOMEM returned without decrementing the counter. Each such failure permanently consumes one slot from the max_connections pool; once cumulative failures reach the cap, atomic_inc_return() hits the threshold on every subsequent accept and every new connection is rejected. The counter is only reset by module reload. An unauthenticated remote attacker can drive the server toward the memory pressure that makes alloc_transport() fail by holding open connections with large RFC1002 lengths up to MAX_STREAM_PROT_LEN (0x00FFFFFF); natural transient allocation failures on a loaded host produce the same drift more slowly. Mirror the existing rollback pattern in ksmbd_kthread_fn(): on the alloc_transport() failure path, decrement active_num_conn gated on server_conf.max_connections. Repro details: with the patch reverted, forced alloc_transport() NULL returns leaked counter slots and subsequent connection attempts -- including legitimate connects issued after the forced-fail window had closed -- were all rejected with "Limit the maximum number of connections". With this patch applied, the same connect sequence produces no rejections and the counter cycles cleanly between zero and one on every accept.	2026-05-01	7.5
CVE-2026-31719	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: krb5enc - fix async decrypt skipping hash verification krb5enc_dispatch_decrypt() sets req->base.complete as the skcipher callback, which is the caller's own completion handler. When the skcipher completes asynchronously, this signals "done" to the caller without executing krb5enc_dispatch_decrypt_hash(), completely bypassing the integrity verification (hash check). Compare with the encrypt path which correctly uses krb5enc_encrypt_done as an intermediate callback to chain into the hash computation on async completion. Fix by adding krb5enc_decrypt_done as an intermediate callback that chains into krb5enc_dispatch_decrypt_hash() upon async skcipher completion, matching the encrypt path's callback pattern. Also fix EBUSY/EINPROGRESS handling throughout: remove krb5enc_request_complete() which incorrectly swallowed EINPROGRESS notifications that must be passed up to callers waiting on backlogged requests, and add missing EBUSY checks in krb5enc_encrypt_ahash_done for the dispatch_encrypt return value. Unset MAY_BACKLOG on the async completion path so the user won't see back-to-back EINPROGRESS notifications.	2026-05-01	7.5
CVE-2026-43029	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix soft lockup in mptcp_recvmsg() syzbot reported a soft lockup in mptcp_recvmsg() [0]. When receiving data with MSG_PEEK | MSG_WAITALL flags, the skb is not	2026-05-01	7.5

		removed from the sk_receive_queue. This causes sk_wait_data() to always find available data and never perform actual waiting, leading to a soft lockup. Fix this by adding a 'last' parameter to track the last peeked skb. This allows sk_wait_data() to make informed waiting decisions and prevent infinite loops when MSG_PEEK is used. [0]: watchdog: BUG: soft lockup - CPU#2 stuck for 156s! [server:1963] Modules linked in: CPU: 2 UID: 0 PID: 1963 Comm: server Not tainted 6.19.0-rc8 #61 PREEMPT(none) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:sk_wait_data+0x15/0x190 Code: 80 00 00 00 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 41 56 41 55 41 54 49 89 f4 55 48 89 d5 53 48 89 fb <48> 83 ec 30 65 48 8b 05 17 a4 6b 01 48 89 44 24 28 31 c0 65 48 8b RSP: 0018:ffffc90000603ca0 EFLAGS: 00000246 RAX: 0000000000000000 RBX: ffff888102bf0800 RCX: 0000000000000001 RDX: 0000000000000000 RSI: ffffc90000603d18 RDI: ffff888102bf0800 RBP: 0000000000000000 R08: 0000000000000002 R09: 0000000000000101 R10: 0000000000000000 R11: 0000000000000075 R12: ffffc90000603d18 R13: ffff888102bf0800 R14: ffff888102bf0800 R15: 0000000000000000 FS: 00007f6e38b8c4c0(0000) GS:ffff8881b877e000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055aa7bff1680 CR3: 0000000105cbe000 CR4: 00000000000006f0 Call Trace: <TASK> mptcp_recvmsg+0x547/0x8c0 net/mptcp/protocol.c:2329 inet_recvmsg+0x11f/0x130 net/ipv4/af_inet.c:891 sock_recvmsg+0x94/0xc0 net/socket.c:1100 __sys_recvfrom+0xb2/0x130 net/socket.c:2256 __x64_sys_recvfrom+0x1f/0x30 net/socket.c:2267 do_syscall_64+0x59/0x2d0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x76/0x7e arch/x86/entry/entry_64.S:131 RIP: 0033:0x7f6e386a4a1d Code: 0f 1f 84 00 00 00 00 00 0f 1f 44 00 00 48 8d 05 f1 de 2c 00 41 89 ca 8b 00 85 c0 75 20 45 31 c9 45 31 c0 b8 2d 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 6b f3 c3 66 0f 1f 84 00 00 00 00 00 41 56 41 RSP: 002b:00007ffc3c4bb078 EFLAGS: 00000246 ORIG_RAX: 000000000000002d RAX: fffffffffffffda RBX: 000000000000861e RCX: 00007f6e386a4a1d RDX: 00000000000003ff RSI: 00007ffc3c4bb150 RDI: 0000000000000004 RBP: 00007ffc3c4bb570 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000103 R11: 0000000000000246 R12: 00005605dbc00be0 R13: 00007ffc3c4bb650 R14: 0000000000000000 R15: 0000000000000000 </TASK>		
[CVE-2026-43031](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: xilinx: axienet: Fix BQL accounting for multi-BD TX packets When a TX packet spans multiple buffer descriptors (scatter-gather), axienet_free_tx_chain sums the per-BD actual length from descriptor status into a caller-provided accumulator. That sum is reset on each NAPI poll. If the BDs for a single packet complete across different polls, the earlier bytes are lost and never credited to BQL. This causes BQL to think bytes are permanently in-flight, eventually stalling the TX queue. The SKB pointer is stored only on the last BD of a packet. When that BD completes, use skb->len for the byte count instead of summing per-BD status lengths. This matches netdev_sent_queue(), which debits skb->len, and naturally survives across polls because no partial packet contributes to the accumulator.	2026-05-01	7.5
[CVE-2026-43055](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: target: file: Use kzalloc_flex for aio_cmd The target_core_file doesn't initialize the aio_cmd->iocb for the ki_write_stream. When a write command fd_execute_rw_aio() is executed, we may get a bogus ki_write_stream value, causing unintended write failure status when checking iocb->ki_write_stream > max_write_streams in the block device. Let's just use kzalloc_flex when allocating the aio_cmd and let ki_write_stream=0 to fix this issue.	2026-05-01	7.5
[CVE-2026-43057](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: correctly handle tunneled traffic on IPV6_CSUM GSO fallback NETIF_F_IPV6_CSUM only advertises support for checksum offload of packets without IPv6 extension headers. Packets with extension	2026-05-01	7.5

		headers must fall back onto software checksumming. Since TSO depends on checksum offload, those must revert to GSO. The below commit introduces that fallback. It always checks network header length. For tunneled packets, the inner header length must be checked instead. Extend the check accordingly. A special case is tunneled packets without inner IP protocol. Such as RFC 6951 SCTP in UDP. Those are not standard IPv6 followed by transport header either, so also must revert to the software GSO path.		
CVE-2026-41603	apache - thrift	Improper Validation of Certificate with Host Mismatch vulnerability in Apache Thrift. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	7.4
CVE-2026-41605	apache - thrift	Integer Overflow or Wraparound vulnerability in Apache Thrift. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	7.3
CVE-2026-5435	gnu - glibc	The deprecated functions ns_printrrf, ns_printrr and fp_nquery in the GNU C Library version 2.2 and newer fail to enforce the caller-supplied buffer length, and can result in an out-of-bounds write when printing TSIG records.	2026-04-28	7.3
CVE-2026-7322	mozilla - multiple products	Memory safety bugs present in Thunderbird ESR 140.10.0 and Thunderbird 150.0.0. Some of these bugs showed evidence of memory corruption and we presume that with enough effort some of these could have been exploited to run arbitrary code. This vulnerability was fixed in Firefox 150.0.1, Firefox ESR 140.10.1, Firefox ESR 115.35.1, Thunderbird 150.0.1, and Thunderbird 140.10.1.	2026-04-28	7.3
CVE-2026-7323	mozilla - multiple products	Memory safety bugs present in Thunderbird ESR 140.10.0 and Thunderbird 150.0.0. Some of these bugs showed evidence of memory corruption and we presume that with enough effort some of these could have been exploited to run arbitrary code. This vulnerability was fixed in Firefox 150.0.1, Firefox ESR 140.10.1, Thunderbird 150.0.1, and Thunderbird 140.10.1.	2026-04-28	7.3
CVE-2026-7324	mozilla - multiple products	Memory safety bugs present in Thunderbird 150.0.0. Some of these bugs showed evidence of memory corruption and we presume that with enough effort some of these could have been exploited to run arbitrary code. This vulnerability was fixed in Firefox 150.0.1 and Thunderbird 150.0.1.	2026-04-28	7.3
CVE-2026-43025	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: ignore explicit helper on new expectations Use the existing master conntrack helper, anything else is not really supported and it just makes validation more complicated, so just ignore what helper userspace suggests for this expectation. This was uncovered when validating CTA_EXPECT_CLASS via different helper provided by userspace than the existing master conntrack helper: BUG: KASAN: slab-out-of-bounds in nf_ct_expect_related_report+0x2479/0x27c0 Read of size 4 at addr ffff8880043fe408 by task poc/102 Call Trace: nf_ct_expect_related_report+0x2479/0x27c0 ctnetlink_create_expect+0x22b/0x3b0 ctnetlink_new_expect+0x4bd/0x5c0 nfnetlink_rcv_msg+0x67a/0x950 netlink_rcv_skb+0x120/0x350 Allowing to read kernel memory bytes off the expectation boundary. CTA_EXPECT_HELP_NAME is still used to offer the helper name to userspace via netlink dump.	2026-05-01	7.3
CVE-2026-1460	zyxel - multiple products	A post-authentication command injection vulnerability in the “DomainName” parameter of the DHCP configuration file in Zyxel DX3301-T0 and EX3301-T0 firmware versions through 5.50(ABVY.7.1)C0 could allow an authenticated attacker with administrator privileges to execute OS commands on an affected device.	2026-04-28	7.2
CVE-2026-35155	dell - idrac10_firmware	Dell iDRAC10, versions 1.20.70.50 and 1.30.05.10, contains an Insufficiently Protected Credentials vulnerability. A race condition vulnerability exists that could allow an authenticated low-privileged attacker to gain elevated access.	2026-04-29	7.1
CVE-2026-6914	mongodb - multiple products	Computing the MD5 checksum of a malformed BSON object under specific conditions may cause loss of availability in MongoDB server. This issue affects all MongoDB Server v8.2 versions, all MongoDB Server v8.1 versions, MongoDB Server v8.0 versions prior to 8.0.21, MongoDB Server v7.0 versions prior to 7.0.32	2026-04-29	7.1
CVE-2026-31697	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp: Don't attempt to copy ID to userspace if PSP command failed When retrieving the ID for the CPU, don't attempt to copy the ID blob to userspace if the firmware command failed. If the failure was due to an invalid length, i.e. the userspace buffer+length was too small, copying the number of bytes _firmware_ requires will overflow the kernel-allocated buffer and leak data to userspace.	2026-05-01	7.1

		BUG: KASAN: slab-out-of-bounds in instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] BUG: KASAN: slab-out-of-bounds in _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] BUG: KASAN: slab-out-of-bounds in _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 Read of size 64 at addr ffff8881867f5960 by task syz.0.906/24388 CPU: 130 UID: 0 PID: 24388 Comm: syz.0.906 Tainted: G U O 7.0.0-smp-DEV #28 PREEMPTLAZY Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 12.62.0-0 11/19/2025 Call Trace: <TASK> dump_stack_lvl+0xc5/0x110 ../lib/dump_stack.c:120 print_address_description ../mm/kasan/report.c:378 [inline] print_report+0xbc/0x260 ../mm/kasan/report.c:482 kasan_report+0xa2/0xe0 ../mm/kasan/report.c:595 check_region_inline ../mm/kasan/generic.c:-1 [inline] kasan_check_range+0x264/0x2c0 ../mm/kasan/generic.c:200 instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 copy_to_user ../include/linux/uaccess.h:236 [inline] sev_ioctl_do_get_id2+0x361/0x490 ../drivers/crypto/ccp/sev-dev.c:2222 sev_ioctl+0x25f/0x490 ../drivers/crypto/ccp/sev-dev.c:2575 vfs_ioctl ../fs/ioctl.c:51 [inline] __do_sys_ioctl ../fs/ioctl.c:597 [inline] __se_sys_ioctl+0x11d/0x1b0 ../fs/ioctl.c:583 do_syscall_x64 ../arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xe0/0x800 ../arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> WARN if the driver says the command succeeded, but the firmware error code says otherwise, as __sev_do_cmd_locked() is expected to return -EIO on any firwmware error.		
CVE-2026-31698	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp: Don't attempt to copy PDH cert to userspace if PSP command failed When retrieving the PDH cert, don't attempt to copy the blobs to userspace if the firmware command failed. If the failure was due to an invalid length, i.e. the userspace buffer+length was too small, copying the number of bytes _firmware_ requires will overflow the kernel-allocated buffer and leak data to userspace. BUG: KASAN: slab-out-of-bounds in instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] BUG: KASAN: slab-out-of-bounds in _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] BUG: KASAN: slab-out-of-bounds in _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 Read of size 2084 at addr ffff8885c4ab8aa0 by task syz.0.186/21033 CPU: 51 UID: 0 PID: 21033 Comm: syz.0.186 Tainted: G U O 7.0.0-smp-DEV #28 PREEMPTLAZY Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 34.84.12-0 11/17/2025 Call Trace: <TASK> dump_stack_lvl+0xc5/0x110 ../lib/dump_stack.c:120 print_address_description ../mm/kasan/report.c:378 [inline] print_report+0xbc/0x260 ../mm/kasan/report.c:482 kasan_report+0xa2/0xe0 ../mm/kasan/report.c:595 check_region_inline ../mm/kasan/generic.c:-1 [inline] kasan_check_range+0x264/0x2c0 ../mm/kasan/generic.c:200 instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 copy_to_user ../include/linux/uaccess.h:236 [inline] sev_ioctl_do_pdh_export+0x3d3/0x7c0 ../drivers/crypto/ccp/sev-dev.c:2347 sev_ioctl+0x2a2/0x490 ../drivers/crypto/ccp/sev-dev.c:2568 vfs_ioctl ../fs/ioctl.c:51 [inline] __do_sys_ioctl ../fs/ioctl.c:597 [inline] __se_sys_ioctl+0x11d/0x1b0 ../fs/ioctl.c:583 do_syscall_x64 ../arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xe0/0x800 ../arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> WARN if the driver says the command succeeded, but the firmware error code	2026-05-01	7.1

		says otherwise, as __sev_do_cmd_locked() is expected to return -EIO on any firwmware error.		
CVE-2026-31699	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp: Don't attempt to copy CSR to userspace if PSP command failed When retrieving the PEK CSR, don't attempt to copy the blob to userspace if the firmware command failed. If the failure was due to an invalid length, i.e. the userspace buffer+length was too small, copying the number of bytes _firmware_ requires will overflow the kernel-allocated buffer and leak data to userspace. BUG: KASAN: slab-out-of-bounds in instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] BUG: KASAN: slab-out-of-bounds in _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] BUG: KASAN: slab-out-of-bounds in _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 Read of size 2084 at addr ffff898144612e20 by task syz.9.219/21405 CPU: 14 UID: 0 PID: 21405 Comm: syz.9.219 Tainted: G U O 7.0.0-smp-DEV #28 PREEMPTLAZY Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 12.62.0-0 11/19/2025 Call Trace: <TASK> dump_stack_lvl+0xc5/0x110 ../lib/dump_stack.c:120 print_address_description ../mm/kasan/report.c:378 [inline] print_report+0xbc/0x260 ../mm/kasan/report.c:482 kasan_report+0xa2/0xe0 ../mm/kasan/report.c:595 check_region_inline ../mm/kasan/generic.c:-1 [inline] kasan_check_range+0x264/0x2c0 ../mm/kasan/generic.c:200 instrument_copy_to_user ../include/linux/instrumented.h:129 [inline] _inline_copy_to_user ../include/linux/uaccess.h:205 [inline] _copy_to_user+0x66/0xa0 ../lib/usercopy.c:26 copy_to_user ../include/linux/uaccess.h:236 [inline] sev_ioctl_do_pek_csr+0x31f/0x590 ../drivers/crypto/ccp/sev-dev.c:1872 sev_ioctl+0x3a4/0x490 ../drivers/crypto/ccp/sev-dev.c:2562 vfs_ioctl ../fs/ioctl.c:51 [inline] __do_sys_ioctl ../fs/ioctl.c:597 [inline] __se_sys_ioctl+0x11d/0x1b0 ../fs/ioctl.c:583 do_syscall_x64 ../arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xe0/0x800 ../arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> WARN if the driver says the command succeeded, but the firmware error code says otherwise, as __sev_do_cmd_locked() is expected to return -EIO on any firwmware error.	2026-05-01	7.1

CVE-2026-31707	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ksmbd: validate response sizes in ipc_validate_msg()
		
		ipc_validate_msg() computes the expected message size for each response type by adding (or multiplying) attacker-controlled fields from the daemon response to a fixed struct size in unsigned int arithmetic. Three cases can overflow:
		
		KSMBD_EVENT_RPC_REQUEST:
		msg_sz = sizeof(struct ksmbd_rpc_command) + resp->payload_sz;
		KSMBD_EVENT_SHARE_CONFIG_REQUEST:
		msg_sz = sizeof(struct ksmbd_share_config_response) + resp->payload_sz;
		KSMBD_EVENT_LOGIN_REQUEST_EXT:
		msg_sz = sizeof(struct ksmbd_login_response_ext) + resp->ngroups * sizeof(gid_t);
		
		resp->payload_sz is __u32 and resp->ngroups is __s32. Each addition can wrap in unsigned int; the multiplication by sizeof(gid_t) mixes signed and size_t, so a negative ngroups is converted to SIZE_MAX before the multiply. A wrapped value of msg_sz that happens to equal entry->msg_sz bypasses the size check on the next line, and downstream consumers (smb2pdu.c:6742 memcpy using rpc_resp->payload_sz, kmemdup in ksmbd_alloc_user using resp_ext->ngroups) then trust the unverified length.
		
		Use check_add_overflow() on the RPC_REQUEST and SHARE_CONFIG_REQUEST paths to detect integer overflow without constraining functional payload size; userspace ksmbd-tools grows NDR responses in 4096-byte chunks for calls like NetShareEnumAll, so a hard transport cap is unworkable on the response side. For LOGIN_REQUEST_EXT, reject resp->ngroups outside the signed [0, NGROUPS_MAX] range up front and
			2026-05-01	7.1

		report the error from ipc_validate_msg() so it fires at the IPC boundary; with that bound the subsequent multiplication and addition stay well below UINT_MAX. The now-redundant ngroups check and pr_err in ksmbd_alloc_user() are removed. This is the response-side analogue of aab98e2dbd64 ("ksmbd: fix integer overflows on 32 bit systems"), which hardened the request side.		
CVE-2026-31766	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: validate doorbell_offset in user queue creation amdgpu_userq_get_doorbell_index() passes the user-provided doorbell_offset to amdgpu_doorbell_index_on_bar() without bounds checking. An arbitrarily large doorbell_offset can cause the calculated doorbell index to fall outside the allocated doorbell BO, potentially corrupting kernel doorbell space. Validate that doorbell_offset falls within the doorbell BO before computing the BAR index, using u64 arithmetic to prevent overflow. (cherry picked from commit de1ef4ffd70e1d15f0bf584fd22b1f28cbd5e2ec)	2026-05-01	7.1
CVE-2026-31774	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring/net: fix slab-out-of-bounds read in io_bundle_nbufs() sqe->len is __u32 but gets stored into sr->len which is int. When userspace passes sqe->len values exceeding INT_MAX (e.g. 0xFFFFFFFF), sr->len overflows to a negative value. This negative value propagates through the bundle recv/send path: 1. io_recv(): sel.val = sr->len (ssize_t gets -1) 2. io_recv_buf_select(): arg.max_len = sel->val (size_t gets 0xFFFFFFFFFFFFFFFF) 3. io_ring_buffers_peek(): buf->len is not clamped because max_len is astronomically large 4. iov[].iov_len = 0xFFFFFFFF flows into io_bundle_nbufs() 5. io_bundle_nbufs(): min_t(int, 0xFFFFFFFF, ret) yields -1, causing ret to increase instead of decrease, creating an infinite loop that reads past the allocated iov[] array This results in a slab-out-of-bounds read in io_bundle_nbufs() from the kmalloc-64 slab, as nbufs increments past the allocated iovec entries. BUG: KASAN: slab-out-of-bounds in io_bundle_nbufs+0x128/0x160 Read of size 8 at addr ffff888100ae05c8 by task exp/145 Call Trace: io_bundle_nbufs+0x128/0x160 io_recv_finish+0x117/0xe20 io_recv+0x2db/0x1160 Fix this by rejecting negative sr->len values early in both io_sendmsg_prep() and io_recvmsg_prep(). Since sqe->len is __u32, any value > INT_MAX indicates overflow and is not a valid length.	2026-05-01	7.1
CVE-2026-43006	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring/rsrc: reject zero-length fixed buffer import validate_fixed_range() admits buf_addr at the exact end of the registered region when len is zero, because the check uses strict greater-than (buf_end > imu->ubuf + imu->len). io_import_fixed() then computes offset == imu->len, which causes the bvec skip logic to advance past the last bio_vec entry and read bv_offset from out-of-bounds slab memory. Return early from io_import_fixed() when len is zero. A zero-length import has no data to transfer and should not walk the bvec array at all. BUG: KASAN: slab-out-of-bounds in io_import_reg_buf+0x697/0x7f0 Read of size 4 at addr ffff888002bcc254 by task poc/103 Call Trace: io_import_reg_buf+0x697/0x7f0 io_write_fixed+0xd9/0x250 __io_issue_sqe+0xad/0x710 io_issue_sqe+0x7d/0x1100 io_submit_sqes+0x86a/0x23c0 __do_sys_io_uring_enter+0xa98/0x1590 Allocated by task 103:	2026-05-01	7.1

		The buggy address is located 12 bytes to the right of allocated 584-byte region [ffff888002bcc000, ffff888002bcc248)		
CVE-2026-43028	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: ensure names are nul-terminated Reject names that lack a \0 character before feeding them to functions that expect c-strings. Fixes tag is the most recent commit that needs this change.	2026-05-01	7.1
CVE-2026-43040	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: ndisc: fix ndisc_ra_useropt to initialize nduseropt_padX fields to zero to prevent an info-leak When processing Router Advertisements with user options the kernel builds an RTM_NEWNDUSEROPT netlink message. The nduseroptmsg struct has three padding fields that are never zeroed and can leak kernel data The fix is simple, just zeroes the padding fields.	2026-05-01	7.1
CVE-2026-43042	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mpls: add seqcount to protect the platform_label{,s} pair The RCU-protected codepaths (mpls_forward, mpls_dump_routes) can have an inconsistent view of platform_labels vs platform_label in case of a concurrent resize (resize_platform_label_table, under platform_mutex). This can lead to OOB accesses. This patch adds a seqcount, so that we get a consistent snapshot. Note that mpls_label_ok is also susceptible to this, so the check against RTA_DST in rtm_to_route_config, done outside platform_mutex, is not sufficient. This value gets passed to mpls_label_ok once more in both mpls_route_add and mpls_route_del, so there is no issue, but that additional check must not be removed.	2026-05-01	7.1
CVE-2026-43052	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: check tdls flag in ieee80211_tdls_oper When NL80211_TDLS_ENABLE_LINK is called, the code only checks if the station exists but not whether it is actually a TDLS station. This allows the operation to proceed for non-TDLS stations, causing unintended side effects like modifying channel context and HT protection before failing. Add a check for sta->sta.tdls early in the ENABLE_LINK case, before any side effects occur, to ensure the operation is only allowed for actual TDLS peers.	2026-05-01	7.1
CVE-2026-40973	vmware - multiple products	A local attacker on the same host as the application may be able to take control of the directory used by `ApplicationTemp`. When `server.servlet.session.persistent` is set to `true` and the attack persists across application restarts, this may allow the attacker to read session information and hijack authenticated users or deploy a gadget chain and execute code as the application's user. Affected: Spring Boot 4.0.0–4.0.5 (fix 4.0.6), 3.5.0–3.5.13 (fix 3.5.14), 3.4.0–3.4.15 (fix 3.4.16), 3.3.0–3.3.18 (fix 3.3.19), 2.7.0–2.7.32 (fix 2.7.33); predictable temp directory / `ApplicationTemp` ownership verification. Versions that are no longer supported are also affected per vendor advisory.	2026-04-28	7
CVE-2026-43050	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: atm: lec: fix use-after-free in sock_def_readable() A race condition exists between lec_atm_close() setting priv->lecd to NULL and concurrent access to priv->lecd in send_to_lecd(), lec_handle_bridge(), and lec_atm_send(). When the socket is freed via RCU while another thread is still using it, a use-after-free occurs in sock_def_readable() when accessing the socket's wait queue. The root cause is that lec_atm_close() clears priv->lecd without any synchronization, while callers dereference priv->lecd without any protection against concurrent teardown. Fix this by converting priv->lecd to an RCU-protected pointer: - Mark priv->lecd as __rcu in lec.h - Use rcu_assign_pointer() in lec_atm_close() and lecd_attach() for safe pointer assignment - Use rcu_access_pointer() for NULL checks that do not dereference the pointer in lec_start_xmit(), lec_push(), send_to_lecd() and lecd_attach() - Use rcu_read_lock/rcu_dereference/rcu_read_unlock in send_to_lecd(), lec_handle_bridge() and lec_atm_send() to safely access lecd	2026-05-01	7

		- Use rcu_assign_pointer() followed by synchronize_rcu() in lec_atm_close() to ensure all readers have completed before proceeding. This is safe since lec_atm_close() is called from vcc_release() which holds lock_sock(), a sleeping lock. - Remove the manual sk_receive_queue drain from lec_atm_close() since vcc_destroy_socket() already drains it after lec_atm_close() returns. v2: Switch from spinlock + sock_hold/put approach to RCU to properly fix the race. The v1 spinlock approach had two issues pointed out by Eric Dumazet: 1. priv->lecd was still accessed directly after releasing the lock instead of using a local copy. 2. The spinlock did not prevent packets being queued after lec_atm_close() drains sk_receive_queue since timer and workqueue paths bypass netif_stop_queue(). Note: Syzbot patch testing was attempted but the test VM terminated unexpectedly with "Connection to localhost closed by remote host", likely due to a QEMU AHCI emulation issue unrelated to this fix. Compile testing with "make W=1 net/atm/lec.o" passes cleanly.		
CVE-2026-21023	samsung - multiple products	Insufficient verification of data authenticity in PackageManagerService prior to SMR Mar-2026 Release 1 allows local attackers to modify the installation restriction of specific application.	2026-04-29	6.9
CVE-2026-0711	zyxel - DX3300-T0 firmware	A post-authentication command injection vulnerability in the EasyMesh-related APIs of Zyxel DX3300-T0 firmware versions through 5.50(ABVY.7.1)C0 could allow an authenticated, adjacent attacker with administrator privileges to execute OS commands on an affected device.	2026-04-28	6.8
CVE-2026-0205	sonicwall - sonicos	A post-authentication Path Traversal vulnerability in SonicOS allows an attacker to interact with usually restricted services.	2026-04-29	6.8
CVE-2026-25908	dell - alienware_command_center	Dell Alienware Command Center (AWCC), versions prior to 6.13.8.0, contain an Execution with Unnecessary Privileges vulnerability in the AWCC. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of Privileges.	2026-04-27	6.7
CVE-2026-41081	apache - storm	Improper Handling of TLS Client Authentication Failure Leading to Anonymous Principal Assignment in Apache Storm Versions Affected: up to 2.8.7 Description: When TLS transport is enabled in Apache Storm without requiring client certificate authentication (the default configuration), the TlsTransportPlugin assigns a fallback principal (CN=ANONYMOUS) if no client certificate is presented or if certificate verification fails. The underlying SSLPeerUnverifiedException is caught and suppressed rather than rejecting the connection. This fail-open behavior means an unauthenticated client can establish a TLS connection and receive a valid principal identity. If the configured authorizer (e.g., SimpleACLAuthorizer) does not explicitly deny access to CN=ANONYMOUS, this may result in unauthorized access to Storm services. The condition is logged at debug level only, reducing visibility in production. Impact: Unauthenticated clients may be assigned a principal identity, potentially bypassing authorization in permissive or misconfigured environments. Mitigation: Users should upgrade to 2.8.7 in which TLS authentication failures are handled in a fail-closed manner. Users who cannot upgrade immediately should: - Enable mandatory client certificate authentication (nimbus.thrift.tls.client.auth.required: true) - Ensure authorization rules explicitly deny access to CN=ANONYMOUS - Review all ACL configurations for implicit default-allow behavior	2026-04-27	6.5
CVE-2026-40980	vmware - multiple products	In Spring AI, a malicious PDF file can be crafted that triggers the allocation of unreasonable amounts of memory when handled by `ForkPDFLayoutTextStripper`. Affected versions: Spring AI: 1.0.0 - 1.0.5 (fixed in 1.0.6), 1.1.0 - 1.1.4 (fixed in 1.1.5)	2026-04-28	6.5
CVE-2026-41607	apache - thrift	Out-of-bounds Read vulnerability in Apache Thrift. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	6.5
CVE-2026-6238	gnu - glibc	The deprecated functions ns_printrrf, ns_printrr and fp_nquery in the GNU C Library version 2.2 and newer fail to validate the RDATA content against the RDATA length in a DNS response when processing LOC, CERT, TKEY or TSIG records, which may allow an attacker to craft a DNS response, causing a target application to crash or read uninitialized memory. These functions are for application debugging only and hence not in the path of code executed by the DNS resolver. Further, they have been deprecated since version 2.34 and should not be used by any new applications. Applications should consider porting away from these interfaces since they may be removed in future versions.	2026-04-28	6.5
CVE-2026-22740	vmware - multiple products	A WebFlux server application that processes multipart requests creates temp files for parts larger than 10 K. Under some circumstances, temp files may remain not deleted after the request is fully processed. This allows an attacker to consume available disk space.	2026-04-29	6.5

		Older, unsupported versions are also affected.		
CVE-2026-42521	jenkins - multiple products	Jenkins Matrix Authorization Strategy Plugin 2.0-beta-1 through 3.2.9 (both inclusive) invokes parameterless constructors of classes specified in configuration when deserializing inheritance strategies, without restricting the classes that can be instantiated, allowing attackers with Item/Configure permission to instantiate arbitrary types, which may lead to information disclosure or other impacts depending on the classes available on the classpath.	2026-04-29	6.5
CVE-2026-3833	gnu - multiple products	A flaw was found in gnutls. This vulnerability occurs because gnutls performs case-sensitive comparisons of `nameConstraints` labels, specifically for `dNSName` (DNS) or `rfc822Name` (email) constraints within `excludedSubtrees` or `permittedSubtrees`. A remote attacker can exploit this by crafting a leaf certificate with casing differences in the Subject Alternative Name (SAN), leading to a policy bypass where a certificate that should be rejected is instead accepted. This could result in unauthorized access or information disclosure.	2026-04-30	6.5
CVE-2026-3340	ibm - Langflow Desktop	IBM Langflow Desktop 1.0.0 through 1.8.4 IBM Langflow is vulnerable to server-side request forgery (SSRF). This may allow an authenticated attacker to send unauthorized requests from the system, potentially leading to network enumeration or facilitating other attacks.	2026-04-30	6.5
CVE-2026-4502	ibm - Langflow Desktop	IBM Langflow Desktop 1.2.0 through 1.8.4 Langflow could allow an authenticated attacker to traverse directories on the system. An attacker could send a specially crafted URL request containing "dot dot" sequences (/../) to write arbitrary files on the system.	2026-04-30	6.5
CVE-2025-36122	ibm - multiple products	IBM Db2 11.5.0 through 11.5.9, and 12.1.0 through 12.1.3 for Linux, UNIX and Windows (includes DB2 Connect Server) could allow an authenticated user to cause a denial of service using a specially crafted SQL query due to improper allocation of system resources.	2026-04-30	6.5
CVE-2026-1577	ibm - multiple products	IBM Db2 11.5.0 through 11.5.9, and 12.1.0 through 12.1.4 for Linux, UNIX and Windows (includes Db2 Connect Server) could allow an authenticated user to cause a denial of service due to improper neutralization of special elements in data query logic.	2026-04-30	6.5
CVE-2026-3345	ibm - Langflow Desktop	IBM Langflow Desktop <=1.8.4 Langflow could allow a remote attacker to traverse directories on the system. An attacker could send a specially crafted URL request containing "dot dot" sequences (/../) to view arbitrary files on the system.	2026-04-30	6.5
CVE-2026-40685	exim - exim	In Exim before 4.99.2, when JSON lookup is enabled, an out-of-bounds heap write can occur when a JSON operator encounters malformed JSON in an untrusted header, because of an incorrect implementation of \ skipping.	2026-04-30	6.5
CVE-2026-28909	apple - container	Users who connect to malicious registries with hostnames matching the bypass patterns will have their registry credentials exposed in plaintext. This issue is fixed in container version 0.12.3.	2026-04-30	6.5
CVE-2026-42404	apache - neethi	Apache Neethi does not impose any restrictions on URIs when manually fetching remote policy references through the PolicyReference API. When an application explicitly calls the API to retrieve a policy from a remote URI, an outbound request is made for arbitrary protocols and internal IP adddresses. From 3.2.2, only http or https URIs are allowed, and link-local/multicast/any-local addresses are forbidden. Users are recommended to upgrade to version 3.2.2, which fixes this issue.	2026-05-01	6.5
CVE-2026-3346	ibm - Langflow Desktop	IBM Langflow Desktop 1.6.0 through 1.8.4 Lanflow is vulnerable to stored cross-site scripting. This vulnerability allows an authenticated user to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session.	2026-04-30	6.4
CVE-2026-2311	ibm - multiple products	IBM i 7.6, 7.5, 7.4, 7.3, and 7.2 s vulnerable to privilege escalation caused by an invalid IBM i Web Administration GUI authorization check. A malicious actor could cause user-controlled code to run with administrator privilege.	2026-04-30	6.4
CVE-2026-27105	dell - dell\/alienware_purchased_apps	Dell/Alienware Purchased Apps, versions prior to 1.1.31.0, contain an Improper Link Resolution Before File Access ('Link Following') vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Arbitrary File Write	2026-04-29	6.3
CVE-2026-7606	trendnet - tew-821dap_firmware	A weakness has been identified in TRENDnet TEW-821DAP 1.12B01. This issue affects the function find_hwid/new_gui_update_firmware of the component Firmware Update Handler. Executing a manipulation of the argument dest can lead to insufficient verification of data authenticity. The attack can be launched remotely. Attacks of this nature are highly complex. The exploitability is assessed as difficult. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	6.3
CVE-2026-7611	trendnet - tew-821dap_firmware	A vulnerability was found in TRENDnet TEW-821DAP up to 1.12B01. This impacts the function platform_do_upgrade_cameo_dev of the file cameo_dev.sh of the component Firmware Update Handler. Performing a manipulation results in insufficient verification of data authenticity. The attack is possible to be carried out remotely. The complexity of an attack is rather high. The exploitability is said to be difficult. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	6.3
CVE-2025-36335	ibm - watsonx.data intelligence	IBM watsonx.data intelligence 5.2.0, 5.2.1, 5.3.0, 5.3.1 stores user credentials in plain text which can be read by a local user.	2026-04-30	6.2
CVE-2026-40979	vmware - multiple products	In Spring AI, having access to a shared environment can expose the ONNX model used by the application. Affected versions: Spring AI: 1.0.0 - 1.0.5 (fixed in 1.0.6), 1.1.0 - 1.1.4 (fixed in 1.1.5)	2026-04-28	6.1
CVE-2025-10503	wso2 - identity_server	The authentication endpoint accepts user-supplied input without enforcing expected validation constraints, leading to a lack of proper output encoding. This allows for the injection of malicious JavaScript payloads, enabling reflected cross-site scripting. An attacker can leverage this vulnerability to redirect the user's browser to a malicious website, modify the user interface of the web page, retrieve information from the browser, or cause other	2026-04-29	6.1

		harmful actions. However, due to the protection of session-related cookies with the httpOnly flag, session hijacking is not possible.		
CVE-2026-7163	redhat - multiple products	A vulnerability in the assisted-service REST API, an optional Assisted Installer (assisted-service) component in the Multicluster Engine (MCE), allows an authenticated user with minimal namespace-scoped privileges to obtain administrative credentials for arbitrary clusters provisioned through the hub. The credentials download endpoint (GET /v2/clusters/{cluster_id}/credentials, which returns the kubeadmin password) and the kubeconfig download endpoint are operational in AUTH_TYPE=local mode, the only authentication mode available in on-premises ACM/MCE hub deployments. The local authenticator unconditionally grants full administrative access to any request bearing a valid JWT, with no per-endpoint restrictions. A valid local JWT is embedded as a plaintext query parameter in InfraEnvStatus.ISODownloadURL and is readable by any user who has get rights on an InfraEnv object in their own namespace. The affected components ship as part of Multicluster Engine (MCE). The Red Hat Advanced Cluster Management (ACM) deployments that include MCE are equally affected. This issue does not affect the hosted SaaS offering (console.redhat.com), which uses a different authentication mode. Successful exploitation gives the attacker the kubeadmin password and kubeconfig for any OpenShift cluster provisioned through the affected hub, granting unrestricted root-level administrative access to those spoke clusters.	2026-04-30	6.1
CVE-2026-40966	vmware - multiple products	In Spring AI, an attacker can bypass conversation isolation and exfiltrate sensitive memory from other users' chat histories, including secrets and credentials, by injecting filter logic through conversationId. Only applications that use VectorStoreChatMemoryAdvisor and pass user-supplied input as a conversationId are affected.	2026-04-28	5.9
CVE-2026-41016	apache - airflow	Apache Airflow's SMTP provider `SmtpHook` called Python's `smtplib.SMTP.starttls()` without an SSL context, so no certificate validation was performed on the TLS upgrade. A man-in-the-middle between the Airflow worker and the SMTP server could present a self-signed certificate, complete the STARTTLS upgrade, and capture the SMTP credentials sent during the subsequent `login()` call. Users are advised to upgrade to the `apache-airflow-providers-smtp` version that contains the fix.	2026-04-30	5.9
CVE-2026-40684	exim - exim	In Exim before 4.99.2, on systems using musl libc (not glibc), an attacker can crash the connection instance when malformed DNS data is present in PTR records. This is caused by a dn_expand oddity in octal printing.	2026-04-30	5.9
CVE-2026-31687	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio: omap: do not register driver in probe() Commit 11a78b794496 ("ARM: OMAP: MPUIO wake updates") registers the omap_mpuio_driver from omap_mpuio_init(), which is called from omap_gpio_probe(). However, it neither makes sense to register drivers from probe() callbacks of other drivers, nor does the driver core allow registering drivers with a device lock already being held. The latter was revealed by commit dc23806a7c47 ("driver core: enforce device_lock for driver_match_device()") leading to a potential deadlock condition described in [1]. Additionally, the omap_mpuio_driver is never unregistered from the driver core, even if the module is unloaded. Hence, register the omap_mpuio_driver from the module initcall and unregister it in module_exit().	2026-04-27	5.5
CVE-2026-31689	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: EDAC/mc: Fix error path ordering in edac_mc_alloc() When the mci->pvt_info allocation in edac_mc_alloc() fails, the error path will call put_device() which will end up calling the device's release function. However, the init ordering is wrong such that device_initialize() happens *after* the failed allocation and thus the device itself and the release function pointer are not initialized yet when they're called: MCE: In-kernel MCE decoding enabled. ------------[cut here]------------ kobject: '(null)': is not initialized, yet kobject_put() is being called. WARNING: lib/kobject.c:734 at kobject_put, CPU#22: systemd-udevd CPU: 22 UID: 0 PID: 538 Comm: systemd-udevd Not tainted 7.0.0-rc1+ #2 PREEMPT(full) RIP: 0010:kobject_put Call Trace: <TASK> edac_mc_alloc+0xbe/0xe0 [edac_core] amd64_edac_init+0x7a4/0xff0 [amd64_edac] ? __pfx_amd64_edac_init+0x10/0x10 [amd64_edac] do_one_initcall	2026-04-27	5.5

		... Reorder the calling sequence so that the device is initialized and thus the release function pointer is properly set before it can be used. This was found by Claude while reviewing another EDAC patch.		
CVE-2026-31691	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: igb: remove napi_synchronize() in igb_down() When an AF_XDP zero-copy application terminates abruptly (e.g., kill -9), the XSK buffer pool is destroyed but NAPI polling continues. igb_clean_rx_irq_zc() repeatedly returns the full budget, preventing napi_complete_done() from clearing NAPI_STATE_SCHED. igb_down() calls napi_synchronize() before napi_disable() for each queue vector. napi_synchronize() spins waiting for NAPI_STATE_SCHED to clear, which never happens. igb_down() blocks indefinitely, the TX watchdog fires, and the TX queue remains permanently stalled. napi_disable() already handles this correctly: it sets NAPI_STATE_DISABLE. After a full-budget poll, __napi_poll() checks napi_disable_pending(). If set, it forces completion and clears NAPI_STATE_SCHED, breaking the loop that napi_synchronize() cannot. napi_synchronize() was added in commit 41f149a285da ("igb: Fix possible panic caused by Rx traffic arrival while interface is down"). napi_disable() provides stronger guarantees: it prevents further scheduling and waits for any active poll to exit. Other Intel drivers (ixgbe, ice, i40e) use napi_disable() without a preceding napi_synchronize() in their down paths. Remove redundant napi_synchronize() call and reorder napi_disable() before igb_set_queue_napi() so the queue-to-NAPI mapping is only cleared after polling has fully stopped.	2026-04-27	5.5
CVE-2026-31692	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rtnetlink: add missing netlink_ns_capable() check for peer netns rtnl_newlink() lacks a CAP_NET_ADMIN capability check on the peer network namespace when creating paired devices (veth, vxcan, netkit). This allows an unprivileged user with a user namespace to create interfaces in arbitrary network namespaces, including init_net. Add a netlink_ns_capable() check for CAP_NET_ADMIN in the peer namespace before allowing device creation to proceed.	2026-04-30	5.5
CVE-2026-31701	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: caiaq: take a reference on the USB device in create_card() The caiaq driver stores a pointer to the parent USB device in cdev->chip.dev but never takes a reference on it. The card's private_free callback, snd_usb_caiaq_card_free(), can run asynchronously via snd_card_free_when_closed() after the USB device has already been disconnected and freed, so any access to cdev->chip.dev in that path dereferences a freed usb_device. On top of the refcounting issue, the current card_free implementation calls usb_reset_device(cdev->chip.dev). A reset in a free callback is inappropriate: the device is going away, the call takes the device lock in a teardown context, and the reset races with the disconnect path that the callback is already cleaning up after. Take a reference on the USB device in create_card() with usb_get_dev(), drop it with usb_put_dev() in the free callback, and remove the usb_reset_device() call.	2026-05-01	5.5
CVE-2026-31704	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: use check_add_overflow() to prevent u16 DACL size overflow set_posix_acl_entries_dacl() and set_ntacl_dacl() accumulate ACE sizes in u16 variables. When a file has many POSIX ACL entries, the accumulated size can wrap past 65535, causing the pointer arithmetic (char *)pndace + *size to land within already-written ACEs. Subsequent writes then overwrite earlier entries, and pndacl->size gets a truncated value. Use check_add_overflow() at each accumulation point to detect the	2026-05-01	5.5

		wrap before it corrupts the buffer, consistent with existing check_mul_overflow() usage elsewhere in smbacl.c.		
CVE-2026-31710	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix dir separator in SMB1 UNIX mounts When calling cifs_mount_get_tcon() with SMB1 UNIX mounts, @cifs_sb->mnt_cifs_flags needs to be read or updated only after calling reset_cifs_unix_caps(), otherwise it might end up with missing CIFS_MOUNT_POSIXACL and CIFS_MOUNT_POSIX_PATHS bits. This fixes the wrong dir separator used in paths caused by the missing CIFS_MOUNT_POSIX_PATHS bit in cifs_sb_info::mnt_cifs_flags.	2026-05-01	5.5
CVE-2026-31713	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fuse: abort on fatal signal during sync init When sync init is used and the server exits for some reason (error, crash) while processing FUSE_INIT, the filesystem creation will hang. The reason is that while all other threads will exit, the mounting thread (or process) will keep the device fd open, which will prevent an abort from happening. This is a regression from the async mount case, where the mount was done first, and the FUSE_INIT processing afterwards, in which case there's no such recursive syscall keeping the fd open.	2026-05-01	5.5
CVE-2026-31714	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid memory leak in f2fs_rename() syzbot reported a f2fs bug as below: BUG: memory leak unreferenced object 0xffff888127f70830 (size 16): comm "syz.0.23", pid 6144, jiffies 4294943712 hex dump (first 16 bytes): 3c af 57 72 5b e6 8f ad 6e 8e fd 33 42 39 03 ff <.Wr[...n..3B9.. backtrace (crc 925f8a80): kmemleak_alloc_recursive include/linux/kmemleak.h:44 [inline] slab_post_alloc_hook mm/slub.c:4520 [inline] slab_alloc_node mm/slub.c:4844 [inline] __do_kmalloc_node mm/slub.c:5237 [inline] __kmalloc_noprof+0x3bd/0x560 mm/slub.c:5250 kmalloc_noprof include/linux/slab.h:954 [inline] fscrypt_setup_filename+0x15e/0x3b0 fs/crypto/fname.c:364 f2fs_setup_filename+0x52/0xb0 fs/f2fs/dir.c:143 f2fs_rename+0x159/0xca0 fs/f2fs/namei.c:961 f2fs_rename2+0xd5/0xf20 fs/f2fs/namei.c:1308 vfs_rename+0x7ff/0x1250 fs/namei.c:6026 filename_renameat2+0x4f4/0x660 fs/namei.c:6144 __do_sys_renameat2 fs/namei.c:6173 [inline] __se_sys_renameat2 fs/namei.c:6168 [inline] __x64_sys_renameat2+0x59/0x80 fs/namei.c:6168 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xe2/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f The root cause is in commit 40b2d55e0452 ("f2fs: fix to create selinux label during whiteout initialization"), we added a call to f2fs_setup_filename() without a matching call to f2fs_free_filename(), fix it.	2026-05-01	5.5
CVE-2026-31721	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_hid: move list and spinlock inits from bind to alloc There was an issue when you did the following: - setup and bind an hid gadget - open /dev/hidg0 - use the resulting fd in EPOLL_CTL_ADD - unbind the UDC - bind the UDC - use the fd in EPOLL_CTL_DEL When CONFIG_DEBUG_LIST was enabled, a list_del corruption was reported within remove_wait_queue (via ep_remove_wait_queue). After some debugging I found out that the queues, which f_hid registers via poll_wait were the problem. These were initialized using init_waitqueue_head inside hidg_bind. So effectively, the bind function re-initialized the queues while there were still items in them. The solution is to move the initialization from hidg_bind to hidg_alloc	2026-05-01	5.5

		to extend their lifetimes to the lifetime of the function instance. Additionally, I found many other possibly problematic init calls in the bind function, which I moved as well.		
CVE-2026-31722	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_rndis: Fix net_device lifecycle with device_move The net_device is allocated during function instance creation and registered during the bind phase with the gadget device as its sysfs parent. When the function unbinds, the parent device is destroyed, but the net_device survives, resulting in dangling sysfs symlinks: console:/ # ls -l /sys/class/net/usb0 lrwxrwxrwx ... /sys/class/net/usb0 -> /sys/devices/platform/.../gadget.0/net/usb0 console:/ # ls -l /sys/devices/platform/.../gadget.0/net/usb0 ls: .../gadget.0/net/usb0: No such file or directory Use device_move() to reparent the net_device between the gadget device tree and /sys/devices/virtual across bind and unbind cycles. During the final unbind, calling device_move(NULL) moves the net_device to the virtual device tree before the gadget device is destroyed. On rebinding, device_move() reparents the device back under the new gadget, ensuring proper sysfs topology and power management ordering. To maintain compatibility with legacy composite drivers (e.g., multi.c), the borrowed_net flag is used to indicate whether the network device is shared and pre-registered during the legacy driver's bind phase.	2026-05-01	5.5
CVE-2026-31723	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_subset: Fix net_device lifecycle with device_move The net_device is allocated during function instance creation and registered during the bind phase with the gadget device as its sysfs parent. When the function unbinds, the parent device is destroyed, but the net_device survives, resulting in dangling sysfs symlinks: console:/ # ls -l /sys/class/net/usb0 lrwxrwxrwx ... /sys/class/net/usb0 -> /sys/devices/platform/.../gadget.0/net/usb0 console:/ # ls -l /sys/devices/platform/.../gadget.0/net/usb0 ls: .../gadget.0/net/usb0: No such file or directory Use device_move() to reparent the net_device between the gadget device tree and /sys/devices/virtual across bind and unbind cycles. During the final unbind, calling device_move(NULL) moves the net_device to the virtual device tree before the gadget device is destroyed. On rebinding, device_move() reparents the device back under the new gadget, ensuring proper sysfs topology and power management ordering. To maintain compatibility with legacy composite drivers (e.g., multi.c), the bound flag is used to indicate whether the network device is shared and pre-registered during the legacy driver's bind phase.	2026-05-01	5.5
CVE-2026-31724	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_eem: Fix net_device lifecycle with device_move The net_device is allocated during function instance creation and registered during the bind phase with the gadget device as its sysfs parent. When the function unbinds, the parent device is destroyed, but the net_device survives, resulting in dangling sysfs symlinks: console:/ # ls -l /sys/class/net/usb0 lrwxrwxrwx ... /sys/class/net/usb0 -> /sys/devices/platform/.../gadget.0/net/usb0 console:/ # ls -l /sys/devices/platform/.../gadget.0/net/usb0 ls: .../gadget.0/net/usb0: No such file or directory Use device_move() to reparent the net_device between the gadget device tree and /sys/devices/virtual across bind and unbind cycles. During the final unbind, calling device_move(NULL) moves the net_device to the virtual device tree before the gadget device is destroyed. On rebinding, device_move() reparents the device back under the new gadget, ensuring proper sysfs topology and power management ordering. To maintain compatibility with legacy composite drivers (e.g., multi.c), the bound flag is used to indicate whether the network device is shared and pre-registered during the legacy driver's bind phase.	2026-05-01	5.5

CVE-2026-31725	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_ecm: Fix net_device lifecycle with device_move The net_device is allocated during function instance creation and registered during the bind phase with the gadget device as its sysfs parent. When the function unbinds, the parent device is destroyed, but the net_device survives, resulting in dangling sysfs symlinks: console:/ # ls -l /sys/class/net/usb0 lrwxrwxrwx ... /sys/class/net/usb0 -> /sys/devices/platform/.../gadget.0/net/usb0 console:/ # ls -l /sys/devices/platform/.../gadget.0/net/usb0 ls: .../gadget.0/net/usb0: No such file or directory Use device_move() to reparent the net_device between the gadget device tree and /sys/devices/virtual across bind and unbind cycles. During the final unbind, calling device_move(NULL) moves the net_device to the virtual device tree before the gadget device is destroyed. On rebinding, device_move() reparents the device back under the new gadget, ensuring proper sysfs topology and power management ordering. To maintain compatibility with legacy composite drivers (e.g., multi.c), the bound flag is used to indicate whether the network device is shared and pre-registered during the legacy driver's bind phase.	2026-05-01	5.5
CVE-2026-31726	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: uvc: fix NULL pointer dereference during unbind race Commit b81ac4395bbe ("usb: gadget: uvc: allow for application to cleanly shutdown") introduced two stages of synchronization waits totaling 1500ms in uvc_function_unbind() to prevent several types of kernel panics. However, this timing-based approach is insufficient during power management (PM) transitions. When the PM subsystem starts freezing user space processes, the wait_event_interruptible_timeout() is aborted early, which allows the unbind thread to proceed and nullify the gadget pointer (cdev->gadget = NULL): [814.123447][T947] configfs-gadget.g1 gadget.0: uvc: uvc_function_unbind() [814.178583][T3173] PM: suspend entry (deep) [814.192487][T3173] Freezing user space processes [814.197668][T947] configfs-gadget.g1 gadget.0: uvc: uvc_function_unbind no clean disconnect, wait for release When the PM subsystem resumes or aborts the suspend and tasks are restarted, the V4L2 release path is executed and attempts to access the already nullified gadget pointer, triggering a kernel panic: [814.292597][C0] PM: pm_system_irq_wakeup: 479 triggered dhdpcie_host_wake [814.386727][T3173] Restarting tasks ... [814.403522][T4558] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000030 [814.404021][T4558] pc : usb_gadget_deactivate+0x14/0xf4 [814.404031][T4558] lr : usb_function_deactivate+0x54/0x94 [814.404078][T4558] Call trace: [814.404080][T4558] usb_gadget_deactivate+0x14/0xf4 [814.404083][T4558] usb_function_deactivate+0x54/0x94 [814.404087][T4558] uvc_function_disconnect+0x1c/0x5c [814.404092][T4558] uvc_v4l2_release+0x44/0xac [814.404095][T4558] v4l2_release+0xcc/0x130 Address the race condition and NULL pointer dereference by: 1. State Synchronization (flag + mutex) Introduce a 'func_unbound' flag in struct uvc_device. This allows uvc_function_disconnect() to safely skip accessing the nullified cdev->gadget pointer. As suggested by Alan Stern, this flag is protected by a new mutex (uvc->lock) to ensure proper memory ordering and prevent instruction reordering or speculative loads. This mutex is also used to protect 'func_connected' for consistent state management. 2. Explicit Synchronization (completion) Use a completion to synchronize uvc_function_unbind() with the uvc_vdev_release() callback. This prevents Use-After-Free (UAF) by ensuring struct uvc_device is freed after all video device resources are released.	2026-05-01	5.5
CVE-2026-31727	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-01	5.5

		usb: gadget: u_ether: Fix NULL pointer deref in eth_get_drvinfo Commit ec35c1969650 ("usb: gadget: f_ncm: Fix net_device lifecycle with device_move") reparents the gadget device to /sys/devices/virtual during unbind, clearing the gadget pointer. If the userspace tool queries on the surviving interface during this detached window, this leads to a NULL pointer dereference. Unable to handle kernel NULL pointer dereference Call trace: eth_get_drvinfo+0x50/0x90 ethtool_get_drvinfo+0x5c/0x1f0 __dev_ethtool+0xaec/0x1fe0 dev_ethtool+0x134/0x2e0 dev_ioctl+0x338/0x560 Add a NULL check for dev->gadget in eth_get_drvinfo(). When detached, skip copying the fw_version and bus_info strings, which is natively handled by ethtool_get_drvinfo for empty strings.		

CVE-2026-31732	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		gpio: Fix resource leaks on errors in gpiochip_add_data_with_key()
		
		Since commit aab5c6f20023 ("gpio: set device type for GPIO chips"), `gdev->dev.release` is unset. As a result, the reference count to `gdev->dev` isn't dropped on the error handling paths.
		
		Drop the reference on errors.
		
		Also reorder the instructions to make the error handling simpler. Now gpiochip_add_data_with_key() roughly looks like:
		
		>>> Some memory allocation. Go to ERR ZONE 1 on errors.
		>>> device_initialize().
		
		gpiodev_release() takes over the responsibility for freeing the resources of `gdev->dev`. The subsequent error handling paths shouldn't go through ERR ZONE 1 again which leads to double free.
		
		>>> Some initialization mainly on `gdev`.
		>>> The rest of initialization. Go to ERR ZONE 2 on errors.
		>>> Chip registration success and exit.
		
		>>> ERR ZONE 2. gpio_device_put() and exit.
		>>> ERR ZONE 1.
			2026-05-01	5.5

CVE-2026-31733	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sched_ext: Fix stale direct dispatch state in ddsp_dsq_id @p->scx.ddsp_dsq_id can be left set (non-SCX_DSQ_INVALID) triggering a spurious warning in mark_direct_dispatch() when the next wakeup's ops.select_cpu() calls scx_bpf_dsq_insert(), such as: WARNING: kernel/sched/ext.c:1273 at scx_dsq_insert_commit+0xcd/0x140 The root cause is that ddsp_dsq_id was only cleared in dispatch_enqueue(), which is not reached in all paths that consume or cancel a direct dispatch verdict. Fix it by clearing it at the right places: - direct_dispatch(): cache the direct dispatch state in local variables and clear it before dispatch_enqueue() on the synchronous path. For the deferred path, the direct dispatch state must remain set until process_ddsp_deferred_locals() consumes them. - process_ddsp_deferred_locals(): cache the dispatch state in local variables and clear it before calling dispatch_to_local_dsq(), which may migrate the task to another rq. - do_enqueue_task(): clear the dispatch state on the enqueue path (local/global/bypass fallbacks), where the direct dispatch verdict is ignored. - dequeue_task_scx(): clear the dispatch state after dispatch_dequeue() to handle both the deferred dispatch cancellation and the holding_cpu race, covering all cases where a pending direct dispatch is cancelled. - scx_disable_task(): clear the direct dispatch state when	2026-05-01	5.5

		transitioning a task out of the current scheduler. Waking tasks may have had the direct dispatch state set by the outgoing scheduler's ops.select_cpu() and then been queued on a wake_list via ttwu_queue_wakelist(), when SCX_OPS_ALLOW_QUEUED_WAKEUP is set. Such tasks are not on the runqueue and are not iterated by scx_bypass(), so their direct dispatch state won't be cleared. Without this clear, any subsequent SCX scheduler that tries to direct dispatch the task will trigger the WARN_ON_ONCE() in mark_direct_dispatch().		
CVE-2026-31734	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sched_ext: Fix is_bpf_migration_disabled() false negative on non-PREEMPT_RCU Since commit 8e4f0b1ebcf2 ("bpf: use rcu_read_lock_dont_migrate() for trampoline.c"), the BPF prolog (__bpf_prog_enter) calls migrate_disable() only when CONFIG_PREEMPT_RCU is enabled, via rcu_read_lock_dont_migrate(). Without CONFIG_PREEMPT_RCU, the prolog never touches migration_disabled, so migration_disabled == 1 always means the task is truly migration-disabled regardless of whether it is the current task. The old unconditional p == current check was a false negative in this case, potentially allowing a migration-disabled task to be dispatched to a remote CPU and triggering scx_error in task_can_run_on_remote_rq(). Only apply the p == current disambiguation when CONFIG_PREEMPT_RCU is enabled, where the ambiguity with the BPF prolog still exists.	2026-05-01	5.5
CVE-2026-31736	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_ppe: avoid NULL deref when gmac0 is disabled If the gmac0 is disabled, the precheck for a valid ingress device will cause a NULL pointer deref and crash the system. This happens because eth->netdev[0] will be NULL but the code will directly try to access netdev_ops. Instead of just checking for the first net_device, it must be checked if any of the mtk_eth net_devices is matching the netdev_ops of the ingress device.	2026-05-01	5.5
CVE-2026-31737	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ftgmac100: fix ring allocation unwind on open failure ftgmac100_alloc_rings() allocates rx_skbs, tx_skbs, rxdes, txdes, and rx_scratch in stages. On intermediate failures it returned -ENOMEM directly, leaking resources allocated earlier in the function. Rework the failure path to use staged local unwind labels and free allocated resources in reverse order before returning -ENOMEM. This matches common netdev allocation cleanup style.	2026-05-01	5.5
CVE-2026-31738	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vxlan: validate ND option lengths in vxlan_na_create vxlan_na_create() walks ND options according to option-provided lengths. A malformed option can make the parser advance beyond the computed option span or use a too-short source LLADDR option payload. Validate option lengths against the remaining NS option area before advancing, and only read source LLADDR when the option is large enough for an Ethernet address.	2026-05-01	5.5
CVE-2026-31740	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: counter: rz-mtu3-cnt: do not use struct rz_mtu3_channel's dev member The counter driver can use HW channels 1 and 2, while the PWM driver can use HW channels 0, 1, 2, 3, 4, 6, 7. The dev member is assigned both by the counter driver and the PWM driver for channels 1 and 2, to their own struct device instance, overwriting the previous value. The sub-drivers race to assign their own struct device pointer to the same struct rz_mtu3_channel's dev member. The dev member of struct rz_mtu3_channel is used by the counter sub-driver for runtime PM. Depending on the probe order of the counter and PWM sub-drivers, the dev member may point to the wrong struct device instance, causing the counter sub-driver to do runtime PM actions on the wrong device.	2026-05-01	5.5

		To fix this, use the parent pointer of the counter, which is assigned during probe to the correct struct device, not the struct device pointer inside the shared struct rz_mtu3_channel.		
CVE-2026-31741	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: counter: rz-mtu3-cnt: prevent counter from being toggled multiple times Runtime PM counter is incremented / decremented each time the sysfs enable file is written to. If user writes 0 to the sysfs enable file multiple times, runtime PM usage count underflows, generating the following message. rz-mtu3-counter rz-mtu3-counter.0: Runtime PM usage count underflow! At the same time, hardware registers end up being accessed with clocks off in rz_mtu3_terminate_counter() to disable an already disabled channel. If user writes 1 to the sysfs enable file multiple times, runtime PM usage count will be incremented each time, requiring the same number of 0 writes to get it back to 0. If user writes 0 to the sysfs enable file while PWM is in progress, PWM is stopped without counter being the owner of the underlying MTU3 channel. Check against the cached count_is_enabled value and exit if the user is trying to set the same enable value.	2026-05-01	5.5
CVE-2026-31744	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PM: EM: Fix NULL pointer dereference when perf domain ID is not found dev_energymodel_nl_get_perf_domains_doit() calls em_perf_domain_get_by_id() but does not check the return value before passing it to __em_nl_get_pd_size(). When a caller supplies a non-existent perf domain ID, em_perf_domain_get_by_id() returns NULL, and __em_nl_get_pd_size() immediately dereferences pd->cpus (struct offset 0x30), causing a NULL pointer dereference. The sister handler dev_energymodel_nl_get_perf_table_doit() already handles this correctly via __em_nl_get_pd_table_id(), which returns NULL and causes the caller to return -EINVAL. Add the same NULL check in the get-perf-domains do handler. [rjw: Subject and changelog edits]	2026-05-01	5.5
CVE-2026-31746	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: s390/zcrypt: Fix memory leak with CCA cards used as accelerator Tests showed that there is a memory leak if CCA cards are used as accelerator for clear key RSA requests (ME and CRT). With the last rework for the memory allocation the AP messages are allocated by ap_init_apmsg() but for some reason on two places (ME and CRT) the older allocation was still in place. So the first allocation simple was never freed.	2026-05-01	5.5
CVE-2026-31749	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: ni_atmio16d: Fix invalid clean-up after failed attach If the driver's COMEDI "attach" handler function (`atmio16d_attach()`) returns an error, the COMEDI core will call the driver's "detach" handler function (`atmio16d_detach()`) to clean up. This calls `reset_atmio16d()` unconditionally, but depending on where the error occurred in the attach handler, the device may not have been sufficiently initialized to call `reset_atmio16d()`. It uses `dev->iobase` as the I/O port base address and `dev->private` as the pointer to the COMEDI device's private data structure. `dev->iobase` may still be set to its initial value of 0, which would result in undesired writes to low I/O port addresses. `dev->private` may still be `NULL`, which would result in null pointer dereferences. Fix `atmio16d_detach()` by checking that `dev->private` is valid (non-null) before calling `reset_atmio16d()`. This implies that `dev->iobase` was set correctly since that is set up before `dev->private`.	2026-05-01	5.5
CVE-2026-31750	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: runflags cannot determine whether to reclaim chanlist	2026-05-01	5.5

		syzbot reported a memory leak [1], because commit 4e1da516debb ("comedi: Add reference counting for Comedi command handling") did not consider the exceptional exit case in do_cmd_ioctl() where runflags is not set. This caused chanlist not to be properly freed by do_become_nonbusy(), as it only frees chanlist when runflags is correctly set. Added a check in do_become_nonbusy() for the case where runflags is not set, to properly free the chanlist memory. [1] BUG: memory leak backtrace (crc 844a0efa): __comedi_get_user_chanlist drivers/comedi/comedi_fops.c:1815 [inline] do_cmd_ioctl.part.0+0x112/0x350 drivers/comedi/comedi_fops.c:1890 do_cmd_ioctl drivers/comedi/comedi_fops.c:1858 [inline]		
CVE-2026-31752	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bridge: br_nd_send: validate ND option lengths br_nd_send() walks ND options according to option-provided lengths. A malformed option can make the parser advance beyond the computed option span or use a too-short source LLADDR option payload. Validate option lengths against the remaining NS option area before advancing, and only read source LLADDR when the option is large enough for an Ethernet address.	2026-05-01	5.5
CVE-2026-31753	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: auxdisplay: line-display: fix NULL dereference in linedisp_release linedisp_release() currently retrieves the enclosing struct linedisp via to_linedisp(). That lookup depends on the attachment list, but the attachment may already have been removed before put_device() invokes the release callback. This can happen in linedisp_unregister(), and can also be reached from some linedisp_register() error paths. In that case, to_linedisp() returns NULL and linedisp_release() dereferences it while freeing the display resources. The struct device released here is the embedded linedisp->dev used by linedisp_register(), so retrieve the enclosing object directly with container_of() instead.	2026-05-01	5.5
CVE-2026-31754	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: cdns3: gadget: fix state inconsistency on gadget init failure When cdns3_gadget_start() fails, the DRD hardware is left in gadget mode while software state remains INACTIVE, creating hardware/software state inconsistency. When switching to host mode via sysfs: echo host > /sys/class/usb_role/13180000.usb-role-switch/role The role state is not set to CDNS_ROLE_STATE_ACTIVE due to the error, so cdns_role_stop() skips cleanup because state is still INACTIVE. This violates the DRD controller design specification (Figure22), which requires returning to idle state before switching roles. This leads to a synchronous external abort in xhci_gen_setup() when setting up the host controller: [516.440698] configfs-gadget 13180000.usb: failed to start g1: -19 [516.442035] cdns-usb3 13180000.usb: Failed to add gadget [516.443278] cdns-usb3 13180000.usb: set role 2 has failed ... [1301.375722] xhci-hcd xhci-hcd.1.auto: xHCI Host Controller [1301.377716] Internal error: synchronous external abort: 96000010 [#1] PREEMPT SMP [1301.382485] pc : xhci_gen_setup+0xa4/0x408 [1301.393391] backtrace: ... xhci_gen_setup+0xa4/0x408 <-- CRASH xhci_plat_setup+0x44/0x58 usb_add_hcd+0x284/0x678 ... cdns_role_set+0x9c/0xbc <-- Role switch Fix by calling cdns_drd_gadget_off() in the error path to properly clean up the DRD gadget state.	2026-05-01	5.5
CVE-2026-31755	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-01	5.5

		usb: cdns3: gadget: fix NULL pointer dereference in ep_queue When the gadget endpoint is disabled or not yet configured, the ep->desc pointer can be NULL. This leads to a NULL pointer dereference when __cdns3_gadget_ep_queue() is called, causing a kernel crash. Add a check to return -ESHUTDOWN if ep->desc is NULL, which is the standard return code for unconfigured endpoints. This prevents potential crashes when ep_queue is called on endpoints that are not ready.		
CVE-2026-31756	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: dwc2: gadget: Fix spin_lock/unlock mismatch in dwc2_hsotg_udc_stop() dwc2_gadget_exit_clock_gating() internally calls call_gadget() macro, which expects hsotg->lock to be held since it does spin_unlock/spin_lock around the gadget driver callback invocation. However, dwc2_hsotg_udc_stop() calls dwc2_gadget_exit_clock_gating() without holding the lock. This leads to: - spin_unlock on a lock that is not held (undefined behavior) - The lock remaining held after dwc2_gadget_exit_clock_gating() returns, causing a deadlock when spin_lock_irqsave() is called later in the same function. Fix this by acquiring hsotg->lock before calling dwc2_gadget_exit_clock_gating() and releasing it afterwards, which satisfies the locking requirement of the call_gadget() macro.	2026-05-01	5.5
CVE-2026-31757	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: misc: usbio: Fix URB memory leak on submit failure When usb_submit_urb() fails in usbio_probe(), the previously allocated URB is never freed, causing a memory leak. Fix this by jumping to err_free_urb label to properly release the URB on the error path.	2026-05-01	5.5
CVE-2026-31760	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpib: lpvo_usb: fix memory leak on disconnect The driver iterates over the registered USB interfaces during GPIB attach and takes a reference to their USB devices until a match is found. These references are never released which leads to a memory leak when devices are disconnected. Fix the leak by dropping the unnecessary references.	2026-05-01	5.5
CVE-2026-31762	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: gyro: mpu3050: Fix irq resource leak The interrupt handler is setup but only a few lines down if iio_trigger_register() fails the function returns without properly releasing the handler. Add cleanup goto to resolve resource leak. Detected by Smatch: drivers/iio/gyro/mpu3050-core.c:1128 mpu3050_trigger_probe() warn: 'irq' from request_threaded_irq() not released on lines: 1124.	2026-05-01	5.5
CVE-2026-31763	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: gyro: mpu3050: Fix incorrect free_irq() variable The handler for the IRQ part of this driver is mpu3050->trig but, in the teardown free_irq() is called with handler mpu3050. Use correct IRQ handler when calling free_irq().	2026-05-01	5.5
CVE-2026-31775	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: ctxfi: Don't enumerate SPDIF1 at DAIO initialization The recent refactoring of xfi driver changed the assignment of atc->daios[] at atc_get_resources(); now it loops over all enum DAIOTYP entries while it looped formerly only a part of them. The problem is that the last entry, SPDIF1, is a special type that is used only for hw20k1 CTSB073X model (as a replacement of SPDIFIO), and there is no corresponding definition for hw20k2. Due to the lack of the info, it caused a kernel crash on hw20k2, which was already	2026-05-01	5.5

		worked around by the commit b045ab3dff97 ("ALSA: ctxfi: Fix missing SPDIFI1 index handling"). This patch addresses the root cause of the regression above properly, simply by skipping the incorrect SPDIF1 type in the parser loop. For making the change clearer, the code is slightly arranged, too.		
CVE-2026-31777	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: ctxfi: Check the error for index mapping The ctxfi driver blindly assumed a proper value returned from daio_device_index(), but it's not always true. Add a proper error check to deal with the error from the function.	2026-05-01	5.5
CVE-2026-43008	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio: qixis-fpga: Fix error handling for devm_regmap_init_mmio() devm_regmap_init_mmio() returns an ERR_PTR() on failure, not NULL. The original code checked for NULL which would never trigger on error, potentially leading to an invalid pointer dereference. Use IS_ERR() and PTR_ERR() to properly handle the error case.	2026-05-01	5.5
CVE-2026-43010	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject sleepable kprobe_multi programs at attach time kprobe.multi programs run in atomic/RCU context and cannot sleep. However, bpf_kprobe_multi_link_attach() did not validate whether the program being attached had the sleepable flag set, allowing sleepable helpers such as bpf_copy_from_user() to be invoked from a non-sleepable context. This causes a "sleeping function called from invalid context" splat: BUG: sleeping function called from invalid context at ./include/linux/uaccess.h:169 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1787, name: sudo preempt_count: 1, expected: 0 RCU nest depth: 2, expected: 0 Fix this by rejecting sleepable programs early in bpf_kprobe_multi_link_attach(), before any further processing.	2026-05-01	5.5
CVE-2026-43012	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix switchdev mode rollback in case of failure If for some internal reason switchdev mode fails, we rollback to legacy mode, before this patch, rollback will unregister the uplink netdev and leave it unregistered causing the below kernel bug. To fix this, we need to avoid netdev unregister by setting the proper rollback flag 'MLX5_PRIV_FLAGS_SWITCH_LEGACY' to indicate legacy mode. devlink (431) used greatest stack depth: 11048 bytes left mlx5_core 0000:00:03.0: E-Switch: Disable: mode(LEGACY), nvfs(0), \ necvfs(0), active vports(0) mlx5_core 0000:00:03.0: E-Switch: Supported tc chains and prios offload mlx5_core 0000:00:03.0: Loading uplink representor for vport 65535 mlx5_core 0000:00:03.0: mlx5_cmd_out_err:816:(pid 456): \ QUERY_HCA_CAP(0x100) op_mod(0x0) failed, \ status bad parameter(0x3), syndrome (0x3a3846), err(-22) mlx5_core 0000:00:03.0 enp0s3np0 (unregistered): Unloading uplink \ representor for vport 65535 ------------[cut here]------------ kernel BUG at net/core/dev.c:12070! Oops: invalid opcode: 0000 [#1] SMP NOPTI CPU: 2 UID: 0 PID: 456 Comm: devlink Not tainted 6.16.0-rc3+ \ #9 PREEMPT(voluntary) RIP: 0010:unregister_netdevice_many_notify+0x123/0xae0 ... Call Trace: [90.923094] unregister_netdevice_queue+0xad/0xf0 [90.923323] unregister_netdev+0x1c/0x40 [90.923522] mlx5e_vport_rep_unload+0x61/0xc6 [90.923736] esw_offloads_enable+0x8e6/0x920 [90.923947] mlx5_eswitch_enable_locked+0x349/0x430 [90.924182] ? is_mp_supported+0x57/0xb0 [90.924376] mlx5_devlink_eswitch_mode_set+0x167/0x350 [90.924628] devlink_nl_eswitch_set_doit+0x6f/0xf0 [90.924862] genl_family_rcv_msg_doit+0xe8/0x140 [90.925088] genl_rcv_msg+0x18b/0x290	2026-05-01	5.5

		[90.925269] ? __pfx_devlink_nl_pre_doit+0x10/0x10 [90.925506] ? __pfx_devlink_nl_eswitch_set_doit+0x10/0x10 [90.925766] ? __pfx_devlink_nl_post_doit+0x10/0x10 [90.926001] ? __pfx_genl_rcv_msg+0x10/0x10 [90.926206] netlink_rcv_skb+0x52/0x100 [90.926393] genl_rcv+0x28/0x40 [90.926557] netlink_unicast+0x27d/0x3d0 [90.926749] netlink_sendmsg+0x1f7/0x430 [90.926942] __sys_sendto+0x213/0x220 [90.927127] ? __sys_recvmsg+0x6a/0xd0 [90.927312] __x64_sys_sendto+0x24/0x30 [90.927504] do_syscall_64+0x50/0x1c0 [90.927687] entry_SYSCALL_64_after_hwframe+0x76/0x7e [90.927929] RIP: 0033:0x7f7d0363e047		
CVE-2026-43013	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: lag: Check for LAG device before creating debugfs __mlx5_lag_dev_add_mdev() may return 0 (success) even when an error occurs that is handled gracefully. Consequently, the initialization flow proceeds to call mlx5_ldev_add_debugfs() even when there is no valid LAG context. mlx5_ldev_add_debugfs() blindly created the debugfs directory and attributes. This exposed interfaces (like the members file) that rely on a valid ldev pointer, leading to potential NULL pointer dereferences if accessed when ldev is NULL. Add a check to verify that mlx5_lag_dev(dev) returns a valid pointer before attempting to create the debugfs entries.	2026-05-01	5.5
CVE-2026-43014	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: macb: properly unregister fixed rate clocks The additional resources allocated with clk_register_fixed_rate() need to be released with clk_unregister_fixed_rate(), otherwise they are lost.	2026-05-01	5.5
CVE-2026-43017	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: validate mesh send advertising payload length mesh_send() currently bounds MGMT_OP_MESH_SEND by total command length, but it never verifies that the bytes supplied for the flexible adv_data[] array actually match the embedded adv_data_len field. MGMT_MESH_SEND_SIZE only covers the fixed header, so a truncated command can still pass the existing 20..50 byte range check and later drive the async mesh send path past the end of the queued command buffer. Keep rejecting zero-length and oversized advertising payloads, but validate adv_data_len explicitly and require the command length to exactly match the flexible array size before queueing the request.	2026-05-01	5.5
CVE-2026-43021	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: fix leaks when hci_cmd_sync_queue_once fails When hci_cmd_sync_queue_once() returns with error, the destroy callback will not be called. Fix leaking references / memory on these failures.	2026-05-01	5.5
CVE-2026-43022	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: hci_cmd_sync_queue_once() return -EEXIST if exists hci_cmd_sync_queue_once() needs to indicate whether a queue item was added, so caller can know if callbacks are called, so it can avoid leaking resources. Change the function to return -EEXIST if queue item already exists. Modify all callsites to handle that.	2026-05-01	5.5
CVE-2026-43024	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: reject immediate NF_QUEUE verdict nft_queue is always used from userspace nftables to deliver the NF_QUEUE verdict. Immediately emitting an NF_QUEUE verdict is never used by the userspace nft tools, so reject immediate NF_QUEUE verdicts. The arp family does not provide queue support, but such an immediate	2026-05-01	5.5

		verdict is still reachable. Globally reject NF_QUEUE immediate verdicts to address this issue.		
CVE-2026-43026	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: zero expect NAT fields when CTA_EXPECT_NAT absent ctnetlink_alloc_expect() allocates expectations from a non-zeroing slab cache via nf_ct_expect_alloc(). When CTA_EXPECT_NAT is not present in the netlink message, saved_addr and saved_proto are never initialized. Stale data from a previous slab occupant can then be dumped to userspace by ctnetlink_exp_dump_expect(), which checks these fields to decide whether to emit CTA_EXPECT_NAT. The safe sibling nf_ct_expect_init(), used by the packet path, explicitly zeroes these fields. Zero saved_addr, saved_proto and dir in the else branch, guarded by IS_ENABLED(CONFIG_NF_NAT) since these fields only exist when NAT is enabled. Confirmed by priming the expect slab with NAT-bearing expectations, freeing them, creating a new expectation without CTA_EXPECT_NAT, and observing that the ctnetlink dump emits a spurious CTA_EXPECT_NAT containing stale data from the prior allocation.	2026-05-01	5.5
CVE-2026-43032	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: NFC: pn533: bound the UART receive buffer pn532_receive_buf() appends every incoming byte to dev->recv_skb and only resets the buffer after pn532_uart_rx_is_frame() recognizes a complete frame. A continuous stream of bytes without a valid PN532 frame header therefore keeps growing the skb until skb_put_u8() hits the tail limit. Drop the accumulated partial frame once the fixed receive buffer is full so malformed UART traffic cannot grow the skb past PN532_UART_SKB_BUFF_LEN.	2026-05-01	5.5
CVE-2026-43034	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: set backing store type from query type bnxt_hwrm_func_backing_store_qcaps_v2() stores resp->type from the firmware response in ctxm->type and later uses that value to index fixed backing-store metadata arrays such as ctx_arr[] and bnxt_bstore_to_trace[]. ctxm->type is fixed by the current backing-store query type and matches the array index of ctx->ctx_arr. Set ctxm->type from the current loop variable instead of depending on resp->type. Also update the loop to advance type from next_valid_type in the for statement, which keeps the control flow simpler for non-valid and unchanged entries.	2026-05-01	5.5
CVE-2026-43035	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: sched: cls_api: fix tc_chain_fill_node to initialize tcm_info to zero to prevent an info-leak When building netlink messages, tc_chain_fill_node() never initializes the tcm_info field of struct tcmsg. Since the allocation is not zeroed, kernel heap memory is leaked to userspace through this 4-byte field. The fix simply zeroes tcm_info alongside the other fields that are already initialized.	2026-05-01	5.5
CVE-2026-43036	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: use skb_header_pointer() for TCPv4 GSO frag_off check Syzbot reported a KMSAN uninit-value warning in gso_features_check() called from netif_skb_features() [1]. gso_features_check() reads iph->frag_off to decide whether to clear mangleid_features. Accessing the IPv4 header via ip_hdr()/inner_ip_hdr() can rely on skb header offsets that are not always safe for direct dereference on packets injected from PF_PACKET paths. Use skb_header_pointer() for the TCPv4 frag_off check so the header read is robust whether data is already linear or needs copying. [1] https://syzkaller.appspot.com/bug?extid=1543a7d954d9c6d00407	2026-05-01	5.5

CVE-2026-43041	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: replace qrtr_tx_flow radix_tree with xarray to fix memory leak __radix_tree_create() allocates and links intermediate nodes into the tree one by one. If a subsequent allocation fails, the already-linked nodes remain in the tree with no corresponding leaf entry. These orphaned internal nodes are never reclaimed because radix_tree_for_each_slot() only visits slots containing leaf values. The radix_tree API is deprecated in favor of xarray. As suggested by Matthew Wilcox, migrate qrtr_tx_flow from radix_tree to xarray instead of fixing the radix_tree itself [1]. xarray properly handles cleanup of internal nodes — xa_destroy() frees all internal xarray nodes when the qrtr_node is released, preventing the leak. [1] https://lore.kernel.org/all/20260225071623.41275-1-jiayuan.chen@linux.dev/T/	2026-05-01	5.5
CVE-2026-43043	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: af-alg - fix NULL pointer dereference in scatterwalk The AF_ALG interface fails to unmark the end of a Scatter/Gather List (SGL) when chaining a new af_alg_tsgl structure. If a sendmsg() fills an SGL exactly to MAX_SGL_ENTS, the last entry is marked as the end. A subsequent sendmsg() allocates a new SGL and chains it, but fails to clear the end marker on the previous SGL's last data entry. This causes the crypto scatterwalk to hit a premature end, returning NULL on sg_next() and leading to a kernel panic during dereference. Fix this by explicitly unmarking the end of the previous SGL when performing sg_chain() in af_alg_alloc_tsgl().	2026-05-01	5.5
CVE-2026-43045	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mshv: Fix error handling in mshv_region_pin The current error handling has two issues: First, pin_user_pages_fast() can return a short pin count (less than requested but greater than zero) when it cannot pin all requested pages. This is treated as success, leading to partially pinned regions being used, which causes memory corruption. Second, when an error occurs mid-loop, already pinned pages from the current batch are not properly accounted for before calling mshv_region_invalidate_pages(), causing a page reference leak. Treat short pins as errors and fix partial batch accounting before cleanup.	2026-05-01	5.5
CVE-2026-43046	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: reject root items with drop_progress and zero drop_level [BUG] When recovering relocation at mount time, merge_reloc_root() and btrfs_drop_snapshot() both use BUG_ON(level == 0) to guard against an impossible state: a non-zero drop_progress combined with a zero drop_level in a root_item, which can be triggered: ------------[cut here]------------ kernel BUG at fs/btrfs/relocation.c:1545! Oops: invalid opcode: 0000 [#1] SMP KASAN NOPTI CPU: 1 UID: 0 PID: 283 ... Tainted: 6.18.0+ #16 PREEMPT(voluntary) Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU Ubuntu 24.04 PC v2, BIOS 1.16.3-debian-1.16.3-2 RIP: 0010:merge_reloc_root+0x1266/0x1650 fs/btrfs/relocation.c:1545 Code: ffff0000 00004589 d7e9acfa ffffe8a1 79bafebe 02000000 Call Trace: merge_reloc_roots+0x295/0x890 fs/btrfs/relocation.c:1861 btrfs_recover_relocation+0xd6e/0x11d0 fs/btrfs/relocation.c:4195 btrfs_start_pre_rw_mount+0xa4d/0x1810 fs/btrfs/disk-io.c:3130 open_ctree+0x5824/0x5fe0 fs/btrfs/disk-io.c:3640 btrfs_fill_super fs/btrfs/super.c:987 [inline] btrfs_get_tree_super fs/btrfs/super.c:1951 [inline] btrfs_get_tree_subvol fs/btrfs/super.c:2094 [inline] btrfs_get_tree+0x111c/0x2190 fs/btrfs/super.c:2128 vfs_get_tree+0x9a/0x370 fs/super.c:1758 fc_mount fs/namespace.c:1199 [inline] do_new_mount_fc fs/namespace.c:3642 [inline] do_new_mount fs/namespace.c:3718 [inline]	2026-05-01	5.5

		path_mount+0x5b8/0x1ea0 fs/namespace.c:4028 do_mount fs/namespace.c:4041 [inline] __do_sys_mount fs/namespace.c:4229 [inline] __se_sys_mount fs/namespace.c:4206 [inline] __x64_sys_mount+0x282/0x320 fs/namespace.c:4206 ... RIP: 0033:0x7f969c9a8fde Code: 0f1f4000 48c7c2b0 fffffff7 d8648902 b8ffffff ffc3660f ---[end trace 0000000000000000]--- The bug is reproducible on 7.0.0-rc2-next-20260310 with our dynamic metadata fuzzing tool that corrupts btrfs metadata at runtime. [CAUSE] A non-zero drop_progress.objectid means an interrupted btrfs_drop_snapshot() left a resume point on disk, and in that case drop_level must be greater than 0 because the checkpoint is only saved at internal node levels. Although this invariant is enforced when the kernel writes the root item, it is not validated when the root item is read back from disk. That allows on-disk corruption to provide an invalid state with drop_progress.objectid != 0 and drop_level == 0. When relocation recovery later processes such a root item, merge_reloc_root() reads drop_level and hits BUG_ON(level == 0). The same invalid metadata can also trigger the corresponding BUG_ON() in btrfs_drop_snapshot(). [FIX] Fix this by validating the root_item invariant in tree-checker when reading root items from disk: if drop_progress.objectid is non-zero, drop_level must also be non-zero. Reject such malformed metadata with -EUCLEAN before it reaches merge_reloc_root() or btrfs_drop_snapshot() and triggers the BUG_ON. After the fix, the same corruption is correctly rejected by tree-checker and the BUG_ON is no longer triggered.		
[CVE-2026-43054](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: target: tcm_loop: Drain commands in target_reset handler tcm_loop_target_reset() violates the SCSI EH contract: it returns SUCCESS without draining any in-flight commands. The SCSI EH documentation (scsi_eh.rst) requires that when a reset handler returns SUCCESS the driver has made lower layers "forget about timed out scmds" and is ready for new commands. Every other SCSI LLD (virtio_scsi, mpt3sas, ipr, scsi_debug, mpi3mr) enforces this by draining or completing outstanding commands before returning SUCCESS. Because tcm_loop_target_reset() doesn't drain, the SCSI EH reuses in-flight scsi_cmnd structures for recovery commands (e.g. TUR) while the target core still has async completion work queued for the old se_cmd. The memset in queuecommand zeroes se_lun and lun_ref_active, causing transport_lun_remove_cmd() to skip its percpu_ref_put(). The leaked LUN reference prevents transport_clear_lun_ref() from completing, hanging configfs LUN unlink forever in D-state: INFO: task rm:264 blocked for more than 122 seconds. rm D 0 264 258 0x00004000 Call Trace: __schedule+0x3d0/0x8e0 schedule+0x36/0xf0 transport_clear_lun_ref+0x78/0x90 [target_core_mod] core_tpg_remove_lun+0x28/0xb0 [target_core_mod] target_fabric_port_unlink+0x50/0x60 [target_core_mod] configfs_unlink+0x156/0x1f0 [configfs] vfs_unlink+0x109/0x290 do_unlinkat+0x1d5/0x2d0 Fix this by making tcm_loop_target_reset() actually drain commands: 1. Issue TMR_LUN_RESET via tcm_loop_issue_tmr() to drain all commands that the target core knows about (those not yet CMD_T_COMPLETE). 2. Use blk_mq_tagset_busy_iter() to iterate all started requests and flush_work() on each se_cmd — this drains any deferred completion work for commands that already had CMD_T_COMPLETE set before the TMR (which the TMR skips via __target_check_io_state()). This is the same pattern	2026-05-01	5.5

		used by mpi3mr, scsi_debug, and libsas to drain outstanding commands during reset.		
CVE-2026-7668	mikrotik - RouterOS	A vulnerability was identified in MikroTik RouterOS 6.49.8. This vulnerability affects the function ASN1_STRING_data in the library nova/lib/www/scep.p of the component SCEP Endpoint. The manipulation of the argument transactionID/messageType leads to out-of-bounds read. The attack may be initiated remotely. The exploit is publicly available and might be used. The vendor was contacted early about this disclosure but did not respond in any way.	2026-05-02	5.5
CVE-2026-7500	redhat - build_of_keycloak	When Keycloak is started with `--features-disabled=account,account-api`, the Account REST API is only partially disabled. Five endpoints under the versioned path `/account/v1alpha1` remain fully functional — including both read and write operations — because they lack the `checkAccountApiEnabled()` gate that correctly blocks four other endpoints in the same REST service class. The user needs to have permissions to use the API.	2026-04-30	5.4
CVE-2026-32655	dell - alienware_command_center	Dell Alienware Command Center (AWCC), versions prior to 6.13.8.0, contain a Least Privilege Violation vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of Privileges.	2026-04-27	5.3
CVE-2026-41606	apache - thrift	Uncontrolled Recursion vulnerability in Apache Thrift. This issue affects Apache Thrift: before 0.23.0. Users are recommended to upgrade to version 0.23.0, which fixes the issue.	2026-04-28	5.3
CVE-2026-22745	vmware - multiple products	Spring MVC and WebFlux applications are vulnerable to Denial of Service attacks when resolving static resources. More precisely, an application can be vulnerable when all the following are true: * the application is using Spring MVC or Spring WebFlux * the application is serving static resources from the file system * the application is running on a Windows platform When all the conditions above are met, the attacker can send malicious requests that are slow to resolve and that can keep HTTP connections in use. This can cause a Denial of Service on the application.	2026-04-29	5.3
CVE-2026-6915	mongodb - multiple products	An authorization flaw in the user management command could allow an authenticated user to make limited changes to authentication-related data associated with another user account. This could affect how authentication is performed for the impacted account.	2026-04-29	5.3
CVE-2025-14688	ibm - multiple products	IBM Db2 11.5.0 through 11.5.9, and 12.1.0 through 12.1.3 for Linux, UNIX and Windows (includes Db2 Connect Server) could allow an authenticated user to cause a denial of service due to improper neutralization of special elements in data query logic when certain configurations exist.	2026-04-30	5.3
CVE-2025-36180	ibm - watsonx.data	IBM watsonx.data 2.2 through 2.3 IBM Lakehouse does not properly restrict communication between pods which could allow an attacker to transfer data between pods without restrictions.	2026-04-30	5.3
CVE-2026-0206	sonicwall - sonicos	A post-authentication Stack-based Buffer Overflow vulnerabilities in SonicOS allows a remote attacker to crash a firewall.	2026-04-29	4.9
CVE-2026-40557	apache - storm_prometheus_reporter	Improper Certificate Validation via Global SSL Context Downgrade in Apache Storm Prometheus Reporter Versions Affected: from 2.6.3 to 2.8.6 Description: In production deployments where an administrator enables storm.daemon.metrics.reporter.plugin.prometheus.skip_tls_validation (by default it is disabled) intending to affect only the Prometheus reporter, the undocumented global side effect creates an attack surface across every TLS-protected communication channel in the Storm daemon. The PrometheusPreparableReporter class implements an INSECURE_TRUST_MANAGER that accepts all SSL certificates without validation, with empty checkClientTrusted and checkServerTrusted methods. Most critically, when the storm.daemon.metrics.reporter.plugin.prometheus.skip_tls_validation configuration option is enabled (default = disabled) for HTTPS Prometheus PushGateway connections, the INSECURE_CONNECTION_FACTORY calls SSLContext.setDefault(sslContext), which globally replaces the JVM's default SSL context rather than applying the insecure context only to the Prometheus connection. This payload flows through storm.yaml configuration → PrometheusPreparableReporter.prepare() → INSECURE_CONNECTION_FACTORY → SSLContext.setDefault(), resulting in a JVM-wide TLS security downgrade. All subsequent HTTPS connections in the process - including ZooKeeper, Thrift, Netty, and UI connections - silently trust all certificates, including self-signed, expired, and attacker-generated ones, enabling man-in-the-middle interception of cluster state, topology submissions, tuple data, and administrative credentials. Mitigation: 2.x users should upgrade to 2.8.7 if the Prometheus Metrics Reporter is used. Prometheus Metrics Reporter Users who cannot upgrade immediately should remove the	2026-04-27	4.8

		storm.daemon.metrics.reporter.plugin.prometheus.skip_tls_validation: true setting from their storm.yaml configuration and instead configure a proper truststore containing the PushGateway's certificate.		
CVE-2026-40975	vmware - multiple products	Values produced by ${random.value} are not suitable for use as secrets. ${random.uuid} is not affected. ${random.int} and ${random.long} should never be used for secrets as they are numeric values with a predictable range. Affected: Spring Boot 4.0.0–4.0.5 (fix 4.0.6), 3.5.0–3.5.13 (fix 3.5.14), 3.4.0–3.4.15 (fix 3.4.16), 3.3.0–3.3.18 (fix 3.3.19), 2.7.0–2.7.32 (fix 2.7.33); random value property source / weak PRNG for secrets. Versions that are no longer supported are also affected per vendor advisory.	2026-04-28	4.8
CVE-2026-1858	gnu - wget2	wget2 accepts a server certificate with incorrect Key Usage (KU) or Extended Key Usage (EKU). If the attackers compromise a certificate (with the associated private key) issued for a different purpose, they may be able to reuse it for TLS server authentication.	2026-04-29	4.8
CVE-2026-40687	exim - exim	In Exim before 4.99.2, when the SPA authentication driver is used with an adversarial SPA resource, there can be an out-of-bounds write that crashes the connection instance, or erroneous data processing that divulges data from uninitialized heap memory.	2026-04-30	4.8
CVE-2026-40977	vmware - multiple products	When an application is configured to use `ApplicationPidFileWriter`, a local attacker with write access to the PID file's location can corrupt one file on the host each time the application is started. Affected: Spring Boot 4.0.0–4.0.5 (fix 4.0.6), 3.5.0–3.5.13 (fix 3.5.14), 3.4.0–3.4.15 (fix 3.4.16), 3.3.0–3.3.18 (fix 3.3.19), 2.7.0–2.7.32 (fix 2.7.33); PID file / symlink behavior (`ApplicationPidFileWriter`). Versions that are no longer supported are also affected per vendor advisory.	2026-04-28	4.7
CVE-2026-31728	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: u_ether: Fix race between gether_disconnect and eth_stop A race condition between gether_disconnect() and eth_stop() leads to a NULL pointer dereference. Specifically, if eth_stop() is triggered concurrently while gether_disconnect() is tearing down the endpoints, eth_stop() attempts to access the cleared endpoint descriptor, causing the following NPE: Unable to handle kernel NULL pointer dereference Call trace: __dwc3_gadget_ep_enable+0x60/0x788 dwc3_gadget_ep_enable+0x70/0xe4 usb_ep_enable+0x60/0x15c eth_stop+0xb8/0x108 Because eth_stop() crashes while holding the dev->lock, the thread running gether_disconnect() fails to acquire the same lock and spins forever, resulting in a hardlockup: Core - Debugging Information for Hardlockup core(7) Call trace: queued_spin_lock_slowpath+0x94/0x488 _raw_spin_lock+0x64/0x6c gether_disconnect+0x19c/0x1e8 ncm_set_alt+0x68/0x1a0 composite_setup+0x6a0/0xc50 The root cause is that the clearing of dev->port_usb in gether_disconnect() is delayed until the end of the function. Move the clearing of dev->port_usb to the very beginning of gether_disconnect() while holding dev->lock. This cuts off the link immediately, ensuring eth_stop() will see dev->port_usb as NULL and safely bail out.	2026-05-01	4.7
CVE-2026-31751	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: dt2815: add hardware detection to prevent crash The dt2815 driver crashes when attached to I/O ports without actual hardware present. This occurs because syzkaller or users can attach the driver to arbitrary I/O addresses via COMEDI_DEVCONFIG ioctl. When no hardware exists at the specified port, inb() operations return 0xff (floating bus), but outb() operations can trigger page faults due to undefined behavior, especially under race conditions: BUG: unable to handle page fault for address: 000000007fffff90 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page RIP: 0010:dt2815_attach+0x6e0/0x1110 Add hardware detection by reading the status register before attempting any write operations. If the read returns 0xff, assume no hardware is present and fail the attach with -ENODEV. This prevents crashes from outb() operations on non-existent hardware.	2026-05-01	4.7

CVE-2026-43053	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: close crash window in attr dabtree inactivation When inactivating an inode with node-format extended attributes, xfs_attr3_node_inactive() invalidates all child leaf/node blocks via xfs_trans_binval(), but intentionally does not remove the corresponding entries from their parent node blocks. The implicit assumption is that xfs_attr_inactive() will truncate the entire attr fork to zero extents afterwards, so log recovery will never reach the root node and follow those stale pointers. However, if a log shutdown occurs after the leaf/node block cancellations commit but before the attr bmap truncation commits, this assumption breaks. Recovery replays the attr bmap intact (the inode still has attr fork extents), but suppresses replay of all cancelled leaf/node blocks, maybe leaving them as stale data on disk. On the next mount, xlog_recover_process_iunlinks() retries inactivation and attempts to read the root node via the attr bmap. If the root node was not replayed, reading the unreplayed root block triggers a metadata verification failure immediately; if it was replayed, following its child pointers to unreplayed child blocks triggers the same failure: XFS (pmem0): Metadata corruption detected at xfs_da3_node_read_verify+0x53/0x220, xfs_da3_node block 0x78 XFS (pmem0): Unmount and run xfs_repair XFS (pmem0): First 128 bytes of corrupted metadata buffer: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... 00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .... XFS (pmem0): metadata I/O error in "xfs_da_read_buf+0x104/0x190" at daddr 0x78 len 8 error 117 Fix this in two places: In xfs_attr3_node_inactive(), after calling xfs_trans_binval() on a child block, immediately remove the entry that references it from the parent node in the same transaction. This eliminates the window where the parent holds a pointer to a cancelled block. Once all children are removed, the now-empty root node is converted to a leaf block within the same transaction. This node-to-leaf conversion is necessary for crash safety. If the system shutdown after the empty node is written to the log but before the second-phase bmap truncation commits, log recovery will attempt to verify the root block on disk. xfs_da3_node_verify() does not permit a node block with count == 0; such a block will fail verification and trigger a metadata corruption shutdown. on the other hand, leaf blocks are allowed to have this transient state. In xfs_attr_inactive(), split the attr fork truncation into two explicit phases. First, truncate all extents beyond the root block (the child extents whose parent references have already been removed above). Second, invalidate the root block and truncate the attr bmap to zero in a single transaction. The two operations in the second phase must be atomic: as long as the attr bmap has any non-zero length, recovery can follow it to the root block, so the root block invalidation must commit together with the bmap-to-zero truncation.	2026-05-01	4.7
CVE-2026-35233	oracle - multiple products	An unprivileged attacker can craft a user-space process with a malicious ELF binary containing an out-of-range sh_link field. When root-level dtrace attaches to -- or instruments -- that process (via dtrace -p , pid probes, or USDT), the ELF parser reads heap memory beyond the allocated section cache array without any bounds check. This results in an uninitialized/out-of-bounds heap read that can cause a NULL pointer dereference crash of the dtrace process (DoS), or -- depending on heap layout -- a read-then-use of a garbage pointer controlled by adjacent allocations, providing a foothold toward further exploitation in a privileged context.	2026-05-01	4.4
CVE-2026-7309	redhat - openshift_container_platform	A flaw was found in the OpenShift Container Platform build system. A user with the `edit` ClusterRole can inject arbitrary environment variables, such as `LD_PRELOAD` or `http_proxy`, into `docker-build` containers through the `buildconfigs/instantiate` API. This incomplete fix for a previous vulnerability allows for information disclosure, specifically impacting the confidentiality of build traffic.	2026-04-28	4.3
CVE-2026-7340	google - chrome	Integer overflow in ANGLE in Google Chrome on Windows prior to 147.0.7727.138 allowed a remote attacker to perform an out of bounds memory read via a crafted HTML page. (Chromium security severity: Medium)	2026-04-28	4.3
CVE-2026-23773	dell - multiple products	Dell Disk Library for Mainframe, version(s) DLm 8700/2700 contain(s) 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.	2026-04-29	4.3

CVE-2026-42519	jenkins - script_security	A missing permission check in Jenkins Script Security Plugin 1399.ve6a_66547f6e1 and earlier allows attackers with Overall/Read permission to enumerate pending and approved Script Security classpaths.	2026-04-29	4.3
CVE-2026-42522	jenkins - github_branch_source	A missing permission check in Jenkins GitHub Branch Source Plugin 1967.vdea_d580c1a_b_a_ and earlier allows attackers with Overall/Read permission to connect to an attacker-specified URL with attacker-specified GitHub App credentials.	2026-04-29	4.3
CVE-2026-42525	jenkins - azure_ad	Jenkins Microsoft Entra ID (previously Azure AD) Plugin 666.v6060de32f87d and earlier does not restrict the redirect URL after login, allowing attackers to perform phishing attacks.	2026-04-29	4.3
CVE-2026-40968	vmware - spring_grpc	When an authenticated user is denied access to a gRPC method, their authenticated identity remains bound to the gRPC worker thread and can be inherited by a subsequent unauthenticated request on the same thread. This may allow the subsequent user to gain escalated permissions. Affected versions: Spring gRPC: 1.0.0 - 1.0.2 (fixed in 1.0.3). Older, unsupported versions are also affected.	2026-04-28	4.2
CVE-2026-40969	vmware - spring_grpc	The raw message of every server-side AuthenticationException is returned to the unauthenticated remote caller in the gRPC status description. This allows an attacker to obtain information about the authentication failure, which may be useful for further attacks. Affected versions: Spring gRPC: 1.0.0 - 1.0.2 (fixed in 1.0.3). Older, unsupported versions are also affected.	2026-04-28	3.7
CVE-2026-40686	exim - exim	In Exim before 4.99.2, when utf8 operators are enabled, there is an out-of-bounds read if large UTF-8 trailing characters are present (malformed UTF-8 header data). Information might be divulged within an error message produced during handling of an unrelated e-mail message.	2026-04-30	3.7
CVE-2026-21996	oracle - multiple products	An unprivileged attacker can reliably trigger a crash of the dtrace process with a malicious ELF binary due to an integer Divide-by-Zero in Pbuild_file_symtab()	2026-05-01	3.3
CVE-2026-7351	google - chrome	Race in MHTML in Google Chrome prior to 147.0.7727.138 allowed an attacker who convinced a user to install a malicious extension to leak cross-origin data via a crafted Chrome Extension. (Chromium security severity: High)	2026-04-28	3.1
CVE-2026-7360	google - chrome	Insufficient validation of untrusted input. in Compositing in Google Chrome prior to 147.0.7727.138 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	2026-04-28	3.1
CVE-2026-22741	vmware - multiple products	Spring MVC and WebFlux applications are vulnerable to cache poisoning when resolving static resources. More precisely, an application can be vulnerable when all the following are true: * the application is using Spring MVC or Spring WebFlux * the application is configuring the resource chain support https://docs.spring.io/spring-framework/reference/web/webmvc/mvc-config/static-resources.html#page-title with caching enabled * the application adds support for encoded resources resolution * the resource cache must be empty when the attacker has access to the application When all the conditions above are met, the attacker can send malicious requests and poison the resource cache with resources using the wrong encoding. This can cause a denial of service by breaking the front-end application for clients.	2026-04-29	3.1
CVE-2026-7610	trendnet - tew-821dap_firmware	A vulnerability has been found in TRENDnet TEW-821DAP 1.12B01. This affects an unknown function of the file /www/cgi/ssi of the component Firmware Update. Such manipulation leads to cleartext transmission of sensitive information. The attack can be executed remotely. This attack is characterized by high complexity. The exploitability is reported as difficult. The exploit has been disclosed to the public and may be used. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	2.9
CVE-2026-7609	trendnet - tew-821dap_firmware	A flaw has been found in TRENDnet TEW-821DAP up to 1.12B01. The impacted element is the function tools_diagnostic of the file /tmp/diagnostic of the component Firmware Udpate. This manipulation causes os command injection. Remote exploitation of the attack is possible. The exploit has been published and may be used. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	2.1
CVE-2026-7608	trendnet - tew-821dap_firmware	A vulnerability was detected in TRENDnet TEW-821DAP up to 1.12B01. The affected element is the function tools_diagnostic. The manipulation results in os command injection. The exploit is now public and may be used. The vendor explains: "That firmware version will only work on our hardware version v1.xR. We have already EOL that product 8 years ago and are no longer selling". This vulnerability only affects products that are no longer supported by the maintainer.	2026-05-02	2

Where NCA provides the vulnerability information as published by NIST's NVD. In addition, it is the entity's or individual's responsibility to ensure the implementation of appropriate recommendations.

وحيث تقدم الهيئة تفاصيل الثغرات كما تم نشرها من قبل NIST's NVD. وإذ تبقى مسؤولية الجهة أو الشخص قائمة للتأكد من تطبيق التوصيات المناسبة.