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 3 مايو إلى 9 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 3rd of May to 9th 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-2026-42826	microsoft - azure_devops	Exposure of sensitive information to an unauthorized actor in Azure DevOps allows an unauthorized attacker to disclose information over a network.	2026-05-07	10.0
CVE-2026-33109	microsoft - azure_managed_instance_for_apache_cassandra	Improper access control in Azure Managed Instance for Apache Cassandra allows an authorized attacker to execute code over a network.	2026-05-07	9.9
CVE-2026-41512	mozilla - 0din_scanner	ai-scanner is an AI model safety scanner built on NVIDIA garak. From version 1.0.0 to before version 1.4.1, there is a remote code execution vulnerability via JavaScript injection in `BrowserAutomation::PlaywrightService`. This issue has been patched in version 1.4.1.	2026-05-08	9.9
CVE-2026-42027	apache - multiple products	Arbitrary Class Instantiation via Model Manifest in Apache OpenNLP ExtensionLoader Versions Affected: before 2.5.9, before 3.0.0-M3 Description: The ExtensionLoader.instantiateExtension(Class, String) method loads a class by its fully-qualified name via Class.forName() and invokes its no-arg constructor, with the class name sourced from the manifest.properties entry of a model archive. The existing isAssignableFrom check correctly rejects classes that are not subtypes of the expected extension interface (BaseToolFactory for factory=, ArtifactSerializer for serializer-class-*), but the check runs after Class.forName() has already loaded and initialized the named class. Class.forName() with default initialization semantics executes the target class's static initializer before returning, so an attacker who can supply a crafted model archive can cause the static initializer of any class on the classpath to run during model loading, regardless of whether that class passes the subsequent type check. Exploitation requires a class with attacker-useful side effects in its static initializer (for example, JNDI lookup, outbound network I/O, or filesystem access) to be present on the classpath, so this is not a drop-in remote code execution; however, the attack surface grows as third-party model distribution becomes more common (community model repositories, Hugging Face-style sharing), where users routinely load model files from origins they do not control. A secondary, narrower vector affects deployments that ship legitimate BaseToolFactory or ArtifactSerializer subclasses with side-effecting no-arg constructors: a malicious manifest can name such a class and force its constructor to run during model load.	2026-05-04	9.8

		Mitigation: * 2.x users should upgrade to 2.5.9. * 3.x users should upgrade to 3.0.0-M3. Note: The fix introduces a package-prefix allowlist that is consulted before Class.forName() is invoked, so the static initializer of a disallowed class is never executed. Classes under the opennlp. prefix remain permitted by default. Deployments that load models referencing factories or serializers outside opennlp.* must opt those packages in, either programmatically via ExtensionLoader.registerAllowedPackage(String) before the first model load, or by setting the OPENNLP_EXT_ALLOWED_PACKAGES system property to a comma-separated list of allowed package prefixes. Users who cannot upgrade immediately should ensure that all model files are sourced from trusted origins and should audit their classpath for classes with side-effecting static initializers or constructors, particularly any that perform JNDI lookups, network requests, or filesystem operations during class initialization.		
CVE-2026-43067	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: handle wraparound when searching for blocks for indirect mapped blocks Commit 4865c768b563 ("ext4: always allocate blocks only from groups inode can use") restricts what blocks will be allocated for indirect block based files to block numbers that fit within 32-bit block numbers. However, when using a review bot running on the latest Gemini LLM to check this commit when backporting into an LTS based kernel, it raised this concern: If ac->ac_g_ex.fe_group is >= ngroups (for instance, if the goal group was populated via stream allocation from s_mb_last_groups), then start will be >= ngroups. Does this allow allocating blocks beyond the 32-bit limit for indirect block mapped files? The commit message mentions that ext4_mb_scan_groups_linear() takes care to not select unsupported groups. However, its loop uses group = *start, and the very first iteration will call ext4_mb_scan_group() with this unsupported group because next_linear_group() is only called at the end of the iteration. After reviewing the code paths involved and considering the LLM review, I determined that this can happen when there is a file system where some files/directories are extent-mapped and others are indirect-block mapped. To address this, add a safety clamp in ext4_mb_scan_groups().	2026-05-05	9.8
CVE-2026-28780	apache - http_server	Heap-based Buffer Overflow vulnerability in mod_proxy_ajp of Apache HTTP Server. If mod_proxy_ajp connects to a malicious AJP server this AJP server can send a malicious AJP message back to mod_proxy_ajp and cause it to write 4 attacker controlled bytes after the end of a heap based buffer. This issue affects Apache HTTP Server: through 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-05	9.8
CVE-2026-43125	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dlm: validate length in dlm_search_rsb_tree The len parameter in dlm_dump_rsb_name() is not validated and comes from network messages. When it exceeds DLM_RESNAME_MAXLEN, it can cause out-of-bounds write in dlm_search_rsb_tree(). Add length validation to prevent potential buffer overflow.	2026-05-06	9.8
CVE-2026-43185	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix signededness bug in smb_direct_prepare_negotiation() smb_direct_prepare_negotiation() casts an unsigned __u32 value from sp->max_recv_size and req->preferred_send_size to a signed int before computing min_t(int, ...). A maliciously provided preferred_send_size of 0x80000000 will return as smaller than max_recv_size, and then be used to set the maximum allowed	2026-05-06	9.8

		alowed receive size for the next message. By sending a second message with a large value (>1420 bytes) the attacker can then achieve a heap buffer overflow. This fix replaces min_t(int, ...) with min_t(u32)		
CVE-2026-43186	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: ioam: fix heap buffer overflow in __ioam6_fill_trace_data() On the receive path, __ioam6_fill_trace_data() uses trace->nodelen to decide how much data to write for each node. It trusts this field as-is from the incoming packet, with no consistency check against trace->type (the 24-bit field that tells which data items are present). A crafted packet can set nodelen=0 while setting type bits 0-21, causing the function to write ~100 bytes past the allocated region (into skb_shared_info), which corrupts adjacent heap memory and leads to a kernel panic. Add a shared helper ioam6_trace_compute_nodelen() in ioam6.c to derive the expected nodelen from the type field, and use it: - in ioam6_iptunnel.c (send path, existing validation) to replace the open-coded computation; - in exthdrs.c (receive path, ipv6_hop_ioam) to drop packets whose nodelen is inconsistent with the type field, before any data is written. Per RFC 9197, bits 12-21 are each short (4-octet) fields, so they are included in IOAM6_MASK_SHORT_FIELDS (changed from 0xff100000 to 0xff1ffc00).	2026-05-06	9.8
CVE-2026-43198	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tcp: fix potential race in tcp_v6_syn_recv_sock() Code in tcp_v6_syn_recv_sock() after the call to tcp_v4_syn_recv_sock() is done too late. After tcp_v4_syn_recv_sock(), the child socket is already visible from TCP ehash table and other cpus might use it. Since newinet->pinet6 is still pointing to the listener ipv6_pinfo bad things can happen as syzbot found. Move the problematic code in tcp_v6_mapped_child_init() and call this new helper from tcp_v4_syn_recv_sock() before the ehash insertion. This allows the removal of one tcp_sync_mss(), since tcp_v4_syn_recv_sock() will call it with the correct context.	2026-05-06	9.8
CVE-2026-43208	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: do not pass flow_id to set_rps_cpu() Blamed commit made the assumption that the RPS table for each receive queue would have the same size, and that it would not change. Compute flow_id in set_rps_cpu(), do not assume we can use the value computed by get_rps_cpu(). Otherwise we risk out-of-bound access and/or crashes.	2026-05-06	9.8
CVE-2026-8091	mozilla - multiple products	Incorrect boundary conditions in the Audio/Video: Playback component. This vulnerability was fixed in Firefox 150, Thunderbird 150, Firefox ESR 140.10.1, Thunderbird 140.10.1, and Firefox ESR 115.35.2.	2026-05-07	9.8
CVE-2026-8094	mozilla - multiple products	Other issue in the WebRTC component. This vulnerability was fixed in Firefox ESR 140.10.2 and Thunderbird 140.10.2.	2026-05-07	9.8
CVE-2026-43304	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: define and enforce CEPH_MAX_KEY_LEN When decoding the key, verify that the key material would fit into a fixed-size buffer in process_auth_done() and generally has a sane length. The new CEPH_MAX_KEY_LEN check replaces the existing check for a key with no key material which is a) not universal since CEPH_CRYPTO_NONE has to be excluded and b) doesn't provide much value since a smaller than needed key is just as invalid as no key -- this has to be handled elsewhere anyway.	2026-05-08	9.8

CVE-2026-43341	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/ipv6: ioam6: prevent schema length wraparound in trace fill ioam6_fill_trace_data() stores the schema contribution to the trace length in a u8. With bit 22 enabled and the largest schema payload, sclen becomes 1 + 1020 / 4, wraps from 256 to 0, and bypasses the remaining-space check. __ioam6_fill_trace_data() then positions the write cursor without reserving the schema area but still copies the 4-byte schema header and the full schema payload, overrunning the trace buffer. Keep sclen in an unsigned int so the remaining-space check and the write cursor calculation both see the full schema length.	2026-05-08	9.8
CVE-2026-43376	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free by using call_rcu() for oplock_info ksmbd currently frees oplock_info immediately using kfree(), even though it is accessed under RCU read-side critical sections in places like opinfo_get() and proc_show_files(). Since there is no RCU grace period delay between nullifying the pointer and freeing the memory, a reader can still access oplock_info structure after it has been freed. This can leads to a use-after-free especially in opinfo_get() where atomic_inc_not_zero() is called on already freed memory. Fix this by switching to deferred freeing using call_rcu().	2026-05-08	9.8
CVE-2026-43379	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb_lazy_parent_lease_break_close() opinfo pointer obtained via rcu_dereference(fp->f_opinfo) is being accessed after rcu_read_unlock() has been called. This creates a race condition where the memory could be freed by a concurrent writer between the unlock and the subsequent pointer dereferences (opinfo->is_lease, etc.), leading to a use-after-free.	2026-05-08	9.8
CVE-2026-43402	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: kthread: consolidate kthread exit paths to prevent use-after-free Guillaume reported crashes via corrupted RCU callback function pointers during KUnit testing. The crash was traced back to the pidfs rhashtable conversion which replaced the 24-byte rb_node with an 8-byte rhash_head in struct pid, shrinking it from 160 to 144 bytes. struct kthread (without CONFIG_BLK_CGROUP) is also 144 bytes. With CONFIG_SLAB_MERGE_DEFAULT and SLAB_HWCACHE_ALIGN both round up to 192 bytes and share the same slab cache. struct pid.rcu.func and struct kthread.affinity_node both sit at offset 0x78. When a kthread exits via make_task_dead() it bypasses kthread_exit() and misses the affinity_node cleanup. free_kthread_struct() frees the memory while the node is still linked into the global kthread_affinity_list. A subsequent list_del() by another kthread writes through dangling list pointers into the freed and reused memory, corrupting the pid's rcu.func pointer. Instead of patching free_kthread_struct() to handle the missed cleanup, consolidate all kthread exit paths. Turn kthread_exit() into a macro that calls do_exit() and add kthread_do_exit() which is called from do_exit() for any task with PF_KTHREAD set. This guarantees that kthread-specific cleanup always happens regardless of the exit path - make_task_dead(), direct do_exit(), or kthread_exit(). Replace __to_kthread() with a new tsk_is_kthread() accessor in the public header. Export do_exit() since module code using the kthread_exit() macro now needs it directly.	2026-05-08	9.8
CVE-2026-43414	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Completely fix fcport double free In qla24xx_els_dcmd_iocb() sp->free is set to qla2x00_els_dcmd_sp_free(). When an error happens, this function is called by qla2x00_sp_release(), when kref_put() releases the first and the last reference. qla2x00_els_dcmd_sp_free() frees fcport by calling qla2x00_free_fcport(). Doing it one more time after kref_put() is a bad idea.	2026-05-08	9.8

CVE-2026-43465	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: RX, Fix XDP multi-buf frag counting for striding RQ XDP multi-buf programs can modify the layout of the XDP buffer when the program calls bpf_xdp_pull_data() or bpf_xdp_adjust_tail(). The referenced commit in the fixes tag corrected the assumption in the mlx5 driver that the XDP buffer layout doesn't change during a program execution. However, this fix introduced another issue: the dropped fragments still need to be counted on the driver side to avoid page fragment reference counting issues. The issue was discovered by the drivers/net/xdp.py selftest, more specifically the test_xdp_native_tx_mb: - The mlx5 driver allocates a page_pool page and initializes it with a frag counter of 64 (pp_ref_count=64) and the internal frag counter to 0. - The test sends one packet with no payload. - On RX (mlx5e_skb_from_cqe_mpwrq_nonlinear()), mlx5 configures the XDP buffer with the packet data starting in the first fragment which is the page mentioned above. - The XDP program runs and calls bpf_xdp_pull_data() which moves the header into the linear part of the XDP buffer. As the packet doesn't contain more data, the program drops the tail fragment since it no longer contains any payload (pp_ref_count=63). - mlx5 device skips counting this fragment. Internal frag counter remains 0. - mlx5 releases all 64 fragments of the page but page pp_ref_count is 63 => negative reference counting error. Resulting splat during the test: WARNING: CPU: 0 PID: 188225 at ./include/net/page_pool/helpers.h:297 mlx5e_page_release_fragmented.isra.0+0xbd/0xe0 [mlx5_core] Modules linked in: [...] CPU: 0 UID: 0 PID: 188225 Comm: ip Not tainted 6.18.0-rc7_for_upstream_min_debug_2025_12_08_11_44 #1 NONE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:mlx5e_page_release_fragmented.isra.0+0xbd/0xe0 [mlx5_core] [...] Call Trace: <TASK> mlx5e_free_rx_mpwqe+0x20a/0x250 [mlx5_core] mlx5e_dealloc_rx_mpwqe+0x37/0xb0 [mlx5_core] mlx5e_free_rx_descs+0x11a/0x170 [mlx5_core] mlx5e_close_rq+0x78/0xa0 [mlx5_core] mlx5e_close_queues+0x46/0x2a0 [mlx5_core] mlx5e_close_channel+0x24/0x90 [mlx5_core] mlx5e_close_channels+0x5d/0xf0 [mlx5_core] mlx5e_safe_switch_params+0x2ec/0x380 [mlx5_core] mlx5e_change_mtu+0x11d/0x490 [mlx5_core] mlx5e_change_nic_mtu+0x19/0x30 [mlx5_core] netif_set_mtu_ext+0xfc/0x240 do_setlink.isra.0+0x226/0x1100 rtnl_newlink+0x7a9/0xba0 rtnetlink_rcv_msg+0x220/0x3c0 netlink_rcv_skb+0x4b/0xf0 netlink_unicast+0x255/0x380 netlink_sendmsg+0x1f3/0x420 __sock_sendmsg+0x38/0x60 ____sys_sendmsg+0x1e8/0x240 ___sys_sendmsg+0x7c/0xb0 [...] __sys_sendmsg+0x5f/0xb0 do_syscall_64+0x55/0xc70 The problem applies for XDP_PASS as well which is handled in a different code path in the driver. This patch fixes the issue by doing page frag counting on all the original XDP buffer fragments for all relevant XDP actions (XDP_TX , XDP_REDIRECT and XDP_PASS). This is basically reverting to the original counting before the commit in the fixes tag. As frag_page is still pointing to the original tail, the nr_frags parameter to xdp_update_skb_frags_info() needs to be calculated in a different way to reflect the new nr_frags.	2026-05-08	9.8

CVE-2026-25293	qualcomm - qca7005_firmware	Buffer overflow due to incorrect authorization in PLC FW	2026-05-04	9.6
CVE-2026-7908	google - chrome	Use after free in Fullscreen in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-05-06	9.6
CVE-2026-7910	google - chrome	Use after free in Views in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	2026-05-06	9.6
CVE-2026-33823	microsoft - teams	Improper authorization in Microsoft Teams allows an authorized attacker to disclose information over a network.	2026-05-07	9.6
CVE-2026-35428	microsoft - azure_cloud_shell	Improper neutralization of special elements used in a command ('command injection') in Azure Cloud Shell allows an unauthorized attacker to perform spoofing over a network.	2026-05-07	9.6
CVE-2026-42809	apache - polaris	Apache Polaris can issue broad temporary ("vended") storage credentials during staged table creation before the effective table location has been validated or durably reserved. Those temporary credentials are meant to limit the scope of accessible table data and metadata, but this scope limitation becomes attacker-directed because the attacker can choose a reachable target location. In the confirmed variant, if the caller supplies a custom `location` during stage create and requests credential vending, Apache Polaris uses that location to construct delegated storage credentials immediately. The stage-create path itself neither runs the normal location validation nor the overlap checks before those credentials are issued. Closely related to that, the staged-create flow also accepts `write.data.path` / `write.metadata.path` in the request properties and feeds those location overrides into the same effective table location set used for credential vending. Those fields are secondary to the main custom-`location` exploit, but they are still attacker-influenced location inputs that should be validated before any credentials are issued.	2026-05-04	9.4
CVE-2026-42810	apache - polaris	Apache Polaris accepts literal `*` characters in namespace and table names. When it later builds temporary S3 access policies for delegated table access, those same characters appear to be reused unescaped in S3 IAM resource patterns and `s3:prefix` conditions. In S3 IAM policy matching, `*` is treated as a wildcard rather than as ordinary text. That means temporary credentials issued for one crafted table can match the storage path of a different table. In private testing against Polaris 1.4.0 using Polaris' AWS S3 temporary-credential path on both MinIO and real AWS S3, credentials returned for crafted tables such as `f*.t1`, `f*.*`, `*.*`, and `foo.*` could reach other tables' S3 locations. The confirmed behavior includes: - reading another table's metadata control file ([Iceberg metadata JSON]); - listing another table's exact S3 table prefix ([table prefix]); - and, when write delegation was returned for the crafted table, creating and deleting an object under another table's exact S3 table prefix. A control case using ordinary different names did not allow the same cross-table access. A least-privilege AWS S3 variant was also confirmed in which the attacker principal had no Polaris permissions on the victim table and only the	2026-05-04	9.4

		minimal permissions required to create and use a crafted wildcard table (namespace-scoped `TABLE_CREATE` and `TABLE_WRITE_DATA` on `*`). In that setup, direct Polaris access to `foo.t1` remained forbidden, but the attacker could still create and load `*.*`, receive delegated S3 credentials, and use those credentials to list, read, create, and delete objects under `foo.t1`. In Iceberg, the metadata JSON file is a control file: it tells readers which data files belong to the table, which snapshots exist, and which table version to read. So unauthorized access to it is already a meaningful confidentiality problem. The confirmed write-capable variant means the issue is not limited to disclosure.		
[CVE-2026-42811](#)	apache - polaris	In plain terms, Apache Polaris is supposed to issue short-lived GCS credentials that only work for one table's files, but a crafted namespace or table name can cause those credentials to work across the configured bucket instead. Apache Polaris builds Google Cloud Storage downscoped credentials by creating a Credential Access Boundary (CAB) with CEL conditions that are intended to restrict access to the requested table's storage path. The relevant CEL string is built from the bucket name and the table path. That table path is derived from namespace and table identifiers. In current code, that path appears to be inserted into the CEL expression without escaping. As a result, a namespace or table identifier containing a single quote and other URI-safe CEL fragments can break out of the intended quoted string and change the meaning of the CEL condition. In private testing against Polaris 1.4.0 on real Google Cloud Storage, it was confirmed that Polaris accepted a crafted identifier and returned delegated GCS credentials whose CEL path restriction had effectively collapsed. Those delegated credentials could then: - list another table's object prefix; - read another table's metadata control file (Iceberg metadata JSON); - create and delete an object under another table's object prefix; - and also list, read, create, and delete objects under an unrelated external prefix in the same bucket that was not part of any table path. That last point is important. The issue is not limited to "another table". In the confirmed setup, once Apache Polaris returned credentials for the crafted table, the path restriction inside the configured bucket was effectively gone. The practical effect is that temporary credentials for one crafted table can be broader than the table Polaris was asked to authorize, and can become effectively bucket-wide within the configured bucket. The current GCS testing used a Polaris principal with broad catalog privileges for setup. A separate least-privilege Polaris RBAC variant has not yet been tested on GCS. However, the storage-credential broadening behavior itself has been confirmed on GCS.	2026-05-04	9.4

		In Apache Iceberg, the table's metadata files are control files: they tell readers which data files belong to the table and which table version to read. `write.metadata.path` is an optional table property that tells Polaris where to write those metadata files. For a table already registered in a Polaris-managed catalog, changing only that property through an `ALTER TABLE`-style settings change (not a row-level `INSERT`, `SELECT`, `UPDATE`, or `DELETE`) bypasses the commit-time branch that is supposed to revalidate storage locations. The full persisted / credential-vending variant requires the affected catalog to have `polaris.config.allow.unstructured.table.location=true`, with `allowedLocations` broad enough to include the attacker-chosen target. `allowedLocations` is the admin-configured allowlist of storage paths that the catalog is allowed to use. Public project materials suggest that this flag is a real supported compatibility / layout mode, not just a contrived lab-only prerequisite. In that configuration, a user who can change table settings can cause Apache Polaris itself to write new table metadata to an attacker-chosen reachable storage location before the intended location-validation branch runs. If the later concrete-path validation also accepts that location, Polaris persists the resulting metadata path into stored table state. Later table-load and credential APIs can then return temporary cloud-storage credentials for the same location without revalidating it. In plain terms, Polaris can later hand out temporary storage access for the same attacker-chosen area. That attacker-chosen area does not need to be limited to the poisoned table's own files. If it is a broader storage prefix, another table's prefix, or, depending on configuration or provider behavior, even a bucket/container root, the resulting disclosure or corruption scope can extend to any data and metadata Polaris can reach there. The practical consequences are therefore similar to the staged-create credential-vending issue already discussed: data and metadata reachable in that storage scope can be exposed and, if write-capable credentials are later issued, modified, corrupted, or removed. Even before that later credential step, Polaris itself performs the metadata write to the unchecked location. So the core issue is not only later credential vending. The primary defect is that Polaris skips its intended location checks before performing a security-sensitive metadata write when only `write.metadata.path` changes. When `polaris.config.allow.unstructured.table.location=false`, current code review suggests the later `updateTableLike(...)` validation usually rejects out-of-tree metadata locations before the unsafe path is persisted. That may reduce the persisted / credential-vending variant, but it does not prevent the underlying defect: Polaris still skips the intended pre-write location check when only `write.metadata.path` changes.		
CVE-2026-42812	apache - polaris		2026-05-04	9.4
CVE-2026-43114	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo_avx2: don't return non-matching entry on expiry New test case fails unexpectedly when avx2 matching functions are used.	2026-05-06	9.4

		The test first loads a ranomly generated pipapo set with 'ipv4 . port' key, i.e. nft -f foo. This works. Then, it reloads the set after a flush: (echo flush set t s; cat foo) | nft -f - This is expected to work, because its the same set after all and it was already loaded once. But with avx2, this fails: nft reports a clashing element. The reported clash is of following form: We successfully re-inserted a . b c . d Then we try to insert a . d avx2 finds the already existing a . d, which (due to 'flush set') is marked as invalid in the new generation. It skips the element and moves to next. Due to incorrect masking, the skip-step finds the next matching element *only considering the first field*, i.e. we return the already reinserted "a . b", even though the last field is different and the entry should not have been matched. No such error is reported for the generic c implementation (no avx2) or when the last field has to use the 'nft_pipapo_avx2_lookup_slow' fallback. Bisection points to 7711f4bb4b36 ("netfilter: nft_set_pipapo: fix range overlap detection") but that fix merely uncovers this bug. Before this commit, the wrong element is returned, but erronously reported as a full, identical duplicate. The root-cause is too early return in the avx2 match functions. When we process the last field, we should continue to process data until the entire input size has been consumed to make sure no stale bits remain in the map.		
CVE-2026-0300	paloaltonetworks - multiple products	A buffer overflow vulnerability in the User-ID™ Authentication Portal (aka Captive Portal) service of Palo Alto Networks PAN-OS software allows an unauthenticated attacker to execute arbitrary code with root privileges on the PA-Series and VM-Series firewalls by sending specially crafted packets. The risk of this issue is greatly reduced if you secure access to the User-ID™ Authentication Portal per the best practice guidelines https://knowledgebase.paloaltonetworks.com/KCSArticleDetail by restricting access to only trusted internal IP addresses. Prisma Access, Cloud NGFW and Panorama appliances are not impacted by this vulnerability.	2026-05-06	9.3
CVE-2022-50994	draytek - Vigor 2960	DrayTek Vigor 2960 firmware versions prior to 1.5.1.4 contain an OS command injection vulnerability in the CGI login handler that allows unauthenticated remote attackers to execute arbitrary commands by injecting shell metacharacters into the formpassword parameter. Attackers can exploit unsanitized input passed to the otp_check.sh script to achieve remote code execution with web server privileges. Exploitation requires knowledge of a valid username and that the target account has MOTP authentication enabled.	2026-05-08	9.2
CVE-2026-40682	apache - multiple products	XML External Entity (XXE) via Unsanitized Dictionary Parsing in Apache OpenNLP DictionaryEntryPersistor Versions Affected: before 2.5.9, before 3.0.0-M3 Description: The DictionaryEntryPersistor class initializes a static SAXParserFactory at class-load time without enabling FEATURE_SECURE_PROCESSING or disabling DTD processing. When create(InputStream, EntryInserter) is invoked, the only feature set on the XMLReader is namespace support — external entity resolution and DOCTYPE declarations remain fully enabled. An attacker who can supply a crafted dictionary file (e.g., a stop-word list or domain dictionary) containing a malicious DOCTYPE declaration can trigger local file disclosure via file:// entity references or server-side request forgery via http:// entity references during SAX parsing, before the application processes a single dictionary entry. This is inconsistent with the project's own XmlUtil.createSaxParser() helper, which correctly sets FEATURE_SECURE_PROCESSING and disallow-doctype-decl and is used by all other XML parsing paths in the codebase. The public Dictionary(InputStream) constructor delegates directly to this method and is the documented API for loading user-supplied dictionaries, making untrusted input a realistic scenario. Mitigation: 2.x users should upgrade to 2.5.9. 3.x users should upgrade to 3.0.0-M3. Users who cannot	2026-05-04	9.1

		upgrade immediately should ensure that all dictionary files are sourced from trusted origins and should consider wrapping the Dictionary(InputStream) constructor with input validation that rejects any XML containing a DOCTYPE declaration before it reaches the parser.		

CVE-2026-43071	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		dcache: Limit the minimal number of bucket to two
		
		There is an OOB read problem on dentry_hashtable when user sets 'dhash_entries=1':
		BUG: unable to handle page fault for address: ffff888b30b774b0
		#PF: supervisor read access in kernel mode
		#PF: error_code(0x0000) - not-present page
		Oops: Oops: 0000 [#1] SMP PTI
		RIP: 0010:__d_lookup+0x56/0x120
		Call Trace:
		d_lookup.cold+0x16/0x5d
		lookup_dcache+0x27/0xf0
		lookup_one_qstr_excl+0x2a/0x180
		start_dirop+0x55/0xa0
		simple_start_creating+0x8d/0xa0
		debugfs_start_creating+0x8c/0x180
		debugfs_create_dir+0x1d/0x1c0
		pinctrl_init+0x6d/0x140
		do_one_initcall+0x6d/0x3d0
		kernel_init_freeable+0x39f/0x460
		kernel_init+0x2a/0x260
		
		There will be only one bucket in dentry_hashtable when dhash_entries is set as one, and d_hash_shift is calculated as 32 by dcache_init(). Then, following process will access more than one buckets(which memory region is not allocated) in dentry_hashtable:
		d_lookup
		b = d_hash(hash)
		dentry_hashtable + ((u32)hashlen >> d_hash_shift)
		// The C standard defines the behavior of right shift amounts
		// exceeding the bit width of the operand as undefined. The
		// result of '(u32)hashlen >> d_hash_shift' becomes 'hashlen',
		// so 'b' will point to an unallocated memory region.
		hlist_bl_for_each_entry_rcu(b)
		hlist_bl_first_rcu(head)
		h->first // read OOB!
		
		Fix it by limiting the minimal number of dentry_hashtable bucket to two, so that 'd_hash_shift' won't exceeds the bit width of type u32.
			2026-05-05	9.1

CVE-2026-40010	apache - multiple products	Missing invocation of Servlet http web request method changeSessionId after session binding can be exploited for a session fixation attack in Apache Wicket. This issue affects Apache Wicket: from 8.0.0 through 8.17.0, 9.0.0, from 10.0.0 through 10.8.0. Users are recommended to upgrade to version 10.9.0, which fixes the issue.	2026-05-06	9.1

CVE-2026-43083	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net: ioam6: fix OOB and missing lock
		
		When trace->type.bit6 is set:
		
		if (trace->type.bit6) {
		...
		queue = skb_get_tx_queue(dev, skb);
		qdisc = rcu_dereference(queue->qdisc);
		
		This code can lead to an out-of-bounds access of the dev->_tx[] array when is_input is true. In such a case, the packet is on the RX path and skb->queue_mapping contains the RX queue index of the ingress device. If the ingress device has more RX queues than the egress device (dev) has TX queues, skb_get_queue_mapping(skb) will exceed dev->num_tx_queues. Add a check to avoid this situation since skb_get_tx_queue() does not clamp the index. This issue has also revealed that per queue visibility cannot be accurate and will be replaced later as a new feature.
		
		While at it, add missing lock around qdisc_qstats_qlen_backlog(). The function __ioam6_fill_trace_data() is called from both softirq and process contexts, hence the use of spin_lock_bh() here.
			2026-05-06	9.1

CVE-2026-43117	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: tracepoints: get correct superblock from dentry in event btrfs_sync_file() If overlay is used on top of btrfs, dentry->d_sb translates to overlay's super block and fsid assignment will lead to a crash.	2026-05-06	9.1

		Use file_inode(file)->i_sb to always get btrfs_sb.		
CVE-2026-43197	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netconsole: avoid OOB reads, msg is not nul-terminated msg passed to netconsole from the console subsystem is not guaranteed to be nul-terminated. Before recent commit 7eab73b18630 ("netconsole: convert to NBCON console infrastructure") the message would be placed in printk_shared_pbufs, a static global buffer, so KASAN had harder time catching OOB accesses. Now we see: printk: console [netcon_ext0] enabled BUG: KASAN: slab-out-of-bounds in string+0x1f7/0x240 Read of size 1 at addr ffff88813b6d4c00 by task pr/netcon_ext0/594 CPU: 65 UID: 0 PID: 594 Comm: pr/netcon_ext0 Not tainted 6.19.0-11754-g4246fd6547c9 Call Trace: kasan_report+0xe4/0x120 string+0x1f7/0x240 vsnprintf+0x655/0xba0 scnprintf+0xba/0x120 netconsole_write+0x3fe/0xa10 nbcon_emit_next_record+0x46e/0x860 nbcon_kthread_func+0x623/0x750 Allocated by task 1: nbcon_alloc+0x1ea/0x450 register_console+0x26b/0xe10 init_netconsole+0xbb0/0xda0 The buggy address belongs to the object at ffff88813b6d4000 which belongs to the cache kmalloc-4k of size 4096 The buggy address is located 0 bytes to the right of allocated 3072-byte region [ffff88813b6d4000, ffff88813b6d4c00)	2026-05-06	9.1
CVE-2026-40982	vmware - multiple products	Spring Cloud Config allows applications to serve arbitrary text and binary files through the spring-cloud-config-server module. A malicious user, or attacker, can send a request using a specially crafted URL that can lead to a directory traversal attack. Spring Cloud Config 3.1.x: affected from 3.1.0 through 3.1.13 (inclusive); upgrade to 3.1.14 or greater (Enterprise Support Only). Spring Cloud Config 4.1.x: affected from 4.1.0 through 4.1.9 (inclusive); upgrade to 4.1.10 or greater (Enterprise Support Only). Spring Cloud Config 4.2.x: affected from 4.2.0 through 4.2.6 (inclusive); upgrade to 4.2.7 or greater (Enterprise Support Only). Spring Cloud Config 4.3.x: affected from 4.3.0 through 4.3.2 (inclusive); upgrade to 4.3.3 or greater. Spring Cloud Config 5.0.x: affected from 5.0.0 through 5.0.2 (inclusive); upgrade to 5.0.3 or greater.	2026-05-07	9.1
CVE-2026-25199	apache - cloudstack	Instances deployed via the Proxmox extension allow unauthorized access to instances belonging to other tenants. This issue affects Apache CloudStack: from 4.21.0.0 through 4.22.0.0. The Proxmox extension for CloudStack improperly uses a user-editable instance setting, proxmox_vmid, to associate CloudStack instances with Proxmox virtual machines. Because this value is not restricted or validated against tenant ownership and Proxmox VM IDs are predictable, a non-privileged attacker can modify the setting to reference a VM belonging to another account. This allows unauthorized cross-tenant access and enables full control over the targeted VM, including starting, stopping, and destroying the virtual machine. Users are recommended to upgrade to version 4.22.0.1, which fixes this issue. As a workaround for the existing installations, editing of the proxmox_vmid instance detail by users can be prevented by adding this detail name to the global configuration parameter - user.vm.denied.details.	2026-05-08	9.1
CVE-2026-43406	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: prevent potential out-of-bounds reads in process_message_header() If the message frame is (maliciously) corrupted in a way that the length of the control segment ends up being less than the size of the message header or a different frame is made to look like a message	2026-05-08	9.1

		frame, out-of-bounds reads may ensue in process_message_header(). Perform an explicit bounds check before decoding the message header.		
CVE-2026-43407	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Fix potential out-of-bounds access in ceph_handle_auth_reply() This patch fixes an out-of-bounds access in ceph_handle_auth_reply() that can be triggered by a message of type CEPH_MSG_AUTH_REPLY. In ceph_handle_auth_reply(), the value of the payload_len field of such a message is stored in a variable of type int. A value greater than INT_MAX leads to an integer overflow and is interpreted as a negative value. This leads to decrementing the pointer address by this value and subsequently accessing it because ceph_decode_need() only checks that the memory access does not exceed the end address of the allocation. This patch fixes the issue by changing the data type of payload_len to u32. Additionally, the data type of result_msg_len is changed to u32, as it is also a variable holding a non-negative length. Also, an additional layer of sanity checks is introduced, ensuring that directly after reading it from the message, payload_len and result_msg_len are not greater than the overall segment length. BUG: KASAN: slab-out-of-bounds in ceph_handle_auth_reply+0x642/0x7a0 [libceph] Read of size 4 at addr ffff88811404df14 by task kworker/20:1/262 CPU: 20 UID: 0 PID: 262 Comm: kworker/20:1 Not tainted 6.19.2 #5 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Workqueue: ceph-msgr ceph_con_workfn [libceph] Call Trace: <TASK> dump_stack_lvl+0x76/0xa0 print_report+0xd1/0x620 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 ? kasan_complete_mode_report_info+0x72/0x210 kasan_report+0xe7/0x130 ? ceph_handle_auth_reply+0x642/0x7a0 [libceph] ? ceph_handle_auth_reply+0x642/0x7a0 [libceph] __asan_report_load_n_noabort+0xf/0x20 ceph_handle_auth_reply+0x642/0x7a0 [libceph] mon_dispatch+0x973/0x23d0 [libceph] ? apparmor_socket_recvmsg+0x6b/0xa0 ? __pfx_mon_dispatch+0x10/0x10 [libceph] ? __kasan_check_write+0x14/0x30i ? mutex_unlock+0x7f/0xd0 ? __pfx_mutex_unlock+0x10/0x10 ? __pfx_do_recvmsg+0x10/0x10 [libceph] ceph_con_process_message+0x1f1/0x650 [libceph] process_message+0x1e/0x450 [libceph] ceph_con_v2_try_read+0x2e48/0x6c80 [libceph] ? __pfx_ceph_con_v2_try_read+0x10/0x10 [libceph] ? save_fpregs_to_fpstate+0xb0/0x230 ? raw_spin_rq_unlock+0x17/0xa0 ? finish_task_switch.isra.0+0x13b/0x760 ? __switch_to+0x385/0xda0 ? __kasan_check_write+0x14/0x30 ? mutex_lock+0x8d/0xe0 ? __pfx_mutex_lock+0x10/0x10 ceph_con_workfn+0x248/0x10c0 [libceph] process_one_work+0x629/0xf80 ? __kasan_check_write+0x14/0x30 worker_thread+0x87f/0x1570 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 ? __pfx_try_to_wake_up+0x10/0x10 ? kasan_print_address_stack_frame+0x1f7/0x280 ? __pfx_worker_thread+0x10/0x10 kthread+0x396/0x830 ? __pfx__raw_spin_lock_irq+0x10/0x10 ? __pfx_kthread+0x10/0x10 ? __kasan_check_write+0x14/0x30 ? recalc_sigpending+0x180/0x210 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x3f7/0x610 ? __pfx_ret_from_fork+0x10/0x10 ? __switch_to+0x385/0xda0 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>	2026-05-08	9.1

		[idryomov: replace if statements with ceph_decode_need() for payload_len and result_msg_len]		
CVE-2026-33844	microsoft - azure_managed_instance_for_apache_cassandra	Improper input validation in Azure Managed Instance for Apache Cassandra allows an authorized attacker to execute code over a network.	2026-05-07	9.0
CVE-2026-5787	ivanti - multiple products	An Improper Certificate Validation in Ivanti EPMM before versions 12.6.1.1, 12.7.0.1, and 12.8.0.1 allows a remote unauthenticated attacker to impersonate registered Sentry hosts and obtain valid CA-signed client certificates.	2026-05-07	8.9
CVE-2026-24072	apache - http_server	An escalation of privilege bug in various modules in Apache HTTP 2.4.66 and earlier allows local .htaccess authors to read files with the privileges of the httpd user. Users are recommended to upgrade to version 2.4.67, which fixes this issue.	2026-05-04	8.8
CVE-2026-23918	apache - http_server	Double Free and possible RCE vulnerability in Apache HTTP Server with the HTTP/2 protocol. This issue affects Apache HTTP Server: 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-04	8.8
CVE-2026-0073	google - multiple products	In adbd_tls_verify_cert of auth.cpp, there is a possible bypass of wireless ADB mutual authentication due to a logic error in the code. This could lead to remote (proximal/adjacent) code execution as the shell user with no additional execution privileges needed. User interaction is not needed for exploitation.	2026-05-04	8.8
CVE-2026-43110	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: validate bsscfg indices in IF events brcmf_fweh_handle_if_event() validates the firmware-provided interface index before it touches drvr->iflist[], but it still uses the raw bsscfgidx field as an array index without a matching range check. Reject IF events whose bsscfg index does not fit in drvr->iflist[] before indexing the interface array. [add missing wifi prefix]	2026-05-06	8.8
CVE-2026-43112	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fs/smb/client: fix out-of-bounds read in cifs_sanitize_prepath When cifs_sanitize_prepath is called with an empty string or a string containing only delimiters (e.g., "/"), the current logic attempts to check *(cursor2 - 1) before cursor2 has advanced. This results in an out-of-bounds read. This patch adds an early exit check after stripping prepended delimiters. If no path content remains, the function returns NULL. The bug was identified via manual audit and verified using a standalone test case compiled with AddressSanitizer, which triggered a SEGV on affected inputs.	2026-05-06	8.8
CVE-2026-43113	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: wl1251: validate packet IDs before indexing tx_frames wl1251_tx_packet_cb() uses the firmware completion ID directly to index the fixed 16-entry wl->tx_frames[] array. The ID is a raw u8 from the completion block, and the callback does not currently verify that it fits the array before dereferencing it. Reject completion IDs that fall outside wl->tx_frames[] and keep the existing NULL check in the same guard. This keeps the fix local to the trust boundary and avoids touching the rest of the completion flow.	2026-05-06	8.8
CVE-2026-43158	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: fix freemap adjustments when adding xattrs to leaf blocks xfs/592 and xfs/794 both trip this assertion in the leaf block freemap adjustment code after ~20 minutes of running on my test VMs: ASSERT(ichdr->firstused >= ichdr->count * sizeof(xfs_attr_leaf_entry_t) + xfs_attr3_leaf_hdr_size(leaf)); Upon enabling quite a lot more debugging code, I narrowed this down to fsstress trying to set a local extended attribute with namelen=3 and valuelen=71. This results in an entry size of 80 bytes. At the start of xfs_attr3_leaf_add_work, the freemap looks like this: i 0 base 448 size 0 rhs 448 count 46 i 1 base 388 size 132 rhs 448 count 46 i 2 base 2120 size 4 rhs 448 count 46	2026-05-06	8.8

		firstused = 520 where "rhs" is the first byte past the end of the leaf entry array. This is inconsistent -- the entries array ends at byte 448, but freemap[1] says there's free space starting at byte 388! By the end of the function, the freemap is in worse shape: i 0 base 456 size 0 rhs 456 count 47 i 1 base 388 size 52 rhs 456 count 47 i 2 base 2120 size 4 rhs 456 count 47 firstused = 440 Important note: 388 is not aligned with the entries array element size of 8 bytes. Based on the incorrect freemap, the name area starts at byte 440, which is below the end of the entries array! That's why the assertion triggers and the filesystem shuts down. How did we end up here? First, recall from the previous patch that the freemap array in an xattr leaf block is not intended to be a comprehensive map of all free space in the leaf block. In other words, it's perfectly legal to have a leaf block with: * 376 bytes in use by the entries array * freemap[0] has [base = 376, size = 8] * freemap[1] has [base = 388, size = 1500] * the space between 376 and 388 is free, but the freemap stopped tracking that some time ago If we add one xattr, the entries array grows to 384 bytes, and freemap[0] becomes [base = 384, size = 0]. So far, so good. But if we add a second xattr, the entries array grows to 392 bytes, and freemap[0] gets pushed up to [base = 392, size = 0]. This is bad, because freemap[1] hasn't been updated, and now the entries array and the free space claim the same space. The fix here is to adjust all freemap entries so that none of them collide with the entries array. Note that this fix relies on commit 2a2b5932db6758 ("xfs: fix attr leaf header freemap.size underflow") and the previous patch that resets zero length freemap entries to have base = 0.		
CVE-2026-43172	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: fix 22000 series SMEM parsing If the firmware were to report three LMACs (which doesn't exist in hardware) then using "fwrt->smem_cfg.lmac[2]" is an overrun of the array. Reject such and use IWL_FW_CHECK instead of WARN_ON in this function.	2026-05-06	8.8
CVE-2026-43176	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: pci: validate release report content before using for RTL8922DE The commit 957eda596c76 ("wifi: rtw89: pci: validate sequence number of TX release report") does validation on existing chips, which somehow a release report of SKB becomes malformed. As no clear cause found, add rules ahead for RTL8922DE to avoid crash if it happens.	2026-05-06	8.8
CVE-2026-43187	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: delete attr leaf freemap entries when empty Back in commit 2a2b5932db6758 ("xfs: fix attr leaf header freemap.size underflow"), Brian Foster observed that it's possible for a small freemap at the end of the end of the xattr entries array to experience a size underflow when subtracting the space consumed by an expansion of the entries array. There are only three freemap entries, which means that it is not a complete index of all free space in the leaf block. This code can leave behind a zero-length freemap entry with a nonzero base. Subsequent setxattr operations can increase the base up to the point that it overlaps with another freemap entry. This isn't in and of itself a problem because the code in _leaf_add that finds free space ignores any freemap entry with zero size. However, there's another bug in the freemap update code in _leaf_add, which is that it fails to update a freemap entry that begins midway through the xattr entry that was just appended to the array. That can	2026-05-06	8.8

		result in the freemap containing two entries with the same base but different sizes (0 for the "pushed-up" entry, nonzero for the entry that's actually tracking free space). A subsequent _leaf_add can then allocate xattr namevalue entries on top of the entries array, leading to data loss. But fixing that is for later. For now, eliminate the possibility of confusion by zeroing out the base of any freemap entry that has zero size. Because the freemap is not intended to be a complete index of free space, a subsequent failure to find any free space for a new xattr will trigger block compaction, which regenerates the freemap. It looks like this bug has been in the codebase for quite a long time.		
CVE-2026-43215	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix locking usage for tcon fields We used to use the cifs_tcp_ses_lock to protect a lot of objects that are not just the server, ses or tcon lists. We later introduced srv_lock, ses_lock and tc_lock to protect fields within the corresponding structs. This was done to provide a more granular protection and avoid unnecessary serialization. There were still a couple of uses of cifs_tcp_ses_lock to provide tcon fields. In this patch, I've replaced them with tc_lock.	2026-05-06	8.8

CVE-2026-43232	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net: wan: farsync: Fix use-after-free bugs caused by unfinished tasklets
		
		When the FarSync T-series card is being detached, the fst_card_info is deallocated in fst_remove_one(). However, the fst_tx_task or fst_int_task may still be running or pending, leading to use-after-free bugs when the already freed fst_card_info is accessed in fst_process_tx_work_q() or fst_process_int_work_q().
		
		A typical race condition is depicted below:
		
		CPU 0 (cleanup) | CPU 1 (tasklet)
		| fst_start_xmit()
		fst_remove_one() | tasklet_schedule()
		unregister_hdlc_device()|
		| fst_process_tx_work_q() //handler
		kfree(card) //free | do_bottom_half_tx()
		| card-> //use
		
		The following KASAN trace was captured:
		
		==========
		BUG: KASAN: slab-use-after-free in do_bottom_half_tx+0xb88/0xd00
		Read of size 4 at addr ffff88800aad101c by task ksoftirqd/3/32
		...
		Call Trace:
		<IRQ>
		dump_stack_lvl+0x55/0x70
		print_report+0xcb/0x5d0
		? do_bottom_half_tx+0xb88/0xd00
		kasan_report+0xb8/0xf0
		? do_bottom_half_tx+0xb88/0xd00
		do_bottom_half_tx+0xb88/0xd00
		? _raw_spin_lock_irqsave+0x85/0xe0
		? __pfx__raw_spin_lock_irqsave+0x10/0x10
		? __pfx___hrtimer_run_queues+0x10/0x10
		fst_process_tx_work_q+0x67/0x90
		tasklet_action_common+0x1fa/0x720
		? hrtimer_interrupt+0x31f/0x780
		handle_softirqs+0x176/0x530
		__irq_exit_rcu+0xab/0xe0
		sysvec_apic_timer_interrupt+0x70/0x80
		...
		
		Allocated by task 41 on cpu 3 at 72.330843s:
		kasan_save_stack+0x24/0x50
		kasan_save_track+0x17/0x60
		__kasan_kmalloc+0x7f/0x90
		fst_add_one+0x1a5/0x1cd0
		local_pci_probe+0xdd/0x190
		pci_device_probe+0x341/0x480
		really_probe+0x1c6/0x6a0
		__driver_probe_device+0x248/0x310
		driver_probe_device+0x48/0x210
			2026-05-06	8.8

		__device_attach_driver+0x160/0x320 bus_for_each_drv+0x101/0x190 __device_attach+0x198/0x3a0 device_initial_probe+0x78/0xa0 pci_bus_add_device+0x81/0xc0 pci_bus_add_devices+0x7e/0x190 enable_slot+0x9b9/0x1130 acpiphp_check_bridge.part.0+0x2e1/0x460 acpiphp_hotplug_notify+0x36c/0x3c0 acpi_device_hotplug+0x203/0xb10 acpi_hotplug_work_fn+0x59/0x80 ... Freed by task 41 on cpu 1 at 75.138639s: kasan_save_stack+0x24/0x50 kasan_save_track+0x17/0x60 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x43/0x70 kfree+0x135/0x410 fst_remove_one+0x2ca/0x540 pci_device_remove+0xa6/0x1d0 device_release_driver_internal+0x364/0x530 pci_stop_bus_device+0x105/0x150 pci_stop_and_remove_bus_device+0xd/0x20 disable_slot+0x116/0x260 acpiphp_disable_and_eject_slot+0x4b/0x190 acpiphp_hotplug_notify+0x230/0x3c0 acpi_device_hotplug+0x203/0xb10 acpi_hotplug_work_fn+0x59/0x80 ... The buggy address belongs to the object at ffff88800aad1000 which belongs to the cache kmalloc-1k of size 1024 The buggy address is located 28 bytes inside of freed 1024-byte region The buggy address belongs to the physical page: page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0xaad0 head: order:3 mapcount:0 entire_mapcount:0 nr_pages_mapped:0 pincount:0 flags: 0x100000000000040(head|node=0|zone=1) page_type: f5(slab) raw: 0100000000000040 ffff888007042dc0 dead000000000122 0000000000000000 raw: 0000000000000000 0000000080100010 00000000f5000000 0000000000000000 head: 0100000000000040 ffff888007042dc0 dead000000000122 0000000000000000 head: 0000000000000000 0000000080100010 00000000f5000000 0000000000000000 head: 0100000000000003 ffffea00002ab401 00000000ffffffff 00000000ffffffff head: 0000000000000000 0000000000000000 00000000ffffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff88800aad0f00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffff88800aad0f80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc >ffff88800aad1000: fa fb ---truncated---		
CVE-2026-43239	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: prevent races in ->query_interfaces() It was possible for two query interface works to be concurrently trying to update the interfaces. Prevent this by checking and updating iface_last_update under iface_lock.	2026-05-06	8.8
CVE-2026-43249	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: 9p/xen: protect xen_9pfs_front_free against concurrent calls The xenwatch thread can race with other back-end change notifications and call xen_9pfs_front_free() twice, hitting the observed general protection fault due to a double-free. Guard the teardown path so only one caller can release the front-end state at a time, preventing the crash. This is a fix for the following double-free: [27.052347] Oops: general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6b6b: 0000 [#1] SMP DEBUG_PAGEALLOC NOPTI [27.052357] CPU: 0 UID: 0 PID: 32 Comm: xenwatch Not tainted 6.18.0-02087-g51ab33fc0a8b-dirty #60 PREEMPT(none) [27.052363] RIP: e030:xen_9pfs_front_free+0x1d/0x150 [27.052368] Code: 90 90 90 90 90 90 90 90 90 90 90 90 41 55 41 54 55 48 89 fd 48 c7 c7 48 d0	2026-05-06	8.8

		92 85 53 e8 cb cb 05 00 48 8b 45 08 48 8b 55 00 <48> 3b 28 0f 85 f9 28 35 fe 48 3b 6a 08 0f 85 ef 28 35 fe 48 89 42 [27.052377] RSP: e02b:ffffc9004016fdd0 EFLAGS: 00010246 [27.052381] RAX: 6b6b6b6b6b6b6b6b RBX: ffff88800d66e400 RCX: 0000000000000000 [27.052385] RDX: 6b6b6b6b6b6b6b6b RSI: 0000000000000000 RDI: 0000000000000000 [27.052389] RBP: ffff88800a887040 R08: 0000000000000000 R09: 0000000000000000 [27.052393] R10: 0000000000000000 R11: 0000000000000000 R12: ffff888009e46b68 [27.052397] R13: 0000000000000200 R14: 0000000000000000 R15: ffff88800a887040 [27.052404] FS: 0000000000000000(0000) GS:ffff88808ca57000(0000) knlGS:0000000000000000 [27.052408] CS: e030 DS: 0000 ES: 0000 CR0: 0000000080050033 [27.052412] CR2: 00007f9714004360 CR3: 0000000004834000 CR4: 0000000000050660 [27.052418] Call Trace: [27.052420] <TASK> [27.052422] xen_9pfs_front_changed+0x5d5/0x720 [27.052426] ? xenbus_otherend_changed+0x72/0x140 [27.052430] ? __pfx_xenwatch_thread+0x10/0x10 [27.052434] xenwatch_thread+0x94/0x1c0 [27.052438] ? __pfx_autoremove_wake_function+0x10/0x10 [27.052442] kthread+0xf8/0x240 [27.052445] ? __pfx_kthread+0x10/0x10 [27.052449] ? __pfx_kthread+0x10/0x10 [27.052452] ret_from_fork+0x16b/0x1a0 [27.052456] ? __pfx_kthread+0x10/0x10 [27.052459] ret_from_fork_asm+0x1a/0x30 [27.052463] </TASK> [27.052465] Modules linked in: [27.052471] ---[end trace 0000000000000000]---		
CVE-2026-43283	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: ec_bhf: Fix dma_free_coherent() dma handle dma_free_coherent() in error path takes priv->rx_buf.alloc_len as the dma handle. This would lead to improper unmapping of the buffer. Change the dma handle to priv->rx_buf.alloc_phys.	2026-05-06	8.8
CVE-2026-20034	cisco - Cisco Unity Connection	A vulnerability in the web-based management interface of Cisco Unity Connection could allow an authenticated, remote attacker to execute arbitrary code on an affected device. This vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by submitting a crafted API request. A successful exploit could allow the attacker to execute arbitrary code as root, possibly resulting in the complete compromise of a targeted device.To exploit this vulnerability, the attacker must have valid user credentials on the affected device.	2026-05-06	8.8
CVE-2026-7896	google - chrome	Integer overflow in Blink in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: Critical)	2026-05-06	8.8
CVE-2026-7898	google - chrome	Use after free in Chromoting in Google Chrome on Linux prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code via malicious network traffic. (Chromium security severity: Critical)	2026-05-06	8.8
CVE-2026-7899	google - chrome	Out of bounds read and write in V8 in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7901	google - chrome	Use after free in ANGLE in Google Chrome on Mac prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7902	google - chrome	Out of bounds memory access in V8 in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7903	google - chrome	Integer overflow in ANGLE in Google Chrome on Mac,Windows prior to 148.0.7778.96 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7906	google - chrome	Use after free in SVG in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7907	google - chrome	Use after free in DOM in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7921	google - chrome	Use after free in Passwords in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7926	google - chrome	Use after free in PresentationAPI in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7927	google - chrome	Type Confusion in Runtime in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7928	google - chrome	Use after free in WebRTC in Google Chrome on Windows prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.8
CVE-2026-7930	google - chrome	Insufficient validation of untrusted input in Cookies in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform privilege escalation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7938	google - chrome	Use after free in CSS in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8

CVE-2026-7940	google - chrome	Use after free in V8 in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to execute arbitrary code inside a sandbox via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7951	google - chrome	Out of bounds write in WebRTC in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7957	google - chrome	Out of bounds write in Media in Google Chrome on Mac, iOS prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7973	google - chrome	Integer overflow in Dawn in Google Chrome on Windows prior to 148.0.7778.96 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7974	google - chrome	Use after free in Blink in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7980	google - chrome	Use after free in WebAudio in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7984	google - chrome	Use after free in ReadingMode in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7987	google - chrome	Use after free in WebRTC in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7988	google - chrome	Type Confusion in WebRTC in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7991	google - chrome	Use after free in UI in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7992	google - chrome	Insufficient validation of untrusted input in UI in Google Chrome on Linux, ChromeOS prior to 148.0.7778.96 allowed a remote attacker who convinced a user to engage in specific UI gestures to execute arbitrary code via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-7995	google - chrome	Out of bounds read in AdFilter in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.8
CVE-2026-8000	google - chrome	Insufficient validation of untrusted input in ChromeDriver in Google Chrome on Windows prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	8.8
CVE-2026-8002	google - chrome	Use after free in Audio in Google Chrome on Mac prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	8.8
CVE-2026-8016	google - chrome	Use after free in WebRTC in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	8.8
CVE-2026-5786	ivanti - multiple products	An Improper Access Control vulnerability in Ivanti EPMM before versions 12.6.1.1, 12.7.0.1, and 12.8.0.1 allows a remote authenticated attacker to gain administrative access.	2026-05-07	8.8
CVE-2026-32207	microsoft - azure_machine_learning	Improper neutralization of input during web page generation ('cross-site scripting') in Azure Machine Learning allows an unauthorized attacker to perform spoofing over a network.	2026-05-07	8.8
CVE-2026-43284	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: esp: avoid in-place decrypt on shared skb frags MSG_SPLICE_PAGES can attach pages from a pipe directly to an skb. TCP marks such skbs with SKBFL_SHARED_FRAG after skb_splice_from_iter(), so later paths that may modify packet data can first make a private copy. The IPv4/IPv6 datagram append paths did not set this flag when splicing pages into UDP skbs. That leaves an ESP-in-UDP packet made from shared pipe pages looking like an ordinary uncloned nonlinear skb. ESP input then takes the no-COW fast path for uncloned skbs without a frag_list and decrypts in place over data that is not owned privately by the skb. Mark IPv4/IPv6 datagram splice frags with SKBFL_SHARED_FRAG, matching TCP. Also make ESP input fall back to skb_cow_data() when the flag is present, so ESP does not decrypt externally backed frags in place. Private nonlinear skb frags still use the existing fast path. This intentionally does not change ESP output. In esp_output_head(), the path that appends the ESP trailer to existing skb tailroom without calling skb_cow_data() is not reachable for nonlinear skbs: skb_tailroom() returns zero when skb->data_len is nonzero, while ESP tailen is positive. Thus ESP output will either use the separate destination-frag path or fall back to skb_cow_data().	2026-05-08	8.8
CVE-2026-25077	apache - multiple products	Account users are allowed by default to register templates to be downloaded directly to the primary storage for deploying instances using the KVM hypervisor. Due to missing file name sanitization, an attacker can register malicious templates to execute arbitrary code on the KVM hosts. This can result in the compromise of resource integrity and confidentiality, data loss, denial of service, and availability of the KVM-based infrastructure managed by CloudStack.	2026-05-08	8.8

		Users are recommended to upgrade to Apache CloudStack versions 4.20.3.0 or 4.22.0.1, or later, which fixes this issue.		
CVE-2026-43322	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: Fix UAF in le_read_features_complete This fixes the following backtrace caused by hci_conn being freed before le_read_features_complete but after hci_le_read_remote_features_sync so hci_conn_del -> hci_cmd_sync_dequeue is not able to prevent it: ========== BUG: KASAN: slab-use-after-free in instrument_atomic_read_write include/linux/instrumented.h:96 [inline] BUG: KASAN: slab-use-after-free in atomic_dec_and_test include/linux/atomic/atomic-instrumented.h:1383 [inline] BUG: KASAN: slab-use-after-free in hci_conn_drop include/net/bluetooth/hci_core.h:1688 [inline] BUG: KASAN: slab-use-after-free in le_read_features_complete+0x5b/0x340 net/bluetooth/hci_sync.c:7344 Write of size 4 at addr ffff8880796b0010 by task kworker/u9:0/52 CPU: 0 UID: 0 PID: 52 Comm: kworker/u9:0 Not tainted syzkaller #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/25/2025 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xcd/0x630 mm/kasan/report.c:482 kasan_report+0xe0/0x110 mm/kasan/report.c:595 check_region_inline mm/kasan/generic.c:194 [inline] kasan_check_range+0x100/0x1b0 mm/kasan/generic.c:200 instrument_atomic_read_write include/linux/instrumented.h:96 [inline] atomic_dec_and_test include/linux/atomic/atomic-instrumented.h:1383 [inline] hci_conn_drop include/net/bluetooth/hci_core.h:1688 [inline] le_read_features_complete+0x5b/0x340 net/bluetooth/hci_sync.c:7344 hci_cmd_sync_work+0x1ff/0x430 net/bluetooth/hci_sync.c:334 process_one_work+0x9ba/0x1b20 kernel/workqueue.c:3257 process_scheduled_works kernel/workqueue.c:3340 [inline] worker_thread+0x6c8/0xf10 kernel/workqueue.c:3421 kthread+0x3c5/0x780 kernel/kthread.c:463 ret_from_fork+0x983/0xb10 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:246 </TASK> Allocated by task 5932: kasan_save_stack+0x33/0x60 mm/kasan/common.c:56 kasan_save_track+0x14/0x30 mm/kasan/common.c:77 poison_kmalloc_redzone mm/kasan/common.c:400 [inline] __kasan_kmalloc+0xaa/0xb0 mm/kasan/common.c:417 kmalloc_noprof include/linux/slab.h:957 [inline] kzalloc_noprof include/linux/slab.h:1094 [inline] __hci_conn_add+0xf8/0x1c70 net/bluetooth/hci_conn.c:963 hci_conn_add_unset+0x76/0x100 net/bluetooth/hci_conn.c:1084 le_conn_complete_evt+0x639/0x1f20 net/bluetooth/hci_event.c:5714 hci_le_enh_conn_complete_evt+0x23d/0x380 net/bluetooth/hci_event.c:5861 hci_le_meta_evt+0x357/0x5e0 net/bluetooth/hci_event.c:7408 hci_event_func net/bluetooth/hci_event.c:7716 [inline] hci_event_packet+0x685/0x11c0 net/bluetooth/hci_event.c:7773 hci_rx_work+0x2c9/0xeb0 net/bluetooth/hci_core.c:4076 process_one_work+0x9ba/0x1b20 kernel/workqueue.c:3257 process_scheduled_works kernel/workqueue.c:3340 [inline] worker_thread+0x6c8/0xf10 kernel/workqueue.c:3421 kthread+0x3c5/0x780 kernel/kthread.c:463 ret_from_fork+0x983/0xb10 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:246 Freed by task 5932: kasan_save_stack+0x33/0x60 mm/kasan/common.c:56 kasan_save_track+0x14/0x30 mm/kasan/common.c:77 __kasan_save_free_info+0x3b/0x60 mm/kasan/generic.c:587 kasan_save_free_info mm/kasan/kasan.h:406 [inline] poison_slab_object mm/kasan/common.c:252 [inline] __kasan_slab_free+0x5f/0x80 mm/kasan/common.c:284 kasan_slab_free include/linux/kasan.h:234 [inline] slab_free_hook mm/slub.c:2540 [inline] slab_free mm/slub.c:6663 [inline]	2026-05-08	8.8

		kfree+0x2f8/0x6e0 mm/slub.c:6871 device_release+0xa4/0x240 drivers/base/core.c:2565 kobject_cleanup lib/kobject.c:689 [inline] kobject_release lib/kobject.c:720 [inline] kref_put include/linux/kref.h:65 [inline] kobject_put+0x1e7/0x590 lib/kobject. ---truncated---		
CVE-2026-43334	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SMP: force responder MITM requirements before building the pairing response smp_cmd_pairing_req() currently builds the pairing response from the initiator auth_req before enforcing the local BT_SECURITY_HIGH requirement. If the initiator omits SMP_AUTH_MITM, the response can also omit it even though the local side still requires MITM. tk_request() then sees an auth value without SMP_AUTH_MITM and may select JUST_CFM, making method selection inconsistent with the pairing policy the responder already enforces. When the local side requires HIGH security, first verify that MITM can be achieved from the IO capabilities and then force SMP_AUTH_MITM in the response in both rsp.auth_req and auth. This keeps the responder auth bits and later method selection aligned.	2026-05-08	8.8
CVE-2026-43391	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nsfs: tighten permission checks for handle opening Even privileged services should not necessarily be able to see other privileged service's namespaces so they can't leak information to each other. Use may_see_all_namespaces() helper that centralizes this policy until the nstree adapts.	2026-05-08	8.8
CVE-2026-43403	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nsfs: tighten permission checks for ns iteration ioctls Even privileged services should not necessarily be able to see other privileged service's namespaces so they can't leak information to each other. Use may_see_all_namespaces() helper that centralizes this policy until the nstree adapts.	2026-05-08	8.8
CVE-2026-39849	pi-hole - ftldns	Pi-hole FTL is the core engine of the Pi-hole network-level advertisement and tracker blocker. In versions before 6.6.1, the `dns.interface` configuration field in Pi-hole FTL accepted newline characters without validation, allowing an attacker to inject arbitrary directives into the generated dnsmasq configuration file. On installations with no admin password set (the default for many deployments), the configuration API is fully accessible without credentials, allowing a network-adjacent attacker to inject the payload, enable the built-in DHCP server, and achieve arbitrary command execution on the host the next time any device on the network requests a DHCP lease. The injected value is persisted to /etc/pihole/pihole.toml and survives restarts. The strncpy in the code path limits the total interface field to 31 bytes, but payloads such as wlan0\ndhcp-script=/tmp/p fit within this constraint. The dnsmasq config validation introduced in FTL 6.6 only checks syntactic validity, so valid directives injected via newline pass validation successfully. This issue has been fixed in version 6.6.1.	2026-05-05	8.7
CVE-2026-43139	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm6: fix uninitialized saddr in xfrm6_get_saddr() xfrm6_get_saddr() does not check the return value of ipv6_dev_get_saddr(). When ipv6_dev_get_saddr() fails to find a suitable source address (returns -EADDRNOTAVAIL), saddr->in6 is left uninitialized, but xfrm6_get_saddr() still returns 0 (success). This causes the caller xfrm_tmpl_resolve_one() to use the uninitialized address in xfrm_state_find(), triggering KMSAN warning: ========== BUG: KMSAN: uninit-value in xfrm_state_find+0x2424/0xa940 xfrm_state_find+0x2424/0xa940 xfrm_resolve_and_create_bundle+0x906/0x5a20 xfrm_lookup_with_ifid+0xcc0/0x3770 xfrm_lookup_route+0x63/0x2b0 ip_route_output_flow+0x1ce/0x270 udp_sendmsg+0x2ce1/0x3400 inet_sendmsg+0x1ef/0x2a0 __sock_sendmsg+0x278/0x3d0 __sys_sendto+0x593/0x720 __x64_sys_sendto+0x130/0x200 x64_sys_call+0x332b/0x3e70 do_syscall_64+0xd3/0xf80 entry_SYSCALL_64_after_hwframe+0x77/0x7f	2026-05-06	8.6

		Local variable tmp.i.i created at: xfrm_resolve_and_create_bundle+0x3e3/0x5a20 xfrm_lookup_with_ifid+0xcc0/0x3770 ========== Fix by checking the return value of ipv6_dev_get_saddr() and propagating the error.		
CVE-2026-35435	microsoft - azure_ai_foundry	Improper access control in Azure AI Foundry M365 published agents allows an unauthorized attacker to elevate privileges over a network.	2026-05-07	8.6
CVE-2026-41705	vmware - multiple products	Spring AI's MilvusVectorStore#doDelete(List) implementation is vulnerable to filter-expression injection via unsanitized document IDs. Spring AI 1.0.x: affected from 1.0.0 through latest 1.0.x; upgrade to 1.0.7 or greater. Spring AI 1.1.x: affected from 1.1.0 through latest 1.1.x; upgrade to 1.1.6 or greater.	2026-05-09	8.6
CVE-2026-6787	watchguard - agent	Use of Hard-coded Cryptographic Key vulnerability in WatchGuard Agent on Windows allows Inclusion of Code in Existing Process.This issue affects WatchGuard Agent: before 1.25.03.0000.	2026-05-06	8.5
CVE-2026-6788	watchguard - agent	Uncontrolled Search Path Element vulnerability in WatchGuard Agent on Windows allows Using Malicious Files.This issue affects WatchGuard Agent before 1.25.03.0000.	2026-05-06	8.5
CVE-2026-43274	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mailbox: mchp-ipc-sbi: fix out-of-bounds access in mchp_ipc_get_cluster_aggr_irq() The cluster_cfg array is dynamically allocated to hold per-CPU configuration structures, with its size based on the number of online CPUs. Previously, this array was indexed using hartid, which may be non-contiguous or exceed the bounds of the array, leading to out-of-bounds access. Switch to using cpuid as the index, as it is guaranteed to be within the valid range provided by for_each_online_cpu().	2026-05-06	8.4
CVE-2026-6266	red hat - multiple products	A flaw was found in the AAP gateway. The user auto-link strategy, introduced in AAP 2.6, automatically links an external Identity Provider (IDP) identity to an existing AAP user account based on email matching without verifying email ownership. This allows a remote attacker to potentially hijack a victim's account or gain unauthorized access to other accounts, including administrative accounts, by manipulating the IDP-provided email.	2026-05-04	8.3
CVE-2026-7900	google - chrome	Heap buffer overflow in ANGLE in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7905	google - chrome	Insufficient validation of untrusted input in Media in Google Chrome on Android prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7911	google - chrome	Use after free in Aura in Google Chrome on Windows prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7914	google - chrome	Type Confusion in Accessibility in Google Chrome on Windows prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7916	google - chrome	Insufficient data validation in InterestGroups in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7917	google - chrome	Use after free in Fullscreen in Google Chrome on Windows prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7918	google - chrome	Use after free in GPU in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7919	google - chrome	Use after free in Aura in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7920	google - chrome	Use after free in Skia in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7922	google - chrome	Use after free in ServiceWorker in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)	2026-05-06	8.3
CVE-2026-7923	google - chrome	Out of bounds write in Skia in Google Chrome prior to 148.0.7778.96 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)	2026-05-06	8.3
CVE-2026-7956	google - chrome	Use after free in Navigation in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3
CVE-2026-7963	google - chrome	Inappropriate implementation in ServiceWorker in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3
CVE-2026-7967	google - chrome	Insufficient validation of untrusted input in Navigation in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3
CVE-2026-7970	google - chrome	Use after free in TopChrome in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3

CVE-2026-7975	google - chrome	Use after free in DevTools in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3
CVE-2026-7985	google - chrome	Use after free in GPU in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	8.3
CVE-2026-8001	google - chrome	Use After Free in Printing in Google Chrome on Linux, Mac, ChromeOS prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	8.3
CVE-2026-43291	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: nfc: nci: Fix parameter validation for packet data Since commit 9c328f54741b ("net: nfc: nci: Add parameter validation for packet data") communication with nci nfc chips is not working any more. The mentioned commit tries to fix access of uninitialized data, but failed to understand that in some cases the data packet is of variable length and can therefore not be compared to the maximum packet length given by the sizeof(struct).	2026-05-08	8.3
CVE-2026-43190	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: xt_tcpmss: check remaining length before reading optlen Quoting reporter: In net/netfilter/xt_tcpmss.c (lines 53-68), the TCP option parser reads op[i+1] directly without validating the remaining option length. If the last byte of the option field is not EOL/NOP (0/1), the code attempts to index op[i+1]. In the case where i + 1 == optlen, this causes an out-of-bounds read, accessing memory past the optlen boundary (either reading beyond the stack buffer _opt or the following payload).	2026-05-06	8.2

CVE-2026-43233	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		netfilter: nf_conntrack_h323: fix OOB read in decode_choice()
		
		In decode_choice(), the boundary check before get_len() uses the variable `len`, which is still 0 from its initialization at the top of the function:
		
		unsigned int type, ext, len = 0;
		...
		if (ext || (son->attr & OPEN)) {
		BYTE_ALIGN(bs);
		if (nf_h323_error_boundary(bs, len, 0)) /* len is 0 here */
		return H323_ERROR_BOUND;
		len = get_len(bs); /* OOB read */
		
		When the bitstream is exactly consumed (bs->cur == bs->end), the check nf_h323_error_boundary(bs, 0, 0) evaluates to (bs->cur + 0 > bs->end), which is false. The subsequent get_len() call then dereferences *bs->cur++, reading 1 byte past the end of the buffer. If that byte has bit 7 set, get_len() reads a second byte as well.
		
		This can be triggered remotely by sending a crafted Q.931 SETUP message with a User-User Information Element containing exactly 2 bytes of PER-encoded data ({0x08, 0x00}) to port 1720 through a firewall with the nf_conntrack_h323 helper active. The decoder fully consumes the PER buffer before reaching this code path, resulting in a 1-2 byte heap-buffer-overflow read confirmed by AddressSanitizer.
		
		Fix this by checking for 2 bytes (the maximum that get_len() may read) instead of the uninitialized `len`. This matches the pattern used at every other get_len() call site in the same file, where the caller checks for 2 bytes of available data before calling get_len().
			2026-05-06	8.2

CVE-2026-34327	microsoft - partner_center	Externally controlled reference to a resource in another sphere in Microsoft Partner Center allows an unauthorized attacker to perform spoofing over a network.	2026-05-07	8.2
CVE-2026-43365	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: fix undersized l_iclog_roundoff values If the superblock doesn't list a log stripe unit, we set the incore log roundoff value to 512. This leads to corrupt logs and unmountable filesystems in generic/617 on a disk with 4k physical sectors... XFS (sda1): Mounting V5 Filesystem ff3121ca-26e6-4b77-b742-aaff9a449e1c XFS (sda1): Torn write (CRC failure) detected at log block 0x318e. Truncating head block from 0x3197. XFS (sda1): failed to locate log tail XFS (sda1): log mount/recovery failed: error -74	2026-05-08	8.2

		XFS (sda1): log mount failed XFS (sda1): Mounting V5 Filesystem ff3121ca-26e6-4b77-b742-aaff9a449e1c XFS (sda1): Ending clean mount ...on the current xfsprogs for-next which has a broken mkfs. xfs_info shows this... meta-data=/dev/sda1 isize=512 agcount=4, agsize=644992 blks = sectsz=4096 attr=2, projid32bit=1 = crc=1 finobt=1, sparse=1, rmapbt=1 = reflink=1 bigtime=1 inobtcount=1 nrext64=1 = exchange=1 metadir=1 data = bsize=4096 blocks=2579968, imaxpct=25 = sunit=0 swidth=0 blks naming =version 2 bsize=4096 ascii-ci=0, ftype=1, parent=1 log =internal log bsize=4096 blocks=16384, version=2 = sectsz=4096 sunit=0 blks, lazy-count=1 realtime =none extsz=4096 blocks=0, rtextents=0 = rgcount=0 rgsize=268435456 extents = zoned=0 start=0 reserved=0 ...observe that the log section has sectsz=4096 sunit=0, which means that the roundoff factor is 512, not 4096 as you'd expect. We should fix mkfs not to generate broken filesystems, but anyone can fuzz the ondisk superblock so we should be more cautious. I think the inadequate logic predates commit a6a65fef5ef8d0, but that's clearly going to require a different backport.		
CVE-2026-43452	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: guard option walkers against 1-byte tail reads When the last byte of options is a non-single-byte option kind, walkers that advance with i += op[i + 1] ? : 1 can read op[i + 1] past the end of the option area. Add an explicit i == optlen - 1 check before dereferencing op[i + 1] in xt_tcpudp and xt_dccp option walkers.	2026-05-08	8.2
CVE-2026-43466	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix DMA FIFO desync on error CQE SQ recovery In case of a TX error CQE, a recovery flow is triggered, mlx5e_reset_txqsq_cc_pc() resets dma_fifo_cc to 0 but not dma_fifo_pc, desyncing the DMA FIFO producer and consumer. After recovery, the producer pushes new DMA entries at the old dma_fifo_pc, while the consumer reads from position 0. This causes us to unmap stale DMA addresses from before the recovery. The DMA FIFO is a purely software construct with no HW counterpart. At the point of reset, all WQEs have been flushed so dma_fifo_cc is already equal to dma_fifo_pc. There is no need to reset either counter, similar to how skb_fifo pc/cc are untouched. Remove the 'dma_fifo_cc = 0' reset. This fixes the following WARNING: WARNING: CPU: 0 PID: 0 at drivers/iommu/dma-iommu.c:1240 iommu_dma_unmap_page+0x79/0x90 Modules linked in: mlx5_vdpa vringh vdpa bonding mlx5_ib mlx5_vfio_pci ipip mlx5_fwctl tunnel4 mlx5_core ib_ipoib geneve ip6_gre ip_gre gre nf_tables ip6_tunnel rdma_ucm ib_uverbs ib_umad vfio_pci vfio_pci_core act_mirred act_skbedit act_vlan vhost_net vhost tap ip6table_mangle ip6table_nat ip6table_filter ip6_tables iptable_mangle cls_matchall nfnetlink_cttimeout act_gact cls_flower sch_ingress vhost_iotlb iptable_raw tunnel6 vfio_iommu_type1 vfio openvswitch nsh rpcsec_gss_krb5 auth_rpcgss oid_registry xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink iptable_nat nf_nat xt_addrtype br_netfilter overlay zram zsmalloc rpcrdma ib_iser libiscsi scsi_transport_iscsi rdma_cm iw_cm ib_cm ib_core fuse [last unloaded: nf_tables] CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.13.0-rc5_for_upstream_min_debug_2024_12_30_21_33 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:iommu_dma_unmap_page+0x79/0x90 Code: 2b 4d 3b 21 72 26 4d 3b 61 08 73 20 49 89 d8 44 89 f9 5b 4c 89 f2 4c 89 e6 48 89 ef 5d 41 5c 41 5d 41 5e 41 5f e9 c7 ae 9e ff <0f> 0b 5b 5d 41 5c 41 5d 41 5e 41 5f c3 66 2e 0f 1f 84 00 00 00 00 Call Trace: <IRQ> ? __warn+0x7d/0x110 ? iommu_dma_unmap_page+0x79/0x90 ? report_bug+0x16d/0x180 ? handle_bug+0x4f/0x90	2026-05-08	8.2

		? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? iommu_dma_unmap_page+0x79/0x90 ? iommu_dma_unmap_page+0x2e/0x90 dma_unmap_page_attrs+0x10d/0x1b0 mlx5e_tx_wi_dma_unmap+0xbe/0x120 [mlx5_core] mlx5e_poll_tx_cq+0x16d/0x690 [mlx5_core] mlx5e_napi_poll+0x8b/0xac0 [mlx5_core] __napi_poll+0x24/0x190 net_rx_action+0x32a/0x3b0 ? mlx5_eq_comp_int+0x7e/0x270 [mlx5_core] ? notifier_call_chain+0x35/0xa0 handle_softirqs+0xc9/0x270 irq_exit_rcu+0x71/0xd0 common_interrupt+0x7f/0xa0 </IRQ> <TASK> asm_common_interrupt+0x22/0x40		
CVE-2026-40563	apache - atlas	Description: Improper Control of Generation of Code ('Code Injection') vulnerability in Apache Atlas Apache Atlas exposes a DSL search endpoint that accepts user-supplied query strings. Attacker can alter Gremlin traversal logic within grammar-allowed characters to access unintended data Affect Version: This issue affects Apache Atlas: from 0.8 through 2.4.0. For the affect version >= 2.0, vulnerability is only when Atlas is deployed with below non-default configuration. atlas.dsl.executor.traversal=false Mitigation: Users are recommended to upgrade to version 2.5.0, which fixes the issue.	2026-05-04	8.1
CVE-2026-43134	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix missing key size check for L2CAP_LE_CONN_REQ This adds a check for encryption key size upon receiving L2CAP_LE_CONN_REQ which is required by L2CAP/LE/CFC/BV-15-C which expects L2CAP_CR_LE_BAD_KEY_SIZE.	2026-05-06	8.1
CVE-2026-7978	google - chrome	Inappropriate implementation in Companion in Google Chrome on Mac prior to 148.0.7778.96 allowed a remote attacker to perform OS-level privilege escalation via malicious network traffic. (Chromium security severity: Medium)	2026-05-06	8.1
CVE-2026-7981	google - chrome	Out of bounds read in Codecs in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to obtain potentially sensitive information from process memory via a malicious file. (Chromium security severity: Medium)	2026-05-06	8.1
CVE-2026-8018	google - chrome	Insufficient policy enforcement in DevTools in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to potentially perform a sandbox escape via malicious network traffic. (Chromium security severity: Low)	2026-05-06	8.1
CVE-2026-8092	mozilla - multiple products	Memory safety bugs present in Firefox ESR 115.35.1, Firefox ESR 140.10.1 and Firefox 150.0.1. 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 150.0.2, Firefox ESR 140.10.2, Firefox ESR 115.35.2, Thunderbird 150.0.2, and Thunderbird 140.10.2.	2026-05-07	8.1
CVE-2026-8093	mozilla - multiple products	Memory safety bugs present in Firefox 150.0.1. 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 150.0.2 and Thunderbird 150.0.2.	2026-05-07	8.1
CVE-2026-41105	microsoft - azure_monitor_action_group_notification_system	Server-side request forgery (ssrf) in Azure Notification Service allows an authorized attacker to elevate privileges over a network.	2026-05-07	8.1
CVE-2025-66172	apache - cloudstack	The CloudStack Backup plugin has an improper access logic in versions 4.21.0.0 and 4.22.0.0. Anyone with authenticated user-account access in CloudStack 4.21.0.0+ environments, where this plugin is enabled and have access to specific APIs can restore a volume from any other user's backups and attach the volume to their own VMs. Backup plugin users using CloudStack 4.21.0.0+ are recommended to upgrade to CloudStack version 4.22.0.1, which fixes this issue.	2026-05-08	8.1
CVE-2026-43362	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix in-place encryption corruption in SMB2_write()	2026-05-08	8.1

		SMB2_write() places write payload in iov[1..n] as part of rq_iov. smb3_init_transform_rq() pointer-shares rq_iov, so crypt_message() encrypts iov[1] in-place, replacing the original plaintext with ciphertext. On a replayable error, the retry sends the same iov[1] which now contains ciphertext instead of the original data, resulting in corruption. The corruption is most likely to be observed when connections are unstable, as reconnects trigger write retries that re-send the already-encrypted data. This affects SFU mknod, MF symlinks, etc. On kernels before 6.10 (prior to the netfs conversion), sync writes also used this path and were similarly affected. The async write path wasn't unaffected as it uses rq_iter which gets deep-copied. Fix by moving the write payload into rq_iter via iov_iter_kvec(), so smb3_init_transform_rq() deep-copies it before encryption.		
CVE-2026-43377	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Don't log keys in SMB3 signing and encryption key generation When KSMBD_DEBUG_AUTH logging is enabled, generate_smb3signingkey() and generate_smb3encryptionkey() log the session, signing, encryption, and decryption key bytes. Remove the logs to avoid exposing credentials.	2026-05-08	8.1
CVE-2025-66467	apache - multiple products	Missing MinIO policy cleanup on bucket deletion via Apache CloudStack allows users to retain access to buckets which they previously owned. If another user creates a new bucket with the same name, the previous owners can gain unauthorized read and write access to it by using the previously generated access and secret keys. Users are recommended to upgrade to Apache CloudStack versions 4.20.3.0 or 4.22.0.1, or later, which fixes this issue.	2026-05-08	8.0
CVE-2026-43133	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: nSVM: Always use vmcb01 in VMLOAD/VMSAVE emulation Commit cc3ed80ae69f ("KVM: nSVM: always use vmcb01 to for vmsave/vmload of guest state") made KVM always use vmcb01 for the fields controlled by VMSAVE/VMLOAD, but it missed updating the VMLOAD/VMSAVE emulation code to always use vmcb01. As a result, if VMSAVE/VMLOAD is executed by an L2 guest and is not intercepted by L1, KVM will mistakenly use vmcb02. Always use vmcb01 instead of the current VMCB.	2026-05-06	7.9
CVE-2025-47405	qualcomm - fastconnect_6900_firmware	Memory corruption when processing camera sensor input/output control codes with invalid output buffers.	2026-05-04	7.8
CVE-2025-47407	qualcomm - cq7790_firmware	Memory corruption while creating a process on the digital signal processor due to allocation failure at the kernel level.	2026-05-04	7.8
CVE-2025-47408	qualcomm - fastconnect_6200_firmware	Memory corruption when another driver calls an IOCTL with invalid input/output buffer.	2026-05-04	7.8
CVE-2026-24082	qualcomm - qxm1096_firmware	Memory Corruption when copying data from a freed source while executing performance counter deselect operation.	2026-05-04	7.8
CVE-2026-43060	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_ct: drop pending enqueued packets on removal Packets sitting in nfqueue might hold a reference to: - templates that specify the conntrack zone, because a percpu area is used and module removal is possible. - conntrack timeout policies and helper, where object removal leave a stale reference. Since these objects can just go away, drop enqueued packets to avoid stale reference to them. If there is a need for finer grain removal, this logic can be revisited to make selective packet drop upon dependencies.	2026-05-05	7.8
CVE-2026-43063	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: don't irele after failing to iget in xfs_attri_recover_work xlog_recovery_iget* never set @ip to a valid pointer if they return an error, so this irele will walk off a dangling pointer. Fix that.	2026-05-05	7.8
CVE-2026-43070	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-05	7.8

		bpf: Reset register ID for BPF_END value tracking When a register undergoes a BPF_END (byte swap) operation, its scalar value is mutated in-place. If this register previously shared a scalar ID with another register (e.g., after an `r1 = r0` assignment), this tie must be broken. Currently, the verifier misses resetting `dst_reg->id` to 0 for BPF_END. Consequently, if a conditional jump checks the swapped register, the verifier incorrectly propagates the learned bounds to the linked register, leading to false confidence in the linked register's value and potentially allowing out-of-bounds memory accesses. Fix this by explicitly resetting `dst_reg->id` to 0 in the BPF_END case to break the scalar tie, similar to how BPF_NEG handles it via `__mark_reg_known`.		
CVE-2026-43074	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: eventpoll: defer struct eventpoll free to RCU grace period In certain situations, ep_free() in eventpoll.c will kfree the epi->ep eventpoll struct while it still being used by another concurrent thread. Defer the kfree() to an RCU callback to prevent UAF.	2026-05-06	7.8
CVE-2026-43075	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix out-of-bounds write in ocfs2_write_end_inline KASAN reports a use-after-free write of 4086 bytes in ocfs2_write_end_inline, called from ocfs2_write_end_nolock during a copy_file_range splice fallback on a corrupted ocfs2 filesystem mounted on a loop device. The actual bug is an out-of-bounds write past the inode block buffer, not a true use-after-free. The write overflows into an adjacent freed page, which KASAN reports as UAF. The root cause is that ocfs2_try_to_write_inline_data trusts the on-disk id_count field to determine whether a write fits in inline data. On a corrupted filesystem, id_count can exceed the physical maximum inline data capacity, causing writes to overflow the inode block buffer. Call trace (crash path): vfs_copy_file_range (fs/read_write.c:1634) do_splice_direct splice_direct_to_actor iter_file_splice_write ocfs2_file_write_iter generic_perform_write ocfs2_write_end ocfs2_write_end_nolock (fs/ocfs2/aops.c:1949) ocfs2_write_end_inline (fs/ocfs2/aops.c:1915) memcpy_from_folio <-- KASAN: write OOB So add id_count upper bound check in ocfs2_validate_inode_block() to alongside the existing i_size check to fix it.	2026-05-06	7.8
CVE-2026-43076	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate inline data i_size during inode read When reading an inode from disk, ocfs2_validate_inode_block() performs various sanity checks but does not validate the size of inline data. If the filesystem is corrupted, an inode's i_size can exceed the actual inline data capacity (id_count). This causes ocfs2_dir_foreach_blk_id() to iterate beyond the inline data buffer, triggering a use-after-free when accessing directory entries from freed memory. In the syzbot report: - i_size was 1099511627576 bytes (~1TB) - Actual inline data capacity (id_count) is typically <256 bytes - A garbage rec_len (54648) caused ctx->pos to jump out of bounds - This triggered a UAF in ocfs2_check_dir_entry() Fix by adding a validation check in ocfs2_validate_inode_block() to ensure inodes with inline data have i_size <= id_count. This catches the corruption early during inode read and prevents all downstream code from operating on invalid data.	2026-05-06	7.8
CVE-2026-43078	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: af_alg - Fix page reassignment overflow in af_alg_pull_tsgl	2026-05-06	7.8

		When page reassignment was added to af_alg_pull_tsgl the original loop wasn't updated so it may try to reassign one more page than necessary. Add the check to the reassignment so that this does not happen. Also update the comment which still refers to the obsolete offset argument.		
CVE-2026-43084	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nfnetlink_queue: make hash table per queue Sharing a global hash table among all queues is tempting, but it can cause crash: BUG: KASAN: slab-use-after-free in nfqnl_recv_verdict+0x11ac/0x15e0 [nfnetlink_queue] [..] nfqnl_recv_verdict+0x11ac/0x15e0 [nfnetlink_queue] nfnetlink_rcv_msg+0x46a/0x930 kmem_cache_alloc_node_noprof+0x11e/0x450 struct nf_queue_entry is freed via kfree, but parallel cpu can still encounter such an nf_queue_entry when walking the list. Alternative fix is to free the nf_queue_entry via kfree_rcu() instead, but as we have to alloc/free for each skb this will cause more mem pressure.	2026-05-06	7.8
CVE-2026-43091	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: Wait for RCU readers during policy netns exit xfrm_policy_fini() frees the policy_bydst hash tables after flushing the policy work items and deleting all policies, but it does not wait for concurrent RCU readers to leave their read-side critical sections first. The policy_bydst tables are published via rcu_assign_pointer() and are looked up through rcu_dereference_check(), so netns teardown must also wait for an RCU grace period before freeing the table memory. Fix this by adding synchronize_rcu() before freeing the policy hash tables.	2026-05-06	7.8
CVE-2026-43093	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xsk: tighten UMEM headroom validation to account for tailroom and min frame The current headroom validation in xdp_umem_reg() could leave us with insufficient space dedicated to even receive minimum-sized ethernet frame. Furthermore if multi-buffer would come to play then skb_shared_info stored at the end of XSK frame would be corrupted. HW typically works with 128-aligned sizes so let us provide this value as bare minimum. Multi-buffer setting is known later in the configuration process so besides accounting for 128 bytes, let us also take care of tailroom space upfront.	2026-05-06	7.8
CVE-2026-43097	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PCI: hv: Fix double ida_free in hv_pci_probe error path If hv_pci_probe() fails after storing the domain number in hbus->bridge->domain_nr, there is a call to free this domain_nr via pci_bus_release_emul_domain_nr(), however, during cleanup, the bridge release callback pci_release_host_bridge_dev() also frees the domain_nr causing ida_free to be called on same ID twice and triggering following warning: ida_free called for id=28971 which is not allocated. WARNING: lib/idr.c:594 at ida_free+0xdf/0x160, CPU#0: kworker/0:2/198 Call Trace: pci_bus_release_emul_domain_nr+0x17/0x20 pci_release_host_bridge_dev+0x4b/0x60 device_release+0x3b/0xa0 kobject_put+0x8e/0x220 devm_pci_alloc_host_bridge_release+0xe/0x20 devres_release_all+0x9a/0xd0 device_unbind_cleanup+0x12/0xa0 really_probe+0x1c5/0x3f0 vmbus_add_channel_work+0x135/0x1a0	2026-05-06	7.8

		Fix this by letting pci core handle the free domain_nr and remove the explicit free called in pci-hyperv driver.		
CVE-2026-43106	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cachefiles: fix incorrect dentry refcount in cachefiles_cull() The patch mentioned below changed cachefiles_bury_object() to expect 2 references to the 'rep' dentry. Three of the callers were changed to use start_removing_dentry() which takes an extra reference so in those cases the call gets the expected references. However there is another call to cachefiles_bury_object() in cachefiles_cull() which did not need to be changed to use start_removing_dentry() and so was not properly considered. It still passed the dentry with just one reference so the net result is that a reference is lost. To meet the expectations of cachefiles_bury_object(), cachefiles_cull() must take an extra reference before the call. It will be dropped by cachefiles_bury_object().	2026-05-06	7.8
CVE-2026-43111	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: roccat: fix use-after-free in roccat_report_event roccat_report_event() iterates over the device->readers list without holding the readers_lock. This allows a concurrent roccat_release() to remove and free a reader while it's still being accessed, leading to a use-after-free. Protect the readers list traversal with the readers_lock mutex.	2026-05-06	7.8
CVE-2026-43116	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: ensure safe access to master conntrack Holding reference on the expectation is not sufficient, the master conntrack object can just go away, making exp->master invalid. To access exp->master safely: - Grab the nf_conntrack_expect_lock, this gets serialized with clean_from_lists() which also holds this lock when the master conntrack goes away. - Hold reference on master conntrack via nf_conntrack_find_get(). Not so easy since the master tuple to look up for the master conntrack is not available in the existing problematic paths. This patch goes for extending the nf_conntrack_expect_lock section to address this issue for simplicity, in the cases that are described below this is just slightly extending the lock section. The add expectation command already holds a reference to the master conntrack from ctnetlink_create_expect(). However, the delete expectation command needs to grab the spinlock before looking up for the expectation. Expand the existing spinlock section to address this to cover the expectation lookup. Note that, the nf_ct_expect_iterate_net() calls already grabs the spinlock while iterating over the expectation table, which is correct. The get expectation command needs to grab the spinlock to ensure master conntrack does not go away. This also expands the existing spinlock section to cover the expectation lookup too. I needed to move the netlink skb allocation out of the spinlock to keep it GFP_KERNEL. For the expectation events, the IPEXP_DESTROY event is already delivered under the spinlock, just move the delivery of IPEXP_NEW under the spinlock too because the master conntrack event cache is reached through exp->master. While at it, add lockdep notations to help identify what codepaths need to grab the spinlock.	2026-05-06	7.8
CVE-2026-43120	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix double free related to rereg_user_mr If IB_MR_REREG_TRANS is set during rereg_user_mr, the umem will be released and a new one will be allocated in irdma_rereg_mr_trans. If any step of irdma_rereg_mr_trans fails after the new umem is allocated, it releases the umem,	2026-05-06	7.8

		but does not set iwmr->region to NULL. The problem is that this failure is propagated to the user, who will then call ibv_dereg_mr (as they should). Then, the dereg_mr path will see a non-NULL umem and attempt to call ib_umem_release again. Fix this by setting iwmr->region to NULL after ib_umem_release. Fixed: 5ac388db27c4 ("RDMA/irdma: Add support to re-register a memory region")		
CVE-2026-43126	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: mixer: oss: Add card disconnect checkpoints ALSA OSS mixer layer calls the kcontrol ops rather individually, and pending calls might be not always caught at disconnecting the device. For avoiding the potential UAF scenarios, add sanity checks of the card disconnection at each entry point of OSS mixer accesses. The rwsem is taken just before that check, hence the rest context should be covered by that properly.	2026-05-06	7.8
CVE-2026-43128	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/umem: Fix double dma_buf_unpin in failure path In ib_umem_dmabuf_get_pinned_with_dma_device(), the call to ib_umem_dmabuf_map_pages() can fail. If this occurs, the dmabuf is immediately unpinned but the umem_dmabuf->pinned flag is still set. Then, when ib_umem_release() is called, it calls ib_umem_dmabuf_revoke() which will call dma_buf_unpin() again. Fix this by removing the immediate unpin upon failure and just let the ib_umem_release/revoke path handle it. This also ensures the proper unmap-unpin unwind ordering if the dmabuf_map_pages call happened to fail due to dma_resv_wait_timeout (and therefore has a non-NULL umem_dmabuf->sgt).	2026-05-06	7.8
CVE-2026-43138	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: reset: gpio: suppress bind attributes in sysfs This is a special device that's created dynamically and is supposed to stay in memory forever. We also currently don't have a devlink between it and the actual reset consumer. Suppress sysfs bind attributes so that user-space can't unbind the device because - as of now - it will cause a use-after-free splat from any user that puts the reset control handle.	2026-05-06	7.8
CVE-2026-43150	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: perf/arm-cmn: Reject unsupported hardware configurations So far we've been fairly lax about accepting both unknown CMN models (at least with a warning), and unknown revisions of those which we do know, as although things do frequently change between releases, typically enough remains the same to be somewhat useful for at least some basic bringup checks. However, we also make assumptions of the maximum supported sizes and numbers of things in various places, and there's no guarantee that something new might not be bigger and lead to nasty array overflows. Make sure we only try to run on things that actually match our assumptions and so will not risk memory corruption. We have at least always failed on completely unknown node types, so update that error message for clarity and consistency too.	2026-05-06	7.8
CVE-2026-43153	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfs: remove xfs_attr_leaf_hasname The calling convention of xfs_attr_leaf_hasname() is problematic, because it returns a NULL buffer when xfs_attr3_leaf_read fails, a valid buffer when xfs_attr3_leaf_lookup_int returns -ENOATTR or -EEXIST, and a non-NULL buffer pointer for an already released buffer when xfs_attr3_leaf_lookup_int fails with other error values. Fix this by simply open coding xfs_attr_leaf_hasname in the callers, so that the buffer release code is done by each caller of xfs_attr3_leaf_read.	2026-05-06	7.8
CVE-2026-43178	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: procfs: fix possible double mmput() in do_procmap_query() When user provides incorrectly sized buffer for build ID for PROCMAP_QUERY we return with -ENAMETOOLONG error. After recent changes this condition happens later, after we unlocked mmap_lock/per-VMA lock and did mmput(),	2026-05-06	7.8

		so original goto out is now wrong and will double-mmput() mm_struct. Fix by jumping further to clean up only vm_file and name_buf.		

CVE-2026-43180	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net: usb: kaweth: remove TX queue manipulation in kaweth_set_rx_mode
		
		kaweth_set_rx_mode(), the ndo_set_rx_mode callback, calls netif_stop_queue() and netif_wake_queue(). These are TX queue flow control functions unrelated to RX multicast configuration.
		
		The premature netif_wake_queue() can re-enable TX while tx_urb is still in-flight, leading to a double usb_submit_urb() on the same URB:
		
		kaweth_start_xmit() {
		  netif_stop_queue();
		  usb_submit_urb(kaweth->tx_urb);
		}
		
		kaweth_set_rx_mode() {
		  netif_stop_queue();
		  netif_wake_queue(); // wakes TX queue before URB is done
		}
		
		kaweth_start_xmit() {
		  netif_stop_queue();
		  usb_submit_urb(kaweth->tx_urb); // URB submitted while active
		}
		
		This triggers the WARN in usb_submit_urb():
		
		"URB submitted while active"
		
		This is a similar class of bug fixed in rtl8150 by
		
		- commit 958baf5eaee3 ("net: usb: Remove disruptive netif_wake_queue in rtl8150_set_multicast").
		
		Also kaweth_set_rx_mode() is already functionally broken, the real set_rx_mode action is performed by kaweth_async_set_rx_mode(), which in turn is not a no-op only at ndo_open() time.
			2026-05-06	7.8

CVE-2026-43196	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: soc: ti: pruss: Fix double free in pruss_clk_mux_setup() In the pruss_clk_mux_setup(), the devm_add_action_or_reset() indirectly calls pruss_of_free_clk_provider(), which calls of_node_put(clk_mux_np) on the error path. However, after the devm_add_action_or_reset() returns, the of_node_put(clk_mux_np) is called again, causing a double free. Fix by returning directly, to avoid the duplicate of_node_put().	2026-05-06	7.8

CVE-2026-43205	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		dpaa2-switch: validate num_ifs to prevent out-of-bounds write
		
		The driver obtains sw_attr.num_ifs from firmware via dpsw_get_attributes() but never validates it against DPSW_MAX_IF (64). This value controls iteration in dpaa2_switch_fdb_get_flood_cfg(), which writes port indices into the fixed-size cfg->if_id[DPSW_MAX_IF] array. When firmware reports num_ifs >= 64, the loop can write past the array bounds.
		
		Add a bound check for num_ifs in dpaa2_switch_init().
		
		dpaa2_switch_fdb_get_flood_cfg() appends the control interface (port num_ifs) after all matched ports. When num_ifs == DPSW_MAX_IF and all ports match the flood filter, the loop fills all 64 slots and the control interface write overflows by one entry.
		
		The check uses >= because num_ifs == DPSW_MAX_IF is also functionally broken.
		
		build_if_id_bitmap() silently drops any ID >= 64:
		  if (id[i] < DPSW_MAX_IF)
		  bmap[id[i] / 64] |= ...
			2026-05-06	7.8

CVE-2026-43206	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix out-of-bounds write in kfd_event_page_set() The kfd_event_page_set() function writes KFD_SIGNAL_EVENT_LIMIT * 8 bytes via memset without checking the buffer size parameter. This allows unprivileged userspace to trigger an out-of bounds kernel memory write	2026-05-06	7.8

		by passing a small buffer, leading to potential privilege escalation.		
CVE-2026-43207	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: mtk-mdp: Fix error handling in probe function Add mtk_mdp_unregister_m2m_device() on the error handling path to prevent resource leak. Add check for the return value of vpu_get_plat_device() to prevent null pointer dereference. And vpu_get_plat_device() increases the reference count of the returned platform device. Add platform_device_put() to prevent reference leak.	2026-05-06	7.8

CVE-2026-43211	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		PCI: Fix pci_slot_trylock() error handling
		
		Commit a4e772898f8b ("PCI: Add missing bridge lock to pci_bus_lock()") delegates the bridge device's pci_dev_trylock() to pci_bus_trylock() in pci_slot_trylock(), but it forgets to remove the corresponding pci_dev_unlock() when pci_bus_trylock() fails.
		
		Before a4e772898f8b, the code did:
		
		if (!pci_dev_trylock(dev)) /* <- lock bridge device */
		goto unlock;
		if (dev->subordinate) {
		if (!pci_bus_trylock(dev->subordinate)) {
		pci_dev_unlock(dev); /* <- unlock bridge device */
		goto unlock;
		}
		}
		
		After a4e772898f8b the bridge-device lock is no longer taken, but the pci_dev_unlock(dev) on the failure path was left in place, leading to the bug.
		
		This yields one of two errors:
		
		1. A warning that the lock is being unlocked when no one holds it.
		2. An incorrect unlock of a lock that belongs to another thread.
		
		Fix it by removing the now-redundant pci_dev_unlock(dev) on the failure path.
		
		[Same patch later posted by Keith at https://patch.msgid.link/20260116184150.3013258-1-kbusch@meta.com]
			2026-05-06	7.8

CVE-2026-43212	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Make cpumask_of_node() robust against NUMA_NO_NODE The arch definition of cpumask_of_node() cannot handle NUMA_NO_NODE - which is a valid index - so add a check for this.	2026-05-06	7.8
CVE-2026-43214	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Add SRCU protection for reading PDPTRs in __get_sregs2() Add SRCU read-side protection when reading PDPTR registers in __get_sregs2(). Reading PDPTRs may trigger access to guest memory: kvm_pdptr_read() -> svm_cache_reg() -> load_pdptrs() -> kvm_vcpu_read_guest_page() -> kvm_vcpu_gfn_to_memslot() kvm_vcpu_gfn_to_memslot() dereferences memslots via __kvm_memslots(), which uses srcu_dereference_check() and requires either kvm->srcu or kvm->slots_lock to be held. Currently only vcpu->mutex is held, triggering lockdep warning: ============================= WARNING: suspicious RCU usage in kvm_vcpu_gfn_to_memslot 6.12.59+ #3 Not tainted include/linux/kvm_host.h:1062 suspicious rcu_dereference_check() usage! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by syz.5.1717/15100: #0: ff1100002f4b00b0 (&vcpu->mutex){+.+.}-{3:3}, at: kvm_vcpu_ioctl+0x1d5/0x1590	2026-05-06	7.8

		Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0xf0/0x120 lib/dump_stack.c:120 lockdep_rcu_suspicious+0x1e3/0x270 kernel/locking/lockdep.c:6824 __kvm_memslots include/linux/kvm_host.h:1062 [inline] __kvm_memslots include/linux/kvm_host.h:1059 [inline] kvm_vcpu_memslots include/linux/kvm_host.h:1076 [inline] kvm_vcpu_gfn_to_memslot+0x518/0x5e0 virt/kvm/kvm_main.c:2617 kvm_vcpu_read_guest_page+0x27/0x50 virt/kvm/kvm_main.c:3302 load_pdptrs+0xff/0x4b0 arch/x86/kvm/x86.c:1065 svm_cache_reg+0x1c9/0x230 arch/x86/kvm/svm/svm.c:1688 kvm_pdptr_read arch/x86/kvm/kvm_cache_regs.h:141 [inline] __get_sregs2 arch/x86/kvm/x86.c:11784 [inline] kvm_arch_vcpu_ioctl+0x3e20/0x4aa0 arch/x86/kvm/x86.c:6279 kvm_vcpu_ioctl+0x856/0x1590 virt/kvm/kvm_main.c:4663 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x18b/0x210 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xbd/0x1d0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Found by Linux Verification Center (linuxtesting.org) with Syzkaller.		
CVE-2026-43222	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: verisilicon: AV1: Fix tile info buffer size Each tile info is composed of: row_sb, col_sb, start_pos and end_pos (4 bytes each). So the total required memory is AV1_MAX_TILES * 16 bytes. Use the correct #define to allocate the buffer and avoid writing tile info in non-allocated memory.	2026-05-06	7.8
CVE-2026-43236	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/atmel-hlcdc: fix use-after-free of drm_crtc_commit after release The atmel_hlcdc_plane_atomic_duplicate_state() callback was copying the atmel_hlcdc_plane state structure without properly duplicating the drm_plane_state. In particular, state->commit remained set to the old state commit, which can lead to a use-after-free in the next drm_atomic_commit() call. Fix this by calling __drm_atomic_helper_duplicate_plane_state(), which correctly clones the base drm_plane_state (including the ->commit pointer). It has been seen when closing and re-opening the device node while another DRM client (e.g. fbdev) is still attached: ========== BUG kmalloc-64 (Not tainted): Poison overwritten ---------- 0xc611b344-0xc611b344 @offset=836. First byte 0x6a instead of 0x6b FIX kmalloc-64: Restoring Poison 0xc611b344-0xc611b344=0x6b Allocated in drm_atomic_helper_setup_commit+0x1e8/0x7bc age=178 cpu=0 pid=29 drm_atomic_helper_setup_commit+0x1e8/0x7bc drm_atomic_helper_commit+0x3c/0x15c drm_atomic_commit+0xc0/0xf4 drm_framebuffer_remove+0x4cc/0x5a8 drm_mode_rmfb_work_fn+0x6c/0x80 process_one_work+0x12c/0x2cc worker_thread+0x2a8/0x400 kthread+0xc0/0xdc ret_from_fork+0x14/0x28 Freed in drm_atomic_helper_commit_hw_done+0x100/0x150 age=8 cpu=0 pid=169 drm_atomic_helper_commit_hw_done+0x100/0x150 drm_atomic_helper_commit_tail+0x64/0x8c commit_tail+0x168/0x18c drm_atomic_helper_commit+0x138/0x15c drm_atomic_commit+0xc0/0xf4 drm_atomic_helper_set_config+0x84/0xb8 drm_mode_setcrtc+0x32c/0x810 drm_ioctl+0x20c/0x488 sys_ioctl+0x14c/0xc20	2026-05-06	7.8

		ret_fast_syscall+0x0/0x54 Slab 0xef8bc360 objects=21 used=16 fp=0xc611b7c0 flags=0x200(workingset|zone=0) Object 0xc611b340 @offset=832 fp=0xc611b7c0		
CVE-2026-43237	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Refactor amdgpu_gem_va_ioctl for Handling Last Fence Update and Timeline Management v4 This commit simplifies the amdgpu_gem_va_ioctl function, key updates include: - Moved the logic for managing the last update fence directly into amdgpu_gem_va_update_vm. - Introduced checks for the timeline point to enable conditional replacement or addition of fences. v2: Addressed review comments from Christian. v3: Updated comments (Christian). v4: The previous version selected the fence too early and did not manage its reference correctly, which could lead to stale or freed fences being used. This resulted in refcount underflows and could crash when updating GPU timelines. The fence is now chosen only after the VA mapping work is completed, and its reference is taken safely. After exporting it to the VM timeline syncobj, the driver always drops its local fence reference, ensuring balanced refcounting and avoiding use-after-free on dma_fence. Crash signature: [205.828135] refcount_t: underflow; use-after-free. [205.832963] WARNING: CPU: 30 PID: 7274 at lib/refcount.c:28 refcount_warn_saturate+0xbe/0x110 ... [206.074014] Call Trace: [206.076488] <TASK> [206.078608] amdgpu_gem_va_ioctl+0x6ea/0x740 [amdgpu] [206.084040] ? __pfx_amdgpu_gem_va_ioctl+0x10/0x10 [amdgpu] [206.089994] drm_ioctl_kernel+0x86/0xe0 [drm] [206.094415] drm_ioctl+0x26e/0x520 [drm] [206.098424] ? __pfx_amdgpu_gem_va_ioctl+0x10/0x10 [amdgpu] [206.104402] amdgpu_drm_ioctl+0x4b/0x80 [amdgpu] [206.109387] __x64_sys_ioctl+0x96/0xe0 [206.113156] do_syscall_64+0x66/0x2d0 ... [206.553351] BUG: unable to handle page fault for address: ffffffffc0dfde90 ... [206.553378] RIP: 0010:dma_fence_signal_timestamp_locked+0x39/0xe0 ... [206.553405] Call Trace: [206.553409] <IRQ> [206.553415] ? __pfx_drm_sched_fence_free_rcu+0x10/0x10 [gpu_sched] [206.553424] dma_fence_signal+0x30/0x60 [206.553427] drm_sched_job_done.isra.0+0x123/0x150 [gpu_sched] [206.553434] dma_fence_signal_timestamp_locked+0x6e/0xe0 [206.553437] dma_fence_signal+0x30/0x60 [206.553441] amdgpu_fence_process+0xd8/0x150 [amdgpu] [206.553854] sdma_v4_0_process_trap_irq+0x97/0xb0 [amdgpu] [206.554353] edac_mce_amd(E) ee1004(E) [206.554270] amdgpu_irq_dispatch+0x150/0x230 [amdgpu] [206.554702] amdgpu_ih_process+0x6a/0x180 [amdgpu] [206.555101] amdgpu_irq_handler+0x23/0x60 [amdgpu] [206.555500] __handle_irq_event_percpu+0x4a/0x1c0 [206.555506] handle_irq_event+0x38/0x80 [206.555509] handle_edge_irq+0x92/0x1e0 [206.555513] __common_interrupt+0x3e/0xb0 [206.555519] common_interrupt+0x80/0xa0 [206.555525] </IRQ> [206.555527] <TASK> ... [206.555650] RIP: 0010:dma_fence_signal_timestamp_locked+0x39/0xe0 ... [206.555667] Kernel panic - not syncing: Fatal exception in interrupt	2026-05-06	7.8
CVE-2026-43248	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vhost: move vdpa group bound check to vhost_vdpa Remove duplication by consolidating these here. This reduces the posibility of a parent driver missing them. While we're at it, fix a bug in vdpa_sim where a valid ASID can be assigned to a group equal to ngroups, causing an out of bound write.	2026-05-06	7.8

CVE-2026-43250	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: udc: fix DMA and SG cleanup in _ep_nuke() The ChipIdea UDC driver can encounter "not page aligned sg buffer" errors when a USB device is reconnected after being disconnected during an active transfer. This occurs because _ep_nuke() returns requests to the gadget layer without properly unmapping DMA buffers or cleaning up scatter-gather bounce buffers. Root cause: When a disconnect happens during a multi-segment DMA transfer, the request's num_mapped_sgs field and sgt.sgl pointer remain set with stale values. The request is returned to the gadget driver with status -ESHUTDOWN but still has active DMA state. If the gadget driver reuses this request on reconnect without reinitializing it, the stale DMA state causes _hardware_enqueue() to skip DMA mapping (seeing non-zero num_mapped_sgs) and attempt to use freed/invalid DMA addresses, leading to alignment errors and potential memory corruption. The normal completion path via _hardware_dequeue() properly calls usb_gadget_unmap_request_by_dev() and sglist_do_debounce() before returning the request. The _ep_nuke() path must do the same cleanup to ensure requests are returned in a clean, reusable state. Fix: Add DMA unmapping and bounce buffer cleanup to _ep_nuke() to mirror the cleanup sequence in _hardware_dequeue(): - Call usb_gadget_unmap_request_by_dev() if num_mapped_sgs is set - Call sglist_do_debounce() with copy=false if bounce buffer exists This ensures that when requests are returned due to endpoint shutdown, they don't retain stale DMA mappings. The 'false' parameter to sglist_do_debounce() prevents copying data back (appropriate for shutdown path where transfer was aborted).	2026-05-06	7.8
CVE-2026-43256	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: qcom: camss: vfe: Fix out-of-bounds access in vfe_isr_reg_update() vfe_isr() iterates using MSM_VFE_IMAGE_MASTERS_NUM(7) as the loop bound and passes the index to vfe_isr_reg_update(). However, vfe->line[] array is defined with VFE_LINE_NUM_MAX(4): struct vfe_line line[VFE_LINE_NUM_MAX]; When index is 4, 5, 6, the access to vfe->line[line_id] exceeds the array bounds and resulting in out-of-bounds memory access. Fix this by using separate loops for output lines and write masters.	2026-05-06	7.8
CVE-2026-43258	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: alpha: fix user-space corruption during memory compaction Alpha systems can suffer sporadic user-space crashes and heap corruption when memory compaction is enabled. Symptoms include SIGSEGV, glibc allocator failures (e.g. "unaligned tcache chunk"), and compiler internal errors. The failures disappear when compaction is disabled or when using global TLB invalidation. The root cause is insufficient TLB shootdown during page migration. Alpha relies on ASN-based MM context rollover for instruction cache coherency, but this alone is not sufficient to prevent stale data or instruction translations from surviving migration. Fix this by introducing a migration-specific helper that combines: - MM context invalidation (ASN rollover), - immediate per-CPU TLB invalidation (TBI), - synchronous cross-CPU shootdown when required. The helper is used only by migration/compaction paths to avoid changing global TLB semantics. Additionally, update flush_tlb_other(), pte_clear(), to use READ_ONCE()/WRITE_ONCE() for correct SMP memory ordering. This fixes observed crashes on both UP and SMP Alpha systems.	2026-05-06	7.8
CVE-2026-43260	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix RSS context delete logic	2026-05-06	7.8

		We need to free the corresponding RSS context VNIC in FW everytime an RSS context is deleted in driver. Commit 667ac333dbb7 added a check to delete the VNIC in FW only when netif_running() is true to help delete RSS contexts with interface down. Having that condition will make the driver leak VNICs in FW whenever close() happens with active RSS contexts. On the subsequent open(), as part of RSS context restoration, we will end up trying to create extra VNICs for which we did not make any reservation. FW can fail this request, thereby making us lose active RSS contexts. Suppose an RSS context is deleted already and we try to process a delete request again, then the HWRM functions will check for validity of the request and they simply return if the resource is already freed. So, even for delete-when-down cases, netif_running() check is not necessary. Remove the netif_running() condition check when deleting an RSS context.		
CVE-2026-43263	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Fix Null reference while testing fluster When multi instances are created/destroyed, many interrupts happens and structures for decoder are removed. "struct vpu_instance" this structure is shared for all flow in the decoder, so if the structure is not protected by lock, Null dereference could happens sometimes. IRQ Handler was spilt to two phases and Lock was added as well.	2026-05-06	7.8
CVE-2026-43276	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix double destroy_workqueue on service rescan PCI path While testing corner cases in the driver, a use-after-free crash was found on the service rescan PCI path. When mana_serv_reset() calls mana_gd_suspend(), mana_gd_cleanup() destroys gc->service_wq. If the subsequent mana_gd_resume() fails with -ETIMEDOUT or -EPROTO, the code falls through to mana_serv_rescan() which triggers pci_stop_and_remove_bus_device(). This invokes the PCI .remove callback (mana_gd_remove), which calls mana_gd_cleanup() a second time, attempting to destroy the already-freed workqueue. Fix this by NULL-checking gc->service_wq in mana_gd_cleanup() and setting it to NULL after destruction. Call stack of issue for reference: [Sat Feb 21 18:53:48 2026] Call Trace: [Sat Feb 21 18:53:48 2026] <TASK> [Sat Feb 21 18:53:48 2026] mana_gd_cleanup+0x33/0x70 [mana] [Sat Feb 21 18:53:48 2026] mana_gd_remove+0x3a/0xc0 [mana] [Sat Feb 21 18:53:48 2026] pci_device_remove+0x41/0xb0 [Sat Feb 21 18:53:48 2026] device_remove+0x46/0x70 [Sat Feb 21 18:53:48 2026] device_release_driver_internal+0x1e3/0x250 [Sat Feb 21 18:53:48 2026] device_release_driver+0x12/0x20 [Sat Feb 21 18:53:48 2026] pci_stop_bus_device+0x6a/0x90 [Sat Feb 21 18:53:48 2026] pci_stop_and_remove_bus_device+0x13/0x30 [Sat Feb 21 18:53:48 2026] mana_do_service+0x180/0x290 [mana] [Sat Feb 21 18:53:48 2026] mana_serv_func+0x24/0x50 [mana] [Sat Feb 21 18:53:48 2026] process_one_work+0x190/0x3d0 [Sat Feb 21 18:53:48 2026] worker_thread+0x16e/0x2e0 [Sat Feb 21 18:53:48 2026] kthread+0xf7/0x130 [Sat Feb 21 18:53:48 2026] ? __pfx_worker_thread+0x10/0x10 [Sat Feb 21 18:53:48 2026] ? __pfx_kthread+0x10/0x10 [Sat Feb 21 18:53:48 2026] ret_from_fork+0x269/0x350 [Sat Feb 21 18:53:48 2026] ? __pfx_kthread+0x10/0x10 [Sat Feb 21 18:53:48 2026] ret_from_fork_asm+0x1a/0x30 [Sat Feb 21 18:53:48 2026] </TASK>	2026-05-06	7.8
CVE-2026-43278	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm: clear cloned request bio pointer when last clone bio completes Stale rq->bio values have been observed to cause double-initialization of cloned bios in request-based device-mapper targets, leading to use-after-free and double-free scenarios. One such case occurs when using dm-multipath on top of a PCIe NVMe	2026-05-06	7.8

		namespace, where cloned request bios are freed during blk_complete_request(), but rq->bio is left intact. Subsequent clone teardown then attempts to free the same bios again via blk_rq_unprep_clone().
		
		The resulting double-free path looks like:
		
		nvme_pci_complete_batch()
		nvme_complete_batch()
		blk_mq_end_request_batch()
		blk_complete_request() // called on a DM clone request
		bio_endio() // first free of all clone bios
		...
		rq->end_io() // end_clone_request()
		dm_complete_request(tio->orig)
		dm_softirq_done()
		dm_done()
		dm_end_request()
		blk_rq_unprep_clone() // second free of clone bios
		
		Fix this by clearing the clone request's bio pointer when the last cloned bio completes, ensuring that later teardown paths do not attempt to free already-released bios.		

CVE-2026-43279	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Add sanity check for OOB writes at silencing At silencing the playback URB packets in the implicit fb mode before the actual playback, we blindly assume that the received packets fit with the buffer size. But when the setup in the capture stream differs from the playback stream (e.g. due to the USB core limitation of max packet size), such an inconsistency may lead to OOB writes to the buffer, resulting in a crash. For addressing it, add a sanity check of the transfer buffer size at prepare_silent_urb(), and stop the data copy if the received data overflows. Also, report back the transfer error properly from there, too. Note that this doesn't fix the root cause of the playback error itself, but this merely covers the kernel Oops.	2026-05-06	7.8
CVE-2026-7913	google - chrome	Insufficient policy enforcement in DevTools in Google Chrome on Android prior to 148.0.7778.96 allowed a local attacker to perform privilege escalation via a malicious file. (Chromium security severity: High)	2026-05-06	7.8
CVE-2026-7925	google - chrome	Use after free in Chromoting in Google Chrome on Windows prior to 148.0.7778.96 allowed a local attacker to perform OS-level privilege escalation via a malicious file. (Chromium security severity: High)	2026-05-06	7.8
CVE-2026-7990	google - chrome	Insufficient validation of untrusted input in Updater in Google Chrome on Windows prior to 148.0.7778.96 allowed a local attacker to perform OS-level privilege escalation via a malicious file. (Chromium security severity: Medium)	2026-05-06	7.8
CVE-2026-7994	google - chrome	Inappropriate implementation in Chromoting in Google Chrome on Windows prior to 148.0.7778.96 allowed a local attacker to perform OS-level privilege escalation via a malicious file. (Chromium security severity: Medium)	2026-05-06	7.8
CVE-2026-7997	google - chrome	Insufficient validation of untrusted input in Updater in Google Chrome on Mac prior to 148.0.7778.96 allowed a local attacker to perform OS-level privilege escalation via a malicious file. (Chromium security severity: Low)	2026-05-06	7.8
CVE-2026-43290	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Return queued buffers on start_streaming() failure Return buffers if streaming fails to start due to uvc_pm_get() error. This bug may be responsible for a warning I got running while :; do yavta -c3 /dev/video0; done on an xHCI controller which failed under this workload. I had no luck reproducing this warning again to confirm. xhci_hcd 0000:09:00.0: HC died; cleaning up usb 13-2: USB disconnect, device number 2 WARNING: CPU: 2 PID: 29386 at drivers/media/common/videobuf2/videobuf2-core.c:1803 vb2_start_streaming+0xac/0x120	2026-05-08	7.8
CVE-2026-43303	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/page_alloc: clear page->private in free_pages_prepare() Several subsystems (slub, shmem, ttm, etc.) use page->private but don't clear it before freeing pages. When these pages are later allocated as high-order pages and split via split_page(), tail pages retain stale	2026-05-08	7.8

		page->private values. This causes a use-after-free in the swap subsystem. The swap code uses page->private to track swap count continuations, assuming freshly allocated pages have page->private == 0. When stale values are present, swap_count_continued() incorrectly assumes the continuation list is valid and iterates over uninitialized page->lru containing LIST_POISON values, causing a crash: KASAN: maybe wild-memory-access in range [0xdead000000000100-0xdead000000000107] RIP: 0010:__do_sys_swapoff+0x1151/0x1860 Fix this by clearing page->private in free_pages_prepare(), ensuring all freed pages have clean state regardless of previous use.		
CVE-2026-43307	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: accel: adxl380: Avoid reading more entries than present in FIFO The interrupt handler reads FIFO entries in batches of N samples, where N is the number of scan elements that have been enabled. However, the sensor fills the FIFO one sample at a time, even when more than one channel is enabled. Therefore,the number of entries reported by the FIFO status registers may not be a multiple of N; if this number is not a multiple, the number of entries read from the FIFO may exceed the number of entries actually present. To fix the above issue, round down the number of FIFO entries read from the status registers so that it is always a multiple of N.	2026-05-08	7.8
CVE-2026-43321	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Properly mark live registers for indirect jumps For a `gotox rX` instruction the rX register should be marked as used in the compute_insn_live_regs() function. Fix this.	2026-05-08	7.8
CVE-2026-43324	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: USB: dummy-hcd: Fix interrupt synchronization error This fixes an error in synchronization in the dummy-hcd driver. The error has a somewhat involved history. The synchronization mechanism was introduced by commit 7dbd8f4cabd9 ("USB: dummy-hcd: Fix erroneous synchronization change"), which added an emulated "interrupts enabled" flag together with code emulating synchronize_irq() (it waits until all current handler callbacks have returned). But the emulated interrupt-disable occurred too late, after the driver containing the handler callback routines had been told that it was unbound and no more callbacks would occur. Commit 4a5d797a9f9c ("usb: gadget: dummy_hcd: fix gpf in gadget_setup") tried to fix this by moving the synchronize_irq() emulation code from dummy_stop() to dummy_pullup(), which runs before the unbind callback. There still were races, though, because the emulated interrupt-disable still occurred too late. It couldn't be moved to dummy_pullup(), because that routine can be called for reasons other than an impending unbind. Therefore commits 7dc0c55e9f30 ("USB: UDC core: Add udc_async_callbacks gadget op") and 04145a03db9d ("USB: UDC: Implement udc_async_callbacks in dummy-hcd") added an API allowing the UDC core to tell dummy-hcd exactly when emulated interrupts and their callbacks should be disabled. That brings us to the current state of things, which is still wrong because the emulated synchronize_irq() occurs before the emulated interrupt-disable! That's no good, beause it means that more emulated interrupts can occur after the synchronize_irq() emulation has run, leading to the possibility that a callback handler may be running when the gadget driver is unbound. To fix this, we have to move the synchronize_irq() emulation code yet again, to the dummy_udc_async_callbacks() routine, which takes care of enabling and disabling emulated interrupt requests. The synchronization will now run immediately after emulated interrupts are disabled, which is where it belongs.	2026-05-08	7.8
CVE-2026-43328	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cpufreq: governor: fix double free in cpufreq_dbs_governor_init() error path When kobject_init_and_add() fails, cpufreq_dbs_governor_init() calls kobject_put(&dbs_data->attr_set.kobj).	2026-05-08	7.8

		The kobject release callback cpufreq_dbs_data_release() calls gov->exit(dbs_data) and kfree(dbs_data), but the current error path then calls gov->exit(dbs_data) and kfree(dbs_data) again, causing a double free. Keep the direct kfree(dbs_data) for the gov->init() failure path, but after kobject_init_and_add() has been called, let kobject_put() handle the cleanup through cpufreq_dbs_data_release().		
CVE-2026-43329	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: strictly check for maximum number of actions The maximum number of flowtable hardware offload actions in IPv6 is: * ethernet mangling (4 payload actions, 2 for each ethernet address) * SNAT (4 payload actions) * DNAT (4 payload actions) * Double VLAN (4 vlan actions, 2 for popping vlan, and 2 for pushing QinQ. * Redirect (1 action) Which makes 17, while the maximum is 16. But act_ct supports for tunnels actions too. Note that payload action operates at 32-bit word level, so mangling an IPv6 address takes 4 payload actions. Update flow_action_entry_next() calls to check for the maximum number of supported actions. While at it, rise the maximum number of actions per flow from 16 to 24 so this works fine with IPv6 setups.	2026-05-08	7.8
CVE-2026-43330	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: caam - fix overflow 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 copying is performed using kmemdup, however this leads to an overflow: reading more bytes (aligned_len - keylen) from the keylen source buffer. Fix this by replacing kmemdup with kmalloc, followed by memcpy.	2026-05-08	7.8
CVE-2026-43332	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Fix thermal zone device registration error path If thermal_zone_device_register_with_trips() fails after registering a thermal zone device, it needs to wait for the tz->removal completion like thermal_zone_device_unregister(), in case user space has managed to take a reference to the thermal zone device's kobject, in which case thermal_release() may not be called by the error path itself and tz may be freed prematurely. Add the missing wait_for_completion() call to the thermal zone device registration error path.	2026-05-08	7.8
CVE-2026-43339	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent possible UaF in addrconf_permanent_addr() The mentioned helper try to warn the user about an exceptional condition, but the message is delivered too late, accessing the ipv6 after its possible deletion. Reorder the statement to avoid the possible UaF; while at it, place the warning outside the idev->lock as it needs no protection.	2026-05-08	7.8
CVE-2026-43352	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: i3c: mipi-i3c-hci: Correct RING_CTRL_ABORT handling in DMA dequeue The logic used to abort the DMA ring contains several flaws: 1. The driver unconditionally issues a ring abort even when the ring has already stopped. 2. The completion used to wait for abort completion is never re-initialized, resulting in incorrect wait behavior. 3. The abort sequence unintentionally clears RING_CTRL_ENABLE, which resets hardware ring pointers and disrupts the controller state. 4. If the ring is already stopped, the abort operation should be considered successful without attempting further action.	2026-05-08	7.8

		Fix the abort handling by checking whether the ring is running before issuing an abort, re-initializing the completion when needed, ensuring that RING_CTRL_ENABLE remains asserted during abort, and treating an already stopped ring as a successful condition.		
CVE-2026-43353	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: i3c: mipi-i3c-hci: Fix race in DMA ring dequeue The HCI DMA dequeue path (hci_dma_dequeue_xfer()) may be invoked for multiple transfers that timeout around the same time. However, the function is not serialized and can race with itself. When a timeout occurs, hci_dma_dequeue_xfer() stops the ring, processes incomplete transfers, and then restarts the ring. If another timeout triggers a parallel call into the same function, the two instances may interfere with each other - stopping or restarting the ring at unexpected times. Add a mutex so that hci_dma_dequeue_xfer() is serialized with respect to itself.	2026-05-08	7.8
CVE-2026-43366	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring/kbuf: check if target buffer list is still legacy on recycle There's a gap between when the buffer was grabbed and when it potentially gets recycled, where if the list is empty, someone could've upgraded it to a ring provided type. This can happen if the request is forced via io-wq. The legacy recycling is missing checking if the buffer_list still exists, and if it's of the correct type. Add those checks.	2026-05-08	7.8
CVE-2026-43368	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix potential overflow of shmem scatterlist length When a scatterlists table of a GEM shmem object of size 4 GB or more is populated with pages allocated from a folio, unsigned int .length attribute of a scatterlist may get overflowed if total byte length of pages allocated to that single scatterlist happens to reach or cross the 4GB limit. As a consequence, users of the object may suffer from hitting unexpected, premature end of the object's backing pages. [278.780187] ------------[cut here]------------ [278.780377] WARNING: CPU: 1 PID: 2326 at drivers/gpu/drm/i915/i915_mm.c:55 remap_sg+0x199/0x1d0 [i915] ... [278.780654] CPU: 1 UID: 0 PID: 2326 Comm: gem_mmap_offset Tainted: G S U 6.17.0-rc1-CI_DRM_16981-ged823aaa0607+ #1 PREEMPT(voluntary) [278.780656] Tainted: [S]=CPU_OUT_OF_SPEC, [U]=USER [278.780658] Hardware name: Intel Corporation Meteor Lake Client Platform/MTL-P LP5x T3 RVP, BIOS MTLPFWI1.R00.3471.D91.2401310918 01/31/2024 [278.780659] RIP: 0010:remap_sg+0x199/0x1d0 [i915] ... [278.780786] Call Trace: [278.780787] <TASK> [278.780788] ? __apply_to_page_range+0x3e6/0x910 [278.780795] ? __pfx_remap_sg+0x10/0x10 [i915] [278.780906] apply_to_page_range+0x14/0x30 [278.780908] remap_io_sg+0x14d/0x260 [i915] [278.781013] vm_fault_cpu+0xd2/0x330 [i915] [278.781137] __do_fault+0x3a/0x1b0 [278.781140] do_fault+0x322/0x640 [278.781143] __handle_mm_fault+0x938/0xfd0 [278.781150] handle_mm_fault+0x12c/0x300 [278.781152] ? lock_mm_and_find_vma+0x4b/0x760 [278.781155] do_user_addr_fault+0x2d6/0x8e0 [278.781160] exc_page_fault+0x96/0x2c0 [278.781165] asm_exc_page_fault+0x27/0x30 ... That issue was apprehended by the author of a change that introduced it, and potential risk even annotated with a comment, but then never addressed. When adding folio pages to a scatterlist table, take care of byte length of any single scatterlist not exceeding max_segment. (cherry picked from commit 06249b4e691a75694c014a61708c007fb5755f60)	2026-05-08	7.8
CVE-2026-43370	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix use-after-free race in VM acquire	2026-05-08	7.8

		Replace non-atomic vm->process_info assignment with cmpxchg() to prevent race when parent/child processes sharing a drm_file both try to acquire the same VM after fork(). (cherry picked from commit c7c573275ec20db05be769288a3e3bb2250ec618)		
CVE-2026-43374	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: nexthop: fix percpu use-after-free in remove_nh_grp_entry When removing a nexthop from a group, remove_nh_grp_entry() publishes the new group via rcu_assign_pointer() then immediately frees the removed entry's percpu stats with free_percpu(). However, the synchronize_net() grace period in the caller remove_nexthop_from_groups() runs after the free. RCU readers that entered before the publish still see the old group and can dereference the freed stats via nh_grp_entry_stats_inc() -> get_cpu_ptr(nhge->stats), causing a use-after-free on percpu memory. Fix by deferring the free_percpu() until after synchronize_net() in the caller. Removed entries are chained via nh_list onto a local deferred free list. After the grace period completes and all RCU readers have finished, the percpu stats are safely freed.	2026-05-08	7.8
CVE-2026-43408	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ceph: add a bunch of missing ceph_path_info initializers ceph_mdsc_build_path() must be called with a zero-initialized ceph_path_info parameter, or else the following ceph_mdsc_free_path_info() may crash. Example crash (on Linux 6.18.12): virt_to_cache: Object is not a Slab page! WARNING: CPU: 184 PID: 2871736 at mm/slub.c:6732 kmem_cache_free+0x316/0x400 [...] Call Trace: [...] ceph_open+0x13d/0x3e0 do_dentry_open+0x134/0x480 vfs_open+0x2a/0xe0 path_openat+0x9a3/0x1160 [...] cache_from_obj: Wrong slab cache. names_cache but object is from ceph_inode_info WARNING: CPU: 184 PID: 2871736 at mm/slub.c:6746 kmem_cache_free+0x2dd/0x400 [...] kernel BUG at mm/slub.c:634! Oops: invalid opcode: 0000 [#1] SMP NOPTI RIP: 0010:__slab_free+0x1a4/0x350 Some of the ceph_mdsc_build_path() callers had initializers, but others had not, even though they were all added by commit 15f519e9f883 ("ceph: fix race condition validating r_parent before applying state"). The ones without initializer are suspectible to random crashes. (I can imagine it could even be possible to exploit this bug to elevate privileges.) Unfortunately, these Ceph functions are undocumented and its semantics can only be derived from the code. I see that ceph_mdsc_build_path() initializes the structure only on success, but not on error. Calling ceph_mdsc_free_path_info() after a failed ceph_mdsc_build_path() call does not even make sense, but that's what all callers do, and for it to be safe, the structure must be zero-initialized. The least intrusive approach to fix this is therefore to add initializers everywhere.	2026-05-08	7.8
CVE-2026-43433	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rust_binder: avoid reading the written value in offsets array When sending a transaction, its offsets array is first copied into the target proc's vma, and then the values are read back from there. This is normally fine because the vma is a read-only mapping, so the target process cannot change the value under us. However, if the target process somehow gains the ability to write to its own vma, it could change the offset before it's read back, causing the kernel to misinterpret what the sender meant. If the sender happens to send a payload with a specific shape, this could in the worst case lead to the receiver being able to privilege escalate into the sender.	2026-05-08	7.8

		The intent is that gaining the ability to change the read-only vma of your own process should not be exploitable, so remove this TOCTOU read even though it's unexploitable without another Binder bug.		
CVE-2026-43434	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rust_binder: check ownership before using vma When installing missing pages (or zapping them), Rust Binder will look up the vma in the mm by address, and then call vm_insert_page (or zap_page_range_single). However, if the vma is closed and replaced with a different vma at the same address, this can lead to Rust Binder installing pages into the wrong vma. By installing the page into a writable vma, it becomes possible to write to your own binder pages, which are normally read-only. Although you're not supposed to be able to write to those pages, the intent behind the design of Rust Binder is that even if you get that ability, it should not lead to anything bad. Unfortunately, due to another bug, that is not the case. To fix this, store a pointer in vm_private_data and check that the vma returned by vma_lookup() has the right vm_ops and vm_private_data before trying to use the vma. This should ensure that Rust Binder will refuse to interact with any other VMA. The plan is to introduce more vma abstractions to avoid this unsafe access to vm_ops and vm_private_data, but for now let's start with the simplest possible fix. C Binder performs the same check in a slightly different way: it provides a vm_ops->close that sets a boolean to true, then checks that boolean after calling vma_lookup(), but this is more fragile than the solution in this patch. (We probably still want to do both, but the vm_ops->close callback will be added later as part of the follow-up vma API changes.) It's still possible to remap the vma so that pages appear in the right vma, but at the wrong offset, but this is a separate issue and will be fixed when Rust Binder gets a vm_ops->close callback.	2026-05-08	7.8
CVE-2026-43437	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: fix use-after-free on linked stream runtime in snd_pcm_drain() In the drain loop, the local variable 'runtime' is reassigned to a linked stream's runtime (runtime = s->runtime at line 2157). After releasing the stream lock at line 2169, the code accesses runtime->no_period_wakeup, runtime->rate, and runtime->buffer_size (lines 2170-2178) — all referencing the linked stream's runtime without any lock or refcount protecting its lifetime. A concurrent close() on the linked stream's fd triggers snd_pcm_release_substream() → snd_pcm_drop() → pcm_release_private() → snd_pcm_unlink() → snd_pcm_detach_substream() → kfree(runtime). No synchronization prevents kfree(runtime) from completing while the drain path dereferences the stale pointer. Fix by caching the needed runtime fields (no_period_wakeup, rate, buffer_size) into local variables while still holding the stream lock, and using the cached values after the lock is released.	2026-05-08	7.8
CVE-2026-43438	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sched_ext: Remove redundant css_put() in scx_cgroup_init() The iterator css_for_each_descendant_pre() walks the cgroup hierarchy under cgroup_lock(). It does not increment the reference counts on yielded css structs. According to the cgroup documentation, css_put() should only be used to release a reference obtained via css_get() or css_tryget_online(). Since the iterator does not use either of these to acquire a reference, calling css_put() in the error path of scx_cgroup_init() causes a refcount underflow. Remove the unbalanced css_put() to prevent a potential Use-After-Free (UAF) vulnerability.	2026-05-08	7.8
CVE-2026-43447	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iavf: fix PTP use-after-free during reset Commit 7c01dbfc8a1c5f ("iavf: periodically cache PHC time") introduced a worker to cache PHC time, but failed to stop it during reset or disable.	2026-05-08	7.8

		This creates a race condition where `iavf_reset_task()` or `iavf_disable_vf()` free adapter resources (AQ) while the worker is still running. If the worker triggers `iavf_queue_ptp_cmd()` during teardown, it accesses freed memory/locks, leading to a crash. Fix this by calling `iavf_ptp_release()` before tearing down the adapter. This ensures `ptp_clock_unregister()` synchronously cancels the worker and cleans up the chardev before the backing resources are destroyed.		
CVE-2026-43454	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: Fix for duplicate device in netdev hooks When handling NETDEV_REGISTER notification, duplicate device registration must be avoided since the device may have been added by nft_netdev_hook_alloc() already when creating the hook.	2026-05-08	7.8
CVE-2026-43456	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bonding: fix type confusion in bond_setup_by_slave() kernel BUG at net/core/skbuff.c:2306! Oops: invalid opcode: 0000 [#1] SMP KASAN NOPTI RIP: 0010:pskb_expand_head+0xa08/0xfe0 net/core/skbuff.c:2306 RSP: 0018:ffffc90004aff760 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffff88807e3c8780 RCX: ffffffff89593e0e RDX: ffff88807b7c4900 RSI: ffffffff89594747 RDI: ffff88807b7c4900 RBP: 0000000000000820 R08: 0000000000000005 R09: 0000000000000000 R10: 00000000961a63e0 R11: 0000000000000000 R12: ffff88807e3c8780 R13: 00000000961a6560 R14: dffffc0000000000 R15: 00000000961a63e0 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fe1a0ed8df0 CR3: 000000002d816000 CR4: 00000000003526f0 Call Trace: <TASK> ipgre_header+0xdd/0x540 net/ipv4/ip_gre.c:900 dev_hard_header include/linux/netdevice.h:3439 [inline] packet_snd net/packet/af_packet.c:3028 [inline] packet_sendmsg+0x3ae5/0x53c0 net/packet/af_packet.c:3108 sock_sendmsg_nosec net/socket.c:727 [inline] __sock_sendmsg net/socket.c:742 [inline] ____sys_sendmsg+0xa54/0xc30 net/socket.c:2592 ___sys_sendmsg+0x190/0x1e0 net/socket.c:2646 __sys_sendmsg+0x170/0x220 net/socket.c:2678 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x106/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fe1a0e6c1a9 When a non-Ethernet device (e.g. GRE tunnel) is enslaved to a bond, bond_setup_by_slave() directly copies the slave's header_ops to the bond device: bond_dev->header_ops = slave_dev->header_ops; This causes a type confusion when dev_hard_header() is later called on the bond device. Functions like ipgre_header(), ip6gre_header(),all use netdev_priv(dev) to access their device-specific private data. When called with the bond device, netdev_priv() returns the bond's private data (struct bonding) instead of the expected type (e.g. struct ip_tunnel), leading to garbage values being read and kernel crashes. Fix this by introducing bond_header_ops with wrapper functions that delegate to the active slave's header_ops using the slave's own device. This ensures netdev_priv() in the slave's header functions always receives the correct device. The fix is placed in the bonding driver rather than individual device drivers, as the root cause is bond blindly inheriting header_ops from the slave without considering that these callbacks expect a specific netdev_priv() layout. The type confusion can be observed by adding a printk in ipgre_header() and running the following commands: ip link add dummy0 type dummy ip addr add 10.0.0.1/24 dev dummy0 ip link set dummy0 up ip link add gre1 type gre local 10.0.0.1 ip link add bond1 type bond mode active-backup ip link set gre1 master bond1 ip link set gre1 up	2026-05-08	7.8

		ip link set bond1 up ip addr add fe80::1/64 dev bond1		
CVE-2026-43461	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: spi: amlogic: spifc-a4: Fix DMA mapping error handling Fix three bugs in aml_sfc_dma_buffer_setup() error paths: 1. Unnecessary goto: When the first DMA mapping (sfc->daddr) fails, nothing needs cleanup. Use direct return instead of goto. 2. Double-unmap bug: When info DMA mapping failed, the code would unmap sfc->daddr inline, then fall through to out_map_data which would unmap it again, causing a double-unmap. 3. Wrong unmap size: The out_map_info label used datalen instead of infolen when unmapping sfc->iaddr, which could lead to incorrect DMA sync behavior.	2026-05-08	7.8
CVE-2026-20167	cisco - Cisco IoT Field Network Director (IoT-FND)	A vulnerability in the web-based management interface of Cisco IoT Field Network Director could allow an authenticated, remote attacker with low privileges to cause a DoS condition on a remotely managed router. This vulnerability is due to improper error handling. An attacker could exploit this vulnerability by submitting crafted input to the web-based management interface. A successful exploit could allow the attacker to request unauthorized files from a remote router, causing the router to reload and resulting in a DoS condition.	2026-05-06	7.7
CVE-2026-20185	cisco - Cisco Small Business Smart and Managed Switches	A vulnerability in the Simple Network Management Protocol (SNMP) subsystem ofCisco 350 Series Managed Switches (SG350) and Cisco 350X Series Stackable Managed Switches (SG350X)firmware could allow an authenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to improper error handling when parsing response data for a specific SNMP request. An attacker could exploit this vulnerability by sending a specific SNMP request to an affected device. A successful exploit could allow the attacker to cause the device to reload unexpectedly, resulting in a DoS condition. This vulnerability affects SNMP versions 1, 2c, and 3. To exploit this vulnerability through SNMPv2c or earlier, the attacker must know a valid read-write or read-only SNMP community string for the affected system. To exploit this vulnerability through SNMPv3, the attacker must have valid SNMP user credentials for the affected system.	2026-05-06	7.7
CVE-2026-43350	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: require a full NFS mode SID before reading mode bits parse_dacl() treats an ACE SID matching sid_unix_NFS_mode as an NFS mode SID and reads sid.sub_auth[2] to recover the mode bits. That assumes the ACE carries three subauthorities, but compare_sids() only compares min(a, b) subauthorities. A malicious server can return an ACE with num_subauth = 2 and sub_auth[] = {88, 3}, which still matches sid_unix_NFS_mode and then drives the sub_auth[2] read four bytes past the end of the ACE. Require num_subauth >= 3 before treating the ACE as an NFS mode SID. This keeps the fix local to the special-SID mode path without changing compare_sids() semantics for the rest of cifsacl.	2026-05-08	7.6
CVE-2026-33846	red hat - multiple products	A heap buffer overflow vulnerability exists in the DTLS handshake fragment reassembly logic of GnuTLS. The issue arises in merge_handshake_packet() where incoming handshake fragments are matched and merged based solely on handshake type, without validating that the message_length field remains consistent across all fragments of the same logical message. An attacker can exploit this by sending crafted DTLS fragments with conflicting message_length values, causing the implementation to allocate a buffer based on a smaller initial fragment and subsequently write beyond its bounds using larger, inconsistent fragments. Because the merge operation does not enforce proper bounds checking against the allocated buffer size, this results in an out-of-bounds write on the heap. The vulnerability is remotely exploitable without authentication via the DTLS handshake path and can lead to application crashes or potential memory corruption.	2026-05-04	7.5
CVE-2026-34059	apache - http_server	Buffer Over-read vulnerability in Apache HTTP Server. This issue affects Apache HTTP Server: through 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-04	7.5
CVE-2026-29169	apache - http_server	A NULL pointer dereference in mod_dav_lock in Apache HTTP Server 2.4.66 and earlier may allow an attacker to crash the server with a malicious request.mod_dav_lock is not used internally by mod_dav or mod_dav_fs. The only known use-case for mod_dav_lock was mod_dav_svn from Apache Subversion earlier than version 1.2.0. Users are recommended to upgrade to version 2.4.66, which fixes this issue, or remove mod_dav_lock.	2026-05-04	7.5
CVE-2026-42440	apache - multiple products	OOM Denial of Service via Unbounded Array Allocation in Apache OpenNLP AbstractModelReader Versions Affected:	2026-05-04	7.5

		before 2.5.9 before 3.0.0-M3 Description: The AbstractModelReader methods getOutcomes(), getOutcomePatterns(), and getPredicates() each read a 32-bit signed integer count field from a binary model stream and pass that value directly to an array allocation (new String[numOutcomes], new int[numOCTypes][], new String[NUM_PREDS]) without validating that the value is non-negative or within a reasonable bound. The count is therefore fully attacker-controlled when the model file originates from an untrusted source. A crafted .bin model file in which any of these count fields is set to Integer.MAX_VALUE (or any value large enough to exhaust the available heap) triggers an OutOfMemoryError at the array allocation itself, before the corresponding label or pattern data is consumed from the stream. The error occurs very early in deserialization: for a GIS model, getOutcomes() is reached after only the model-type string, the correction constant, and the correction parameter have been read; so the attacker pays no meaningful size cost to weaponize a payload, and a single small file can crash a JVM that loads it. Any code path that deserializes a .bin model is affected, including direct use of GenericModelReader and any higher-level component that delegates to it during model load. The practical impact is denial of service against processes that load model files from untrusted or semi-trusted origins. Mitigation: * 2.x users should upgrade to 2.5.9. * 3.x users should upgrade to 3.0.0-M3. Note: The fix introduces an upper bound on each of the three count fields, checked before array allocation; counts that are negative or exceed the bound cause an IllegalArgumentException to be thrown and the read to fail fast with no large allocation. The default bound is 10,000,000, which is well above the entry counts of legitimate OpenNLP models but far below any value that would threaten heap exhaustion. Deployments that legitimately need to load models with more entries than the default can raise the limit at JVM startup by setting the OPENNLP_MAX_ENTRIES system property to the desired positive integer (e.g. -DOPENNLP_MAX_ENTRIES=50000000); invalid or non-positive values fall back to the default. Users who cannot upgrade immediately should treat all .bin model files as untrusted input unless their provenance is verified, and should avoid loading models supplied by end users or fetched from third-party repositories without integrity checks.		
CVE-2025-71251	google - multiple products	In IMS, there is a possible system crash due to improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2025-71252	google - multiple products	In Modem IMS, there is a possible improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2025-71253	google - multiple products	In Modem IMS, there is a possible improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2025-71254	google - multiple products	In Modem IMS, there is a possible improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2025-71255	google - multiple products	In Modem IMS, there is a possible improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2025-71256	google - multiple products	In nr modem, there is a possible improper input validation. This could lead to remote denial of service with no additional execution privileges needed.	2026-05-06	7.5
CVE-2026-43099	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv4: icmp: fix null-ptr-deref in icmp_build_probe() ipv6_stub->ipv6_dev_find() may return ERR_PTR(-EAFNOSUPPORT) when the IPv6 stack is not active (CONFIG_IPV6=m and not loaded), and passing this error pointer to dev_hold() will cause a kernel crash with null-ptr-deref. Instead, silently discard the request. RFC 8335 does not appear to define a specific response for the case where an IPv6 interface identifier is syntactically valid but the implementation cannot perform the lookup at runtime, and silently dropping the request may safer than misreporting "No Such Interface".	2026-05-06	7.5
CVE-2026-43101	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-06	7.5

		ipv6: ioam: fix potential NULL dereferences in __ioam6_fill_trace_data() We need to check __in6_dev_get() for possible NULL value, as suggested by Yiming Qian. Also add skb_dst_dev_rcu() instead of skb_dst_dev(), and two missing READ_ONCE(). Note that @dev can't be NULL.		
CVE-2026-43646	apache - multiple products	Exposure of Sensitive Information to an Unauthorized Actor vulnerability in Apache Wicket. This issue affects Apache Wicket: from 8.0.0 through 8.17.0, from 9.0.0 through 9.22.0, from 10.0.0 through 10.8.0. Users are recommended to upgrade to version 10.9.0, which fixes the issue.	2026-05-06	7.5
CVE-2026-43164	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: udplite: Fix null-ptr-deref in __udp_enqueue_schedule_skb(). syzbot reported null-ptr-deref of udp_sk(sk)->udp_prod_queue. [0] Since the cited commit, udp_lib_init_sock() can fail, as can udp_init_sock() and udpv6_init_sock(). Let's handle the error in udplite_sk_init() and udplitev6_sk_init(). [0]: BUG: KASAN: null-ptr-deref in instrument_atomic_read include/linux/instrumented.h:82 [inline] BUG: KASAN: null-ptr-deref in atomic_read include/linux/atomic/atomic-instrumented.h:32 [inline] BUG: KASAN: null-ptr-deref in __udp_enqueue_schedule_skb+0x151/0x1480 net/ipv4/udp.c:1719 Read of size 4 at addr 0000000000000008 by task syz.2.18/2944 CPU: 1 UID: 0 PID: 2944 Comm: syz.2.18 Not tainted syzkaller #0 PREEMPTLAZY Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/25/2025 Call Trace: <IRQ> dump_stack_lvl+0xe8/0x150 lib/dump_stack.c:120 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_atomic_read include/linux/instrumented.h:82 [inline] atomic_read include/linux/atomic/atomic-instrumented.h:32 [inline] __udp_enqueue_schedule_skb+0x151/0x1480 net/ipv4/udp.c:1719 __udpv6_queue_rcv_skb net/ipv6/udp.c:795 [inline] udpv6_queue_rcv_one_skb+0xa2e/0x1ad0 net/ipv6/udp.c:906 udp6_unicast_rcv_skb+0x227/0x380 net/ipv6/udp.c:1064 ip6_protocol_deliver_rcu+0xe17/0x1540 net/ipv6/ip6_input.c:438 ip6_input_finish+0x191/0x350 net/ipv6/ip6_input.c:489 NF_HOOK+0x354/0x3f0 include/linux/netfilter.h:318 ip6_input+0x16c/0x2b0 net/ipv6/ip6_input.c:500 NF_HOOK+0x354/0x3f0 include/linux/netfilter.h:318 __netif_receive_skb_one_core net/core/dev.c:6149 [inline] __netif_receive_skb+0xd3/0x370 net/core/dev.c:6262 process_backlog+0x4d6/0x1160 net/core/dev.c:6614 __napi_poll+0xae/0x320 net/core/dev.c:7678 napi_poll net/core/dev.c:7741 [inline] net_rx_action+0x60d/0xdc0 net/core/dev.c:7893 handle_softirqs+0x209/0x8d0 kernel/softirq.c:622 do_softirq+0x52/0x90 kernel/softirq.c:523 </IRQ> <TASK> __local_bh_enable_ip+0xe7/0x120 kernel/softirq.c:450 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:924 [inline] __dev_queue_xmit+0x109c/0x2dc0 net/core/dev.c:4856 __ip6_finish_output net/ipv6/ip6_output.c:-1 [inline] ip6_finish_output+0x158/0x4e0 net/ipv6/ip6_output.c:219 NF_HOOK_COND include/linux/netfilter.h:307 [inline] ip6_output+0x342/0x580 net/ipv6/ip6_output.c:246 ip6_send_skb+0x1d7/0x3c0 net/ipv6/ip6_output.c:1984 udp_v6_send_skb+0x9a5/0x1770 net/ipv6/udp.c:1442 udp_v6_push_pending_frames+0xa2/0x140 net/ipv6/udp.c:1469 udpv6_sendmsg+0xfe0/0x2830 net/ipv6/udp.c:1759 sock_sendmsg_nosec net/socket.c:727 [inline] __sock_sendmsg+0xe5/0x270 net/socket.c:742 __sys_sendto+0x3eb/0x580 net/socket.c:2206 __do_sys_sendto net/socket.c:2213 [inline] __se_sys_sendto net/socket.c:2209 [inline] __x64_sys_sendto+0xde/0x100 net/socket.c:2209 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline]	2026-05-06	7.5

		do_syscall_64+0xd2/0xf20 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f67b4d9c629 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:00007f67b5c98028 EFLAGS: 00000246 ORIG_RAX: 000000000000002c RAX: ffffffffffffffda RBX: 00007f67b5015fa0 RCX: 00007f67b4d9c629 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003 RBP: 00007f67b4e32b39 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000040000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007f67b5016038 R14: 00007f67b5015fa0 R15: 00007ffe3cb66dd8 </TASK>		
CVE-2026-43184	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rnbd-srv: Zero the rsp buffer before using it Before using the data buffer to send back the response message, zero it completely. This prevents any stray bytes to be picked up by the client side when there the message is exchanged between different protocol versions.	2026-05-06	7.5
CVE-2026-43194	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: consume xmit errors of GSO frames udpgro_frglist.sh and udpgro_bench.sh are the flakiest tests currently in NIPA. They fail in the same exact way, TCP GRO test stalls occasionally and the test gets killed after 10min. These tests use veth to simulate GRO. They attach a trivial ("return XDP_PASS;") XDP program to the veth to force TSO off and NAPI on. Digging into the failure mode we can see that the connection is completely stuck after a burst of drops. The sender's snd_nxt is at sequence number N [1], but the receiver claims to have received (rcv_nxt) up to N + 3 * MSS [2]. Last piece of the puzzle is that senders rtx queue is not empty (let's say the block in the rtx queue is at sequence number N - 4 * MSS [3]). In this state, sender sends a retransmission from the rtx queue with a single segment, and sequence numbers N-4*MSS:N-3*MSS [3]. Receiver sees it and responds with an ACK all the way up to N + 3 * MSS [2]. But sender will reject this ack as TCP_ACK_UNSENT_DATA because it has no recollection of ever sending data that far out [1]. And we are stuck. The root cause is the mess of the xmit return codes. veth returns an error when it can't xmit a frame. We end up with a loss event like this: -- | GSO super frame 1 | GSO super frame 2 | |--| | seg 1 | seg 2 | seg 3 | seg 4 | seg 5 | seg 6 | seg 7 | seg 8 | -- x ok ok <ok>| ok ok ok <x> \\ snd_nxt "x" means packet lost by veth, and "ok" means it went thru. Since veth has TSO disabled in this test it sees individual segments. Segment 1 is on the retransmit queue and will be resent. So why did the sender not advance snd_nxt even tho it clearly did send up to seg 8? tcp_write_xmit() interprets the return code from the core to mean that data has not been sent at all. Since TCP deals with GSO super frames, not individual segment the crux of the problem is that loss of a single segment can be interpreted as loss of all. TCP only sees the last return code for the last segment of the GSO frame (in <> brackets in the diagram above). Of course for the problem to occur we need a setup or a device without a Qdisc. Otherwise Qdisc layer disconnects the protocol layer from the device errors completely. We have multiple ways to fix this. 1) make veth not return an error when it lost a packet. While this is what I think we did in the past, the issue keeps	2026-05-06	7.5

		reappearing and it's annoying to debug. The game of whack a mole is not great. 2) fix the damn return codes We only talk about NETDEV_TX_OK and NETDEV_TX_BUSY in the documentation, so maybe we should make the return code from ndo_start_xmit() a boolean. I like that the most, but perhaps some ancient, not-really-networking protocol would suffer. 3) make TCP ignore the errors It is not entirely clear to me what benefit TCP gets from interpreting the result of ip_queue_xmit()? Specifically once the connection is established and we're pushing data - packet loss is just packet loss? 4) this fix Ignore the rc in the Qdisc-less+GSO case, since it's unreliable. We already always return OK in the TCQ_F_CAN_BYPASS case. In the Qdisc-less case let's be a bit more conservative and only mask the GSO errors. This path is taken by non-IP-"networks" like CAN, MCTP etc, so we could regress some ancient thing. This is the simplest, but also maybe the hackiest fix? Similar fix has been proposed by Eric in the past but never committed because original reporter was working with an OOT driver and wasn't providing feedback (see Link).		
CVE-2026-43199	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix "scheduling while atomic" in IPsec MAC address query Fix a "scheduling while atomic" bug in mlx5e_ipsec_init_macs() by replacing mlx5_query_mac_address() with ether_addr_copy() to get the local MAC address directly from netdev->dev_addr. The issue occurs because mlx5_query_mac_address() queries the hardware which involves mlx5_cmd_exec() that can sleep, but it is called from the mlx5e_ipsec_handle_event workqueue which runs in atomic context. The MAC address is already available in netdev->dev_addr, so no need to query hardware. This avoids the sleeping call and resolves the bug. Call trace: BUG: scheduling while atomic: kworker/u112:2/69344/0x00000200 __schedule+0x7ab/0xa20 schedule+0x1c/0xb0 schedule_timeout+0x6e/0xf0 __wait_for_common+0x91/0x1b0 cmd_exec+0xa85/0xff0 [mlx5_core] mlx5_cmd_exec+0x1f/0x50 [mlx5_core] mlx5_query_nic_vport_mac_address+0x7b/0xd0 [mlx5_core] mlx5_query_mac_address+0x19/0x30 [mlx5_core] mlx5e_ipsec_init_macs+0xc1/0x720 [mlx5_core] mlx5e_ipsec_build_accel_xfrm_attrs+0x422/0x670 [mlx5_core] mlx5e_ipsec_handle_event+0x2b9/0x460 [mlx5_core] process_one_work+0x178/0x2e0 worker_thread+0x2ea/0x430	2026-05-06	7.5
CVE-2026-43203	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: atm: fore200e: fix use-after-free in tasklets during device removal When the PCA-200E or SBA-200E adapter is being detached, the fore200e is deallocated. However, the tx_tasklet or rx_tasklet may still be running or pending, leading to use-after-free bug when the already freed fore200e is accessed again in fore200e_tx_tasklet() or fore200e_rx_tasklet(). One of the race conditions can occur as follows: CPU 0 (cleanup) | CPU 1 (tasklet) fore200e_pca_remove_one() | fore200e_interrupt() fore200e_shutdown() | tasklet_schedule() kfree(fore200e) | fore200e_tx_tasklet() | fore200e-> // UAF Fix this by ensuring tx_tasklet or rx_tasklet is properly canceled before the fore200e is released. Add tasklet_kill() in fore200e_shutdown() to synchronize with any pending or running tasklets. Moreover, since fore200e_reset() could prevent further interrupts or data transfers, the tasklet_kill() should be placed after fore200e_reset() to prevent the tasklet from being rescheduled in fore200e_interrupt(). Finally, it only needs to do tasklet_kill() when the fore200e state is greater	2026-05-06	7.5

		than or equal to FORE200E_STATE_IRQ, since tasklets are uninitialized in earlier states. In a word, the tasklet_kill() should be placed in the FORE200E_STATE_IRQ branch within the switch...case structure. This bug was identified through static analysis.		
CVE-2026-43213	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: pci: validate sequence number of TX release report Hardware rarely reports abnormal sequence number in TX release report, which will access out-of-bounds of wd_ring->pages array, causing NULL pointer dereference. BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 1 PID: 1085 Comm: irq/129-rtw89_p Tainted: G S U 6.1.145-17510-g2f3369c91536 #1 (HASH:69e8 1) Call Trace: <IRQ> rtw89_pci_release_tx+0x18f/0x300 [rtw89_pci (HASH:4c83 2)] rtw89_pci_napi_poll+0xc2/0x190 [rtw89_pci (HASH:4c83 2)] net_rx_action+0xfc/0x460 net/core/dev.c:6578 net/core/dev.c:6645 net/core/dev.c:6759 handle_softirqs+0xbe/0x290 kernel/softirq.c:601 ? rtw89_pci_interrupt_threadfn+0xc5/0x350 [rtw89_pci (HASH:4c83 2)] __local_bh_enable_ip+0xeb/0x120 kernel/softirq.c:499 kernel/softirq.c:423 </IRQ> <TASK> rtw89_pci_interrupt_threadfn+0xf8/0x350 [rtw89_pci (HASH:4c83 2)] ? irq_thread+0xa7/0x340 kernel/irq/manage.c:0 irq_thread+0x177/0x340 kernel/irq/manage.c:1205 kernel/irq/manage.c:1314 ? thaw_kernel_threads+0xb0/0xb0 kernel/irq/manage.c:1202 ? irq_forced_thread_fn+0x80/0x80 kernel/irq/manage.c:1220 kthread+0xea/0x110 kernel/kthread.c:376 ? synchronize_irq+0x1a0/0x1a0 kernel/irq/manage.c:1287 ? kthread_associate_blkcg+0x80/0x80 kernel/kthread.c:331 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:295 </TASK> To prevent crash, validate rpp_info.seq before using.	2026-05-06	7.5
CVE-2026-43226	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/rds: No shortcut out of RDS_CONN_ERROR RDS connections carry a state "rds_conn_path::cp_state" and transitions from one state to another and are conditional upon an expected state: "rds_conn_path_transition." There is one exception to this conditionality, which is "RDS_CONN_ERROR" that can be enforced by "rds_conn_path_drop" regardless of what state the condition is currently in. But as soon as a connection enters state "RDS_CONN_ERROR", the connection handling code expects it to go through the shutdown-path. The RDS/TCP multipath changes added a shortcut out of "RDS_CONN_ERROR" straight back to "RDS_CONN_CONNECTING" via "rds_tcp_accept_one_path" (e.g. after "rds_tcp_state_change"). A subsequent "rds_tcp_reset_callbacks" can then transition the state to "RDS_CONN_RESETTING" with a shutdown-worker queued. That'll trip up "rds_conn_init_shutdown", which was never adjusted to handle "RDS_CONN_RESETTING" and subsequently drops the connection with the dreaded "DR_INV_CONN_STATE", which leaves "RDS_SHUTDOWN_WORK_QUEUED" on forever. So we do two things here: a) Don't shortcut "RDS_CONN_ERROR", but take the longer path through the shutdown code. b) Add "RDS_CONN_RESETTING" to the expected states in "rds_conn_init_shutdown" so that we won't error out and get stuck, if we ever hit weird state transitions like this again."	2026-05-06	7.5

CVE-2026-43230	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/rds: Clear reconnect pending bit When canceling the reconnect worker, care must be taken to reset the reconnect-pending bit. If the reconnect worker has not yet been scheduled before it is canceled, the reconnect-pending bit will stay on forever.	2026-05-06	7.5
CVE-2026-43245	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntfs: ->d_compare() must not block ... so don't use __getname() there. Switch it (and ntfs_d_hash(), while we are at it) to kmalloc(PATH_MAX, GFP_NOWAIT). Yes, ntfs_d_hash() almost certainly can do with smaller allocations, but let ntfs folks deal with that - keep the allocation size as-is for now. Stop abusing names_cachep in ntfs, period - various uses of that thing in there have nothing to do with pathnames; just use k[mz]alloc() and be done with that. For now let's keep sizes as-in, but AFAICS none of the users actually want PATH_MAX.	2026-05-06	7.5
CVE-2026-43253	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: move wait_on_sem() out of spinlock With iommu.strict=1, the existing completion wait path can cause soft lockups under stressed environment, as wait_on_sem() busy-waits under the spinlock with interrupts disabled. Move the completion wait in iommu_completion_wait() out of the spinlock. wait_on_sem() only polls the hardware-updated cmd_sem and does not require iommu->lock, so holding the lock during the busy wait unnecessarily increases contention and extends the time with interrupts disabled.	2026-05-06	7.5
CVE-2026-43254	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ovpn: tcp - fix packet extraction from stream When processing TCP stream data in ovpn_tcp_recv, we receive large cloned skbs from __strp_rcv that may contain multiple coalesced packets. The current implementation has two bugs: 1. Header offset overflow: Using pskb_pull with large offsets on coalesced skbs causes skb->data - skb->head to exceed the u16 storage of skb->network_header. This causes skb_reset_network_header to fail on the inner decapsulated packet, resulting in packet drops. 2. Unaligned protocol headers: Extracting packets from arbitrary positions within the coalesced TCP stream provides no alignment guarantees for the packet data causing performance penalties on architectures without efficient unaligned access. Additionally, openvpn's 2-byte length prefix on TCP packets causes the subsequent 4-byte opcode and packet ID fields to be inherently misaligned. Fix both issues by allocating a new skb for each openvpn packet and using skb_copy_bits to extract only the packet content into the new buffer, skipping the 2-byte length prefix. Also, check the length before invoking the function that performs the allocation to avoid creating an invalid skb. If the packet has to be forwarded to userspace the 2-byte prefix can be pushed to the head safely, without misalignment. As a side effect, this approach also avoids the expensive linearization that pskb_pull triggers on cloned skbs with page fragments. In testing, this resulted in TCP throughput improvements of up to 74%.	2026-05-06	7.5
CVE-2026-7897	google - chrome	Use after free in Mobile in Google Chrome on iOS prior to 148.0.7778.96 allowed a remote attacker who convinced a user to engage in specific UI gestures to execute arbitrary code via a crafted HTML page. (Chromium security severity: Critical)	2026-05-06	7.5
CVE-2026-7929	google - chrome	Use after free in MediaRecording in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who convinced a user to engage in specific UI gestures to execute arbitrary code via a crafted HTML page. (Chromium security severity: High)	2026-05-06	7.5
CVE-2026-7948	google - chrome	Race in Chromoting in Google Chrome on Windows prior to 148.0.7778.96 allowed a local attacker to perform privilege escalation via a malicious file. (Chromium security severity: Medium)	2026-05-06	7.5
CVE-2026-7976	google - chrome	Use after free in Views in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to execute arbitrary code via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	7.5
CVE-2026-8007	google - chrome	Insufficient validation of untrusted input in Cast in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform privilege escalation via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	7.5

CVE-2026-40981	vmware - multiple products	When using Google Secrets Manager as a backend for the Spring Cloud Config server a client can craft a request to the config server potentially exposing secrets from unintended GCP projects. Spring Cloud Config 3.1.x: affected from 3.1.0 through 3.1.13 (inclusive); upgrade to 3.1.14 or greater (Enterprise Support Only). Spring Cloud Config 4.1.x: affected from 4.1.0 through 4.1.9 (inclusive); upgrade to 4.1.10 or greater (Enterprise Support Only). Spring Cloud Config 4.2.x: affected from 4.2.0 through 4.2.6 (inclusive); upgrade to 4.2.7 or greater (Enterprise Support Only). Spring Cloud Config 4.3.x: affected from 4.3.0 through 4.3.2 (inclusive); upgrade to 4.3.3 or greater. Spring Cloud Config 5.0.x: affected from 5.0.0 through 5.0.2 (inclusive); upgrade to 5.0.3 or greater.	2026-05-07	7.5
CVE-2026-26129	microsoft - 365_copilot_chat	Improper neutralization of special elements in M365 Copilot allows an unauthorized attacker to disclose information over a network.	2026-05-07	7.5
CVE-2026-26164	microsoft - 365_copilot_chat	Improper neutralization of special elements in output used by a downstream component ('injection') in M365 Copilot allows an unauthorized attacker to disclose information over a network.	2026-05-07	7.5
CVE-2026-33111	microsoft - copilot_chat	Improper neutralization of special elements used in a command ('command injection') in Copilot Chat (Microsoft Edge) allows an unauthorized attacker to disclose information over a network.	2026-05-07	7.5
CVE-2026-39816	apache - nifi	The optional extension component TinkerpopClientService is missing the Restricted annotation with the Execute Code Required Permission in Apache NiFi 2.0.0-M1 through 2.8.0. The TinkerpopClientService supports configuration of ByteCode Submission for the Script Submission Type, enabling Groovy Script execution in the service prior to submitting the query. The missing Restricted annotation allows users without the Execute Code Permission to configure the Service in installations that use fine-grained authorization and have the optional TinkerpopClientService installed. Apache NiFi installations that do not have the nifi-other-graph-services-nar installed are not subject to this vulnerability. Upgrading to Apache NiFi 2.9.0 is the recommended mitigation.	2026-05-08	7.5
CVE-2026-43296	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: octeontx2-af: Workaround SQM/PSE stalls by disabling sticky NIX SQ manager sticky mode is known to cause stalls when multiple SQs share an SMQ and transmit concurrently. Additionally, PSE may deadlock on transitions between sticky and non-sticky transmissions. There is also a credit drop issue observed when certain condition clocks are gated. work around these hardware errata by: - Disabling SQM sticky operation: - Clear TM6 (bit 15) - Clear TM11 (bit 14) - Disabling sticky → non-sticky transition path that can deadlock PSE: - Clear TM5 (bit 23) - Preventing credit drops by keeping the control-flow clock enabled: - Set TM9 (bit 21) These changes are applied via NIX_AF_SQM_DBG_CTL_STATUS. With this configuration the SQM/PSE maintain forward progress under load without credit loss, at the cost of disabling sticky optimizations.	2026-05-08	7.5
CVE-2026-43336	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: lib/crypto: chacha: Zeroize permuted_state before it leaves scope Since the ChaCha permutation is invertible, the local variable 'permuted_state' is sufficient to compute the original 'state', and thus the key, even after the permutation has been done. While the kernel is quite inconsistent about zeroizing secrets on the stack (and some prominent userspace crypto libraries don't bother at all since it's not guaranteed to work anyway), the kernel does try to do it as a best practice, especially in cases involving the RNG. Thus, explicitly zeroize 'permuted_state' before it goes out of scope.	2026-05-08	7.5
CVE-2026-43345	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ipa: fix event ring index not programmed for IPA v5.0+ For IPA v5.0+, the event ring index field moved from CH_C_CNTXT_0 to CH_C_CNTXT_1. The v5.0 register definition intended to define this field in the CH_C_CNTXT_1 fmask array but used the old identifier of ERINDEX instead of CH_ERINDEX. Without a valid event ring, GSI channels could never signal transfer completions. This caused gsi_channel_trans_quiesce() to block forever in wait_for_completion(). At least for IPA v5.2 this resolves an issue seen where runtime suspend, system suspend, and remoteproc stop all hanged forever. It also meant the IPA data path was completely non functional.	2026-05-08	7.5
CVE-2026-43347	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: arm64: dts: qcom: monaco: Reserve full Gunyah metadata region We observe spurious "Synchronous External Abort" exceptions (ESR=0x96000010) and kernel crashes on Monaco-based platforms.	2026-05-08	7.5

		These faults are caused by the kernel inadvertently accessing hypervisor-owned memory that is not properly marked as reserved. >From boot log, The Qualcomm hypervisor reports the memory range at 0x91a80000 of size 0x80000 (512 KiB) as hypervisor-owned: qhee_hyp_assign_remove_memory: 0x91a80000/0x80000 -> ret 0 However, the EFI memory map provided by firmware only reserves the subrange 0x91a40000–0x91a87fff (288 KiB). The remaining portion (0x91a88000–0x91affff) is incorrectly reported as conventional memory (from efi debug): efi: 0x000091a40000-0x000091a87fff [Reserved...] efi: 0x000091a88000-0x0000938fffff [Conventional...] As a result, the allocator may hand out PFNs inside the hypervisor owned region, causing fatal aborts when the kernel accesses those addresses. Add a reserved-memory carveout for the Gunyah hypervisor metadata at 0x91a80000 (512 KiB) and mark it as no-map so Linux does not map or allocate from this area. For the record: Hyp version: gunyah-e78adb36e debug (2025-11-17 05:38:05 UTC) UEFI Ver: 6.0.260122.BOOT.MXF.1.0.c1-00449-KODIAKLA-1		
CVE-2026-43373	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ncsi: fix skb leak in error paths Early return paths in NCSI RX and AEN handlers fail to release the received skb, resulting in a memory leak. Specifically, ncsi_aen_handler() returns on invalid AEN packets without consuming the skb. Similarly, ncsi_rcv_rsp() exits early when failing to resolve the NCSI device, response handler, or request, leaving the skb unfreed.	2026-05-08	7.5
CVE-2026-43405	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Use u32 for non-negative values in ceph_monmap_decode() This patch fixes unnecessary implicit conversions that change signedness of blob_len and num_mon in ceph_monmap_decode(). Currently blob_len and num_mon are (signed) int variables. They are used to hold values that are always non-negative and get assigned in ceph_decode_32_safe(), which is meant to assign u32 values. Both variables are subsequently used as unsigned values, and the value of num_mon is further assigned to monmap->num_mon, which is of type u32. Therefore, both variables should be of type u32. This is especially relevant for num_mon. If the value read from the incoming message is very large, it is interpreted as a negative value, and the check for num_mon > CEPH_MAX_MON does not catch it. This leads to the attempt to allocate a very large chunk of memory for monmap, which will most likely fail. In this case, an unnecessary attempt to allocate memory is performed, and -ENOMEM is returned instead of -EINVAL.	2026-05-08	7.5
CVE-2026-43441	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: bonding: Fix nd_tbl NULL dereference when IPv6 is disabled When booting with the 'ipv6.disable=1' parameter, the nd_tbl is never initialized because inet6_init() exits before ndisc_init() is called which initializes it. If bonding ARP/NS validation is enabled, an IPv6 NS/NA packet received on a slave can reach bond_validate_na(), which calls bond_has_this_ip6(). That path calls ipv6_chk_addr() and can crash in __ipv6_chk_addr_and_flags(). BUG: kernel NULL pointer dereference, address: 00000000000005d8 Oops: Oops: 0000 [#1] SMP NOPTI RIP: 0010:__ipv6_chk_addr_and_flags+0x69/0x170 Call Trace: <IRQ> ipv6_chk_addr+0x1f/0x30 bond_validate_na+0x12e/0x1d0 [bonding] ? __pfx_bond_handle_frame+0x10/0x10 [bonding] bond_rcv_validate+0x1a0/0x450 [bonding] bond_handle_frame+0x5e/0x290 [bonding] ? srso_alias_return_thunk+0x5/0xfbef5 __netif_receive_skb_core.constprop.0+0x3e8/0xe50 ? srso_alias_return_thunk+0x5/0xfbef5 ? update_cfs_rq_load_avg+0x1a/0x240 ? srso_alias_return_thunk+0x5/0xfbef5	2026-05-08	7.5

		? __enqueue_entity+0x5e/0x240 __netif_receive_skb_one_core+0x39/0xa0 process_backlog+0x9c/0x150 __napi_poll+0x30/0x200 ? srso_alias_return_thunk+0x5/0xfbef5 net_rx_action+0x338/0x3b0 handle_softirqs+0xc9/0x2a0 do_softirq+0x42/0x60 </IRQ> <TASK> __local_bh_enable_ip+0x62/0x70 __dev_queue_xmit+0x2d3/0x1000 ? srso_alias_return_thunk+0x5/0xfbef5 ? srso_alias_return_thunk+0x5/0xfbef5 ? packet_parse_headers+0x10a/0x1a0 packet_sendmsg+0x10da/0x1700 ? kick_pool+0x5f/0x140 ? srso_alias_return_thunk+0x5/0xfbef5 ? __queue_work+0x12d/0x4f0 __sys_sendto+0x1f3/0x220 __x64_sys_sendto+0x24/0x30 do_syscall_64+0x101/0xf80 ? exc_page_fault+0x6e/0x170 ? srso_alias_return_thunk+0x5/0xfbef5 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> Fix this by checking ipv6_mod_enabled() before dispatching IPv6 packets to bond_na_rcv(). If IPv6 is disabled, return early from bond_rcv_validate() and avoid the path to ipv6_chk_addr().		
CVE-2026-43462	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: spacemit: Fix error handling in emac_tx_mem_map() The DMA mappings were leaked on mapping error. Free them with the existing emac_free_tx_buf() function.	2026-05-08	7.5
CVE-2026-43464	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: RX, Fix XDP multi-buf frag counting for legacy RQ XDP multi-buf programs can modify the layout of the XDP buffer when the program calls bpf_xdp_pull_data() or bpf_xdp_adjust_tail(). The referenced commit in the fixes tag corrected the assumption in the mlx5 driver that the XDP buffer layout doesn't change during a program execution. However, this fix introduced another issue: the dropped fragments still need to be counted on the driver side to avoid page fragment reference counting issues. Such issue can be observed with the test_xdp_native_adjst_tail_shrnk_data selftest when using a payload of 3600 and shrinking by 256 bytes (an upcoming selftest patch): the last fragment gets released by the XDP code but doesn't get tracked by the driver. This results in a negative pp_ref_count during page release and the following splat: WARNING: include/net/page_pool/helpers.h:297 at mlx5e_page_release_fragmented.isra.0+0x4a/0x50 [mlx5_core], CPU#12: ip/3137 Modules linked in: [...] CPU: 12 UID: 0 PID: 3137 Comm: ip Not tainted 6.19.0-rc3+ #12 NONE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:mlx5e_page_release_fragmented.isra.0+0x4a/0x50 [mlx5_core] [...] Call Trace: <TASK> mlx5e_dealloc_rx_wqe+0xcb/0x1a0 [mlx5_core] mlx5e_free_rx_descs+0x7f/0x110 [mlx5_core] mlx5e_close_rq+0x50/0x60 [mlx5_core] mlx5e_close_queues+0x36/0x2c0 [mlx5_core] mlx5e_close_channel+0x1c/0x50 [mlx5_core] mlx5e_close_channels+0x45/0x80 [mlx5_core] mlx5e_safe_switch_params+0x1a5/0x230 [mlx5_core] mlx5e_change_mtu+0xf3/0x2f0 [mlx5_core] netif_set_mtu_ext+0xf1/0x230 do_setlink.isra.0+0x219/0x1180 rtnl_newlink+0x79f/0xb60 rtnetlink_rcv_msg+0x213/0x3a0 netlink_rcv_skb+0x48/0xf0 netlink_unicast+0x24a/0x350 netlink_sendmsg+0x1ee/0x410	2026-05-08	7.5

		__sock_sendmsg+0x38/0x60 ____sys_sendmsg+0x232/0x280 ___sys_sendmsg+0x78/0xb0 __sys_sendmsg+0x5f/0xb0 [...] do_syscall_64+0x57/0xc50 This patch fixes the issue by doing page frag counting on all the original XDP buffer fragments for all relevant XDP actions (XDP_TX , XDP_REDIRECT and XDP_PASS). This is basically reverting to the original counting before the commit in the fixes tag. As frag_page is still pointing to the original tail, the nr_frags parameter to xdp_update_skb_frags_info() needs to be calculated in a different way to reflect the new nr_frags.		
CVE-2026-43469	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: Decrement re_receiving on the early exit paths In the event that rpcrdma_post_recvs() fails to create a work request (due to memory allocation failure, say) or otherwise exits early, we should decrement ep->re_receiving before returning. Otherwise we will hang in rpcrdma_xprt_drain() as re_receiving will never reach zero and the completion will never be triggered. On a system with high memory pressure, this can appear as the following hung task: INFO: task kworker/u385:17:8393 blocked for more than 122 seconds. Tainted: G S E 6.19.0 #3 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:kworker/u385:17 state:D stack:0 pid:8393 tgid:8393 ppid:2 task_flags:0x4248060 flags:0x00080000 Workqueue: xprtiod xprt_autoclose [sunrpc] Call Trace: <TASK> __schedule+0x48b/0x18b0 ? ib_post_send_mad+0x247/0xae0 [ib_core] schedule+0x27/0xf0 schedule_timeout+0x104/0x110 __wait_for_common+0x98/0x180 ? __pfx_schedule_timeout+0x10/0x10 wait_for_completion+0x24/0x40 rpcrdma_xprt_disconnect+0x444/0x460 [rpcrdma] xprt_rdma_close+0x12/0x40 [rpcrdma] xprt_autoclose+0x5f/0x120 [sunrpc] process_one_work+0x191/0x3e0 worker_thread+0x2e3/0x420 ? __pfx_worker_thread+0x10/0x10 kthread+0x10d/0x230 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x273/0x2b0 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30	2026-05-08	7.5
CVE-2026-42011	red hat - multiple products	A flaw was found in gnutls. This vulnerability occurs because permitted name constraints were incorrectly ignored when previous Certificate Authorities (CAs) only had excluded name constraints. A remote attacker could exploit this to bypass critical name constraint checks during certificate validation. This bypass could lead to the acceptance of invalid certificates, potentially enabling spoofing or man-in-the-middle attacks against affected systems.	2026-05-07	7.4
CVE-2026-7821	ivanti - multiple products	Improper certificate validation in Ivanti EPMM before versions 12.6.1.1, 12.7.0.1, and 12.8.0.1 allows a remote unauthenticated attacker to enroll a device belonging to a restricted set of unenrolled devices, leading to information disclosure about EPMM appliance and impacting on the integrity of the newly enrolled device identity.	2026-05-07	7.4
CVE-2026-43869	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.	2026-05-05	7.3
CVE-2026-43870	apache - thrift	Origin Validation Error, Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'), Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Request/Response Splitting'), Uncontrolled Resource Consumption 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.	2026-05-05	7.3
CVE-2026-29168	apache - http_server	Allocation of Resources Without Limits or Throttling vulnerability in Apache HTTP Server's mod_md via OCSP response data. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.66.	2026-05-05	7.3

		Users are recommended to upgrade to version 2.4.67, which fixes the issue.		
CVE-2026-23926	zabbix - Zabbix	An authenticated (non-super) administrator can create a maintenance period with a JavaScript payload that is executed by any user that opens tooltip for that maintenance period in the Host navigator widget. This can allow the attacker to perform unauthorized actions depending on which user opens the tooltip.	2026-05-06	7.3
CVE-2026-23928	zabbix - Zabbix	The Item history widget (in Zabbix 7.0+) or the Plain text widget (in Zabbix 6.0) can execute injected JavaScript when HTML display is enabled. This can allow an attacker to perform unauthorized actions depending on which user opens a dashboard containing these widgets. The malicious JavaScript would have to come from a monitored host controlled by the attacker. Note: the Item history widget is a replacement for the Plain text widget since Zabbix 7.0.	2026-05-06	7.3
CVE-2026-41288	watchguard - agent	Incorrect permission assignment for a resource in the patch management component of the WatchGuard Agent on Windows allows an authenticated local user to elevate their privileges to NT AUTHORITY\\SYSTEM.	2026-05-06	7.3
CVE-2026-8090	mozilla - multiple products	Use-after-free in the DOM: Networking component. This vulnerability was fixed in Firefox 150.0.2, Firefox ESR 140.10.2, Firefox ESR 115.35.2, Thunderbird 150.0.2, and Thunderbird 140.10.2.	2026-05-07	7.3
CVE-2026-43459	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: soc-core: flush delayed work before removing DAIs and widgets When a sound card is unbound while a PCM stream is open, a use-after-free can occur in snd_soc_dapm_stream_event(), called from the close_delayed_work workqueue handler. During unbind, snd_soc_unbind_card() flushes delayed work and then calls soc_cleanup_card_resources(). Inside cleanup, snd_card_disconnect_sync() releases all PCM file descriptors, and the resulting PCM close path can call snd_soc_dapm_stream_stop() which schedules new delayed work with a pmdown_time timer delay. Since this happens after the flush in snd_soc_unbind_card(), the new work is not caught. soc_remove_link_components() then frees DAPM widgets before this work fires, leading to the use-after-free. The existing flush in soc_free_pcm_runtime() also cannot help as it runs after soc_remove_link_components() has already freed the widgets. Add a flush in soc_cleanup_card_resources() after snd_card_disconnect_sync() (after which no new PCM closes can schedule further delayed work) and before soc_remove_link_dais() and soc_remove_link_components() (which tear down the structures the delayed work accesses).	2026-05-08	7.3
CVE-2026-20035	cisco - Cisco Unity Connection	A vulnerability in the web UI of Cisco Unity Connection Web Inbox could allow an unauthenticated, remote attacker to conduct SSRF attacks through an affected device. This vulnerability is due to improper input validation for specific HTTP requests. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to send arbitrary network requests that are sourced from the affected device.	2026-05-06	7.2
CVE-2026-41002	vmware - multiple products	The base directory (`spring.cloud.config.server.git.basedir`) used by the Spring Cloud Config Server to clone Git repositories to is susceptible to time-of-check-time-of-use (TOCTOU) attacks. Spring Cloud Config 3.1.x: affected from 3.1.0 through 3.1.13 (inclusive); upgrade to 3.1.14 or greater (Enterprise Support Only). Spring Cloud Config 4.1.x: affected from 4.1.0 through 4.1.9 (inclusive); upgrade to 4.1.10 or greater (Enterprise Support Only). Spring Cloud Config 4.2.x: affected from 4.2.0 through 4.2.6 (inclusive); upgrade to 4.2.7 or greater (Enterprise Support Only). Spring Cloud Config 4.3.x: affected from 4.3.0 through 4.3.2 (inclusive); upgrade to 4.3.3 or greater. Spring Cloud Config 5.0.x: affected from 5.0.0 through 5.0.2 (inclusive); upgrade to 5.0.3 or greater.	2026-05-07	7.2
CVE-2026-6973	ivanti - multiple products	An Improper Input Validation in Ivanti EPMM before versions 12.6.1.1, 12.7.0.1, and 12.8.0.1 allows a remotely authenticated user with administrative access to achieve remote code execution.	2026-05-07	7.2
CVE-2026-3828	hikvision - multiple products	Some Hikvision switch products (discontinued since December 2023) are vulnerable to authenticated remote command execution due to insufficient input validation. Attackers with valid credentials can exploit this flaw by sending crafted packets containing malicious commands to affected devices, leading to arbitrary command execution.	2026-05-09	7.2
CVE-2026-43062	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix type confusion in l2cap_ecred_reconf_rsp() l2cap_ecred_reconf_rsp() casts the incoming data to struct l2cap_ecred_conn_rsp (the ECRED *connection* response, 8 bytes with result at offset 6) instead of struct l2cap_ecred_reconf_rsp (2 bytes with result at offset 0). This causes two problems: - The sizeof(*rsp) length check requires 8 bytes instead of the correct 2, so valid L2CAP_ECRED_RECONF_RSP packets are rejected with -EPROTO. - rsp->result reads from offset 6 instead of offset 0, returning wrong data when the packet is large enough to pass the check.	2026-05-05	7.1

		Fix by using the correct type. Also pass the already byte-swapped result variable to BT_DBG instead of the raw __le16 field.		
CVE-2026-43141	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntb: ntb_hw_switchtec: Fix shift-out-of-bounds for 0 mw lut Number of MW LUTs depends on NTB configuration and can be set to zero, in such scenario rounddown_pow_of_two will cause undefined behaviour and should not be performed. This patch ensures that rounddown_pow_of_two is called on valid value.	2026-05-06	7.1
CVE-2026-43166	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: erofs: fix interlaced plain identification for encoded extents Only plain data whose start position and on-disk physical length are both aligned to the block size should be classified as interlaced plain extents. Otherwise, it must be treated as shifted plain extents. This issue was found by syzbot using a crafted compressed image containing plain extents with unaligned physical lengths, which can cause OOB read in z_erofs_transform_plain().	2026-05-06	7.1
CVE-2026-43241	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntb: ntb_hw_switchtec: Fix array-index-out-of-bounds access Number of MW LUTs depends on NTB configuration and can be set to MAX_MWS, This patch protects against invalid index out of bounds access to mw_sizes When invalid access print message to user that configuration is not valid.	2026-05-06	7.1
CVE-2026-43280	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Add bounds check on pat_index to prevent OOB kernel read in madvise When user provides a bogus pat_index value through the madvise IOCTL, the xe_pat_index_get_coh_mode() function performs an array access without validating bounds. This allows a malicious user to trigger an out-of-bounds kernel read from the xe->pat.table array. The vulnerability exists because the validation in madvise_args_are_sane() directly calls xe_pat_index_get_coh_mode(xe, args->pat_index.val) without first checking if pat_index is within [0, xe->pat.n_entries). Although xe_pat_index_get_coh_mode() has a WARN_ON to catch this in debug builds, it still performs the unsafe array access in production kernels. v2(Matthew Auld) - Using array_index_nospec() to mitigate spectre attacks when the value is used v3(Matthew Auld) - Put the declarations at the start of the block (cherry picked from commit 944a3329b05510d55c69c2ef455136e2fc02de29)	2026-05-06	7.1
CVE-2026-43281	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mailbox: Prevent out-of-bounds access in fw_mbox_index_xlate() Although it is guided that `#mbox-cells` must be at least 1, there are many instances of `#mbox-cells = <0>;` in the device tree. If that is the case and the corresponding mailbox controller does not provide `fw_xlate` and of_xlate` function pointers, `fw_mbox_index_xlate()` will be used by default and out-of-bounds accesses could occur due to lack of bounds check in that function.	2026-05-06	7.1
CVE-2026-41287	watchguard - agent	Stack-based Buffer Overflow vulnerability in the WatchGuard Agent discovery service on Windows allows Overflow Buffers. An unauthenticated attacker on the same local network could exploit this vulnerability to crash the agent service.	2026-05-06	7.1
CVE-2026-41286	watchguard - agent	Stack-based Buffer Overflow vulnerability in the WatchGuard Agent discovery service on Windows allows Overflow Buffers. An unauthenticated attacker on the same local network could exploit this vulnerability to crash the agent service.	2026-05-06	7.1
CVE-2026-8063	mongodb - mongodb	An authenticated user can crash mongod when running $rankFusion or $scoreFusion with an empty pipeline on a view. When resolving a view, the server inspects the aggregation pipeline to determine whether it begins with an Atlas Search stage. For $rankFusion and $scoreFusion, this inspection reads the first element on each stage's input pipeline array without first verifying that the array is non-empty. Supplying an empty pipeline causes a null pointer dereference and crashes the server. This issue affects MongoDB Server 8.2 versions prior to 8.2.7.	2026-05-07	7.1
CVE-2026-42010	gnu - multiple products	A flaw was found in gnutls. Servers configured with RSA-PSK (Rivest–Shamir–Adleman – Pre-Shared Key) wrongfully matched usernames containing a NUL character with truncated usernames. A remote	2026-05-07	7.1

		attacker could exploit this by sending a specially crafted username, leading to an authentication bypass. This vulnerability allows an attacker to gain unauthorized access by circumventing the authentication process.		
CVE-2026-43442	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring: fix physical SQE bounds check for SQE_MIXED 128-byte ops When IORING_SETUP_SQE_MIXED is used without IORING_SETUP_NO_SQARRAY, the boundary check for 128-byte SQE operations in io_init_req() validated the logical SQ head position rather than the physical SQE index. The existing check: !(ctx->cached_sq_head & (ctx->sq_entries - 1)) ensures the logical position isn't at the end of the ring, which is correct for NO_SQARRAY rings where physical == logical. However, when sq_array is present, an unprivileged user can remap any logical position to an arbitrary physical index via sq_array. Setting sq_array[N] = sq_entries - 1 places a 128-byte operation at the last physical SQE slot, causing the 128-byte memcpy in io_uring_cmd_sqe_copy() to read 64 bytes past the end of the SQE array. Replace the cached_sq_head alignment check with a direct validation of the physical SQE index, which correctly handles both sq_array and NO_SQARRAY cases.	2026-05-08	7.1
CVE-2026-5788	ivanti - multiple products	An Improper Access Control in Ivanti EPMM before versions 12.6.1.1, 12.7.0.1, and 12.8.0.1 allows a remote unauthenticated attacker to invoke arbitrary methods.	2026-05-07	7.0
CVE-2026-3291	hp - samsung_print_service_plugin	Samsung Print Service Plugin for Android is potentially vulnerable to information disclosure when using an outdated version of the application via mobile devices. HP is releasing updates to mitigate these potential vulnerabilities.	2026-05-06	6.9
CVE-2026-1749	hikvision - HikCentral Professional	There is an Access Control Vulnerability in some HikCentral Professional versions. This could allow an unauthenticated user to obtain the admin permission.	2026-05-09	6.8
CVE-2026-35255	oracle - cloud_native_environment_command_line_interface	Vulnerability in the Oracle Cloud Native Environment Command Line Interface product of Oracle Open Source Projects. The supported versions that is affected is v2.3.2. Easily exploitable vulnerability allows unauthenticated attacker to compromise Oracle Cloud Native Environment Command Line Interface product via a malicious environment variable. Successful attacks of this vulnerability can result in Oracle Cloud Native Environment Command Line Interface allowing users to execute arbitrary code.	2026-05-06	6.6
CVE-2026-33523	apache - http_server	HTTP response splitting vulnerability in multiple Apache HTTP Server modules with untrusted or compromised backend servers. This issue affects Apache HTTP Server: from through 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-04	6.5
CVE-2025-47401	qualcomm - fastconnect_6200_firmware	Transient DOS when processing target power rate tables during channel configuration.	2026-05-04	6.5
CVE-2025-47403	qualcomm - snapdragon_x65_5g_modem-rf_firmware	Transient DOS when processing a malformed Fast Transition response frame with an invalid header structure during wireless roaming.	2026-05-04	6.5
CVE-2025-47404	qualcomm - qca8695au_firmware	Memory corruption when dynamically changing the size of a previously allocated buffer while its contents are being modified.	2026-05-04	6.5
CVE-2025-42611	mikrotik - RouterOS	RouterOS provides various services that rely on correct verification of client and server certificates to secure confidentiality and integrity of communications. This includes OpenVPN, CAPsMAN, Dot1x (802.1X), among others. The vulnerability lies in shared certificate validation logic which uses the system certificate store that is shared and equally trusted by all system services. This causes confusion of scope, allowing any certificate authority present in the system-wide trust store to be trusted in any context (with some exceptions), allowing partial or full authentication bypass in CAPsMAN, OpenVPN, Dot1X and potentially others.	2026-05-05	6.5
CVE-2026-43975	apache - multiple products	FolderUploadsFileManager in Apache Wicket does not validate or sanitize the uploadFieldId parameter or the clientFileName before constructing file paths, allowing an unauthenticated attacker to write arbitrary files outside the intended upload directory or read files from arbitrary locations on the server. This issue affects Apache Wicket: from 8.0.0 through 8.17.0, from 9.0.0 through 9.22.0, from 10.0.0 through 10.8.0.	2026-05-06	6.5

		Users are recommended to upgrade to version 10.9.0, which fixes the issue.		
CVE-2026-20168	cisco - Cisco IoT Field Network Director (IoT-FND)	A vulnerability in the web-based management interface of Cisco IoT Field Network Director could allow an authenticated, remote attacker with low privileges to retrieve files that they do not have permission to access. This vulnerability is due to insufficient file access checks. An attacker could exploit this vulnerability by submitting crafted input in the web-based management interface. A successful exploit could allow the attacker to read files that they are not authorized to access.	2026-05-06	6.5
CVE-2026-7924	google - chrome	Uninitialized Use in Dawn in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: High)	2026-05-06	6.5
CVE-2026-7982	google - chrome	Uninitialized Use in WebCodecs in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	6.5
CVE-2025-66170	apache - cloudstack	The CloudStack Backup plugin has an improper authorization logic in versions 4.21.0.0 and 4.22.0.0. Anyone with authenticated user-account access in CloudStack 4.21.0.0+ environments, where this plugin is enabled and has access to specific APIs can list backups from any account in the environment. This vulnerability does not allow them to see the contents of the backup. Users are recommended to upgrade to version 4.22.0.1, which fixes the issue.	2026-05-08	6.5
CVE-2025-66171	apache - cloudstack	The CloudStack Backup plugin has an improper access logic in versions 4.21.0.0 and 4.22.0.0. Anyone with authenticated user-account access in CloudStack 4.21.0.0+ environments, where this plugin is enabled and have access to specific APIs can create new VMs using backups of any other user of the environment. Backup plugin users using CloudStack 4.21.0.0+ are recommended to upgrade to CloudStack version 4.22.0.1, which fixes this issue.	2026-05-08	6.5
CVE-2025-69233	apache - multiple products	Due to multiple time-of-check time-of-use race conditions in the resource count check and increment logic, as well as missing validations, users of the platform are able to exceed the allocation limits configured for their accounts/domains. This can be used by an attacker to degrade the infrastructure's resources and lead to denial of service conditions. Users are recommended to upgrade to Apache CloudStack versions 4.20.3.0 or 4.22.0.1, or later, which fixes this issue.	2026-05-08	6.5
CVE-2026-20169	cisco - Cisco IoT Field Network Director (IoT-FND)	A vulnerability in the web-based management interface of Cisco IoT Field Network Director could allow an authenticated, remote attacker with low privileges to access files and execute commands on a remote router. This vulnerability is due to insufficient input validation of user-supplied data. An attacker could exploit this vulnerability by submitting crafted input in the web-based management interface. A successful exploit could allow the attacker to create, read, or delete files and execute limited commands inuser EXEC mode on a remote router.	2026-05-06	6.4
CVE-2026-6420	red hat - multiple products	A flaw was found in Keylime. An attacker with root access on an enrolled monitored machine, where the Keylime agent runs, can exploit a vulnerability in the Keylime verifier. The verifier uses a hardcoded challenge nonce for Trusted Platform Module (TPM) quote attestation instead of a cryptographically random value. This allows the attacker to stockpile valid TPM quotes and replay them to evade detection after compromising the system. This issue affects only the push model deployment.	2026-05-06	6.3
CVE-2026-7971	google - chrome	Inappropriate implementation in ORB in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	6.3
CVE-2026-7977	google - chrome	Inappropriate implementation in Canvas in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to bypass same origin policy via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	6.3
CVE-2026-8010	google - chrome	Insufficient validation of untrusted input in SiteIsolation in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	6.3
CVE-2025-47406	qualcomm - cologne_firmware	Information Disclosure while processing IOCTL handler callbacks without verifying buffer size.	2026-05-04	6.1
CVE-2026-35254	oracle - cloud_infrastructure_cli	Vulnerability in the Oracle OCI CLI product of Oracle Open Source Projects. The supported versions that is affected is 3.77. Easily exploitable vulnerability allows unauthenticated attacker with network access to compromise Oracle OCI CLI. Successful attacks of this vulnerability can result in Oracle OCI CLI allowing users to place imported files outside the intended directory.	2026-05-06	6.1
CVE-2026-42509	apache - multiple products	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') vulnerability in Apache Wicket. This issue affects Apache Wicket: from 8.0.0 through 8.17.0, 9.0.0, from 10.0.0 through 10.8.0. Users are recommended to upgrade to version 10.9.0, which fixes the issue.	2026-05-06	6.1
CVE-2026-7953	google - chrome	Insufficient validation of untrusted input in Omnibox in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to inject arbitrary scripts or HTML (UXSS) via malicious network traffic. (Chromium security severity: Medium)	2026-05-06	6.1
CVE-2026-34956	red hat - multiple products	A flaw was found in Open vSwitch. When Open vSwitch is configured with a conntrack flow using FTP helpers over the userspace datapath, a remote attacker can send a specially crafted FTP stream with an EPASV command exceeding 255 characters. This heap access error can lead to a crash, resulting in a Denial of Service (DoS) for the affected system.	2026-05-05	5.9
CVE-2026-25266	qualcomm - cologne_firmware	Memory corruption while processing IOCTL command when device is in power-save state.	2026-05-04	5.5
CVE-2026-43098	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-06	5.5

		nfc: s3fwrn5: allocate rx skb before consuming bytes s3fwrn82_uart_read() reports the number of accepted bytes to the serdev core. The current code consumes bytes into recv_skb and may already deliver a complete frame before allocating a fresh receive buffer. If that alloc_skb() fails, the callback returns 0 even though it has already consumed bytes, and it leaves recv_skb as NULL for the next receive callback. That breaks the receive_buf() accounting contract and can also lead to a NULL dereference on the next skb_put_u8(). Allocate the receive skb lazily before consuming the next byte instead. If allocation fails, return the number of bytes already accepted.		
CVE-2026-43100	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bridge: guard local VLAN-0 FDB helpers against NULL vlan group When CONFIG_BRIDGE_VLAN_FILTERING is not set, br_vlan_group() and nbp_vlan_group() return NULL (br_private.h stub definitions). The BR_BOOLOPT_FDB_LOCAL_VLAN_0 toggle code is compiled unconditionally and reaches br_fdb_delete_locals_per_vlan_port() and br_fdb_insert_locals_per_vlan_port(), where the NULL vlan group pointer is dereferenced via list_for_each_entry(v, &vg->vlan_list, vlist). The observed crash is in the delete path, triggered when creating a bridge with IFLA_BR_MULTI_BOOLOPT containing BR_BOOLOPT_FDB_LOCAL_VLAN_0 via RTM_NEWLINK. The insert helper has the same bug pattern. Oops: general protection fault, probably for non-canonical address 0xdffffc0000000056: 0000 [#1] KASAN NOPTI KASAN: null-ptr-deref in range [0x00000000000002b0-0x00000000000002b7] RIP: 0010:br_fdb_delete_locals_per_vlan+0x2b9/0x310 Call Trace: br_fdb_toggle_local_vlan_0+0x452/0x4c0 br_toggle_fdb_local_vlan_0+0x31/0x80 net/bridge/br.c:276 br_boolopt_toggle net/bridge/br.c:313 br_boolopt_multi_toggle net/bridge/br.c:364 br_changelink net/bridge/br_netlink.c:1542 br_dev_newlink net/bridge/br_netlink.c:1575 Add NULL checks for the vlan group pointer in both helpers, returning early when there are no VLANs to iterate. This matches the existing pattern used by other bridge FDB functions such as br_fdb_add() and br_fdb_delete().	2026-05-06	5.5
CVE-2026-43102	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: airoha: Fix memory leak in airoha_qdma_rx_process() If an error occurs on the subsequents buffers belonging to the non-linear part of the skb (e.g. due to an error in the payload length reported by the NIC or if we consumed all the available fragments for the skb), the page_pool fragment will not be linked to the skb so it will not return to the pool in the airoha_qdma_rx_process() error path. Fix the memory leak partially reverting commit 'd6d2b0e1538d ("net: airoha: Fix page recycling in airoha_qdma_rx_process()")' and always running page_pool_put_full_page routine in the airoha_qdma_rx_process() error path.	2026-05-06	5.5
CVE-2026-43103	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: lapbether: handle NETDEV_PRE_TYPE_CHANGE lapbeth_data_transmit() expects the underlying device type to be ARPHRD_ETHER. Returning NOTIFY_BAD from lapbeth_device_event() makes sure bonding driver can not break this expectation.	2026-05-06	5.5
CVE-2026-43104	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: Fix a memory leak in hang state error path When vc4_save_hang_state() encounters an early return condition, it returns without freeing the previously allocated `kernel_state`, leaking memory. Add the missing kfree() calls by consolidating the early return paths into a single place.	2026-05-06	5.5
CVE-2026-43105	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: Fix memory leak of BO array in hang state	2026-05-06	5.5

		The hang state's BO array is allocated separately with kzalloc() in vc4_save_hang_state() but never freed in vc4_free_hang_state(). Add the missing kfree() for the BO array before freeing the hang state struct.		
CVE-2026-43107	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: account XFRMA_IF_ID in aevent size calculation xfrm_get_ae() allocates the reply skb with xfrm_aevent_msgsize(), then build_aevent() appends attributes including XFRMA_IF_ID when x->if_id is set. xfrm_aevent_msgsize() does not include space for XFRMA_IF_ID. For states with if_id, build_aevent() can fail with -EMSGSIZE and hit BUG_ON(err < 0) in xfrm_get_ae(), turning a malformed netlink interaction into a kernel panic. Account XFRMA_IF_ID in the size calculation unconditionally and replace the BUG_ON with normal error unwinding.	2026-05-06	5.5
CVE-2026-43108	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: pd-mapper: Fix element length in servreg_loc_pfr_req_ei It looks element length declared in servreg_loc_pfr_req_ei for reason not matching servreg_loc_pfr_req's reason field due which we could observe decoding error on PD crash. qmi_decode_string_elem: String len 81 >= Max Len 65 Fix this by matching with servreg_loc_pfr_req's reason field.	2026-05-06	5.5
CVE-2026-43109	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86: shadow stacks: proper error handling for mmap lock 김영민 reports that shstk_pop_sigframe() doesn't check for errors from mmap_read_lock_killable(), which is a silly oversight, and also shows that we haven't marked those functions with "__must_check", which would have immediately caught it. So let's fix both issues.	2026-05-06	5.5
CVE-2026-43115	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: srcu: Use irq_work to start GP in tiny SRCU Tiny SRCU's srcu_gp_start_if_needed() directly calls schedule_work(), which acquires the workqueue pool->lock. This causes a lockdep splat when call_srcu() is called with a scheduler lock held, due to: call_srcu() [holding pi_lock] srcu_gp_start_if_needed() schedule_work() -> pool->lock workqueue_init() / create_worker() [holding pool->lock] wake_up_process() -> try_to_wake_up() -> pi_lock Also add irq_work_sync() to cleanup_srcu_struct() to prevent a use-after-free if a queued irq_work fires after cleanup begins. Tested with rcutorture SRCU-T and no lockdep warnings. [Thanks to Boqun for similar fix in patch "rcu: Use an intermediate irq_work to start process_srcu()"]	2026-05-06	5.5
CVE-2026-43118	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix zero size inode with non-zero size after log replay When logging that an inode exists, as part of logging a new name or logging new dir entries for a directory, we always set the generation of the logged inode item to 0. This is to signal during log replay (in overwrite_item()), that we should not set the i_size since we only logged that an inode exists, so the i_size of the inode in the subvolume tree must be preserved (as when we log new names or that an inode exists, we don't log extents). This works fine except when we have already logged an inode in full mode or it's the first time we are logging an inode created in a past transaction, that inode has a new i_size of 0 and then we log a new name for the inode (due to a new hardlink or a rename), in which case we log	2026-05-06	5.5

		an i_size of 0 for the inode and a generation of 0, which causes the log replay code to not update the inode's i_size to 0 (in overwrite_item()). An example scenario: mkdir /mnt/dir xfs_io -f -c "pwrite 0 64K" /mnt/dir/foo sync xfs_io -c "truncate 0" -c "fsync" /mnt/dir/foo ln /mnt/dir/foo /mnt/dir/bar xfs_io -c "fsync" /mnt/dir <power fail> After log replay the file remains with a size of 64K. This is because when we first log the inode, when we fsync file foo, we log its current i_size of 0, and then when we create a hard link we log again the inode in exists mode (LOG_INODE_EXISTS) but we set a generation of 0 for the inode item we add to the log tree, so during log replay overwrite_item() sees that the generation is 0 and i_size is 0 so we skip updating the inode's i_size from 64K to 0. Fix this by making sure at fill_inode_item() we always log the real generation of the inode if it was logged in the current transaction with the i_size we logged before. Also if an inode created in a previous transaction is logged in exists mode only, make sure we log the i_size stored in the inode item located from the commit root, so that if we log multiple times that the inode exists we get the correct i_size. A test case for fstests will follow soon.		
CVE-2026-43119	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: annotate data-races around hdev->req_status __hci_cmd_sync_sk() sets hdev->req_status under hdev->req_lock: hdev->req_status = HCI_REQ_PEND; However, several other functions read or write hdev->req_status without holding any lock: - hci_send_cmd_sync() reads req_status in hci_cmd_work (workqueue) - hci_cmd_sync_complete() reads/writes from HCI event completion - hci_cmd_sync_cancel() / hci_cmd_sync_cancel_sync() read/write - hci_abort_conn() reads in connection abort path Since __hci_cmd_sync_sk() runs on hdev->req_workqueue while hci_send_cmd_sync() runs on hdev->workqueue, these are different workqueues that can execute concurrently on different CPUs. The plain C accesses constitute a data race. Add READ_ONCE()/WRITE_ONCE() annotations on all concurrent accesses to hdev->req_status to prevent potential compiler optimizations that could affect correctness (e.g., load fusing in the wait_event condition or store reordering).	2026-05-06	5.5
CVE-2025-71271	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hfsplus: ensure sb->s_fs_info is always cleaned up When hfsplus was converted to the new mount api a bug was introduced by changing the allocation pattern of sb->s_fs_info. If setup_bdev_super() fails after a new superblock has been allocated by sget_fc(), but before hfsplus_fill_super() takes ownership of the filesystem-specific s_fs_info data it was leaked. Fix this by freeing sb->s_fs_info in hfsplus_kill_super().	2026-05-06	5.5
CVE-2025-71272	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: most: core: fix resource leak in most_register_interface error paths The function most_register_interface() did not correctly release resources if it failed early (before registering the device). In these cases, it returned an error code immediately, leaking the memory allocated for the interface. Fix this by initializing the device early via device_initialize() and	2026-05-06	5.5

		calling put_device() on all error paths. The most_register_interface() is expected to call put_device() on error which frees the resources allocated in the caller. The put_device() either calls release_mdev() or dim2_release(), depending on the caller. Switch to using device_add() instead of device_register() to handle the split initialization.		
CVE-2025-71273	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: Use devm_kmemdup() in rtw_set_supported_band() Simplify the code by using device managed memory allocations. This also fixes a memory leak in rtw_register_hw(). The supported bands were not freed in the error path. Copied from commit 145df52a8671 ("wifi: rtw89: Convert rtw89_core_set_supported_band to use devm_*").	2026-05-06	5.5
CVE-2025-71285	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: Drop the MHI auto_queue feature for IPCR DL channels MHI stack offers the 'auto_queue' feature, which allows the MHI stack to auto queue the buffers for the RX path (DL channel). Though this feature simplifies the client driver design, it introduces race between the client drivers and the MHI stack. For instance, with auto_queue, the 'dl_callback' for the DL channel may get called before the client driver is fully probed. This means, by the time the dl_callback gets called, the client driver's structures might not be initialized, leading to NULL ptr dereference. Currently, the drivers have to workaround this issue by initializing the internal structures before calling mhi_prepare_for_transfer_autoqueue(). But even so, there is a chance that the client driver's internal code path may call the MHI queue APIs before mhi_prepare_for_transfer_autoqueue() is called, leading to similar NULL ptr dereference. This issue has been reported on the Qcom X1E80100 CRD machines affecting boot. So to properly fix all these races, drop the MHI 'auto_queue' feature altogether and let the client driver (QRTR) manage the RX buffers manually. In the QRTR driver, queue the RX buffers based on the ring length during probe and recycle the buffers in 'dl_callback' once they are consumed. This also warrants removing the setting of 'auto_queue' flag from controller drivers. Currently, this 'auto_queue' feature is only enabled for IPCR DL channel. So only the QRTR client driver requires the modification.	2026-05-06	5.5
CVE-2025-71286	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Correct the allocation size for bytes controls The size of the data behind of scontrol->ipc_control_data for bytes controls is: [1] sizeof(struct sof_ipc4_control_data) + // kernel only struct [2] sizeof(struct sof_abi_hdr)) + payload The max_size specifies the size of [2] and it is coming from topology. Change the function to take this into account and allocate adequate amount of memory behind scontrol->ipc_control_data. With the change we will allocate [1] amount more memory to be able to hold the full size of data.	2026-05-06	5.5
CVE-2025-71287	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: memory: mtk-smi: fix device leak on larb probe Make sure to drop the reference taken when looking up the SMI device during larb probe on late probe failure (e.g. probe deferral) and on driver unbind.	2026-05-06	5.5
CVE-2025-71288	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: memory: mtk-smi: fix device leaks on common probe Make sure to drop the reference taken when looking up the SMI device during common probe on late probe failure (e.g. probe deferral) and on driver unbind.	2026-05-06	5.5
CVE-2025-71289	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved:	2026-05-06	5.5

		fs/ntfs3: handle attr_set_size() errors when truncating files If attr_set_size() fails while truncating down, the error is silently ignored and the inode may be left in an inconsistent state.		
CVE-2025-71290	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: misc: ti_fpc202: fix a potential memory leak in probe function Use for_each_child_of_node_scoped() to simplify the code and ensure the device node reference is automatically released when the loop scope ends.	2026-05-06	5.5

CVE-2025-71291	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		misc: bcm_vk: Fix possible null-pointer dereferences in bcm_vk_read()
		
		In the function bcm_vk_read(), the pointer entry is checked, indicating that it can be NULL. If entry is NULL and rc is set to -EMSGSIZE, the following code may cause null-pointer dereferences:
		
		struct vk_msg_blk tmp_msg = entry->to_h_msg[0];
		set_msg_id(&tmp_msg, entry->usr_msg_id);
		tmp_msg.size = entry->to_h_blks - 1;
		
		To prevent these possible null-pointer dereferences, copy to_h_msg, usr_msg_id, and to_h_blks from iter into temporary variables, and return these temporary variables to the application instead of accessing them through a potentially NULL entry.
			2026-05-06	5.5

CVE-2025-71292	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: jfs: nlink overflow in jfs_rename If nlink is maximal for a directory (-1) and inside that directory you perform a rename for some child directory (not moving from the parent), then the nlink of the first directory is first incremented and later decremented. Normally this is fine, but when nlink = -1 this causes a wrap around to 0, and then drop_nlink issues a warning. After applying the patch syzbot no longer issues any warnings. I also ran some basic fs tests to look for any regressions.	2026-05-06	5.5
CVE-2025-71293	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/ras: Move ras data alloc before bad page check In the rare event if eeprom has only invalid address entries, allocation is skipped, this causes following NULL pointer issue [547.103445] BUG: kernel NULL pointer dereference, address: 0000000000000010 [547.118897] #PF: supervisor read access in kernel mode [547.130292] #PF: error_code(0x0000) - not-present page [547.141689] PGD 124757067 P4D 0 [547.148842] Oops: 0000 [#1] PREEMPT SMP NOPTI [547.158504] CPU: 49 PID: 8167 Comm: cat Tainted: G OE 6.8.0-38-generic #38-Ubuntu [547.177998] Hardware name: Supermicro AS -8126GS-TNMR/H14DSG-OD, BIOS 1.7 09/12/2025 [547.195178] RIP: 0010:amdgpu_ras_sysfs_badpages_read+0x2f2/0x5d0 [amdgpu] [547.210375] Code: e8 63 78 82 c0 45 31 d2 45 3b 75 08 48 8b 45 a0 73 44 44 89 f1 48 8b 7d 88 48 89 ca 48 c1 e2 05 48 29 ca 49 8b 4d 00 48 01 d1 <48> 83 79 10 00 74 17 49 63 f2 48 8b 49 08 41 83 c2 01 48 8d 34 76 [547.252045] RSP: 0018:ffa0000067287ac0 EFLAGS: 00010246 [547.263636] RAX: ff11000167c28130 RBX: ff11000127600000 RCX: 0000000000000000 [547.279467] RDX: 0000000000000000 RSI: 0000000000000000 RDI: ff11000125b1c800 [547.295298] RBP: ffa0000067287b50 R08: 0000000000000000 R09: 0000000000000000 [547.311129] R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000 [547.326959] R13: ff11000217b1de00 R14: 0000000000000000 R15: 0000000000000092 [547.342790] FS: 0000746e59d14740(0000) GS:ff11017dfda80000(0000) knlGS:0000000000000000 [547.360744] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [547.373489] CR2: 0000000000000010 CR3: 000000019585e001 CR4: 0000000000f71ef0 [547.389321] PKRU: 55555554 [547.395316] Call Trace: [547.400737] <TASK> [547.405386] ? show_regs+0x6d/0x80 [547.412929] ? __die+0x24/0x80 [547.419697] ? page_fault_oops+0x99/0x1b0 [547.428588] ? do_user_addr_fault+0x2ee/0x6b0 [547.438249] ? exc_page_fault+0x83/0x1b0 [547.446949] ? asm_exc_page_fault+0x27/0x30 [547.456225] ? amdgpu_ras_sysfs_badpages_read+0x2f2/0x5d0 [amdgpu] [547.470040] ? mas_wr_modify+0xcd/0x140 [547.478548] sysfs_kf_bin_read+0x63/0xb0 [547.487248] kernfs_file_read_iter+0xa1/0x190 [547.496909] kernfs_fop_read_iter+0x25/0x40 [547.506182] vfs_read+0x255/0x390	2026-05-06	5.5

		This also result in space left assigned to negative values. Moving data alloc call before bad page check resolves both the issue.		
CVE-2025-71294	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix NULL pointer issue buffer funcs If SDMA block not enabled, buffer_funcs will not initialize, fix the null pointer issue if buffer_funcs not initialized.	2026-05-06	5.5
CVE-2025-71295	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fs/buffer: add alert in try_to_free_buffers() for folios without buffers try_to_free_buffers() can be called on folios with no buffers attached when filemap_release_folio() is invoked on a folio belonging to a mapping with AS_RELEASE_ALWAYS set but no release_folio operation defined. In such cases, folio_needs_release() returns true because of the AS_RELEASE_ALWAYS flag, but the folio has no private buffer data. This causes try_to_free_buffers() to call drop_buffers() on a folio with no buffers, leading to a null pointer dereference. Adding a check in try_to_free_buffers() to return early if the folio has no buffers attached, with WARN_ON_ONCE() to alert about the misconfiguration. This provides defensive hardening.	2026-05-06	5.5
CVE-2026-43122	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ACPI: processor: Update cpuidle driver check in __acpi_processor_start() Commit 7a8c994cbb2d ("ACPI: processor: idle: Optimize ACPI idle driver registration") moved the ACPI idle driver registration to acpi_processor_driver_init() and acpi_processor_power_init() does not register an idle driver any more. Accordingly, the cpuidle driver check in __acpi_processor_start() needs to be updated to avoid calling acpi_processor_power_init() without a cpuidle driver, in which case the registration of the cpuidle device in that function would lead to a NULL pointer dereference in __cpuidle_register_device().	2026-05-06	5.5
CVE-2026-43123	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fbcon: check return value of con2fb_acquire_newinfo() If fbcon_open() fails when called from con2fb_acquire_newinfo() then info->fbcon_par pointer remains NULL which is later dereferenced. Add check for return value of the function con2fb_acquire_newinfo() to avoid it. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2026-05-06	5.5
CVE-2026-43124	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: pstore: ram_core: fix incorrect success return when vmap() fails In persistent_ram_vmap(), vmap() may return NULL on failure. If offset is non-zero, adding offset_in_page(start) causes the function to return a non-NULL pointer even though the mapping failed. persistent_ram_buffer_map() therefore incorrectly returns success. Subsequent access to prz->buffer may dereference an invalid address and cause crashes. Add proper NULL checking for vmap() failures.	2026-05-06	5.5
CVE-2026-43127	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntfs3: fix circular locking dependency in run_unpack_ex Syzbot reported a circular locking dependency between wnd->rw_lock (sbi->used.bitmap) and ni->file.run_lock. The deadlock scenario: 1. ntfs_extend_mft() takes ni->file.run_lock then wnd->rw_lock. 2. run_unpack_ex() takes wnd->rw_lock then tries to acquire ni->file.run_lock inside ntfs_refresh_zone(). This creates an AB-BA deadlock. Fix this by using down_read_trylock() instead of down_read() when acquiring run_lock in run_unpack_ex(). If the lock is contended,	2026-05-06	5.5

		skip ntfs_refresh_zone() - the MFT zone will be refreshed on the next MFT operation. This breaks the circular dependency since we never block waiting for run_lock while holding wnd->rw_lock.		
CVE-2026-43129	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ima: verify the previous kernel's IMA buffer lies in addressable RAM Patch series "Address page fault in ima_restore_measurement_list()", v3. When the second-stage kernel is booted via kexec with a limiting command line such as "mem=<size>" we observe a pafe fault that happens. BUG: unable to handle page fault for address: ffff97793ff47000 RIP: ima_restore_measurement_list+0xdc/0x45a #PF: error_code(0x0000) not-present page This happens on x86_64 only, as this is already fixed in aarch64 in commit: cbf9c4b9617b ("of: check previous kernel's ima-kexec-buffer against memory bounds") This patch (of 3): When the second-stage kernel is booted with a limiting command line (e.g. "mem=<size>"), the IMA measurement buffer handed over from the previous kernel may fall outside the addressable RAM of the new kernel. Accessing such a buffer can fault during early restore. Introduce a small generic helper, ima_validate_range(), which verifies that a physical [start, end] range for the previous-kernel IMA buffer lies within addressable memory: - On x86, use pfn_range_is_mapped(). - On OF based architectures, use page_is_ram().	2026-05-06	5.5
CVE-2026-43130	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Flush dev-IOTLB only when PCIe device is accessible in scalable mode Commit 4fc82cd907ac ("iommu/vt-d: Don't issue ATS Invalidation request when device is disconnected") relies on pci_dev_is_disconnected() to skip ATS invalidation for safely-removed devices, but it does not cover link-down caused by faults, which can still hard-lock the system. For example, if a VM fails to connect to the PCIe device, "virsh destroy" is executed to release resources and isolate the fault, but a hard-lockup occurs while releasing the group fd. Call Trace: qi_submit_sync qi_flush_dev_iotlb intel_pasid_tear_down_entry device_block_translation blocking_domain_attach_dev __iommu_attach_device __iommu_device_set_domain __iommu_group_set_domain_internal iommu_detach_group vfio_iommu_type1_detach_group vfio_group_detach_container vfio_group_fops_release __fput Although pci_device_is_present() is slower than pci_dev_is_disconnected(), it still takes only ~70 µs on a ConnectX-5 (8 GT/s, x2) and becomes even faster as PCIe speed and width increase. Besides, devtlb_invalidation_with_pasid() is called only in the paths below, which are far less frequent than memory map/unmap. 1. mm-struct release 2. {attach,release}_dev 3. set/remove PASID 4. dirty-tracking setup The gain in system stability far outweighs the negligible cost of using pci_device_is_present() instead of pci_dev_is_disconnected() to decide when to skip ATS invalidation, especially under GDR high-load conditions.	2026-05-06	5.5

CVE-2026-43131	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Fix null pointer dereference issue If SMU is disabled, during RAS initialization, there will be null pointer dereference issue here.	2026-05-06	5.5
CVE-2026-43132	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm-verity: correctly handle dm_bufio_client_create() failure If either of the calls to dm_bufio_client_create() in verity_fec_ctr() fails, then dm_bufio_client_destroy() is later called with an ERR_PTR() argument. That causes a crash. Fix this.	2026-05-06	5.5
CVE-2026-43135	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: cx23885: Add missing unmap in snd_cx23885_hw_params() In error path, add cx23885_alsa_dma_unmap() to release the resource acquired by cx23885_alsa_dma_map().	2026-05-06	5.5
CVE-2026-43136	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: logitech-hidpp: Check maxfield in hidpp_get_report_length() Do not crash when a report has no fields. Fake USB gadgets can send their own HID report descriptors and can define report structures without valid fields. This can be used to crash the kernel over USB.	2026-05-06	5.5
CVE-2026-43137	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: Fix NULL pointer dereference If there's a mismatch between the DAI links in the machine driver and the topology, it is possible that the playback/capture widget is not set, especially in the case of loopback capture for echo reference where we use the dummy DAI link. Return the error when the widget is not set to avoid a null pointer dereference like below when the topology is broken. RIP: 0010:hda_dai_get_ops.isra.0+0x14/0xa0 [snd_sof_intel_hda_common]	2026-05-06	5.5
CVE-2026-43140	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: magicmouse: Do not crash on missing msc->input Fake USB devices can send their own report descriptors for which the input_mapping() hook does not get called. In this case, msc->input stays NULL, leading to a crash at a later time. Detect this condition in the input_configured() hook and reject the device. This is not supposed to happen with actual magic mouse devices, but can be provoked by imposing as a magic mouse USB device.	2026-05-06	5.5
CVE-2026-43142	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: iris: gen1: Destroy internal buffers after FW releases After the firmware releases internal buffers, the driver was not destroying them. This left stale allocations that were no longer used, especially across resolution changes where new buffers are allocated per the updated requirements. As a result, memory was wasted until session close. Destroy internal buffers once the release response is received from the firmware.	2026-05-06	5.5
CVE-2026-43143	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mfd: core: Add locking around 'mfd_of_node_list' Manipulating a list in the kernel isn't safe without some sort of mutual exclusion. Add a mutex any time we access / modify 'mfd_of_node_list' to prevent possible crashes.	2026-05-06	5.5
CVE-2026-43144	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Fix potential kernel oops when probe fails When probe of the sdio brcmfmac device fails for some reasons (i.e. missing firmware), the sdiodev->bus is set to error instead of NULL, thus the cleanup later in brcmf_sdio_remove() tries to free resources via invalid bus pointer. This happens because sdiodev->bus is set 2 times: first in brcmf_sdio_probe() and second time in brcmf_sdiod_probe(). Fix	2026-05-06	5.5

		this by chaning the brcmf_sdio_probe() function to return the error code and set sdio->bus only there.		
CVE-2026-43145	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: remoteproc: imx_rproc: Fix invalid loaded resource table detection imx_rproc_elf_find_loaded_rsc_table() may incorrectly report a loaded resource table even when the current firmware does not provide one. When the device tree contains a "rsc-table" entry, priv->rsc_table is non-NULL and denotes where a resource table would be located if one is present in memory. However, when the current firmware has no resource table, rproc->table_ptr is NULL. The function still returns priv->rsc_table, and the remoteproc core interprets this as a valid loaded resource table. Fix this by returning NULL from imx_rproc_elf_find_loaded_rsc_table() when there is no resource table for the current firmware (i.e. when rproc->table_ptr is NULL). This aligns the function's semantics with the remoteproc core: a loaded resource table is only reported when a valid table_ptr exists. With this change, starting firmware without a resource table no longer triggers a crash.	2026-05-06	5.5
CVE-2026-43146	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: iris: Add buffer to list only after successful allocation Move `list_add_tail()` to after `dma_alloc_attrs()` succeeds when creating internal buffers. Previously, the buffer was enqueued in `buffers->list` before the DMA allocation. If the allocation failed, the function returned `-ENOMEM` while leaving a partially initialized buffer in the list, which could lead to inconsistent state and potential leaks. By adding the buffer to the list only after `dma_alloc_attrs()` succeeds, we ensure the list contains only valid, fully initialized buffers.	2026-05-06	5.5

CVE-2026-43147	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		Revert "PCI/IOV: Add PCI rescan-remove locking when enabling/disabling SR-IOV"
		
		This reverts commit 05703271c3cd ("PCI/IOV: Add PCI rescan-remove locking when enabling/disabling SR-IOV"), which causes a deadlock by recursively taking pci_rescan_remove_lock when sriov_del_vfs() is called as part of pci_stop_and_remove_bus_device(). For example with the following sequence of commands:
		
		$ echo <NUM> > /sys/bus/pci/devices/<pf>/sriov_numvfs
		$ echo 1 > /sys/bus/pci/devices/<pf>/remove
		
		A trimmed trace of the deadlock on a mlx5 device is as below:
		
		zsh/5715 is trying to acquire lock:
		000002597926ef50 (pci_rescan_remove_lock){+.+.}-{3:3}, at: sriov_disable+0x34/0x140
		
		but task is already holding lock:
		000002597926ef50 (pci_rescan_remove_lock){+.+.}-{3:3}, at: pci_stop_and_remove_bus_device_locked+0x24/0x80
		...
		Call Trace:
		[<00000259778c4f90>] dump_stack_lvl+0xc0/0x110
		[<00000259779c844e>] print_deadlock_bug+0x31e/0x330
		[<00000259779c1908>] __lock_acquire+0x16c8/0x32f0
		[<00000259779bffac>] lock_acquire+0x14c/0x350
		[<00000259789643a6>] __mutex_lock_common+0xe6/0x1520
		[<000002597896413c>] mutex_lock_nested+0x3c/0x50
		[<00000259784a07e4>] sriov_disable+0x34/0x140
		[<00000258f7d6dd80>] mlx5_sriov_disable+0x50/0x80 [mlx5_core]
		[<00000258f7d5745e>] remove_one+0x5e/0xf0 [mlx5_core]
		[<00000259784857fc>] pci_device_remove+0x3c/0xa0
		[<000002597851012e>] device_release_driver_internal+0x18e/0x280
		[<000002597847ae22>] pci_stop_bus_device+0x82/0xa0
		[<000002597847afce>] pci_stop_and_remove_bus_device_locked+0x5e/0x80
		[<00000259784972c2>] remove_store+0x72/0x90
		[<0000025977e6661a>] kernfs_fop_write_iter+0x15a/0x200
		[<0000025977d7241c>] vfs_write+0x24c/0x300
		[<0000025977d72696>] ksys_write+0x86/0x110
		[<000002597895b61c>] __do_syscall+0x14c/0x400
		[<000002597896e0ee>] system_call+0x6e/0x90
			2026-05-06	5.5

		This alone is not a complete fix as it restores the issue the cited commit tried to solve. A new fix will be provided as a follow on.		
CVE-2026-43148	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: powerpc/smp: Add check for kcalloc() failure in parse_thread_groups() As kcalloc() may fail, check its return value to avoid a NULL pointer dereference when passing it to of_property_read_u32_array().	2026-05-06	5.5
CVE-2026-43149	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: wan/fsl_ucc_hdlc: Fix dma_free_coherent() in uhdlc_memclean() The priv->rx_buffer and priv->tx_buffer are alloc'd together as contiguous buffers in uhdlc_init() but freed as two buffers in uhdlc_memclean(). Change the cleanup to only call dma_free_coherent() once on the whole buffer.	2026-05-06	5.5
CVE-2026-43151	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Revert "media: iris: Add sanity check for stop streaming" This reverts commit ad699fa78b59241c9d71a8cafb51525f3dab04d4. Revert the check that skipped stop_streaming when the instance was in IRIS_INST_ERROR, as it caused multiple regressions: 1. Buffers were not returned to vb2 when the instance was already in error state, triggering warnings in the vb2 core because buffer completion was skipped. 2. If a session failed early (e.g. unsupported configuration), the instance transitioned to IRIS_INST_ERROR. When userspace attempted to stop streaming for cleanup, stop_streaming was skipped due to the added check, preventing proper teardown and leaving the firmware in an inconsistent state.	2026-05-06	5.5
CVE-2026-43152	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: hid-pl: handle probe errors Errors in init must be reported back or we'll follow a NULL pointer the first time FF is used.	2026-05-06	5.5
CVE-2026-43154	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: erofs: fix incorrect early exits in volume label handling Crafted EROFS images containing valid volume labels can trigger incorrect early returns, leading to folio reference leaks. However, this does not cause system crashes or other severe issues.	2026-05-06	5.5
CVE-2026-43155	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mux: mmio: fix regmap leak on probe failure The mmio regmap that may be allocated during probe is never freed. Switch to using the device managed allocator so that the regmap is released on probe failures (e.g. probe deferral) and on driver unbind.	2026-05-06	5.5
CVE-2026-43156	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: usb: pegasus: enable basic endpoint checking pegasus_probe() fills URBs with hardcoded endpoint pipes without verifying the endpoint descriptors: - usb_rcvbulkpipe(dev, 1) for RX data - usb_sndbulkpipe(dev, 2) for TX data - usb_rcvintpipe(dev, 3) for status interrupts A malformed USB device can present these endpoints with transfer types that differ from what the driver assumes. Add a pegasus_usb_ep enum for endpoint numbers, replacing magic constants throughout. Add usb_check_bulk_endpoints() and usb_check_int_endpoints() calls before any resource allocation to verify endpoint types before use, rejecting devices with mismatched descriptors at probe time, and avoid triggering assertion. Similar fix to	2026-05-06	5.5

		- commit 90b7f2961798 ("net: usb: rtl8150: enable basic endpoint checking") - commit 9e7021d2aeae ("net: usb: catc: enable basic endpoint checking")		
CVE-2026-43157	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: octeontx2-af: CGX: fix bitmap leaks The RX/TX flow-control bitmaps (rx_fc_pfvf_bmap and tx_fc_pfvf_bmap) are allocated by cgx_lmac_init() but never freed in cgx_lmac_exit(). Unbinding and rebinding the driver therefore triggers kmemleak: unreferenced object (size 16): backtrace: rvu_alloc_bitmap cgx_probe Free both bitmaps during teardown.	2026-05-06	5.5
CVE-2026-43159	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: fix null dereference in find_network The variable pwlan has the possibility of being NULL when passed into rtw_free_network_nolock() which would later dereference the variable.	2026-05-06	5.5
CVE-2026-43160	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mfd: macsmc: Initialize mutex Initialize struct apple_smc's mutex in apple_smc_probe(). Using the mutex uninitialized surprisingly resulted only in occasional NULL pointer dereferences in apple_smc_read() calls from the probe() functions of sub devices.	2026-05-06	5.5
CVE-2026-43161	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Skip dev-iotlb flush for inaccessible PCIe device without scalable mode PCIe endpoints with ATS enabled and passed through to userspace (e.g., QEMU, DPDK) can hard-lock the host when their link drops, either by surprise removal or by a link fault. Commit 4fc82cd907ac ("iommu/vt-d: Don't issue ATS Invalidation request when device is disconnected") adds pci_dev_is_disconnected() to devtlb_invalidation_with_pasid() so ATS invalidation is skipped only when the device is being safely removed, but it applies only when Intel IOMMU scalable mode is enabled. With scalable mode disabled or unsupported, a system hard-lock occurs when a PCIe endpoint's link drops because the Intel IOMMU waits indefinitely for an ATS invalidation that cannot complete. Call Trace: qi_submit_sync qi_flush_dev_iotlb __context_flush_dev_iotlb.part.0 domain_context_clear_one_cb pci_for_each_dma_alias device_block_translation blocking_domain_attach_dev iommu_deinit_device __iommu_group_remove_device iommu_release_device iommu_bus_notifier blocking_notifier_call_chain bus_notify device_del pci_remove_bus_device pci_stop_and_remove_bus_device pciehp_unconfigure_device pciehp_disable_slot pciehp_handle_presence_or_link_change pciehp_ist Commit 81e921fd3216 ("iommu/vt-d: Fix NULL domain on device release") adds intel_pasid_teardown_sm_context() to intel_iommu_release_device(), which calls qi_flush_dev_iotlb() and can also hard-lock the system when a PCIe endpoint's link drops. Call Trace: qi_submit_sync qi_flush_dev_iotlb __context_flush_dev_iotlb.part.0 intel_context_flush_no_pasid	2026-05-06	5.5

		device_pasid_table_teardown pci_pasid_table_teardown pci_for_each_dma_alias intel_pasid_teardown_sm_context intel_iommu_release_device iommu_deinit_device __iommu_group_remove_device iommu_release_device iommu_bus_notifier blocking_notifier_call_chain bus_notify device_del pci_remove_bus_device pci_stop_and_remove_bus_device pciehp_unconfigure_device pciehp_disable_slot pciehp_handle_presence_or_link_change pciehp_ist Sometimes the endpoint loses connection without a link-down event (e.g., due to a link fault); killing the process (virsh destroy) then hard-locks the host. Call Trace: qi_submit_sync qi_flush_dev_iotlb __context_flush_dev_iotlb.part.0 domain_context_clear_one_cb pci_for_each_dma_alias device_block_translation blocking_domain_attach_dev __iommu_attach_device __iommu_device_set_domain __iommu_group_set_domain_internal iommu_detach_group vfio_iommu_type1_detach_group vfio_group_detach_container vfio_group_fops_release __fput pci_dev_is_disconnected() only covers safe-removal paths; pci_device_is_present() tests accessibility by reading vendor/device IDs and internally calls pci_dev_is_disconnected(). On a ConnectX-5 (8 GT/s, x2) this costs ~70 µs. Since __context_flush_dev_iotlb() is only called on {attach,release}_dev paths (not hot), add pci_device_is_present() there to skip inaccessible devices and avoid the hard-lock.		
CVE-2026-43162	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: tegra-video: Fix memory leak in __tegra_channel_try_format() The state object allocated by __v4l2_subdev_state_alloc() must be freed with __v4l2_subdev_state_free() when it is no longer needed. In __tegra_channel_try_format(), two error paths return directly after v4l2_subdev_call() fails, without freeing the allocated 'sd_state' object. This violates the requirement and causes a memory leak. Fix this by introducing a cleanup label and using goto statements in the error paths to ensure that __v4l2_subdev_state_free() is always called before the function returns.	2026-05-06	5.5
CVE-2026-43165	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hwmon: (nct7363) Fix a resource leak in nct7363_present_pwm_fanin When calling of_parse_phandle_with_args(), the caller is responsible to call of_node_put() to release the reference of device node. In nct7363_present_pwm_fanin, it does not release the reference, causing a resource leak.	2026-05-06	5.5
CVE-2026-43167	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: always flush state and policy upon NETDEV_UNREGISTER event syzbot is reporting that "struct xfrm_state" refcount is leaking. unregister_netdevice: waiting for netdevsim0 to become free. Usage count = 2 ref_tracker: netdev@ffff888052f24618 has 1/1 users at __netdev_tracker_alloc include/linux/netdevice.h:4400 [inline] netdev_tracker_alloc include/linux/netdevice.h:4412 [inline]	2026-05-06	5.5

		xfrm_dev_state_add+0x3a5/0x1080 net/xfrm/xfrm_device.c:316 xfrm_state_construct net/xfrm/xfrm_user.c:986 [inline] xfrm_add_sa+0x34ff/0x5fa0 net/xfrm/xfrm_user.c:1022 xfrm_user_rcv_msg+0x58e/0xc00 net/xfrm/xfrm_user.c:3507 netlink_rcv_skb+0x158/0x420 net/netlink/af_netlink.c:2550 xfrm_netlink_rcv+0x71/0x90 net/xfrm/xfrm_user.c:3529 netlink_unicast_kernel net/netlink/af_netlink.c:1318 [inline] netlink_unicast+0x5aa/0x870 net/netlink/af_netlink.c:1344 netlink_sendmsg+0x8c8/0xdd0 net/netlink/af_netlink.c:1894 sock_sendmsg_nosec net/socket.c:727 [inline] __sock_sendmsg net/socket.c:742 [inline] ____sys_sendmsg+0xa5d/0xc30 net/socket.c:2592 ___sys_sendmsg+0x134/0x1d0 net/socket.c:2646 __sys_sendmsg+0x16d/0x220 net/socket.c:2678 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f This is because commit d77e38e612a0 ("xfrm: Add an IPsec hardware offloading API") implemented xfrm_dev_unregister() as no-op despite xfrm_dev_state_add() from xfrm_state_construct() acquires a reference to "struct net_device". I guess that that commit expected that NETDEV_DOWN event is fired before NETDEV_UNREGISTER event fires, and also assumed that xfrm_dev_state_add() is called only if (dev->features & NETIF_F_HW_ESP) != 0. Sabrina Dubroca identified steps to reproduce the same symptoms as below. echo 0 > /sys/bus/netdevsim/new_device dev=$(ls -1 /sys/bus/netdevsim/devices/netdevsim0/net/) ip xfrm state add src 192.168.13.1 dst 192.168.13.2 proto esp \ spi 0x1000 mode tunnel aead 'rfc4106(gcm(aes))' $key 128 \ offload crypto dev $dev dir out ethtool -K $dev esp-hw-offload off echo 0 > /sys/bus/netdevsim/del_device Like these steps indicate, the NETIF_F_HW_ESP bit can be cleared after xfrm_dev_state_add() acquired a reference to "struct net_device". Also, xfrm_dev_state_add() does not check for the NETIF_F_HW_ESP bit when acquiring a reference to "struct net_device". Commit 03891f820c21 ("xfrm: handle NETDEV_UNREGISTER for xfrm device") re-introduced the NETDEV_UNREGISTER event to xfrm_dev_event(), but that commit for unknown reason chose to share xfrm_dev_down() between the NETDEV_DOWN event and the NETDEV_UNREGISTER event. I guess that that commit missed the behavior in the previous paragraph. Therefore, we need to re-introduce xfrm_dev_unregister() in order to release the reference to "struct net_device" by unconditionally flushing state and policy.		
CVE-2026-43168	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix reflink preserve cleanup issue commit c06c303832ec ("ocfs2: fix xattr array entry __counted_by error") doesn't handle all cases and the cleanup job for preserved xattr entries still has bug: - the 'last' pointer should be shifted by one unit after cleanup an array entry. - current code logic doesn't cleanup the first entry when xh_count is 1. Note, commit c06c303832ec is also a bug fix for 0fe9b66c65f3.	2026-05-06	5.5
CVE-2026-43169	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/buddy: Prevent BUG_ON by validating rounded allocation When DRM_BUDDY_CONTIGUOUS_ALLOCATION is set, the requested size is rounded up to the next power-of-two via roundup_pow_of_two(). Similarly, for non-contiguous allocations with large min_block_size, the size is aligned up via round_up(). Both operations can produce a rounded size that exceeds mm->size, which later triggers BUG_ON(order > mm->max_order). Example scenarios: - 9G CONTIGUOUS allocation on 10G VRAM memory: roundup_pow_of_two(9G) = 16G > 10G - 9G allocation with 8G min_block_size on 10G VRAM memory: round_up(9G, 8G) = 16G > 10G Fix this by checking the rounded size against mm->size. For	2026-05-06	5.5

		non-contiguous or range allocations where size > mm->size is invalid, return -EINVAL immediately. For contiguous allocations without range restrictions, allow the request to fall through to the existing __alloc_contig_try_harder() fallback. This ensures invalid user input returns an error or uses the fallback path instead of hitting BUG_ON. v2: (Matt A) - Add Fixes, Cc stable, and Closes tags for context		
CVE-2026-43170	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: Move vbus draw to workqueue context Currently dwc3_gadget_vbus_draw() can be called from atomic context, which in turn invokes power-supply-core APIs. And some these PMIC APIs have operations that may sleep, leading to kernel panic. Fix this by moving the vbus_draw into a workqueue context.	2026-05-06	5.5
CVE-2026-43171	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: EFI/CPER: don't dump the entire memory region The current logic at cper_print_fw_err() doesn't check if the error record length is big enough to handle offset. On a bad firmware, if the ofset is above the actual record, length -= offset will underflow, making it dump the entire memory. The end result can be: - the logic taking a lot of time dumping large regions of memory; - data disclosure due to the memory dumps; - an OOPS, if it tries to dump an unmapped memory region. Fix it by checking if the section length is too small before doing a hex dump. [rjw: Subject tweaks]	2026-05-06	5.5
CVE-2026-43173	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: xscale: Check for PTP support properly In ixp4xx_get_ts_info() ixp46x_ptp_find() is called unconditionally despite this feature only existing on ixp46x, leading to the following splat from tcpdump: root@OpenWrt:~# tcpdump -vv -X -i eth0 (...) Unable to handle kernel NULL pointer dereference at virtual address 00000238 when read (...) Call trace: ptp_clock_index from ixp46x_ptp_find+0x1c/0x38 ixp46x_ptp_find from ixp4xx_get_ts_info+0x4c/0x64 ixp4xx_get_ts_info from __ethtool_get_ts_info+0x90/0x108 __ethtool_get_ts_info from __dev_ethtool+0xa00/0x2648 __dev_ethtool from dev_ethtool+0x160/0x234 dev_ethtool from dev_ioctl+0x2cc/0x460 dev_ioctl from sock_ioctl+0x1ec/0x524 sock_ioctl from sys_ioctl+0x51c/0xa94 sys_ioctl from ret_fast_syscall+0x0/0x44 (...) Segmentation fault Check for ixp46x in ixp46x_ptp_find() before trying to set up PTP to avoid this. To avoid altering the returned error code from ixp4xx_hwtstamp_set() which before this patch was -EOPNOTSUPP, we return -EOPNOTSUPP from ixp4xx_hwtstamp_set() if ixp46x_ptp_find() fails no matter the error code. The helper function ixp46x_ptp_find() helper returns -ENODEV.	2026-05-06	5.5
CVE-2026-43174	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: io_uring/zcrx: fix post open error handling Closing a queue doesn't guarantee that all associated page pools are terminated right away, let the refcounting do the work instead of releasing the zcrx ctx directly.	2026-05-06	5.5

CVE-2026-43175	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: clk: rs9: Reserve 8 struct clk_hw slots for for 9FGV0841 The 9FGV0841 has 8 outputs and registers 8 struct clk_hw, make sure there are 8 slots for those newly registered clk_hw pointers, else there is going to be out of bounds write when pointers 4..7 are set into struct rs9_driver_data .clk_dif[4..7] field. Since there are other structure members past this struct clk_hw pointer array, writing to .clk_dif[4..7] fields corrupts both the struct rs9_driver_data content and data around it, sometimes without crashing the kernel. However, the kernel does surely crash when the driver is unbound or during suspend. Fix this, increase the struct clk_hw pointer array size to the maximum output count of 9FGV0841, which is the biggest chip that is supported by this driver.	2026-05-06	5.5
CVE-2026-43177	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: ipu6: Fix RPM reference leak in probe error paths Several error paths in ipu6_pci_probe() were jumping directly to out_ipu6_bus_del_devices without releasing the runtime PM reference. Add pm_runtime_put_sync() before cleaning up other resources.	2026-05-06	5.5
CVE-2026-43179	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: erofs: fix incorrect early exits for invalid metabox-enabled images Crafted EROFS images with metadata compression enabled can trigger incorrect early returns, leading to folio reference leaks. However, this does not cause system crashes or other severe issues.	2026-05-06	5.5
CVE-2026-43181	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio: sysfs: fix chip removal with GPIOs exported over sysfs Currently if we export a GPIO over sysfs and unbind the parent GPIO controller, the exported attribute will remain under /sys/class/gpio because once we remove the parent device, we can no longer associate the descriptor with it in gpiod_unexport() and never drop the final reference. Rework the teardown code: provide an unlocked variant of gpiod_unexport() and remove all exported GPIOs with the sysfs_lock taken before unregistering the parent device itself. This is done to prevent any new exports happening before we unregister the device completely.	2026-05-06	5.5
CVE-2026-43182	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: ccs: Avoid possible division by zero Calculating maximum M for scaler configuration involves dividing by MIN_X_OUTPUT_SIZE limit register's value. Albeit the value is presumably non-zero, the driver was missing the check it in fact was. Fix this.	2026-05-06	5.5
CVE-2026-43183	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: cx25821: Fix a resource leak in cx25821_dev_setup() Add release_mem_region() if ioremap() fails to release the memory region obtained by cx25821_get_resources().	2026-05-06	5.5
CVE-2026-43188	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ceph: do not propagate page array emplacement errors as batch errors When fscrypt is enabled, move_dirty_folio_in_page_array() may fail because it needs to allocate bounce buffers to store the encrypted versions of each folio. Each folio beyond the first allocates its bounce buffer with GFP_NOWAIT. Failures are common (and expected) under this allocation mode; they should flush (not abort) the batch. However, ceph_process_folio_batch() uses the same `rc` variable for its own return code and for capturing the return codes of its routine calls; failing to reset `rc` back to 0 results in the error being propagated out to the main writeback loop, which cannot actually tolerate any errors here: once `ceph_wbc.pages` is allocated, it must be passed to ceph_submit_write() to be freed. If it survives until the next iteration (e.g. due to the goto being followed), ceph_allocate_page_array()'s BUG_ON() will oops the worker. Note that this failure mode is currently masked due to another bug	2026-05-06	5.5

		(addressed next in this series) that prevents multiple encrypted folios from being selected for the same write. For now, just reset `rc` when redirtying the folio to prevent errors in move_dirty_folio_in_page_array() from propagating. Note that move_dirty_folio_in_page_array() is careful never to return errors on the first folio, so there is no need to check for that. After this change, ceph_process_folio_batch() no longer returns errors; its only remaining failure indicator is `locked_pages == 0`, which the caller already handles correctly.		
CVE-2026-43189	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-async: Fix error handling on steps after finding a match Once an async connection is found to be matching with an fwnode, a sub-device may be registered (in case it wasn't already), its bound operation is called, ancillary links are created, the async connection is added to the sub-device's list of connections and removed from the global waiting connection list. Further on, the sub-device's possible own notifier is searched for possible additional matches. Fix these specific issues: - If v4l2_async_match_notify() failed before the sub-notifier handling, the async connection was unbound and its entry removed from the sub-device's async connection list. The latter part was also done in v4l2_async_match_notify(). - The async connection's sd field was only set after creating ancillary links in v4l2_async_match_notify(). It was however dereferenced in v4l2_async_unbind_subdev_one(), which was called on error path of v4l2_async_match_notify() failure.	2026-05-06	5.5
CVE-2026-43191	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Adjust PHY FSM transition to TX_EN-to-PLL_ON for TMDS on DCN35 [Why] A backport of the change made for DCN401 that addresses an issue where we turn off the PHY PLL when disabling TMDS output, which causes the OTG to remain stuck. The OTG being stuck can lead to a hang in the DCHVM's ability to ACK invalidations when it thinks the HUBP is still on but it's not receiving global sync. The transition to PLL_ON needs to be atomic as there's no guarantee that the thread isn't pre-empted or is able to complete before the IOMMU watchdog times out. [How] Backport the implementation from dcn401 back to dcn35. There's a functional difference in when the eDP output is disabled in dcn401 code so we don't want to utilize it directly.	2026-05-06	5.5
CVE-2026-43192	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: dm mpath: Add missing dm_put_device when failing to get scsi dh name When commit fd81bc5cca8f ("scsi: device_handler: Return error pointer in scsi_dh_attached_handler_name()") added code to fail parsing the path if scsi_dh_attached_handler_name() failed with -ENOMEM, it didn't clean up the reference to the path device that had just been taken. Fix this, and steamline the error paths of parse_path() a little.	2026-05-06	5.5
CVE-2026-43193	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix nfs4_file refcount leak in nfsd_get_dir_deleg() Claude pointed out that there is a nfs4_file refcount leak in nfsd_get_dir_deleg(). Ensure that the reference to "fp" is released before returning.	2026-05-06	5.5
CVE-2026-43195	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: validate user queue size constraints Add validation to ensure user queue sizes meet hardware requirements: - Size must be a power of two for efficient ring buffer wrapping - Size must be at least AMDGPU_GPU_PAGE_SIZE to prevent undersized allocations This prevents invalid configurations that could lead to GPU faults or unexpected behavior.	2026-05-06	5.5

CVE-2026-43200	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: Fix swapped parameters in pci_{primary/secondary}_epc_epf_unlink() functions struct configfs_item_operations callbacks are defined like the following: int (*allow_link)(struct config_item *src, struct config_item *target); void (*drop_link)(struct config_item *src, struct config_item *target); While pci_primary_epc_epf_link() and pci_secondary_epc_epf_link() specify the parameters in the correct order, pci_primary_epc_epf_unlink() and pci_secondary_epc_epf_unlink() specify the parameters in the wrong order, leading to the below kernel crash when using the unlink command in configfs: Unable to handle kernel paging request at virtual address 0000000300000857 Mem abort info: ... pc : string+0x54/0x14c lr : vsnprintf+0x280/0x6e8 ... string+0x54/0x14c vsnprintf+0x280/0x6e8 vprintk_default+0x38/0x4c vprintk+0xc4/0xe0 pci_epf_unbind+0xdc/0x108 configfs_unlink+0xe0/0x208+0x44/0x74 vfs_unlink+0x120/0x29c __arm64_sys_unlinkat+0x3c/0x90 invoke_syscall+0x48/0x134 do_el0_svc+0x1c/0x30prop.0+0xd0/0xf0 [mani: cced stable, changed commit message as per https://lore.kernel.org/linux-pci/aV9joi3jF1R6ca02@ryzen]	2026-05-06	5.5
CVE-2026-43201	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: APEI/GHES: ARM processor Error: don't go past allocated memory If the BIOS generates a very small ARM Processor Error, or an incomplete one, the current logic will fail to deferrence err->section_length and ctx_info->size Add checks to avoid that. With such changes, such GHESv2 records won't cause OOPSes like this: [1.492129] Internal error: Oops: 0000000096000005 [#1] SMP [1.495449] Modules linked in: [1.495820] CPU: 0 UID: 0 PID: 9 Comm: kworker/0:0 Not tainted 6.18.0-rc1-00017-gabadcc3553dd-dirty #18 PREEMPT [1.496125] Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 02/02/2022 [1.496433] Workqueue: kacpi_notify acpi_os_execute_deferred [1.496967] pstate: 814000c5 (Nzcv daIF +PAN -UAO -TCO +DIT -SSBS BTYPE=--) [1.497199] pc : log_arm_hw_error+0x5c/0x200 [1.497380] lr : ghes_handle_arm_hw_error+0x94/0x220 0xffff8000811c5324 is in log_arm_hw_error (../drivers/ras/ras.c:75). 70 err_info = (struct cper_arm_err_info *)(err + 1); 71 ctx_info = (struct cper_arm_ctx_info *)(err_info + err->err_info_num); 72 ctx_err = (u8 *)ctx_info; 73 74 for (n = 0; n < err->context_info_num; n++) { 75 sz = sizeof(struct cper_arm_ctx_info) + ctx_info->size; 76 ctx_info = (struct cper_arm_ctx_info *)((long)ctx_info + sz); 77 ctx_len += sz; 78 } 79 and similar ones while trying to access section_length on an error dump with too small size. [rjw: Subject tweaks]	2026-05-06	5.5
CVE-2026-43202	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fbdev: vt8500lcdfb: fix missing dma_free_coherent() fbi->fb.screen_buffer is allocated with dma_alloc_coherent() but is not freed if the error path is reached.	2026-05-06	5.5

CVE-2026-43204	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: q6asm: drop DSP responses for closed data streams 'Commit a354f030dbce ("ASoC: qcom: q6asm: handle the responses after closing")' attempted to ignore DSP responses arriving after a stream had been closed. However, those responses were still handled, causing lockups. Fix this by unconditionally dropping all DSP responses associated with closed data streams.	2026-05-06	5.5
CVE-2026-43209	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: minix: Add required sanity checking to minix_check_superblock() The fs/minix implementation of the minix filesystem does not currently support any other value for s_log_zone_size than 0. This is also the only value supported in util-linux; see mkfs.minix.c line 511. In addition, this patch adds some sanity checking for the other minix superblock fields, and moves the minix_blocks_needed() checks for the zmap and imap also to minix_check_super_block(). This also closes a related syzbot bug report.	2026-05-06	5.5
CVE-2026-43210	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tracing: ring-buffer: Fix to check event length before using Check the event length before adding it for accessing next index in rb_read_data_buffer(). Since this function is used for validating possibly broken ring buffers, the length of the event could be broken. In that case, the new event (e + len) can point a wrong address. To avoid invalid memory access at boot, check whether the length of each event is in the possible range before using it.	2026-05-06	5.5
CVE-2026-43216	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: Drop the lock in skb_may_tx_timestamp() skb_may_tx_timestamp() may acquire sock::sk_callback_lock. The lock must not be taken in IRQ context, only softirq is okay. A few drivers receive the timestamp via a dedicated interrupt and complete the TX timestamp from that handler. This will lead to a deadlock if the lock is already write-locked on the same CPU. Taking the lock can be avoided. The socket (pointed by the skb) will remain valid until the skb is released. The ->sk_socket and ->file member will be set to NULL once the user closes the socket which may happen before the timestamp arrives. If we happen to observe the pointer while the socket is closing but before the pointer is set to NULL then we may use it because both pointer (and the file's cred member) are RCU freed. Drop the lock. Use READ_ONCE() to obtain the individual pointer. Add a matching WRITE_ONCE() where the pointer are cleared.	2026-05-06	5.5
CVE-2026-43217	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: iris: gen2: Add sanity check for session stop In iris_kill_session, inst->state is set to IRIS_INST_ERROR and session_close is executed, which will kfree(inst_hfi_gen2->packet). If stop_streaming is called afterward, it will cause a crash. Add a NULL check for inst_hfi_gen2->packet before sendling STOP packet to firmware to fix that.	2026-05-06	5.5
CVE-2026-43218	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: i2c/tw9903: Fix potential memory leak in tw9903_probe() In one of the error paths in tw9903_probe(), the memory allocated in v4l2_ctrl_handler_init() and v4l2_ctrl_new_std() is not freed. Fix that by calling v4l2_ctrl_handler_free() on the handler in that error path.	2026-05-06	5.5
CVE-2026-43219	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: cpsw_new: Fix potential unregister of netdev that has not been registered yet If an error occurs during register_netdev() for the first MAC in cpsw_register_ports(), even though cpsw->slaves[0].ndev is set to NULL, cpsw->slaves[1].ndev would remain unchanged. This could later cause cpsw_unregister_ports() to attempt unregistering the second MAC. To address this, add a check for ndev->reg_state before calling	2026-05-06	5.5

		unregister_netdev(). With this change, setting cpsw->slaves[i].ndev to NULL becomes unnecessary and can be removed accordingly.		
CVE-2026-43220	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: serialize sequence allocation under concurrent TLB invalidations With concurrent TLB invalidations, completion wait randomly gets timed out because cmd_sem_val was incremented outside the IOMMU spinlock, allowing CMD_COMPL_WAIT commands to be queued out of sequence and breaking the ordering assumption in wait_on_sem(). Move the cmd_sem_val increment under iommu->lock so completion sequence allocation is serialized with command queuing. And remove the unnecessary return.	2026-05-06	5.5
CVE-2026-43221	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipmi: ipmb: initialise event handler read bytes IPMB doesn't use i2c reads, but the handler needs to set a value. Otherwise an i2c read will return an uninitialised value from the bus driver.	2026-05-06	5.5
CVE-2026-43223	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: pvrusb2: fix URB leak in pvr2_send_request_ex When pvr2_send_request_ex() submits a write URB successfully but fails to submit the read URB (e.g. returns -ENOMEM), it returns immediately without waiting for the write URB to complete. Since the driver reuses the same URB structure, a subsequent call to pvr2_send_request_ex() attempts to submit the still-active write URB, triggering a 'URB submitted while active' warning in usb_submit_urb(). Fix this by ensuring the write URB is unlinked and waited upon if the read URB submission fails.	2026-05-06	5.5
CVE-2026-43224	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring/zcrx: fix sgtable leak on mapping failures In an unlikely case when io_populate_area_dma() fails, which could only happen on a PAGE_POOL_32BIT_ARCH_WITH_64BIT_DMA machine, io_zcrx_map_area() will have an initialised and not freed table. It was supposed to be cleaned up in the error path, but !is_mapped prevents that.	2026-05-06	5.5
CVE-2026-43225	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: fix memory leak on failure path cfg80211_inform_bss_frame() may return NULL on failure. In that case, the allocated buffer 'buf' is not freed and the function returns early, leading to potential memory leak. Fix this by ensuring that 'buf' is freed on both success and failure paths.	2026-05-06	5.5
CVE-2026-43227	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: clocksource/drivers/sh_tmu: Always leave device running after probe The TMU device can be used as both a clocksource and a clockevent provider. The driver tries to be smart and power itself on and off, as well as enabling and disabling its clock when it's not in operation. This behavior is slightly altered if the TMU is used as an early platform device in which case the device is left powered on after probe, but the clock is still enabled and disabled at runtime. This has worked for a long time, but recent improvements in PREEMPT_RT and PROVE_LOCKING have highlighted an issue. As the TMU registers itself as a clockevent provider, clockevents_register_device(), it needs to use raw spinlocks internally as this is the context of which the clockevent framework interacts with the TMU driver. However in the context of holding a raw spinlock the TMU driver can't really manage its power state or clock with calls to pm_runtime_*() and clk_*() as these calls end up in other platform drivers using regular spinlocks to control power and clocks. This mix of spinlock contexts trips a lockdep warning. ============================= [BUG: Invalid wait context] 6.18.0-arm64-renesas-09926-gee959e7c5e34 #1 Not tainted ----------------------------- swapper/0/0 is trying to lock: ffff000008c9e180 (&dev->power.lock){-...}-{3:3}, at: __pm_runtime_resume+0x38/0x88 other info that might help us debug this:	2026-05-06	5.5

		context-{5:5} 1 lock held by swapper/0/0: ccree e6601000.crypto: ARM CryptoCell 630P Driver: HW version 0xAF400001/0xDCC63000, Driver version 5.0 #0: ffff8000817ec298 ccree e6601000.crypto: ARM ccree device initialized (tick_broadcast_lock){-...}-{2:2}, at: __tick_broadcast_oneshot_control+0xa4/0x3a8 stack backtrace: CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.18.0-arm64-renesas-09926-gee959e7c5e34 #1 PREEMPT Hardware name: Renesas Salvator-X 2nd version board based on r8a77965 (DT) Call trace: show_stack+0x14/0x1c (C) dump_stack_lvl+0x6c/0x90 dump_stack+0x14/0x1c __lock_acquire+0x904/0x1584 lock_acquire+0x220/0x34c _raw_spin_lock_irqsave+0x58/0x80 __pm_runtime_resume+0x38/0x88 sh_tmu_clock_event_set_oneshot+0x84/0xd4 clockevents_switch_state+0xfc/0x13c tick_broadcast_set_event+0x30/0xa4 __tick_broadcast_oneshot_control+0x1e0/0x3a8 tick_broadcast_oneshot_control+0x30/0x40 cpuidle_enter_state+0x40c/0x680 cpuidle_enter+0x30/0x40 do_idle+0x1f4/0x280 cpu_startup_entry+0x34/0x40 kernel_init+0x0/0x130 do_one_initcall+0x0/0x230 __primary_switched+0x88/0x90 For non-PREEMPT_RT builds this is not really an issue, but for PREEMPT_RT builds where normal spinlocks can sleep this might be an issue. Be cautious and always leave the power and clock running after probe.		
CVE-2026-43228	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: hfs: Replace BUG_ON with error handling for CNID count checks In a06ec283e125 next_id, folder_count, and file_count in the super block info were expanded to 64 bits, and BUG_ONs were added to detect overflow. This triggered an error reported by syzbot: if the MDB is corrupted, the BUG_ON is triggered. This patch replaces this mechanism with proper error handling and resolves the syzbot reported bug. Singed-off-by: Jori Koolstra <jkoolstra@xs4all.nl>	2026-05-06	5.5
CVE-2026-43229	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Fix device cleanup order to prevent kernel panic Move video device unregistration to the beginning of the remove function to ensure all video operations are stopped before cleaning up the worker thread and disabling PM runtime. This prevents hardware register access after the device has been powered down. In polling mode, the hrtimer periodically triggers wave5_vpu_timer_callback() which queues work to the kthread worker. The worker executes wave5_vpu_irq_work_fn() which reads hardware registers via wave5_vdi_read_register(). The original cleanup order disabled PM runtime and powered down hardware before unregistering video devices. When autosuspend triggers and powers off the hardware, the video devices are still registered and the worker thread can still be triggered by the hrtimer, causing it to attempt reading registers from powered-off hardware. This results in a bus error (synchronous external abort) and kernel panic. This causes random kernel panics during encoding operations: Internal error: synchronous external abort: 0000000096000010 [#1] PREEMPT SMP Modules linked in: wave5 rpmsg_ctrl rpmsg_char ... CPU: 0 UID: 0 PID: 1520 Comm: vpu_irq_thread Tainted: G M W pc : wave5_vdi_read_register+0x10/0x38 [wave5] lr : wave5_vpu_irq_work_fn+0x28/0x60 [wave5] Call trace: wave5_vdi_read_register+0x10/0x38 [wave5] kthread_worker_fn+0xd8/0x238	2026-05-06	5.5

		kthread+0x104/0x120 ret_from_fork+0x10/0x20 Code: aa1e03e9 d503201f f9416800 8b214000 (b9400000) ---[end trace 0000000000000000]--- Kernel panic - not syncing: synchronous external abort: Fatal exception		
CVE-2026-43231	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: radio-keene: fix memory leak in error path Fix a memory leak in usb_keene_probe(). The v4l2 control handler is initialized and controls are added, but if v4l2_device_register() or video_register_device() fails afterward, the handler was never freed, leaking memory. Add v4l2_ctrl_handler_free() call in the err_v4l2 error path to ensure the control handler is properly freed for all error paths after it is initialized.	2026-05-06	5.5
CVE-2026-43234	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: team: avoid NETDEV_CHANGEMTU event when unregistering slave syzbot is reporting unregister_netdevice: waiting for netdevsim0 to become free. Usage count = 3 ref_tracker: netdev@ffff88807dcf8618 has 1/2 users at __netdev_tracker_alloc include/linux/netdevice.h:4400 [inline] netdev_hold include/linux/netdevice.h:4429 [inline] inetdev_init+0x201/0x4e0 net/ipv4/devinet.c:286 inetdev_event+0x251/0x1610 net/ipv4/devinet.c:1600 notifier_call_chain+0x19d/0x3a0 kernel/notifier.c:85 call_netdevice_notifiers_mtu net/core/dev.c:2318 [inline] netif_set_mtu_ext+0x5aa/0x800 net/core/dev.c:9886 netif_set_mtu+0xd7/0x1b0 net/core/dev.c:9907 dev_set_mtu+0x126/0x260 net/core/dev_api.c:248 team_port_del+0xb07/0xcb0 drivers/net/team/team_core.c:1333 team_del_slave drivers/net/team/team_core.c:1936 [inline] team_device_event+0x207/0x5b0 drivers/net/team/team_core.c:2929 notifier_call_chain+0x19d/0x3a0 kernel/notifier.c:85 call_netdevice_notifiers_extack net/core/dev.c:2281 [inline] call_netdevice_notifiers net/core/dev.c:2295 [inline] __dev_change_net_namespace+0xcb7/0x2050 net/core/dev.c:12592 do_setlink+0x2ce/0x4590 net/core/rtnetlink.c:3060 rtnl_changelink net/core/rtnetlink.c:3776 [inline] __rtnl_newlink net/core/rtnetlink.c:3935 [inline] rtnl_newlink+0x15a9/0x1be0 net/core/rtnetlink.c:4072 rtnetlink_rcv_msg+0x7d5/0xbe0 net/core/rtnetlink.c:6958 netlink_rcv_skb+0x232/0x4b0 net/netlink/af_netlink.c:2550 netlink_unicast_kernel net/netlink/af_netlink.c:1318 [inline] netlink_unicast+0x80f/0x9b0 net/netlink/af_netlink.c:1344 netlink_sendmsg+0x813/0xb40 net/netlink/af_netlink.c:1894 problem. Ido Schimmel found steps to reproduce ip link add name team1 type team ip link add name dummy1 mtu 1499 master team1 type dummy ip netns add ns1 ip link set dev dummy1 netns ns1 ip -n ns1 link del dev dummy1 and also found that the same issue was fixed in the bond driver in commit f51048c3e07b ("bonding: avoid NETDEV_CHANGEMTU event when unregistering slave"). Let's do similar thing for the team driver, with commit ad7c7b2172c3 ("net: hold netdev instance lock during sysfs operations") and commit 303a8487a657 ("net: s/__dev_set_mtu/__netif_set_mtu/") also applied.	2026-05-06	5.5
CVE-2026-43235	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: iris: Add missing platform data entries for SM8750 Two platform-data fields for SM8750 were missed: - get_vpu_buffer_size = iris_vpu33_buf_size Without this, the driver fails to allocate the required internal buffers, leading to basic decode/encode failures during session bring-up. - max_core_mbps = ((7680 * 4320) / 256) * 60	2026-05-06	5.5

		Without this capability exposed, capability checks are incomplete and v4l2-compliance for encoder fails.		

CVE-2026-43238	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net/sched: act_skbedit: fix divide-by-zero in tcf_skbedit_hash()
		
		Commit 38a6f0865796 ("net: sched: support hash selecting tx queue") added SKBEDIT_F_TXQ_SKBHASH support. The inclusive range size is computed as:
		
		mapping_mod = queue_mapping_max - queue_mapping + 1;
		
		The range size can be 65536 when the requested range covers all possible u16 queue IDs (e.g. queue_mapping=0 and queue_mapping_max=U16_MAX). That value cannot be represented in a u16 and previously wrapped to 0, so tcf_skbedit_hash() could trigger a divide-by-zero:
		
		queue_mapping += skb_get_hash(skb) % params->mapping_mod;
		
		Compute mapping_mod in a wider type and reject ranges larger than U16_MAX to prevent params->mapping_mod from becoming 0 and avoid the crash.
			2026-05-06	5.5

CVE-2026-43240	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86/kexec: add a sanity check on previous kernel's ima kexec buffer When the second-stage kernel is booted via kexec with a limiting command line such as "mem=<size>", the physical range that contains the carried over IMA measurement list may fall outside the truncated RAM leading to a kernel panic. BUG: unable to handle page fault for address: ffff97793ff47000 RIP: ima_restore_measurement_list+0xdc/0x45a #PF: error_code(0x0000) – not-present page Other architectures already validate the range with page_is_ram(), as done in commit cbf9c4b9617b ("of: check previous kernel's ima-kexec-buffer against memory bounds") do a similar check on x86. Without carrying the measurement list across kexec, the attestation would fail.	2026-05-06	5.5
CVE-2026-43242	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: soc: ti: k3-socinfo: Fix regmap leak on probe failure The mmio regmap allocated during probe is never freed. Switch to using the device managed allocator so that the regmap is released on probe failures (e.g. probe deferral) and on driver unbind.	2026-05-06	5.5
CVE-2026-43243	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add signal type check for dcn401 get_phyd32clk_src Trying to access link enc on a dpia link will cause a crash otherwise	2026-05-06	5.5
CVE-2026-43244	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: kcm: fix zero-frag skb in frag_list on partial sendmsg error Syzkaller reported a warning in kcm_write_msgs() when processing a message with a zero-fragment skb in the frag_list. When kcm_sendmsg() fills MAX_SKB_FRAGS fragments in the current skb, it allocates a new skb (tskb) and links it into the frag_list before copying data. If the copy subsequently fails (e.g. -EFAULT from user memory), tskb remains in the frag_list with zero fragments: head skb (msg being assembled, NOT yet in sk_write_queue) +-----------+ | frags[17] | (MAX_SKB_FRAGS, all filled with data) | frag_list-+--> tskb +-----------+ +----------+ | frags[0] | (empty! copy failed before filling) +----------+ For SOCK_SEQPACKET with partial data already copied, the error path saves this message via partial_message for later completion. For SOCK_SEQPACKET, sock_write_iter() automatically sets MSG_EOR, so a subsequent zero-length write(fd, NULL, 0) completes the message and queues it to sk_write_queue. kcm_write_msgs() then walks the frag_list and hits:	2026-05-06	5.5

		WARN_ON(!skb_shinfo(skb)->nr_frags) TCP has a similar pattern where skbs are enqueued before data copy and cleaned up on failure via tcp_remove_empty_skb(). KCM was missing the equivalent cleanup. Fix this by tracking the predecessor skb (frag_prev) when allocating a new frag_list entry. On error, if the tail skb has zero frags, use frag_prev to unlink and free it in O(1) without walking the singly-linked frag_list. frag_prev is safe to dereference because the entire message chain is only held locally (or in kcm->seq_skb) and is not added to sk_write_queue until MSG_EOR, so the send path cannot free it underneath us. Also change the WARN_ON to WARN_ON_ONCE to avoid flooding the log if the condition is somehow hit repeatedly. There are currently no KCM selftests in the kernel tree; a simple reproducer is available at [1]. [1] https://gist.github.com/mrpre/a94d431c757e8d6f168f4dd1a3749daa		
CVE-2026-43246	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: i2c/tw9906: Fix potential memory leak in tw9906_probe() In one of the error paths in tw9906_probe(), the memory allocated in v4l2_ctrl_handler_init() and v4l2_ctrl_new_std() is not freed. Fix that by calling v4l2_ctrl_handler_free() on the handler in that error path.	2026-05-06	5.5
CVE-2026-43247	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Fix SError of kernel panic when closed SError of kernel panic rarely happened while testing fluster. The root cause was to enter suspend mode because timeout of autosuspend delay happened. [48.834439] SError Interrupt on CPU0, code 0x00000000bf000000 -- SError [48.834455] CPU: 0 UID: 0 PID: 1067 Comm: v4l2h265dec0:sr Not tainted 6.12.9-gc9e21a1ebd75-dirty #7 [48.834461] Hardware name: ti Texas Instruments J721S2 EVM/Texas Instruments J721S2 EVM, BIOS 2025.01-00345-gbaf3aaa8ecfa 01/01/2025 [48.834464] pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [48.834468] pc : wave5_dec_clr_disp_flag+0x40/0x80 [wave5] [48.834488] lr : wave5_dec_clr_disp_flag+0x40/0x80 [wave5] [48.834495] sp : ffff8000856e3a30 [48.834497] x29: ffff8000856e3a30 x28: ffff0008093f6010 x27: ffff000809158130 [48.834504] x26: 0000000000000000 x25: ffff00080b625000 x24: ffff000804a9ba80 [48.834509] x23: ffff000802343028 x22: ffff000809158150 x21: ffff000802218000 [48.834513] x20: ffff0008093f6000 x19: ffff0008093f6000 x18: 0000000000000000 [48.834518] x17: 0000000000000000 x16: 0000000000000000 x15: 0000ffff74009618 [48.834523] x14: 000000010000000c x13: 0000000000000000 x12: 0000000000000000 [48.834527] x11: ffffffffffffffff x10: ffffffffffffffff x9 : ffff000802343028 [48.834532] x8 : ffff00080b6252a0 x7 : 0000000000000038 x6 : 0000000000000000 [48.834536] x5 : ffff00080b625060 x4 : 0000000000000000 x3 : 0000000000000000 [48.834541] x2 : 0000000000000000 x1 : ffff800084bf0118 x0 : ffff800084bf0000 [48.834547] Kernel panic - not syncing: Asynchronous SError Interrupt [48.834549] CPU: 0 UID: 0 PID: 1067 Comm: v4l2h265dec0:sr Not tainted 6.12.9-gc9e21a1ebd75-dirty #7 [48.834554] Hardware name: ti Texas Instruments J721S2 EVM/Texas Instruments J721S2 EVM, BIOS 2025.01-00345-gbaf3aaa8ecfa 01/01/2025 [48.834556] Call trace: [48.834559] dump_backtrace+0x94/0xec [48.834574] show_stack+0x18/0x24 [48.834579] dump_stack_lvl+0x38/0x90 [48.834585] dump_stack+0x18/0x24 [48.834588] panic+0x35c/0x3e0 [48.834592] nmi_panic+0x40/0x8c [48.834595] arm64_serror_panic+0x64/0x70 [48.834598] do_serror+0x3c/0x78 [48.834601] el1h_64_error_handler+0x34/0x4c [48.834605] el1h_64_error+0x64/0x68 [48.834608] wave5_dec_clr_disp_flag+0x40/0x80 [wave5] [48.834615] wave5_vpu_dec_clr_disp_flag+0x54/0x80 [wave5] [48.834622] wave5_vpu_dec_buf_queue+0x19c/0x1a0 [wave5] [48.834628] __enqueue_in_driver+0x3c/0x74 [videobuf2_common] [48.834639] vb2_core_qbuf+0x508/0x61c [videobuf2_common] [48.834646] vb2_qbuf+0xa4/0x168 [videobuf2_v4l2] [48.834656] v4l2_m2m_qbuf+0x80/0x238 [v4l2_mem2mem] [48.834666] v4l2_m2m_ioctl_qbuf+0x18/0x24 [v4l2_mem2mem] [48.834673] v4l_qbuf+0x48/0x5c [videodev]	2026-05-06	5.5

		[48.834704] __video_do_ioctl+0x180/0x3f0 [videodev] [48.834725] video_usercopy+0x2ec/0x68c [videodev] [48.834745] video_ioctl2+0x18/0x24 [videodev] [48.834766] v4l2_ioctl+0x40/0x60 [videodev] [48.834786] __arm64_sys_ioctl+0xa8/0xec [48.834793] invoke_syscall+0x44/0x100 [48.834800] el0_svc_common.constprop.0+0xc0/0xe0 [48.834804] do_el0_svc+0x1c/0x28 [48.834809] el0_svc+0x30/0xd0 [48.834813] el0t_64_sync_handler+0xc0/0xc4 [48.834816] el0t_64_sync+0x190/0x194 [48.834820] SMP: stopping secondary CPUs [48.834831] Kernel Offset: disabled [48.834833] CPU features: 0x08,00002002,80200000,4200421b [48.834837] Memory Limit: none [49.161404] ---[end Kernel panic - not syncing: Asynchronous SError Interrupt]---		
CVE-2026-43251	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: prodikeys: Check presence of pm->input_ep82 Fake USB devices can send their own report descriptors for which the input_mapping() hook does not get called. In this case, pm->input_ep82 stays NULL, which leads to a crash later. This does not happen with the real device, but can be provoked by imposing as one.	2026-05-06	5.5
CVE-2026-43252	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: in-kernel: always set ID as avail when rm endp Syzkaller managed to find a combination of actions that was generating this warning: WARNING: net/mptcp/pm_kernel.c:1074 at __mark_subflow_endp_available net/mptcp/pm_kernel.c:1074 [inline], CPU#1: syz.7.48/2535 WARNING: net/mptcp/pm_kernel.c:1074 at mptcp_pm_nl_fullmesh net/mptcp/pm_kernel.c:1446 [inline], CPU#1: syz.7.48/2535 WARNING: net/mptcp/pm_kernel.c:1074 at mptcp_pm_nl_set_flags_all net/mptcp/pm_kernel.c:1474 [inline], CPU#1: syz.7.48/2535 WARNING: net/mptcp/pm_kernel.c:1074 at mptcp_pm_nl_set_flags+0x5de/0x640 net/mptcp/pm_kernel.c:1538, CPU#1: syz.7.48/2535 Modules linked in: CPU: 1 UID: 0 PID: 2535 Comm: syz.7.48 Not tainted 6.18.0-03987-gea5f5e676cf5 #17 PREEMPT(voluntary) Hardware name: QEMU Ubuntu 25.10 PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 RIP: 0010:__mark_subflow_endp_available net/mptcp/pm_kernel.c:1074 [inline] RIP: 0010:mptcp_pm_nl_fullmesh net/mptcp/pm_kernel.c:1446 [inline] RIP: 0010:mptcp_pm_nl_set_flags_all net/mptcp/pm_kernel.c:1474 [inline] RIP: 0010:mptcp_pm_nl_set_flags+0x5de/0x640 net/mptcp/pm_kernel.c:1538 Code: 89 c7 e8 c5 8c 73 fe e9 f7 fd ff ff 49 83 ef 80 e8 b7 8c 73 fe 4c 89 ff be 03 00 00 00 e8 4a 29 e3 fe eb ac e8 a3 8c 73 fe 90 <0f> 0b 90 e9 3d ff ff ff e8 95 8c 73 fe b8 a1 ff ff ff eb 1a e8 89 RSP: 0018:ffffc9001535b820 EFLAGS: 00010287 netdevsim0: tun_chr_ioctl cmd 1074025677 RAX: ffffffff82da294d RBX: 0000000000000001 RCX: 0000000000080000 RDX: ffffc900096d0000 RSI: 00000000000006d6 RDI: 00000000000006d7 netdevsim0: linktype set to 823 RBP: ffff88802cdb2240 R08: 00000000000104ae R09: ffffffffffffffff R10: ffffffff82da27d4 R11: 0000000000000000 R12: 0000000000000000 R13: ffff88801246d8c0 R14: ffffc9001535b8b8 R15: ffff88802cdb1800 FS: 00007fc6ac5a76c0(0000) GS:ffff8880f90c8000(0000) knlGS:0000000000000000 netlink: 'syz.3.50': attribute type 5 has an invalid length. CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 netlink: 1232 bytes leftover after parsing attributes in process `syz.3.50'. CR2: 0000200000010000 CR3: 0000000025b1a000 CR4: 0000000000350ef0 Call Trace: <TASK> mptcp_pm_set_flags net/mptcp/pm_netlink.c:277 [inline] mptcp_pm_nl_set_flags_doit+0x1d7/0x210 net/mptcp/pm_netlink.c:282 genl_family_rcv_msg_doit+0x117/0x180 net/netlink/genetlink.c:1115 genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline] genl_rcv_msg+0x3a8/0x3f0 net/netlink/genetlink.c:1210 netlink_rcv_skb+0x16d/0x240 net/netlink/af_netlink.c:2550 genl_rcv+0x28/0x40 net/netlink/genetlink.c:1219 netlink_unicast_kernel net/netlink/af_netlink.c:1318 [inline] netlink_unicast+0x3e9/0x4c0 net/netlink/af_netlink.c:1344 netlink_sendmsg+0x4ab/0x5b0 net/netlink/af_netlink.c:1894 sock_sendmsg_nosec net/socket.c:718 [inline] __sock_sendmsg+0xc9/0xf0 net/socket.c:733 ____sys_sendmsg+0x272/0x3b0 net/socket.c:2608	2026-05-06	5.5

		___sys_sendmsg+0x2de/0x320 net/socket.c:2662 __sys_sendmsg net/socket.c:2694 [inline] __do_sys_sendmsg net/socket.c:2699 [inline] __se_sys_sendmsg net/socket.c:2697 [inline] __x64_sys_sendmsg+0x110/0x1a0 net/socket.c:2697 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xed/0x360 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fc6adb66f6d Code: ff c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa 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:00007fc6ac5a6ff8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffffffffda RBX: 00007fc6addf5fa0 RCX: 00007fc6adb66f6d RDX: 0000000000048084 RSI: 00002000000002c0 RDI: 000000000000000e RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 000000000000 ---truncated---		
CVE-2026-43255	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: libertas: fix WARNING in usb_tx_block The function usb_tx_block() submits cardp->tx_urb without ensuring that any previous transmission on this URB has completed. If a second call occurs while the URB is still active (e.g. during rapid firmware loading), usb_submit_urb() detects the active state and triggers a warning: 'URB submitted while active'. Fix this by enforcing serialization: call usb_kill_urb() before submitting the new request. This ensures the URB is idle and safe to reuse.	2026-05-06	5.5
CVE-2026-43257	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: cx88: Add missing unmap in snd_cx88_hw_params() In error path, add cx88_alsa_dma_unmap() to release resource acquired by cx88_alsa_dma_map().	2026-05-06	5.5
CVE-2026-43259	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: phy: fsl-imx8mq-usb: set platform driver data Add missing platform_set_drvdata() as the data will be used in remove().	2026-05-06	5.5
CVE-2026-43261	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: arm64: Add support for TSV110 Spectre-BHB mitigation The TSV110 processor is vulnerable to the Spectre-BHB (Branch History Buffer) attack, which can be exploited to leak information through branch prediction side channels. This commit adds the MIDR of TSV110 to the list for software mitigation.	2026-05-06	5.5
CVE-2026-43262	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gfs2: fiemap page fault fix In gfs2_fiemap(), we are calling iomap_fiemap() while holding the inode glock. This can lead to recursive glock taking if the fiemap buffer is memory mapped to the same inode and accessing it triggers a page fault. Fix by disabling page faults for iomap_fiemap() and faulting in the buffer by hand if necessary. Fixes xfstest generic/742.	2026-05-06	5.5
CVE-2026-43264	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fbdev: of: display_timing: fix refcount leak in of_get_display_timings() of_parse_phandle() returns a device_node with refcount incremented, which is stored in 'entry' and then copied to 'native_mode'. When the error paths at lines 184 or 192 jump to 'entryfail', native_mode's refcount is not decremented, causing a refcount leak. Fix this by changing the goto target from 'entryfail' to 'timingfail', which properly calls of_node_put(native_mode) before cleanup.	2026-05-06	5.5
CVE-2026-43265	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Ignore -EBUSY when checking nested events from vcpu_block() Ignore -EBUSY when checking nested events after exiting a blocking state while L2 is active, as exiting to userspace will generate a spurious userspace exit, usually with KVM_EXIT_UNKNOWN, and likely lead to the VM's demise. Continuing with the wakeup isn't perfect either, as *something* has gone sideways if a vCPU is awakened in L2 with an injected event (or	2026-05-06	5.5

		worse, a nested run pending), but continuing on gives the VM a decent chance of surviving without any major side effects. As explained in the Fixes commits, it _should_ be impossible for a vCPU to be put into a blocking state with an already-injected event (exception, IRQ, or NMI). Unfortunately, userspace can stuff MP_STATE and/or injected events, and thus put the vCPU into what should be an impossible state. Don't bother trying to preserve the WARN, e.g. with an anti-syzkaller Kconfig, as WARNs can (hopefully) be added in paths where _KVM_ would be violating x86 architecture, e.g. by WARNing if KVM attempts to inject an exception or interrupt while the vCPU isn't running.		
CVE-2026-43266	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: EFI/CPER: don't go past the ARM processor CPER record buffer There's a logic inside GHES/CPER to detect if the section_length is too small, but it doesn't detect if it is too big. Currently, if the firmware receives an ARM processor CPER record stating that a section length is big, kernel will blindly trust section_length, producing a very long dump. For instance, a 67 bytes record with ERR_INFO_NUM set 46198 and section length set to 854918320 would dump a lot of data going a way past the firmware memory-mapped area. Fix it by adding a logic to prevent it to go past the buffer if ERR_INFO_NUM is too big, making it report instead: [Hardware Error]: Hardware error from APEI Generic Hardware Error Source: 1 [Hardware Error]: event severity: recoverable [Hardware Error]: Error 0, type: recoverable [Hardware Error]: section_type: ARM processor error [Hardware Error]: MIDR: 0xff304b2f8476870a [Hardware Error]: section length: 854918320, CPER size: 67 [Hardware Error]: section length is too big [Hardware Error]: firmware-generated error record is incorrect [Hardware Error]: ERR_INFO_NUM is 46198 [rjw: Subject and changelog tweaks]	2026-05-06	5.5
CVE-2026-43267	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix potential zero beacon interval in beacon tracking During fuzz testing, it was discovered that bss_conf->beacon_int might be zero, which could result in a division by zero error in subsequent calculations. Set a default value of 100 TU if the interval is zero to ensure stability.	2026-05-06	5.5
CVE-2026-43268	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hfsplus: pretend special inodes as regular files Since commit af153bb63a33 ("vfs: catch invalid modes in may_open()") requires any inode be one of S_IFDIR/S_IFLNK/S_IFREG/S_IFCHR/S_IFBLK/S_IFIFO/S_IFSOCK type, use S_IFREG for special inodes.	2026-05-06	5.5
CVE-2026-43269	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/atmel-hlcdc: fix memory leak from the atomic_destroy_state callback After several commits, the slab memory increases. Some drm_crtc_commit objects are not freed. The atomic_destroy_state callback only put the framebuffer. Use the __drm_atomic_helper_plane_destroy_state() function to put all the objects that are no longer needed. It has been seen after hours of usage of a graphics application or using kmemleak: unreferenced object 0xc63a6580 (size 64): comm "egt_basic", pid 171, jiffies 4294940784 hex dump (first 32 bytes): 40 50 34 c5 01 00 00 00 ff ff ff ff 8c 65 3a c6 @P4.....e:. 8c 65 3a c6 ff ff ff ff 98 65 3a c6 98 65 3a c6 .e:.....e:..e:. backtrace (crc c25aa925): kmemleak_alloc+0x34/0x3c __kmalloc_cache_noprof+0x150/0x1a4 drm_atomic_helper_setup_commit+0x1e8/0x7bc drm_atomic_helper_commit+0x3c/0x15c drm_atomic_commit+0xc0/0xf4 drm_atomic_helper_set_config+0x84/0xb8 drm_mode_setcrtc+0x32c/0x810	2026-05-06	5.5

		drm_ioctl+0x20c/0x488 sys_ioctl+0x14c/0xc20 ret_fast_syscall+0x0/0x54		
CVE-2026-43270	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: mtk-mdp: Fix a reference leak bug in mtk_mdp_remove() In mtk_mdp_probe(), vpu_get_plat_device() increases the reference count of the returned platform device. Add platform_device_put() to prevent reference leak.	2026-05-06	5.5
CVE-2026-43271	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: md-cluster: fix NULL pointer dereference in process_metadata_update The function process_metadata_update() blindly dereferences the 'thread' pointer (acquired via rcu_dereference_protected) within the wait_event() macro. While the code comment states "daemon thread must exist", there is a valid race condition window during the MD array startup sequence (md_run): 1. bitmap_load() is called, which invokes md_cluster_ops->join(). 2. join() starts the "cluster_recv" thread (recv_daemon). 3. At this point, recv_daemon is active and processing messages. 4. However, mddev->thread (the main MD thread) is not initialized until later in md_run(). If a METADATA_UPDATED message is received from a remote node during this specific window, process_metadata_update() will be called while mddev->thread is still NULL, leading to a kernel panic. To fix this, we must validate the 'thread' pointer. If it is NULL, we release the held lock (no_new_dev_lockres) and return early, safely ignoring the update request as the array is not yet fully ready to process it.	2026-05-06	5.5
CVE-2026-43272	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix possible dereference of uninitialized pointer There is a pointer head_page in rb_meta_validate_events() which is not initialized at the beginning of a function. This pointer can be dereferenced if there is a failure during reader page validation. In this case the control is passed to "invalid" label where the pointer is dereferenced in a loop. To fix the issue initialize orig_head and head_page before calling rb_validate_buffer. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2026-05-06	5.5
CVE-2026-43273	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ceph: supply snapshot context in ceph_zero_partial_object() The ceph_zero_partial_object function was missing proper snapshot context for its OSD write operations, which could lead to data inconsistencies in snapshots. Reproducer: ../src/vstart.sh --new -x --localhost --bluestore ./bin/ceph auth caps client.fs_a mds 'allow rwps fsname=a' mon 'allow r fsname=a' osd 'allow rw tag cephfs data=a' mount -t ceph fs_a@.a=/ /mnt/mycephfs/ -o conf=./ceph.conf dd if=/dev/urandom of=/mnt/mycephfs/foo bs=64K count=1 mkdir /mnt/mycephfs/.snap/snap1 md5sum /mnt/mycephfs/.snap/snap1/foo fallocate -p -o 0 -l 4096 /mnt/mycephfs/foo echo 3 > /proc/sys/vm/drop/caches md5sum /mnt/mycephfs/.snap/snap1/foo # get different md5sum!!	2026-05-06	5.5
CVE-2026-43277	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: APEI/GHES: ensure that won't go past CPER allocated record The logic at ghes_new() prevents allocating too large records, by checking if they're bigger than GHES_ESTATUS_MAX_SIZE (currently, 64KB). Yet, the allocation is done with the actual number of pages from the CPER bios table location, which can be smaller. Yet, a bad firmware could send data with a different size, which might be bigger than the allocated memory, causing an OOPS: Unable to handle kernel paging request at virtual address fff00000f9b40000	2026-05-06	5.5

		Mem abort info: ESR = 0x0000000096000007 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x07: level 3 translation fault Data abort info: ISV = 0, ISS = 0x00000007, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 swapper pgtable: 4k pages, 52-bit VAs, pgdp=000000008ba16000 [fff00000f9b40000] pgd=180000013ffff403, p4d=180000013fffe403, pud=180000013f85b403, pmd=180000013f68d403, pte=0000000000000000 Internal error: Oops: 0000000096000007 [#1] SMP Modules linked in: CPU: 0 UID: 0 PID: 303 Comm: kworker/0:1 Not tainted 6.19.0-rc1-00002-gda407d200220 #34 PREEMPT Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 02/02/2022 Workqueue: kacpi_notify acpi_os_execute_deferred pstate: 214020c5 (nzCv daIF +PAN -UAO -TCO +DIT -SSBS BTYPE=--) pc : hex_dump_to_buffer+0x30c/0x4a0 lr : hex_dump_to_buffer+0x328/0x4a0 sp : ffff800080e13880 x29: ffff800080e13880 x28: ffffac9aba86f6a8 x27: 0000000000000083 x26: fff00000f9b3fffc x25: 0000000000000004 x24: 0000000000000004 x23: ffff800080e13905 x22: 0000000000000010 x21: 0000000000000083 x20: 0000000000000001 x19: 0000000000000008 x18: 0000000000000010 x17: 0000000000000001 x16: 00000007c7f20fec x15: 0000000000000020 x14: 0000000000000008 x13: 0000000000081020 x12: 0000000000000008 x11: ffff800080e13905 x10: ffff800080e13988 x9 : 0000000000000000 x8 : 0000000000000000 x7 : 0000000000000001 x6 : 0000000000000020 x5 : 0000000000000030 x4 : 00000000ffffffe x3 : 0000000000000000 x2 : ffffac9aba78c1c8 x1 : ffffac9aba76d0a8 x0 : 0000000000000008 Call trace: hex_dump_to_buffer+0x30c/0x4a0 (P) print_hex_dump+0xac/0x170 cper_estatus_print_section+0x90c/0x968 cper_estatus_print+0xf0/0x158 __ghes_print_estatus+0xa0/0x148 ghes_proc+0x1bc/0x220 ghes_notify_hed+0x5c/0xb8 notifier_call_chain+0x78/0x148 blocking_notifier_call_chain+0x4c/0x80 acpi_hed_notify+0x28/0x40 acpi_ev_notify_dispatch+0x50/0x80 acpi_os_execute_deferred+0x24/0x48 process_one_work+0x15c/0x3b0 worker_thread+0x2d0/0x400 kthread+0x148/0x228 ret_from_fork+0x10/0x20 Code: 6b14033f 540001ad a94707e2 f100029f (b8747b44) ---[end trace 0000000000000000]--- Prevent that by taking the actual allocated are into account when checking for CPER length. [rjw: Subject tweaks]		
CVE-2026-43282	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/ionic: Fix potential NULL pointer dereference in ionic_query_port The function ionic_query_port() calls ib_device_get_netdev() without checking the return value which could lead to NULL pointer dereference, Fix it by checking the return value and return -ENODEV if the 'ndev' is NULL.	2026-05-06	5.5
CVE-2025-71296	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/tests: shmem: Hold reservation lock around purge Acquire and release the GEM object's reservation lock around calls to the object's purge operation. The tests use drm_gem_shmem_purge_locked(), which led to errors such as show below. [58.709128] WARNING: CPU: 1 PID: 1354 at drivers/gpu/drm/drm_gem_shmem_helper.c:515 drm_gem_shmem_purge_locked+0x51c/0x740 Only export the new helper drm_gem_shmem_purge() for Kunit tests. This is not an interface for regular drivers.	2026-05-08	5.5
CVE-2025-71297	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-05-08	5.5

		wifi: rtw88: 8822b: Avoid WARNING in rtw8822b_config_trx_mode() rtw8822b_set_antenna() can be called from userspace when the chip is powered off. In that case a WARNING is triggered in rtw8822b_config_trx_mode() because trying to read the RF registers when the chip is powered off returns an unexpected value. Call rtw8822b_config_trx_mode() in rtw8822b_set_antenna() only when the chip is powered on. ------------[cut here]------------ write RF mode table fail WARNING: CPU: 0 PID: 7183 at rtw8822b.c:824 rtw8822b_config_trx_mode.constprop.0+0x835/0x840 [rtw88_8822b] CPU: 0 UID: 0 PID: 7183 Comm: iw Tainted: G W OE 6.17.5-arch1-1 #1 PREEMPT(full) 01c39fc421df2af799dd5e9180b572af860b40c1 Tainted: [W]=WARN, [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: LENOVO 82KR/LNVNB161216, BIOS HBCN18WW 08/27/2021 RIP: 0010:rtw8822b_config_trx_mode.constprop.0+0x835/0x840 [rtw88_8822b] Call Trace: <TASK> rtw8822b_set_antenna+0x57/0x70 [rtw88_8822b 370206f42e5890d8d5f48eb358b759efa37c422b] rtw_ops_set_antenna+0x50/0x80 [rtw88_core 711c8fb4f686162be4625b1d0b8e8c6a5ac850fb] ieee80211_set_antenna+0x60/0x100 [mac80211 f1845d85d2ecacf3b71867635a050ece90486cf3] nl80211_set_wiphy+0x384/0xe00 [cfg80211 296485ee85696d2150309a6d21a7fbca83d3dbda] ? netdev_run_todo+0x63/0x550 genl_family_rcv_msg_doit+0xfc/0x160 genl_rcv_msg+0x1aa/0x2b0 ? __pfx_nl80211_pre_doit+0x10/0x10 [cfg80211 296485ee85696d2150309a6d21a7fbca83d3dbda] ? __pfx_nl80211_set_wiphy+0x10/0x10 [cfg80211 296485ee85696d2150309a6d21a7fbca83d3dbda] ? __pfx_nl80211_post_doit+0x10/0x10 [cfg80211 296485ee85696d2150309a6d21a7fbca83d3dbda] ? __pfx_genl_rcv_msg+0x10/0x10 netlink_rcv_skb+0x59/0x110 genl_rcv+0x28/0x40 netlink_unicast+0x285/0x3c0 ? __alloc_skb+0xdb/0x1a0 netlink_sendmsg+0x20d/0x430 ____sys_sendmsg+0x39f/0x3d0 ? import_iovec+0x2f/0x40 ___sys_sendmsg+0x99/0xe0 ? refill_obj_stock+0x12e/0x240 __sys_sendmsg+0x8a/0xf0 do_syscall_64+0x81/0x970 ? do_syscall_64+0x81/0x970 ? ksys_read+0x73/0xf0 ? do_syscall_64+0x81/0x970 ? count_memcg_events+0xc2/0x190 ? handle_mm_fault+0x1d7/0x2d0 ? do_user_addr_fault+0x21a/0x690 ? exc_page_fault+0x7e/0x1a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> ---[end trace 0000000000000000]---		
CVE-2025-71298	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/tests: shmem: Hold reservation lock around madvise Acquire and release the GEM object's reservation lock around calls to the object's madvide operation. The tests use drm_gem_shmem_madvise_locked(), which led to errors such as show below. [58.339389] WARNING: CPU: 1 PID: 1352 at drivers/gpu/drm/drm_gem_shmem_helper.c:499 drm_gem_shmem_madvise_locked+0xde/0x140 Only export the new helper drm_gem_shmem_madvise() for Kunit tests. This is not an interface for regular drivers.	2026-05-08	5.5
CVE-2025-71299	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: spi: cadence-quadspi: Parse DT for flashes with the rest of the DT parsing The recent refactoring of where runtime PM is enabled done in commit f1eb4e792bb1 ("spi: spi-cadence-quadspi: Enable pm runtime earlier to avoid imbalance") made the fact that when we do a pm_runtime_disable() in the error paths of probe() we can trigger a runtime disable which in turn results in duplicate clock disables. This is particularly likely to happen when there is missing or broken DT description for the flashes attached to the controller.	2026-05-08	5.5

		Early on in the probe function we do a pm_runtime_get_noresume() since the probe function leaves the device in a powered up state but in the error path we can't assume that PM is enabled so we also manually disable everything, including clocks. This means that when runtime PM is active both it and the probe function release the same reference to the main clock for the IP, triggering warnings from the clock subsystem: [8.693719] clk:75:7 already disabled [8.693791] WARNING: CPU: 1 PID: 185 at /usr/src/kernel/drivers/clk/clk.c:1188 clk_core_disable+0xa0/0xb ... [8.694261] clk_core_disable+0xa0/0xb4 (P) [8.694272] clk_disable+0x38/0x60 [8.694283] cqspi_probe+0x7c8/0xc5c [spi_cadence_quadspi] [8.694309] platform_probe+0x5c/0xa4 Dealing with this issue properly is complicated by the fact that we don't know if runtime PM is active so can't tell if it will disable the clocks or not. We can, however, sidestep the issue for the flash descriptions by moving their parsing to when we parse the controller properties which also save us doing a bunch of setup which can never be used so let's do that.		
CVE-2025-71300	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Revert "arm64: zynqmp: Add an OP-TEE node to the device tree" This reverts commit 06d22ed6b6635b17551f386b50bb5aaff9b75fbe. OP-TEE logic in U-Boot automatically injects a reserved-memory node along with optee firmware node to kernel device tree. The injection logic is dependent on that there is no manually defined optee node. Having the node in zynqmp.dtsi effectively breaks OP-TEE's insertion of the reserved-memory node, causing memory access violations during runtime.	2026-05-08	5.5
CVE-2025-71301	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/tests: shmem: Hold reservation lock around vmap/vunmap Acquire and release the GEM object's reservation lock around vmap and vunmap operations. The tests use vmap_locked, which led to errors such as show below. [122.292030] WARNING: CPU: 3 PID: 1413 at drivers/gpu/drm/drm_gem_shmem_helper.c:390 drm_gem_shmem_vmap_locked+0x3a3/0x6f0 [122.468066] WARNING: CPU: 3 PID: 1413 at drivers/gpu/drm/drm_gem_shmem_helper.c:293 drm_gem_shmem_pin_locked+0x1fe/0x350 [122.563504] WARNING: CPU: 3 PID: 1413 at drivers/gpu/drm/drm_gem_shmem_helper.c:234 drm_gem_shmem_get_pages_locked+0x23c/0x370 [122.662248] WARNING: CPU: 2 PID: 1413 at drivers/gpu/drm/drm_gem_shmem_helper.c:452 drm_gem_shmem_vunmap_locked+0x101/0x330 Only export the new vmap/vunmap helpers for Kunit tests. These are not interfaces for regular drivers.	2026-05-08	5.5
CVE-2025-71302	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/panthor: fix for dma-fence safe access rules Commit 506aa8b02a8d6 ("dma-fence: Add safe access helpers and document the rules") details the dma-fence safe access rules. The most common culprit is that drm_sched_fence_get_timeline_name may race with group_free_queue.	2026-05-08	5.5
CVE-2026-43285	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/slab: do not access current->mems_allowed_seq if !allow_spin Lockdep complains when get_from_any_partial() is called in an NMI context, because current->mems_allowed_seq is seqcount_spinlock_t and not NMI-safe: ===== WARNING: inconsistent lock state 6.19.0-rc5-kfree-rcu+ #315 Tainted: G N --- inconsistent {INITIAL USE} -> {IN-NMI} usage. kunit_try_catch/9989 [HC1[1]:SC0[0]:HE0:SE1] takes: ffff889085799820 (&____s->seqcount#3){.-.-}-{0:0}, at: ___slab_alloc+0x58f/0xc00 {INITIAL USE} state was registered at:	2026-05-08	5.5

		lock_acquire+0x185/0x320 kernel_init_freeable+0x391/0x1150 kernel_init+0x1f/0x220 ret_from_fork+0x736/0x8f0 ret_from_fork_asm+0x1a/0x30 irq event stamp: 56 hardirqs last enabled at (55): [<ffffffff850a68d7>] _raw_spin_unlock_irq+0x27/0x70 hardirqs last disabled at (56): [<ffffffff850858ca>] __schedule+0x2a8a/0x6630 softirqs last enabled at (0): [<ffffffff81536711>] copy_process+0x1dc1/0x6a10 softirqs last disabled at (0): [<0000000000000000>] 0x0 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&____s->seqcount#3); <Interrupt> lock(&____s->seqcount#3); *** DEADLOCK *** According to Documentation/locking/seqlock.rst, seqcount_t is not NMI-safe and seqcount_latch_t should be used when read path can interrupt the write-side critical section. In this case, do not access current->mems_allowed_seq and avoid retry.		
CVE-2026-43286	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: restore failed global reservations to subpool Commit a833a693a490 ("mm: hugetlb: fix incorrect fallback for subpool") fixed an underflow error for hstate->resv_huge_pages caused by incorrectly attributing globally requested pages to the subpool's reservation. Unfortunately, this fix also introduced the opposite problem, which would leave spool->used_hpages elevated if the globally requested pages could not be acquired. This is because while a subpool's reserve pages only accounts for what is requested and allocated from the subpool, its "used" counter keeps track of what is consumed in total, both from the subpool and globally. Thus, we need to adjust spool->used_hpages in the other direction, and make sure that globally requested pages are uncharged from the subpool's used counter. Each failed allocation attempt increments the used_hpages counter by how many pages were requested from the global pool. Ultimately, this renders the subpool unusable, as used_hpages approaches the max limit. The issue can be reproduced as follows: 1. Allocate 4 hugetlb pages 2. Create a hugetlb mount with max=4, min=2 3. Consume 2 pages globally 4. Request 3 pages from the subpool (2 from subpool + 1 from global) 4.1 hugepage_subpool_get_pages(spool, 3) succeeds. used_hpages += 3 4.2 hugetlb_acct_memory(h, 1) fails: no global pages left used_hpages -= 2 5. Subpool now has used_hpages = 1, despite not being able to successfully allocate any hugepages. It believes it can now only allocate 3 more hugepages, not 4. With each failed allocation attempt incrementing the used counter, the subpool eventually reaches a point where its used counter equals its max counter. At that point, any future allocations that try to allocate hugeTLB pages from the subpool will fail, despite the subpool not having any of its hugeTLB pages consumed by any user. Once this happens, there is no way to make the subpool usable again, since there is no way to decrement the used counter as no process is really consuming the hugeTLB pages. The underflow issue that the original commit fixes still remains fixed as well. Without this fix, used_hpages would keep on leaking if hugetlb_acct_memory() fails.	2026-05-08	5.5
CVE-2026-43287	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm: Account property blob allocations to memcg DRM_IOCTL_MODE_CREATEPROPBLOB allows userspace to allocate arbitrary-sized	2026-05-08	5.5

		property blobs backed by kernel memory. Currently, the blob data allocation is not accounted to the allocating process's memory cgroup, allowing unprivileged users to trigger unbounded kernel memory consumption and potentially cause system-wide OOM. Mark the property blob data allocation with GFP_KERNEL_ACCOUNT so that the memory is properly charged to the caller's memcg. This ensures existing cgroup memory limits apply and prevents uncontrolled kernel memory growth without introducing additional policy or per-file limits.		
CVE-2026-43288	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: move ext4_percpu_param_init() before ext4_mb_init() When running `kvm-xfstests -c ext4/1k -C 1 generic/383` with the `DOUBLE_CHECK` macro defined, the following panic is triggered: ========== EXT4-fs error (device vdc): ext4_validate_block_bitmap:423: comm mount: bg 0: bad block bitmap checksum BUG: unable to handle page fault for address: ff110000fa2cc000 PGD 3e01067 P4D 3e02067 PUD 0 Oops: Oops: 0000 [#1] SMP NOPTI CPU: 0 UID: 0 PID: 2386 Comm: mount Tainted: G W 6.18.0-gba65a4e7120a-dirty #1152 PREEMPT(none) RIP: 0010:percpu_counter_add_batch+0x13/0xa0 Call Trace: <TASK> ext4_mark_group_bitmap_corrupted+0xcb/0xe0 ext4_validate_block_bitmap+0x2a1/0x2f0 ext4_read_block_bitmap+0x33/0x50 mb_group_bb_bitmap_alloc+0x33/0x80 ext4_mb_add_groupinfo+0x190/0x250 ext4_mb_init_backend+0x87/0x290 ext4_mb_init+0x456/0x640 __ext4_fill_super+0x1072/0x1680 ext4_fill_super+0xd3/0x280 get_tree_bdev_flags+0x132/0x1d0 vfs_get_tree+0x29/0xd0 vfs_cmd_create+0x59/0xe0 __do_sys_fsconfig+0x4f6/0x6b0 do_syscall_64+0x50/0x1f0 entry_SYSCALL_64_after_hwframe+0x76/0x7e ========== This issue can be reproduced using the following commands: mkfs.ext4 -F -q -b 1024 /dev/sda 5G tune2fs -O quota,project /dev/sda mount /dev/sda /tmp/test With DOUBLE_CHECK defined, mb_group_bb_bitmap_alloc() reads and validates the block bitmap. When the validation fails, ext4_mark_group_bitmap_corrupted() attempts to update sbi->s_freeclusters_counter. However, this percpu_counter has not been initialized yet at this point, which leads to the panic described above. Fix this by moving the execution of ext4_percpu_param_init() to occur before ext4_mb_init(), ensuring the per-CPU counters are initialized before they are used.	2026-05-08	5.5
CVE-2026-43289	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: kexec: derive purgatory entry from symbol kexec_load_purgatory() derives image->start by locating e_entry inside an SHF_EXECINSTR section. If the purgatory object contains multiple executable sections with overlapping sh_addr, the entrypoint check can match more than once and trigger a WARN. Derive the entry section from the purgatory_start symbol when present and compute image->start from its final placement. Keep the existing e_entry fallback for purgatories that do not expose the symbol. WARNING: kernel/kexec_file.c:1009 at kexec_load_purgatory+0x395/0x3c0, CPU#10: kexec/1784 Call Trace: <TASK> bzImage64_load+0x133/0xa00 __do_sys_kexec_file_load+0x2b3/0x5c0 do_syscall_64+0x81/0x610 entry_SYSCALL_64_after_hwframe+0x76/0x7e	2026-05-08	5.5

		[me@linux.beauty: move helper to avoid forward declaration, per Baoquan]		
CVE-2026-43292	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/vmalloc: prevent RCU stalls in kasan_release_vmalloc_node When CONFIG_PAGE_OWNER is enabled, freeing KASAN shadow pages during vmalloc cleanup triggers expensive stack unwinding that acquires RCU read locks. Processing a large purge_list without rescheduling can cause the task to hold CPU for extended periods (10+ seconds), leading to RCU stalls and potential OOM conditions. The issue manifests in purge_vmap_node() -> kasan_release_vmalloc_node() where iterating through hundreds or thousands of vmap_area entries and freeing their associated shadow pages causes: rcu: INFO: rcu_preempt detected stalls on CPUs/tasks: rcu: Tasks blocked on level-0 rcu_node (CPUs 0-1): P6229/1:b..l ... task:kworker/0:17 state:R running task stack:28840 pid:6229 ... kasan_release_vmalloc_node+0x1ba/0xad0 mm/vmalloc.c:2299 purge_vmap_node+0x1ba/0xad0 mm/vmalloc.c:2299 Each call to kasan_release_vmalloc() can free many pages, and with page_owner tracking, each free triggers save_stack() which performs stack unwinding under RCU read lock. Without yielding, this creates an unbounded RCU critical section. Add periodic cond_resched() calls within the loop to allow: - RCU grace periods to complete - Other tasks to run - Scheduler to preempt when needed The fix uses need_resched() for immediate response under load, with a batch count of 32 as a guaranteed upper bound to prevent worst-case stalls even under light load.	2026-05-08	5.5
CVE-2026-43293	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Fix kthread worker destruction in polling mode Fix the cleanup order in polling mode (irq < 0) to prevent kernel warnings during module removal. Cancel the hrtimer before destroying the kthread worker to ensure work queues are empty. In polling mode, the driver uses hrtimer to periodically trigger wave5_vpu_timer_callback() which queues work via kthread_queue_work(). The kthread_destroy_worker() function validates that both work queues are empty with WARN_ON(!list_empty(&worker->work_list)) and WARN_ON(!list_empty(&worker->delayed_work_list)). The original code called kthread_destroy_worker() before hrtimer_cancel(), creating a race condition where the timer could fire during worker destruction and queue new work, triggering the WARN_ON. This causes the following warning on every module unload in polling mode: ------------[cut here]------------ WARNING: CPU: 2 PID: 1034 at kernel/kthread.c:1430 kthread_destroy_worker+0x84/0x98 Modules linked in: wave5(-) rpmsg_ctrl rpmsg_char ... Call trace: kthread_destroy_worker+0x84/0x98 wave5_vpu_remove+0xc8/0xe0 [wave5] platform_remove+0x30/0x58 ... ---[end trace 0000000000000000]---	2026-05-08	5.5
CVE-2026-43294	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm: renesas: rz-du: mipi_dsi: fix kernel panic when rebooting for some panels Since commit 56de5e305d4b ("clk: renesas: r9a07g044: Add MSTOP for RZ/G2L") we may get the following kernel panic, for some panels, when rebooting: systemd-shutdown[1]: Rebooting. Call trace: ... do_serror+0x28/0x68 el1h_64_error_handler+0x34/0x50 el1h_64_error+0x6c/0x70	2026-05-08	5.5

		rzg2l_mipi_dsi_host_transfer+0x114/0x458 (P) mipi_dsi_device_transfer+0x44/0x58 mipi_dsi_dcs_set_display_off_multi+0x9c/0xc4 ili9881c_unprepare+0x38/0x88 drm_panel_unprepare+0xbc/0x108 This happens for panels that need to send MIPI-DSI commands in their unprepare() callback. Since the MIPI-DSI interface is stopped at that point, rzg2l_mipi_dsi_host_transfer() triggers the kernel panic. Fix by moving rzg2l_mipi_dsi_stop() to new callback function rzg2l_mipi_dsi_atomic_post_disable(). With this change we now have the correct power-down/stop sequence: systemd-shutdown[1]: Rebooting. rzg2l-mipi-dsi 10850000.dsi: rzg2l_mipi_dsi_atomic_disable(): entry ili9881c-dsi 10850000.dsi.0: ili9881c_unprepare(): entry rzg2l-mipi-dsi 10850000.dsi: rzg2l_mipi_dsi_atomic_post_disable(): entry reboot: Restarting system		
CVE-2026-43295	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rapidio: replace rio_free_net() with kfree() in rio_scan_alloc_net() When idtab allocation fails, net is not registered with rio_add_net() yet, so kfree(net) is sufficient to release the memory. Set mport->net to NULL to avoid dangling pointer.	2026-05-08	5.5
CVE-2026-43297	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: rockchip: rga: Fix possible ERR_PTR dereference in rga_buf_init() rga_get_frame() can return ERR_PTR(-EINVAL) when buffer type is unsupported or invalid. rga_buf_init() does not check the return value and unconditionally dereferences the pointer when accessing f->size. Add proper ERR_PTR checking and return the error to prevent dereferencing an invalid pointer.	2026-05-08	5.5
CVE-2026-43298	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Skip vcn poison irq release on VF VF doesn't enable VCN poison irq in VCNv2.5. Skip releasing it and avoid call trace during deinitialization. [71.913601] [drm] clean up the vf2pf work item [71.915088] ------------[cut here]------------ [71.915092] WARNING: CPU: 3 PID: 1079 at /tmp/amd.aFkFvSQl/amd/amdgpu/amdgpu_irq.c:641 amdgpu_irq_put+0xc6/0xe0 [amdgpu] [71.915355] Modules linked in: amdgpu(OE-) amddrm_ttm_helper(OE) amdttm(OE) amddrm_buddy(OE) amdxcp(OE) amddrm_exec(OE) amd_sched(OE) amdkcl(OE) drm_suballoc_helper drm_display_helper cec rc_core i2c_algo_bit video wmi binfmt_misc nls_iso8859_1 intel_rapl_msr intel_rapl_common input_leds joydev serio_raw mac_hid qemu_fw_cfg sch_fq_codel dm_multipath scsi_dh_rdac scsi_dh_emc scsi_dh_alua efi_pstore ip_tables x_tables autofs4 btrfs blake2b_generic raid10 raid456 async_raid6_recov async_memcpy async_pq async_xor async_tx xor raid6_pq libcrc32c raid1 raid0 hid_generic crct10dif_pclmul crc32_pclmul polyval_clmulni polyval_generic ghash_clmulni_intel usbhid 8139too sha256_ssse3 sha1_ssse3 hid psmouse bochs i2c_i801 ahci drm_vram_helper libahci i2c_smbus lpc_ich drm_ttm_helper 8139cp mii ttm aesni_intel crypto_simd cryptd [71.915484] CPU: 3 PID: 1079 Comm: rmmod Tainted: G OE 6.8.0-87-generic #88~22.04.1-Ubuntu [71.915489] Hardware name: Red Hat KVM/RHEL, BIOS 1.16.3-2.el9_5.1 04/01/2014 [71.915492] RIP: 0010:amdgpu_irq_put+0xc6/0xe0 [amdgpu] [71.915768] Code: 75 84 b8 ea ff ff ff eb d4 44 89 ea 48 89 de 4c 89 e7 e8 fd fc ff ff 5b 41 5c 41 5d 41 5e 5d 31 d2 31 f6 31 ff e9 55 30 3b c7 <0f> 0b eb d4 b8 fe ff ff ff eb a8 e9 b7 3b 8a 00 66 2e 0f 1f 84 00 [71.915771] RSP: 0018:ffffcf0800eafa30 EFLAGS: 00010246 [71.915775] RAX: 0000000000000000 RBX: ffff891bda4b0668 RCX: 0000000000000000 [71.915777] RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 [71.915779] RBP: ffffcf0800eafa50 R08: 0000000000000000 R09: 0000000000000000 [71.915781] R10: 0000000000000000 R11: 0000000000000000 R12: ffff891bda480000 [71.915782] R13: 0000000000000000 R14: 0000000000000001 R15: 0000000000000000 [71.915792] FS: 000070cff87c4c40(0000) GS:ffff893abfb80000(0000) knlGS:0000000000000000 [71.915795] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [71.915797] CR2: 00005fa13073e478 CR3: 000000010d634006 CR4: 0000000000770ef0 [71.915800] PKRU: 55555554 [71.915802] Call Trace: [71.915805] <TASK> [71.915809] vcn_v2_5_hw_fini+0x19e/0x1e0 [amdgpu]	2026-05-08	5.5
CVE-2026-43299	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved:	2026-05-08	5.5

		btrfs: do not ASSERT() when the fs flips RO inside btrfs_repair_io_failure() [BUG] There is a bug report that when btrfs hits ENOSPC error in a critical path, btrfs flips RO (this part is expected, although the ENOSPC bug still needs to be addressed). The problem is after the RO flip, if there is a read repair pending, we can hit the ASSERT() inside btrfs_repair_io_failure() like the following: BTRFS info (device vdc): relocating block group 30408704 flags metadata|raid1 ------------[cut here]------------ BTRFS: Transaction aborted (error -28) WARNING: fs/btrfs/extent-tree.c:3235 at __btrfs_free_extent.isra.0+0x453/0xfd0, CPU#1: btrfs/383844 Modules linked in: kvm_intel kvm irqbypass [...] ---[end trace 0000000000000000]--- BTRFS info (device vdc state EA): 2 enospc errors during balance BTRFS info (device vdc state EA): balance: ended with status: -30 BTRFS error (device vdc state EA): parent transid verify failed on logical 30556160 mirror 2 wanted 8 found 6 BTRFS error (device vdc state EA): bdev /dev/nvme0n1 errs: wr 0, rd 0, flush 0, corrupt 10, gen 0 [...] assertion failed: !(fs_info->sb->s_flags & SB_RDONLY) :: 0, in fs/btrfs/bio.c:938 ------------[cut here]------------ assertion failed: !(fs_info->sb->s_flags & SB_RDONLY) :: 0, in fs/btrfs/bio.c:938 kernel BUG at fs/btrfs/bio.c:938! Oops: invalid opcode: 0000 [#1] SMP NOPTI CPU: 0 UID: 0 PID: 868 Comm: kworker/u8:13 Tainted: G W N 6.19.0-rc6+ #4788 PREEMPT(full) Tainted: [W]=WARN, [N]=TEST Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.17.0-0-gb52ca86e094d-prebuilt.qemu.org 04/01/2014 Workqueue: btrfs-endio simple_end_io_work RIP: 0010:btrfs_repair_io_failure.cold+0xb2/0x120 RSP: 0000:ffffc90001d2bcf0 EFLAGS: 00010246 RAX: 0000000000000051 RBX: 0000000000001000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffffff8305cf42 RDI: 00000000ffffffff RBP: 0000000000000002 R08: 00000000fffeffff R09: ffffffff837fa988 R10: ffffffff8327a9e0 R11: 6f69747265737361 R12: ffff88813018d310 R13: ffff888168b8a000 R14: ffffc90001d2bd90 R15: ffff88810a169000 FS: 0000000000000000(0000) GS:ffff8885e752c000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 ------------[cut here]------------ [CAUSE] The cause of -ENOSPC error during the test case btrfs/124 is still unknown, although it's known that we still have cases where metadata can be over-committed but can not be fulfilled correctly, thus if we hit such ENOSPC error inside a critical path, we have no choice but abort the current transaction. This will mark the fs read-only. The problem is inside the btrfs_repair_io_failure() path that we require the fs not to be mount read-only. This is normally fine, but if we are doing a read-repair meanwhile the fs flips RO due to a critical error, we can enter btrfs_repair_io_failure() with super block set to read-only, thus triggering the above crash. [FIX] Just replace the ASSERT() with a proper return if the fs is already read-only.		

CVE-2026-43300	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		drm/panel: Fix a possible null-pointer dereference in jdi_panel_dsi_remove()
		
		In jdi_panel_dsi_remove(), jdi is explicitly checked, indicating that it may be NULL:
		
		if (!jdi)
		mipi_dsi_detach(dsi);
		
		However, when jdi is NULL, the function does not return and continues by calling jdi_panel_disable():
		
		err = jdi_panel_disable(&jdi->base);
		
		Inside jdi_panel_disable(), jdi is dereferenced unconditionally, which can lead to a NULL-pointer dereference:
			2026-05-08	5.5

		struct jdi_panel *jdi = to_panel_jdi(panel);
		backlight_disable(jdi->backlight);
		
		To prevent such a potential NULL-pointer dereference, return early from jdi_panel_dsi_remove() when jdi is NULL.		

CVE-2026-43301	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: chips-media: wave5: Fix PM runtime usage count underflow Replace pm_runtime_put_sync() with pm_runtime_dont_use_autosuspend() in the remove path to properly pair with pm_runtime_use_autosuspend() from probe. This allows pm_runtime_disable() to handle reference count cleanup correctly regardless of current suspend state. The driver calls pm_runtime_put_sync() unconditionally in remove, but the device may already be suspended due to autosuspend configured in probe. When autosuspend has already suspended the device, the usage count is 0, and pm_runtime_put_sync() decrements it to -1. This causes the following warning on module unload: ------------[cut here]------------ WARNING: CPU: 1 PID: 963 at kernel/kthread.c:1430 kthread_destroy_worker+0x84/0x98 ... vdec 30210000.video-codec: Runtime PM usage count underflow!	2026-05-08	5.5
CVE-2026-43302	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Set DMA segment size to avoid debug warnings When using V3D rendering with CONFIG_DMA_API_DEBUG enabled, the kernel occasionally reports a segment size mismatch. This is because 'max_seg_size' is not set. The kernel defaults to 64K. setting 'max_seg_size' to the maximum will prevent 'debug_dma_map_sg()' from complaining about the over-mapping of the V3D segment length. DMA-API: v3d 1002000000.v3d: mapping sg segment longer than device claims to support [len=8290304] [max=65536] WARNING: CPU: 0 PID: 493 at kernel/dma/debug.c:1179 debug_dma_map_sg+0x330/0x388 CPU: 0 UID: 0 PID: 493 Comm: Xorg Not tainted 6.12.53-yocto-standard #1 Hardware name: Raspberry Pi 5 Model B Rev 1.0 (DT) pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : debug_dma_map_sg+0x330/0x388 lr : debug_dma_map_sg+0x330/0x388 sp : ffff8000829a3ac0 x29: ffff8000829a3ac0 x28: 0000000000000001 x27: ffff8000813fe000 x26: ffffc1ffc0000000 x25: ffff00010fdeb760 x24: 0000000000000000 x23: ffff8000816a9bf0 x22: 0000000000000001 x21: 0000000000000002 x20: 0000000000000002 x19: ffff00010185e810 x18: ffffffffffffffff x17: 69766564206e6168 x16: 74207265676e6f6c x15: 20746e656d676573 x14: 20677320676e6970 x13: 5d34303334393134 x12: 0000000000000000 x11: 00000000000000c0 x10: 00000000000009c0 x9 : ffff8000800e0b7c x8 : ffff00010a315ca0 x7 : ffff8000816a5110 x6 : 0000000000000001 x5 : 000000000000002b x4 : 0000000000000002 x3 : 0000000000000008 x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff00010a315280 Call trace: debug_dma_map_sg+0x330/0x388 __dma_map_sg_attrs+0xc0/0x278 dma_map_sgtable+0x30/0x58 drm_gem_shmem_get_pages_sgt+0xb4/0x140 v3d_bo_create_finish+0x28/0x130 [v3d] v3d_create_bo_ioctl+0x54/0x180 [v3d] drm_ioctl_kernel+0xc8/0x140 drm_ioctl+0x2d4/0x4d8	2026-05-08	5.5
CVE-2026-43305	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix mismatched unlock for DMUB HW lock in HWSS fast path [Why] The evaluation for whether we need to use the DMUB HW lock isn't the same as whether we need to unlock which results in a hang when the fast path is used for ASIC without FAMS support. [How] Store a flag that indicates whether we should use the lock and use that same flag to specify whether unlocking is needed.	2026-05-08	5.5
CVE-2026-43306	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: crypto: Use the correct destructor kfunc type	2026-05-08	5.5

		With CONFIG_CFI enabled, the kernel strictly enforces that indirect function calls use a function pointer type that matches the target function. I ran into the following type mismatch when running BPF self-tests: CFI failure at bpf_obj_free_fields+0x190/0x238 (target: bpf_crypto_ctx_release+0x0/0x94; expected type: 0xa488ebfc) Internal error: Oops - CFI: 00000000f2008228 [#1] SMP ... As bpf_crypto_ctx_release() is also used in BPF programs and using a void pointer as the argument would make the verifier unhappy, add a simple stub function with the correct type and register it as the destructor kfunc instead.		
CVE-2026-43308	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't BUG() on unexpected delayed ref type in run_one_delayed_ref() There is no need to BUG(), we can just return an error and log an error message.	2026-05-08	5.5
CVE-2026-43309	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: md raid: fix hang when stopping arrays with metadata through dm-raid When using device-mapper's dm-raid target, stopping a RAID array can cause the system to hang under specific conditions. This occurs when: - A dm-raid managed device tree is suspended from top to bottom (the top-level RAID device is suspended first, followed by its underlying metadata and data devices) - The top-level RAID device is then removed Removing the top-level device triggers a hang in the following sequence: the dm-raid destructor calls md_stop(), which tries to flush the write-intent bitmap by writing to the metadata sub-devices. However, these devices are already suspended, making them unable to complete the write-intent operations and causing an indefinite block. Fix: - Prevent bitmap flushing when md_stop() is called from dm-raid destructor context and avoid a quiescing/unquescing cycle which could also cause I/O - Still allow write-intent bitmap flushing when called from dm-raid suspend context This ensures that RAID array teardown can complete successfully even when the underlying devices are in a suspended state. This second patch uses md_is_rdwr() to distinguish between suspend and destructor paths as elaborated on above.	2026-05-08	5.5
CVE-2026-43310	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: media: verisilicon: Avoid G2 bus error while decoding H.264 and HEVC For the i.MX8MQ platform, there is a hardware limitation: the g1 VPU and g2 VPU cannot decode simultaneously; otherwise, it will cause below bus error and produce corrupted pictures, even potentially lead to system hang. [110.527986] hantro-vpu 38310000.video-codec: frame decode timed out. [110.583517] hantro-vpu 38310000.video-codec: bus error detected. Therefore, it is necessary to ensure that g1 and g2 operate alternately. This allows for successful multi-instance decoding of H.264 and HEVC. To achieve this, g1 and g2 share the same v4l2_m2m_dev, and then the v4l2_m2m_dev can handle the scheduling.	2026-05-08	5.5
CVE-2026-43311	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: soc/tegra: pmc: Fix unsafe generic_handle_irq() call Currently, when resuming from system suspend on Tegra platforms, the following warning is observed: WARNING: CPU: 0 PID: 14459 at kernel/irq/irqdesc.c:666	2026-05-08	5.5

		Call trace: handle_irq_desc+0x20/0x58 (P) tegra186_pmc_wake_syscore_resume+0xe4/0x15c syscore_resume+0x3c/0xb8 suspend_devices_and_enter+0x510/0x540 pm_suspend+0x16c/0x1d8 The warning occurs because generic_handle_irq() is being called from a non-interrupt context which is considered as unsafe. Fix this warning by deferring generic_handle_irq() call to an IRQ work which gets executed in hard IRQ context where generic_handle_irq() can be called safely. When PREEMPT_RT kernels are used, regular IRQ work (initialized with init_irq_work) is deferred to run in per-CPU kthreads in preemptible context rather than hard IRQ context. Hence, use the IRQ_WORK_INIT_HARD variant so that with PREEMPT_RT kernels, the IRQ work is processed in hardirq context instead of being deferred to a thread which is required for calling generic_handle_irq(). On non-PREEMPT_RT kernels, both init_irq_work() and IRQ_WORK_INIT_HARD() execute in IRQ context, so this change has no functional impact for standard kernel configurations. [treding@nvidia.com: miscellaneous cleanups]		
CVE-2026-43312	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: i2c: ov5647: Initialize subdev before controls In ov5647_init_controls() we call v4l2_get_subdevdata, but it is initialized by v4l2_i2c_subdev_init() in the probe, which currently happens after init_controls(). This can result in a segfault if the error condition is hit, and we try to access i2c_client, so fix the order.	2026-05-08	5.5

CVE-2026-43313	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ACPI: processor: Fix NULL-pointer dereference in acpi_processor_errata_piix4()
		
		In acpi_processor_errata_piix4(), the pointer dev is first assigned an IDE device and then reassigned an ISA device:
		
		dev = pci_get_subsys(..., PCI_DEVICE_ID_INTEL_82371AB, ...);
		dev = pci_get_subsys(..., PCI_DEVICE_ID_INTEL_82371AB_0, ...);
		
		If the first lookup succeeds but the second fails, dev becomes NULL. This leads to a potential null-pointer dereference when dev_dbg() is called:
		
		if (errata.piix4.bmisx)
		dev_dbg(&dev->dev, ...);
		
		To prevent this, use two temporary pointers and retrieve each device independently, avoiding overwriting dev with a possible NULL value.
		
		[rjw: Subject adjustment, added an empty code line]
			2026-05-08	5.5

CVE-2026-43314	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm: remove fake timeout to avoid leak request Since commit 15f73f5b3e59 ("blk-mq: move failure injection out of blk_mq_complete_request"), drivers are responsible for calling blk_should_fake_timeout() at appropriate code paths and opportunities. However, the dm driver does not implement its own timeout handler and relies on the timeout handling of its slave devices. If an io-timeout-fail error is injected to a dm device, the request will be leaked and never completed, causing tasks to hang indefinitely. Reproduce: 1. prepare dm which has iscsi slave device 2. inject io-timeout-fail to dm echo 1 >/sys/class/block/dm-0/io-timeout-fail echo 100 >/sys/kernel/debug/fail_io_timeout/probability echo 10 >/sys/kernel/debug/fail_io_timeout/times 3. read/write dm 4. iscsiadm -m node -u Result: hang task like below [862.243768] INFO: task kworker/u514:2:151 blocked for more than 122 seconds.	2026-05-08	5.5

		[862.244133] Tainted: G E 6.19.0-rc1+ #51 [862.244337] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [862.244718] task:kworker/u514:2 state:D stack:0 pid:151 tgid:151 ppid:2 task_flags:0x4288060 flags:0x00080000 [862.245024] Workqueue: iscsi_ctrl_3:1 __iscsi_unbind_session [scsi_transport_iscsi] [862.245264] Call Trace: [862.245587] <TASK> [862.245814] __schedule+0x810/0x15c0 [862.246557] schedule+0x69/0x180 [862.246760] blk_mq_freeze_queue_wait+0xde/0x120 [862.247688] elevator_change+0x16d/0x460 [862.247893] elevator_set_none+0x87/0xf0 [862.248798] blk_unregister_queue+0x12e/0x2a0 [862.248995] __del_gendisk+0x231/0x7e0 [862.250143] del_gendisk+0x12f/0x1d0 [862.250339] sd_remove+0x85/0x130 [sd_mod] [862.250650] device_release_driver_internal+0x36d/0x530 [862.250849] bus_remove_device+0x1dd/0x3f0 [862.251042] device_del+0x38a/0x930 [862.252095] __scsi_remove_device+0x293/0x360 [862.252291] scsi_remove_target+0x486/0x760 [862.252654] __iscsi_unbind_session+0x18a/0x3e0 [scsi_transport_iscsi] [862.252886] process_one_work+0x633/0xe50 [862.253101] worker_thread+0x6df/0xf10 [862.253647] kthread+0x36d/0x720 [862.254533] ret_from_fork+0x2a6/0x470 [862.255852] ret_from_fork_asm+0x1a/0x30 [862.256037] </TASK> Remove the blk_should_fake_timeout() check from dm, as dm has no native timeout handling and should not attempt to fake timeouts.		

CVE-2026-43315	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		KVM: nSVM: Remove a user-triggerable WARN on nested_svm_load_cr3() succeeding
		
		Drop the WARN in svm_set_nested_state() on nested_svm_load_cr3() failing as it is trivially easy to trigger from userspace by modifying CPUID after loading CR3. E.g. modifying the state restoration selftest like so:
		
		--- 	
		tools/testing/selftests/kvm/x86/state_test.c
		+++
		tools/testing/selftests/kvm/x86/state_test.c
		@@ -280,7 +280,16 @@ int main(int argc, char *argv[])
		
		/* Restore state in a new VM. */
		vcpu = vm_recreate_with_one_vcpu(vm);
		- vcpu_load_state(vcpu, state);
		+
		+ if (stage == 4) {
		+ state->sregs.cr3 = BIT(44);
		+ vcpu_load_state(vcpu, state);
		+
		+ vcpu_set_cpuid_property(vcpu, X86_PROPERTY_MAX_PHY_ADDR, 36);
		+ __vcpu_nested_state_set(vcpu, &state->nested);
		+ } else {
		+ vcpu_load_state(vcpu, state);
		+ }
		
		/*
		* Restore XSAVE state in a dummy vCPU, first without doing
		
		generates:
		
		WARNING: CPU: 30 PID: 938 at arch/x86/kvm/svm/nested.c:1877 svm_set_nested_state+0x34a/0x360 [kvm_amd]
		Modules linked in: kvm_amd kvm irqbypass [last unloaded: kvm]
		CPU: 30 UID: 1000 PID: 938 Comm: state_test Tainted: G W 6.18.0-rc7-58e10b63777d-next-vm
		Tainted: [W]=WARN
		Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 0.0.0 02/06/2015
		RIP: 0010:svm_set_nested_state+0x34a/0x360 [kvm_amd]
		Call Trace:
		<TASK>
		kvm_arch_vcpu_ioctl+0xf33/0x1700 [kvm]
		kvm_vcpu_ioctl+0x4e6/0x8f0 [kvm]
		__x64_sys_ioctl+0x8f/0xd0
		do_syscall_64+0x61/0xad0
		entry_SYSCALL_64_after_hwframe+0x4b/0x53
		
		Simply delete the WARN instead of trying to prevent userspace from shoving "illegal" state into CR3. For better or worse, KVM's ABI allows userspace
			2026-05-08	5.5

		to set CPUID after SREGS, and vice versa, and KVM is very permissive when it comes to guest CPUID. I.e. attempting to enforce the virtual CPU model when setting CPUID could break userspace. Given that the WARN doesn't provide any meaningful protection for KVM or benefit for userspace, simply drop it even though the odds of breaking userspace are minuscule. Opportunistically delete a spurious newline.		
CVE-2026-43316	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: solo6x10: Check for out of bounds chip_id Clang with CONFIG_UBSAN_SHIFT=y noticed a condition where a signed type (literal "1" is an "int") could end up being shifted beyond 32 bits, so instrumentation was added (and due to the double is_tw286x() call seen via inlining), Clang decides the second one must now be undefined behavior and elides the rest of the function[1]. This is a known problem with Clang (that is still being worked on), but we can avoid the entire problem by actually checking the existing max chip ID, and now there is no runtime instrumentation added at all since everything is known to be within bounds. Additionally use an unsigned value for the shift to remove the instrumentation even without the explicit bounds checking. [hverkuil: fix checkpatch warning for is_tw286x]	2026-05-08	5.5
CVE-2026-43317	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: most: core: fix leak on early registration failure A recent commit fixed a resource leak on early registration failures but for some reason left out the first error path which still leaks the resources associated with the interface. Fix up also the first error path so that the interface is always released on errors.	2026-05-08	5.5
CVE-2026-43318	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix sync handling in amdgpu_dma_buf_move_notify Invalidating a dmabuf will impact other users of the shared BO. In the scenario where process A moves the BO, it needs to inform process B about the move and process B will need to update its page table. The commit fixes a synchronisation bug caused by the use of the ticket: it made amdgpu_vm_handle_moved behave as if updating the page table immediately was correct but in this case it's not. An example is the following scenario, with 2 GPUs and glxgears running on GPU0 and Xorg running on GPU1, on a system where P2P PCI isn't supported: glxgears: export linear buffer from GPU0 and import using GPU1 submit frame rendering to GPU0 submit tiled->linear blit Xorg: copy of linear buffer The sequence of jobs would be: drm_sched_job_run # GPU0, frame rendering drm_sched_job_queue # GPU0, blit drm_sched_job_done # GPU0, frame rendering drm_sched_job_run # GPU0, blit move linear buffer for GPU1 access # amdgpu_dma_buf_move_notify -> update pt # GPU0 It this point the blit job on GPU0 is still running and would likely produce a page fault.	2026-05-08	5.5
CVE-2026-43319	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: spi: spidev: fix lock inversion between spi_lock and buf_lock The spidev driver previously used two mutexes, spi_lock and buf_lock, but acquired them in different orders depending on the code path: write()/read(): buf_lock -> spi_lock ioctl(): spi_lock -> buf_lock This AB-BA locking pattern triggers lockdep warnings and can	2026-05-08	5.5

		cause real deadlocks: WARNING: possible circular locking dependency detected spidev_ioctl() -> mutex_lock(&spidev->buf_lock) spidev_sync_write() -> mutex_lock(&spidev->spi_lock) *** DEADLOCK *** The issue is reproducible with a simple userspace program that performs write() and SPI_IOC_WR_MAX_SPEED_HZ ioctl() calls from separate threads on the same spidev file descriptor. Fix this by simplifying the locking model and removing the lock inversion entirely. spidev_sync() no longer performs any locking, and all callers serialize access using spi_lock. buf_lock is removed since its functionality is fully covered by spi_lock, eliminating the possibility of lock ordering issues. This removes the lock inversion and prevents deadlocks without changing userspace ABI or behaviour.		
CVE-2026-43320	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix dsc eDP issue [why] Need to add function hook check before use	2026-05-08	5.5
CVE-2026-43323	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sched/fair: Fix zero_vruntime tracking fix John reported that stress-ng-yield could make his machine unhappy and managed to bisect it to commit b3d99f43c72b ("sched/fair: Fix zero_vruntime tracking"). The combination of yield and that commit was specific enough to hypothesize the following scenario: Suppose we have 2 runnable tasks, both doing yield. Then one will be eligible and one will not be, because the average position must be in between these two entities. Therefore, the runnable task will be eligible, and be promoted a full slice (all the tasks do is yield after all). This causes it to jump over the other task and now the other task is eligible and current is no longer. So we schedule. Since we are runnable, there is no {de,en}queue. All we have is the __{en,de}queue_entity() from {put_prev,set_next}_task(). But per the fingered commit, those two no longer move zero_vruntime. All that moves zero_vruntime are tick and full {de,en}queue. This means, that if the two tasks playing leapfrog can reach the critical speed to reach the overflow point inside one tick's worth of time, we're up a creek. Additionally, when multiple cgroups are involved, there is no guarantee the tick will in fact hit every cgroup in a timely manner. Statistically speaking it will, but that same statistics does not rule out the possibility of one cgroup not getting a tick for a significant amount of time -- however unlikely. Therefore, just like with the yield() case, force an update at the end of every slice. This ensures the update is never more than a single slice behind and the whole thing is within 2 lag bounds as per the comment on entity_key().	2026-05-08	5.5
CVE-2026-43325	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't send a 6E related command when not supported MCC_ALLOWED_AP_TYPE_CMD is related to 6E support. Do not send it if the device doesn't support 6E. Apparently, the firmware is mistakenly advertising support for this command even on AX201 which does not support 6E and then the firmware crashes.	2026-05-08	5.5
CVE-2026-43326	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sched_ext: Fix SCX_KICK_WAIT deadlock by deferring wait to balance callback SCX_KICK_WAIT busy-waits in kick_cpus_irq_workfn() using	2026-05-08	5.5

		smp_cond_load_acquire() until the target CPU's kick_sync advances. Because the irq_work runs in hardirq context, the waiting CPU cannot reschedule and its own kick_sync never advances. If multiple CPUs form a wait cycle, all CPUs deadlock. Replace the busy-wait in kick_cpus_irq_workfn() with resched_curr() to force the CPU through do_pick_task_scx(), which queues a balance callback to perform the wait. The balance callback drops the rq lock and enables IRQs following the sched_core_balance() pattern, so the CPU can process IPIs while waiting. The local CPU's kick_sync is advanced on entry to do_pick_task_scx() and continuously during the wait, ensuring any CPU that starts waiting for us sees the advancement and cannot form cyclic dependencies.		
CVE-2026-43327	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: USB: dummy-hcd: Fix locking/synchronization error Syzbot testing was able to provoke an addressing exception and crash in the usb_gadget_udc_reset() routine in drivers/usb/gadgets/udc/core.c, resulting from the fact that the routine was called with a second ("driver") argument of NULL. The bad caller was set_link_state() in dummy_hcd.c, and the problem arose because of a race between a USB reset and driver unbind. These sorts of races were not supposed to be possible; commit 7dbd8f4cabd9 ("USB: dummy-hcd: Fix erroneous synchronization change"), along with a few followup commits, was written specifically to prevent them. As it turns out, there are (at least) two errors remaining in the code. Another patch will address the second error; this one is concerned with the first. The error responsible for the syzbot crash occurred because the stop_activity() routine will sometimes drop and then re-acquire the dum->lock spinlock. A call to stop_activity() occurs in set_link_state() when handling an emulated USB reset, after the test of dum->ints_enabled and before the increment of dum->callback_usage. This allowed another thread (doing a driver unbind) to sneak in and grab the spinlock, and then clear dum->ints_enabled and dum->driver. Normally this other thread would have to wait for dum->callback_usage to go down to 0 before it would clear dum->driver, but in this case it didn't have to wait since dum->callback_usage had not yet been incremented. The fix is to increment dum->callback_usage _before_ calling stop_activity() instead of after. Then the thread doing the unbind will not clear dum->driver until after the call to usb_gadget_udc_reset() safely returns and dum->callback_usage has been decremented again.	2026-05-08	5.5

CVE-2026-43331	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		x86/kexec: Disable KCOV instrumentation after load_segments()
		
		The load_segments() function changes segment registers, invalidating GS base (which KCOV relies on for per-cpu data). When CONFIG_KCOV is enabled, any subsequent instrumented C code call (e.g. native_gdt_invalidate()) begins crashing the kernel in an endless loop.
		
		To reproduce the problem, it's sufficient to do kexec on a KCOV-instrumented kernel:
		
		$ kexec -l /boot/otherKernel
		$ kexec -e
		
		The real-world context for this problem is enabling crash dump collection in syzkaller. For this, the tool loads a panic kernel before fuzzing and then calls makedumpfile after the panic. This workflow requires both CONFIG_KEXEC and CONFIG_KCOV to be enabled simultaneously.
		
		Adding safeguards directly to the KCOV fast-path (__sanitizer_cov_trace_pc()) is also undesirable as it would introduce an extra performance overhead.
		
		Disabling instrumentation for the individual functions would be too fragile, so disable KCOV instrumentation for the entire machine_kexec_64.c and physaddr.c. If coverage-guided fuzzing ever needs these components in the future, other approaches should be considered.
		
		The problem is not relevant for 32 bit kernels as CONFIG_KCOV is not supported there.
		
		[bp: Space out comment for better readability.]
			2026-05-08	5.5

CVE-2026-43333	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: reject direct access to nullable PTR_TO_BUF pointers check_mem_access() matches PTR_TO_BUF via base_type() which strips PTR_MAYBE_NULL, allowing direct dereference without a null check. Map iterator ctx->key and ctx->value are PTR_TO_BUF | PTR_MAYBE_NULL. On stop callbacks these are NULL, causing a kernel NULL dereference. Add a type_may_be_null() guard to the PTR_TO_BUF branch, matching the existing PTR_TO_BTF_ID pattern.	2026-05-08	5.5
CVE-2026-43335	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: interconnect: qcom: sm8450: Fix NULL pointer dereference in icc_link_nodes() The change to dynamic IDs for SM8450 platform interconnects left two links unconverted, fix it to avoid the NULL pointer dereference in runtime, when a pointer to a destination interconnect is not valid: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 <...> Call trace: icc_link_nodes+0x3c/0x100 (P) qcom_icc_rpmh_probe+0x1b4/0x528 platform_probe+0x64/0xc0 really_probe+0xc4/0x2a8 __driver_probe_device+0x80/0x140 driver_probe_device+0x48/0x170 __device_attach_driver+0xc0/0x148 bus_for_each_drv+0x88/0xf0 __device_attach+0xb0/0x1c0 device_initial_probe+0x58/0x68 bus_probe_device+0x40/0xb8 deferred_probe_work_func+0x90/0xd0 process_one_work+0x15c/0x3c0 worker_thread+0x2e8/0x400 kthread+0x150/0x208 ret_from_fork+0x10/0x20 Code: 900310f4 911d6294 91008280 94176078 (f94002a0) ---[end trace 0000000000000000]--- Kernel panic - not syncing: Oops: Fatal exception	2026-05-08	5.5
CVE-2026-43337	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix NULL pointer dereference in dcn401_init_hw() dcn401_init_hw() assumes that update_bw_bounding_box() is valid when entering the update path. However, the existing condition: ((!fams2_enable && update_bw_bounding_box) || freq_changed) does not guarantee this, as the freq_changed branch can evaluate to true independently of the callback pointer. This can result in calling update_bw_bounding_box() when it is NULL. Fix this by separating the update condition from the pointer checks and ensuring the callback, dc->clk_mgr, and bw_params are validated before use. Fixes the below: ../dc/hwss/dcn401/dcn401_hwseq.c:367 dcn401_init_hw() error: we previously assumed 'dc->res_pool->funcs->update_bw_bounding_box' could be null (see line 362) (cherry picked from commit 86117c5ab42f21562fedb0a64bffea3ee5fcd477)	2026-05-08	5.5
CVE-2026-43338	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: reserve enough transaction items for qgroup ioctls Currently our qgroup ioctls don't reserve any space, they just do a transaction join, which does not reserve any space, neither for the quota tree updates nor for the delayed refs generated when updating the quota tree. The quota root uses the global block reserve, which is fine most of the time since we don't expect a lot of updates to the quota root, or to be too close to -ENOSPC such that other critical metadata updates need to resort to the global reserve. However this is not optimal, as not reserving proper space may result in a transaction abort due to not reserving space for delayed refs and then abusing the use of the global block reserve.	2026-05-08	5.5

		For example, the following reproducer (which is unlikely to model any real world use case, but just to illustrate the problem), triggers such a transaction abort due to -ENOSPC when running delayed refs:
		
		$ cat test.sh
		#!/bin/bash
		
		DEV=/dev/nullb0
		MNT=/mnt/nullb0
		
		umount $DEV &> /dev/null
		# Limit device to 1G so that it's much faster to reproduce the issue.
		mkfs.btrfs -f -b 1G $DEV
		mount -o commit=600 $DEV $MNT
		
		fallocate -l 800M $MNT/filler
		btrfs quota enable $MNT
		
		for ((i = 1; i <= 400000; i++)); do
		  btrfs qgroup create 1/$i $MNT
		done
		
		umount $MNT
		
		When running this, we can see in dmesg/syslog that a transaction abort happened:
		
		[436.490] BTRFS error (device nullb0): failed to run delayed ref for logical 30408704 num_bytes 16384 type 176 action 1 ref_mod 1: -28
		[436.493] ------------[cut here]------------
		[436.494] BTRFS: Transaction aborted (error -28)
		[436.495] WARNING: fs/btrfs/extent-tree.c:2247 at btrfs_run_delayed_refs+0xd9/0x110 [btrfs], CPU#4: umount/2495372
		[436.497] Modules linked in: btrfs loop (...)
		[436.508] CPU: 4 UID: 0 PID: 2495372 Comm: umount Tainted: G W 6.19.0-rc8-btrfs-next-225+ #1 PREEMPT(full)
		[436.510] Tainted: [W]=WARN
		[436.511] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.2-0-gea1b7a073390-prebuilt.qemu.org 04/01/2014
		[436.513] RIP: 0010:btrfs_run_delayed_refs+0xdf/0x110 [btrfs]
		[436.514] Code: 0f 82 ea (...)
		[436.518] RSP: 0018:ffffd511850b7d78 EFLAGS: 00010292
		[436.519] RAX: 00000000fffffffe4 RBX: ffff8f120dad37e0 RCX: 0000000002040001
		[436.520] RDX: 0000000000000002 RSI: 00000000fffffffe4 RDI: ffffffffc090fd80
		[436.522] RBP: 0000000000000000 R08: 0000000000000001 R09: ffffffffc04d1867
		[436.523] R10: ffff8f18dc1fffa8 R11: 0000000000000003 R12: ffff8f173aa89400
		[436.524] R13: 0000000000000000 R14: ffff8f173aa89400 R15: 0000000000000000
		[436.526] FS: 00007fe59045d840(0000) GS:ffff8f192e22e000(0000) knlGS:0000000000000000
		[436.527] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
		[436.528] CR2: 00007fe5905ff2b0 CR3: 0000000060710a002 CR4: 0000000000370ef0
		[436.530] Call Trace:
		[436.530] <TASK>
		[436.530] btrfs_commit_transaction+0x73/0xc00 [btrfs]
		[436.531] ? btrfs_attach_transaction_barrier+0x1e/0x70 [btrfs]
		[436.532] sync_filesystem+0x7a/0x90
		[436.533] generic_shutdown_super+0x28/0x180
		[436.533] kill_anon_super+0x12/0x40
		[436.534] btrfs_kill_super+0x12/0x20 [btrfs]
		[436.534] deactivate_locked_super+0x2f/0xb0
		[436.534] cleanup_mnt+0xea/0x180
		[436.535] task_work_run+0x58/0xa0
		[436.535] exit_to_user_mode_loop+0xed/0x480
		[436.536] ? __x64_sys_umount+0x68/0x80
		[436.536] do_syscall_64+0x2a5/0xf20
		[436.537] entry_SYSCALL_64_after_hwframe+0x76/0x7e
		[436.537] RIP: 0033:0x7fe5906b6217
		[436.538] Code: 0d 00 f7 (...)
		[436.540] RSP: 002b:00007ffcd87a61f8 EFLAGS: 00000246 ORIG_RAX: 00000000000000a6
		[436.541] RAX: 0000000000000000 RBX: 00005618b9ecadc8 RCX: 00007fe5906b6217
		[436.541] RDX: 0000000000000000 RSI: 0000000000000000 RDI: 00005618b9ecb100
		[436.542] RBP: 0000000000000000 R08: 00007ffcd87a4fe0 R09: 00000000ffffffff
		[436.544] R10: 0000000000000103 R11:
		---truncated---		

[CVE-2026-43340](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: comedi: Reinit dev->spinlock between attachments to low-level drivers `struct comedi_device` is the main controlling structure for a COMEDI device created by the COMEDI subsystem. It contains a member `spinlock`	2026-05-08	5.5

		containing a spin-lock that is initialized by the COMEDI subsystem, but is reserved for use by a low-level driver attached to the COMEDI device (at least since commit 25436dc9d84f ("Staging: comedi: remove RT code")). Some COMEDI devices (those created on initialization of the COMEDI subsystem when the "comedi.comedi_num_legacy_minors" parameter is non-zero) can be attached to different low-level drivers over their lifetime using the `COMEDI_DEVCONFIG` ioctl command. This can result in inconsistent lock states being reported when there is a mismatch in the spin-lock locking levels used by each low-level driver to which the COMEDI device has been attached. Fix it by reinitializing `dev->spinlock` before calling the low-level driver's `attach` function pointer if `CONFIG_LOCKDEP` is enabled.		
CVE-2026-43343	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_subset: Fix unbalanced refcnt in geth_free geth_alloc() increments the reference count, but geth_free() fails to decrement it. This prevents the configuration of attributes via configfs after unlinking the function. Decrement the reference count in geth_free() to ensure proper cleanup.	2026-05-08	5.5
CVE-2026-43344	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/uncore: Fix die ID init and look up bugs In snbep_pci2phy_map_init(), in the nr_node_ids > 8 path, uncore_device_to_die() may return -1 when all CPUs associated with the UBOX device are offline. Remove the WARN_ON_ONCE(die_id == -1) check for two reasons: - The current code breaks out of the loop. This is incorrect because pci_get_device() does not guarantee iteration in domain or bus order, so additional UBOX devices may be skipped during the scan. - Returning -EINVAL is incorrect, since marking offline buses with die_id == -1 is expected and should not be treated as an error. Separately, when NUMA is disabled on a NUMA-capable platform, pcibus_to_node() returns NUMA_NO_NODE, causing uncore_device_to_die() to return -1 for all PCI devices. As a result, spr_update_device_location(), used on Intel SPR and EMR, ignores the corresponding PMON units and does not add them to the RB tree. Fix this by using uncore_pcibus_to_dieid(), which retrieves topology from the UBOX GIDNIDMAP register and works regardless of whether NUMA is enabled in Linux. This requires snbep_pci2phy_map_init() to be added in spr_uncore_pci_init(). Keep uncore_device_to_die() only for the nr_node_ids > 8 case, where NUMA is expected to be enabled.	2026-05-08	5.5
CVE-2026-43346	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ice: ptp: don't WARN when controlling PF is unavailable In VFIO passthrough setups, it is possible to pass through only a PF which doesn't own the source timer. In that case the PTP controlling PF (adapter->ctrl_pf) is never initialized in the VM, so ice_get_ctrl_ptp() returns NULL and triggers WARN_ON() in ice_ptp_setup_pf(). Since this is an expected behavior in that configuration, replace WARN_ON() with an informational message and return -EOPNOTSUPP.	2026-05-08	5.5
CVE-2026-43348	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: mshv_vtl: Fix vmemmap_shift exceeding MAX_FOLIO_ORDER When registering VTL0 memory via MSHV_ADD_VTL0_MEMORY, the kernel computes pgmap->vmemmap_shift as the number of trailing zeros in the OR of start_pfn and last_pfn, intending to use the largest compound page order both endpoints are aligned to. However, this value is not clamped to MAX_FOLIO_ORDER, so a sufficiently aligned range (e.g. physical range [0x800000000000, 0x800080000000), corresponding to start_pfn=0x800000000 with 35 trailing zeros) can produce a shift larger than what memremap_pages() accepts, triggering a WARN and returning -EINVAL: WARNING: ... memremap_pages+0x512/0x650	2026-05-08	5.5

		requested folio size unsupported The MAX_FOLIO_ORDER check was added by commit 646b67d57589 ("mm/memremap: reject unreasonable folio/compound page sizes in memremap_pages()"). Fix this by clamping vmemmap_shift to MAX_FOLIO_ORDER so we always request the largest order the kernel supports, in those cases, rather than an out-of-range value. Also fix the error path to propagate the actual error code from devm_memremap_pages() instead of hard-coding -EFAULT, which was masking the real -EINVAL return.		
CVE-2026-43349	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid uninit-value access in f2fs_sanity_check_node_footer syzbot reported a f2fs bug as below: BUG: KMSAN: uninit-value in f2fs_sanity_check_node_footer+0x374/0xa20 fs/f2fs/node.c:1520 f2fs_sanity_check_node_footer+0x374/0xa20 fs/f2fs/node.c:1520 f2fs_finish_read_bio+0xe1e/0x1d60 fs/f2fs/data.c:177 f2fs_read_end_io+0x6ab/0x2220 fs/f2fs/data.c:-1 bio_endio+0x1006/0x1160 block/bio.c:1792 submit_bio_noacct+0x533/0x2960 block/blk-core.c:891 submit_bio+0x57a/0x620 block/blk-core.c:926 blk_crypto_submit_bio include/linux/blk-crypto.h:203 [inline] f2fs_submit_read_bio+0x12c/0x360 fs/f2fs/data.c:557 f2fs_submit_page_bio+0xee2/0x1450 fs/f2fs/data.c:775 read_node_folio+0x384/0x4b0 fs/f2fs/node.c:1481 __get_node_folio+0x5db/0x15d0 fs/f2fs/node.c:1576 f2fs_get_inode_folio+0x40/0x50 fs/f2fs/node.c:1623 do_read_inode fs/f2fs/inode.c:425 [inline] f2fs_iget+0x1209/0x9380 fs/f2fs/inode.c:596 f2fs_fill_super+0x8f5a/0xb2e0 fs/f2fs/super.c:5184 get_tree_bdev_flags+0x6e6/0x920 fs/super.c:1694 get_tree_bdev+0x38/0x50 fs/super.c:1717 f2fs_get_tree+0x35/0x40 fs/f2fs/super.c:5436 vfs_get_tree+0xb3/0x5d0 fs/super.c:1754 fc_mount fs/namespace.c:1193 [inline] do_new_mount_fc fs/namespace.c:3763 [inline] do_new_mount+0x885/0x1dd0 fs/namespace.c:3839 path_mount+0x7a2/0x20b0 fs/namespace.c:4159 do_mount fs/namespace.c:4172 [inline] __do_sys_mount fs/namespace.c:4361 [inline] __se_sys_mount+0x704/0x7f0 fs/namespace.c:4338 __x64_sys_mount+0xe4/0x150 fs/namespace.c:4338 x64_sys_call+0x39f0/0x3ea0 arch/x86/include/generated/asm/syscalls_64.h:166 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x134/0xf80 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f The root cause is: in f2fs_finish_read_bio(), we may access uninit data in folio if we failed to read the data from device into folio, let's add a check condition to avoid such issue.	2026-05-08	5.5
CVE-2026-43351	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Eagerly init vgic dist/redist on vgic creation If vgic_allocate_private_irqs_locked() fails for any odd reason, we exit kvm_vgic_create() early, leaving dist->rd_regions uninitialised. kvm_vgic_dist_destroy() then comes along and walks into the weeds trying to free the RDs. Got to love this stuff. Solve it by moving all the static initialisation early, and make sure that if we fail halfway, we're in a reasonable shape to perform the rest of the teardown. While at it, reset the vgic model on failure, just in case...	2026-05-08	5.5
CVE-2026-43354	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: proximity: hx9023s: Protect against division by zero in set_samp_freq Avoid division by zero when sampling frequency is unspecified.	2026-05-08	5.5
CVE-2026-43355	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: light: bh1780: fix PM runtime leak on error path Move pm_runtime_put_autosuspend() before the error check to ensure the PM runtime reference count is always decremented after	2026-05-08	5.5

		pm_runtime_get_sync(), regardless of whether the read operation succeeds or fails.		
CVE-2026-43356	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: imu: adis: Fix NULL pointer dereference in adis_init The adis_init() function dereferences adis->ops to check if the individual function pointers (write, read, reset) are NULL, but does not first check if adis->ops itself is NULL. Drivers like adis16480, adis16490, adis16545 and others do not set custom ops and rely on adis_init() assigning the defaults. Since struct adis is zero-initialized by devm_iio_device_alloc(), adis->ops is NULL when adis_init() is called, causing a NULL pointer dereference: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 pc : adis_init+0xc0/0x118 Call trace: adis_init+0xc0/0x118 adis16480_probe+0xe0/0x670 Fix this by checking if adis->ops is NULL before dereferencing it, falling through to assign the default ops in that case.	2026-05-08	5.5
CVE-2026-43357	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: gyro: mpu3050-core: fix pm_runtime error handling The return value of pm_runtime_get_sync() is not checked, allowing the driver to access hardware that may fail to resume. The device usage count is also unconditionally incremented. Use pm_runtime_resume_and_get() which propagates errors and avoids incrementing the usage count on failure. In preenable, add pm_runtime_put_autosuspend() on set_8khz_samplerate() failure since postdisable does not run when preenable fails.	2026-05-08	5.5
CVE-2026-43358	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: add missing RCU unlock in error path in try_release_subpage_extent_buffer() Call rcu_read_lock() before exiting the loop in try_release_subpage_extent_buffer() because there is a rcu_read_unlock() call past the loop. This has been detected by the Clang thread-safety analyzer.	2026-05-08	5.5
CVE-2026-43359	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix transaction abort on set received ioctl due to item overflow If the set received ioctl fails due to an item overflow when attempting to add the BTRFS_UUID_KEY_RECEIVED_SUBVOL we have to abort the transaction since we did some metadata updates before. This means that if a user calls this ioctl with the same received UUID field for a lot of subvolumes, we will hit the overflow, trigger the transaction abort and turn the filesystem into RO mode. A malicious user could exploit this, and this ioctl does not even requires that a user has admin privileges (CAP_SYS_ADMIN), only that he/she owns the subvolume. Fix this by doing an early check for item overflow before starting a transaction. This is also race safe because we are holding the subvol_sem semaphore in exclusive (write) mode. A test case for fstests will follow soon.	2026-05-08	5.5
CVE-2026-43360	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix transaction abort on file creation due to name hash collision If we attempt to create several files with names that result in the same hash, we have to pack them in same dir item and that has a limit inherent to the leaf size. However if we reach that limit, we trigger a transaction abort and turns the filesystem into RO mode. This allows for a malicious user to disrupt a system, without the need to have administration privileges/capabilities. Reproducer: $ cat exploit-hash-collisions.sh #!/bin/bash DEV=/dev/sdi	2026-05-08	5.5

		MNT=/mnt/sdi # Use smallest node size to make the test faster and require fewer file # names that result in hash collision. mkfs.btrfs -f --nodesize 4K $DEV mount $DEV $MNT # List of names that result in the same crc32c hash for btrfs. declare -a names=('foobar' '%a8tYkxfGMLWRGr55QSeQc4PBNH9PCLIvR6jZnkDtUUru1t@RouaUe_L:@xGkbO3nCwvLNYeK9vhE628gss:T$yZjZ5l-Nbd6CbC$M=hqE-ujhJICXyIxBvYrIU9-TDC' 'AQci3EUB%shMsg-N%frgU:02ByLs=IPJU0OpgiWit5nexSyxZDncY6WB:=zKZuk5Zy0DD$Ua78%MelgBuMqaHGyKsJUFf9s=UW80PcJmKctb46KveLSiUtNmqrMiL9-Y0I_l5Fnam04CGIg=8@U:Z' 'CvVqJpJzueKcuA$wqwePfyu7VxuWNN3ho$p0zi2H8QFYK$7YlEqOhhb%:hHgjhIjW5vnqWHKNP4' 'ET:vk@rFU4tsvMB0$C_p=xQHaYZjvoF%-BTc%wkFW8yaDAPcCYoR%x$FH5O:' 'HwTon%v7SGSP4FE08jBwwiu5aot2CFKXHTeEAa@38fUcNGOWvE@Mz6WBeDH_VooaZ6AgsXPkVGwy9l@@ZbNXabUU9csiWrrOp0MWUdfi$EZ3w9GkIqtz7I_eOsByOkBOO' 'Ij%2VlFGXSuPvxJGf5UWy6O@1svxGha%b@=%wjkq:ClgE6u7eJOjmQY5qTtxE2Rjbis9@us' 'KBkjG5%9R8K9sOG8UTnAYjxLNAvBmvV5vz3IiZaPmKuLYO03-6asI9lJ_j4@6Xo$KZicaLWJ3Pv8XEwVeUPMwbHYWwbx0pYvNlGMO9F:ZhHAwyctnGy%_eujl%WPd4U2BI7qooOSr85J-C2V$LfY' 'NcRfDfuUQ2=zP8K3CCF5dFcpfiOm6mwenShsAb_F%n6GAGC7fT2JFFn:c35X-3aYwoq7jNX5$ZJ6hI3wnZs$7KgGi7wjulffhHNUxAT0fRRLF39vJ@NvaEMxsMO' 'Oj42AQAEzRoTxa5OuSKIr=A_lwGMy132v4g3Pdq1GvUG9874YseIFQ6QU' 'Ono7avN5GjC:_6dBJ_' 'WHmN2gnmaN-9dVDy4aWo:yNGFzz8qsJyJhWEWcud7$QzN2D9R0efIWWEdu5kwWr73NZm4=@CoCDxrrZnRITr-kGtU_cfW2:%2_am' 'WiFnuTEhAG9FEC6zopQmj-A-$LDQ0T3WULz%ox3UZAPybSV6v1Z$b4L_XBi4M4BMBtJZpz93r9xafpB77r:lbwvitWRyo$odnAUYlYMmU4RvgnNd--e=l5hiEjGLETTtaScWlQp8mYsBovZwM2k' 'XKyH=OsOAF3p%uziGF_ZVr$ivrvhVgD@1u%5RtrV-gl_vqAwHkK@x7YwlxX3qT6WKKQ%PR56NrUBU2dOAOAdzr2=5nJuKPM-T-$ZpQfCL7phxQbUcb:BZOTPaFExc-qK-gDRCDW2' 'd3uUR6OFEwZr%ns1XH_@tbxA@cCPmbBRLdyh7p6V45H$P2$F%w0RqrD3M0g8aGvWpoTFMiBdOTJXjD:JF7=h9a_43xBywYAP%r$SPZi%zDg%ql-KvkdUCtF9OLaQlxmd' 'ePTpbnit%hyNm@WELlpKzNZYOzOTf8EQ$sEfkMy1VOfIUu3coyvIr13-Y7Sv5v-Ivax2Go_GQRFMU1b3362nktT9WOJf3SpT%z8sZmM3gvYQBDgmKI%%RM-G7hyrhgYflOw%z::ZRcv5O:lDCFm' 'evqk743Y@dvZAiG5J05L_ROFV@$2%rVWJ2%3nxV72-W7$e$-SK3tuSHA2mBt$qloC5jwNx33GmQUjD%akhBPu=VJ5g$xhlZiaFtTrjeeM5x7dt4cHpX0cZkmfImndYzGmvwQG:$euFYmXn$_2rA9mKZ' 'gkgUtnihWXsZQTEkrMAWIxir09k3t7jk_IK25t1:cy1XWN0GGqC%FrySdcmU7M8MuPO_ppkLw3=Dfr0UuBAL4%GFk2$Ma10V1jDRGJje%Xx9EV2ERaWKtjpwiZwh0gCSJsj5UL7CR8RtW5opCVFKGGy8Cky' 'hNgsG_8lNRik3PvphqPm0yEH3P%%fYG:kQLY=6O-61Wa6nrV_WVGR6TLB09vHOv%g4VQRP8Gzx7VXUY1qvZyS' 'isA7JVzN12xCxVPJZ_qoLm-pTBuhjjHMvV7o=F:EaClfYNyFGlsfw-Kf%uxdqW-kwk1sPl2vhbjyHU1A6$hz' 'kiJ_fgcdZFDiOptjgH5PN9-PSyLO4fbk_:u5_2tz35lV_iXiJ6cx7pwjTtKy-XGaQ5IefmpJ4N_ZqGsqCsKuqOOBgf9LkUdffHet@Wu' 'lvwtxyhE9:%Q3UxeHiViUyNzJsy:fm38pg_b6s25JvdhOAT=1s0$pG25x=LZ2rlHTszj=gN6M4zHZYr_qrB49i=pA--@WqWLIuX7o1S_SfS@2FSiUZN' 'rC24cw3UBDZ=5qJBUMs9e$=S4Y94ni%Z8639vnrGp=0Hv4z3dNFL0fBLmQ40=EYIY:Z=SLc@QLMSt2zsss2ZXrP7j4=' 'uwGl2s-fFrf@GqS=DQqq2I0LJSsOmM%xzTjS:lzXguE3wChdMoHYtLRKPvfaPOZF2fER@j53evbKa7R%A7r4%YEkD=kicJe@SFiGtXHbKe4gCgPAYbnVn' 'UG37U6KKua2bgc:IHzRs7BnB6FD:2Mt5Cc5NdlsW%$1tyvnfz7S27FvNkroXwAW:mBZLA1@qa9WnDbHCDmQmfPMC9z-Eq6QT0jhhPpqyymaD:R02ghwYo%yx7SAaaq-:x33LYpei$5g8DMl3C' 'y2vjek0FE1PDJC0qpfnN:x8k2wCFZ9xiUF2ege=JnP98R%wxjKkdfEiLWvQzmnW' '8-HCSgH5B%K7P8_jaVtQhBXpBk:pE-$P7ts58U0J@iR9YZntMPl7j$s62yAJO@_9eanFPS54b=UTw$94C-t=HLxT8n6o9P=QnIxq-f1=Ne2dvhe6WbjEQtc' 'YPPh:IFt2mtR6XWSmjHptXL_hbSYu8bMw-JP8@PNyaFkdNFsk$M=xfL6LDKCDM-mSyGA_2MBwZ8Dr4=R1D%7-mC ---truncated---		
CVE-2026-43361	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix transaction abort when snapshotting received subvolumes Currently a user can trigger a transaction abort by snapshotting a	2026-05-08	5.5

		previously received snapshot a bunch of times until we reach a BTRFS_UUID_KEY_RECEIVED_SUBVOL item overflow (the maximum item size we can store in a leaf). This is very likely not common in practice, but if it happens, it turns the filesystem into RO mode. The snapshot, send and set_received_subvol and subvol_setflags (used by receive) don't require CAP_SYS_ADMIN, just inode_owner_or_capable(). A malicious user could use this to turn a filesystem into RO mode and disrupt a system.
		
		Reproducer script:
		
		$ cat test.sh
		#!/bin/bash
		
		DEV=/dev/sdi
		MNT=/mnt/sdi
		
		# Use smallest node size to make the test faster.
		mkfs.btrfs -f --nodesize 4K $DEV
		mount $DEV $MNT
		
		# Create a subvolume and set it to RO so that it can be used for send.
		btrfs subvolume create $MNT/sv
		touch $MNT/sv/foo
		btrfs property set $MNT/sv ro true
		
		# Send and receive the subvolume into snaps/sv.
		mkdir $MNT/snaps
		btrfs send $MNT/sv | btrfs receive $MNT/snaps
		
		# Now snapshot the received subvolume, which has a received_uuid, a
		# lot of times to trigger the leaf overflow.
		total=500
		for ((i = 1; i <= $total; i++)); do
		  echo -ne "\rCreating snapshot $i/$total"
		  btrfs subvolume snapshot -r $MNT/snaps/sv $MNT/snaps/sv_$i > /dev/null
		done
		echo
		
		umount $MNT
		
		When running the test:
		
		$ ./test.sh
		(...)
		Create subvolume '/mnt/sdi/sv'
		At subvol /mnt/sdi/sv
		At subvol sv
		Creating snapshot 496/500ERROR: Could not create subvolume: Value too large for defined data type
		Creating snapshot 497/500ERROR: Could not create subvolume: Read-only file system
		Creating snapshot 498/500ERROR: Could not create subvolume: Read-only file system
		Creating snapshot 499/500ERROR: Could not create subvolume: Read-only file system
		Creating snapshot 500/500ERROR: Could not create subvolume: Read-only file system
		
		And in dmesg/syslog:
		
		$ dmesg
		(...)
		[251067.627338] BTRFS warning (device sdi): insert uuid item failed -75 (0x4628b21c4ac8d898, 0x2598bee2b1515c91) type 252!
		[251067.629212] ------------[cut here]------------
		[251067.630033] BTRFS: Transaction aborted (error -75)
		[251067.630871] WARNING: fs/btrfs/transaction.c:1907 at create_pending_snapshot.cold+0x52/0x465 [btrfs], CPU#10: btrfs/615235
		[251067.632851] Modules linked in: btrfs dm_zero (...)
		[251067.644071] CPU: 10 UID: 0 PID: 615235 Comm: btrfs Tainted: G W 6.19.0-rc8-btrfs-next-225+ #1 PREEMPT(full)
		[251067.646165] Tainted: [W]=WARN
		[251067.646733] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.2-0-gea1b7a073390-prebuilt.qemu.org 04/01/2014
		[251067.648735] RIP: 0010:create_pending_snapshot.cold+0x55/0x465 [btrfs]
		[251067.649984] Code: f0 48 0f (...)
		[251067.653313] RSP: 0018:ffffce644908fae8 EFLAGS: 00010292
		[251067.653987] RAX: 00000000ffffff01 RBX: ffff8e5639e63a80 RCX: 00000000ffffffd3
		[251067.655042] RDX: ffff8e53faa76b00 RSI: 00000000ffffffb5 RDI: ffffffffc0919750
		[251067.656077] RBP: ffffce644908fbd8 R08: 0000000000000000 R09: ffffce644908f820
		[251067.657068] R10: ffff8e5adc1fffa8 R11: 0000000000000003 R12: ffff8e53c0431bd0
		[251067.658050] R13: ffff8e5414593600 R14: ffff8e55efafd000 R15: 00000000ffffffb5
		[251067.659019] FS: 00007f2a4944b3c0(0000) GS:ffff8e5b27dae000(0000) knlGS:0000000000000000		

		[251067.660115] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [251067.660943] CR2: 00007ffc5aa57898 CR3: 00000005813a2003 CR4: 0000000000370ef0 [251067.661972] Call Trace: [251067.662292] <TASK> [251067.662653] create_pending_snapshots+0x97/0xc0 [btrfs] [251067.663413] btrfs_commit_transaction+0x26e/0xc00 [btrfs] [251067.664257] ? btrfs_qgroup_convert_reserved_meta+0x35/0x390 [btrfs] [251067.665238] ? _raw_spin_unlock+0x15/0x30 [251067.665837] ? record_root_ ---truncated---		
CVE-2026-43363	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86/apic: Disable x2apic on resume if the kernel expects so When resuming from s2ram, firmware may re-enable x2apic mode, which may have been disabled by the kernel during boot either because it doesn't support IRQ remapping or for other reasons. This causes the kernel to continue using the xapic interface, while the hardware is in x2apic mode, which causes hangs. This happens on defconfig + bare metal + s2ram. Fix this in lapic_resume() by disabling x2apic if the kernel expects it to be disabled, i.e. when x2apic_mode = 0. The ACPI v6.6 spec, Section 16.3 [1] says firmware restores either the pre-sleep configuration or initial boot configuration for each CPU, including MSR state: When executing from the power-on reset vector as a result of waking from an S2 or S3 sleep state, the platform firmware performs only the hardware initialization required to restore the system to either the state the platform was in prior to the initial operating system boot, or to the pre-sleep configuration state. In multiprocessor systems, non-boot processors should be placed in the same state as prior to the initial operating system boot. (further ahead) If this is an S2 or S3 wake, then the platform runtime firmware restores minimum context of the system before jumping to the waking vector. This includes: CPU configuration. Platform runtime firmware restores the pre-sleep configuration or initial boot configuration of each CPU (MSR, MTRR, firmware update, SMBase, and so on). Interrupts must be disabled (for IA-32 processors, disabled by CLI instruction). (and other things) So at least as per the spec, re-enablement of x2apic by the firmware is allowed if "x2apic on" is a part of the initial boot configuration. [1] https://uefi.org/specs/ACPI/6.6/16_Waking_and_Sleeping.html#initialization [bp: Massage.]	2026-05-08	5.5
CVE-2026-43364	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ublk: fix NULL pointer dereference in ublk_ctrl_set_size() ublk_ctrl_set_size() unconditionally dereferences ub->ub_disk via set_capacity_and_notify() without checking if it is NULL. ub->ub_disk is NULL before UBLK_CMD_START_DEV completes (it is only assigned in ublk_ctrl_start_dev()) and after UBLK_CMD_STOP_DEV runs (ublk_detach_disk() sets it to NULL). Since the UBLK_CMD_UPDATE_SIZE handler performs no state validation, a user can trigger a NULL pointer dereference by sending UPDATE_SIZE to a device that has been added but not yet started, or one that has been stopped. Fix this by checking ub->ub_disk under ub->mutex before dereferencing it, and returning -ENODEV if the disk is not available.	2026-05-08	5.5
CVE-2026-43367	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix a few more NULL pointer dereference in device cleanup I found a few more paths that cleanup fails due to a NULL version pointer on unsupported hardware. Add NULL checks as applicable. (cherry picked from commit f5a05f8414fc10f307eb965f303580c7778f8dd2)	2026-05-08	5.5

CVE-2026-43369	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix NULL pointer dereference in device cleanup When GPU initialization fails due to an unsupported HW block IP blocks may have a NULL version pointer. During cleanup in amdgpu_device_fini_hw, the code calls amdgpu_device_set_pg_state and amdgpu_device_set_cg_state which iterate over all IP blocks and access adev->ip_blocks[i].version without NULL checks, leading to a kernel NULL pointer dereference. Add NULL checks for adev->ip_blocks[i].version in both amdgpu_device_set_cg_state and amdgpu_device_set_pg_state to prevent dereferencing NULL pointers during GPU teardown when initialization has failed. (cherry picked from commit b7ac77468cda92eecae560b05f62f997a12fe2f2)	2026-05-08	5.5
CVE-2026-43371	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: macb: Shuffle the tx ring before enabling tx Quanyang observed that when using an NFS rootfs on an AMD ZynqMp board, the rootfs may take an extended time to recover after a suspend. Upon investigation, it was determined that the issue originates from a problem in the macb driver. According to the Zynq UltraScale TRM [1], when transmit is disabled, the transmit buffer queue pointer resets to point to the address specified by the transmit buffer queue base address register. In the current implementation, the code merely resets `queue->tx_head` and `queue->tx_tail` to '0'. This approach presents several issues: - Packets already queued in the tx ring are silently lost, leading to memory leaks since the associated skbs cannot be released. - Concurrent write access to `queue->tx_head` and `queue->tx_tail` may occur from `macb_tx_poll()` or `macb_start_xmit()` when these values are reset to '0'. - The transmission may become stuck on a packet that has already been sent out, with its 'TX_USED' bit set, but has not yet been processed. However, due to the manipulation of 'queue->tx_head' and 'queue->tx_tail', `macb_tx_poll()` incorrectly assumes there are no packets to handle because `queue->tx_head == queue->tx_tail`. This issue is only resolved when a new packet is placed at this position. This is the root cause of the prolonged recovery time observed for the NFS root filesystem. To resolve this issue, shuffle the tx ring and tx skb array so that the first unsent packet is positioned at the start of the tx ring. Additionally, ensure that updates to `queue->tx_head` and `queue->tx_tail` are properly protected with the appropriate lock. [1] https://docs.amd.com/v/u/en-US/ug1085-zynq-ultrascale-trm	2026-05-08	5.5
CVE-2026-43372	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: dsa: microchip: Fix error path in PTP IRQ setup If request_threaded_irq() fails during the PTP message IRQ setup, the newly created IRQ mapping is never disposed. Indeed, the ksz_ptp_irq_setup()'s error path only frees the mappings that were successfully set up. Dispose the newly created mapping if the associated request_threaded_irq() fails at setup.	2026-05-08	5.5
CVE-2026-43375	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mctp: fix device leak on probe failure Driver core holds a reference to the USB interface and its parent USB device while the interface is bound to a driver and there is no need to take additional references unless the structures are needed after disconnect. This driver takes a reference to the USB device during probe but does not to release it on probe failures. Drop the redundant device reference to fix the leak, reduce cargo culting, make it easier to spot drivers where an extra reference is needed, and reduce the risk of further memory leaks.	2026-05-08	5.5

CVE-2026-20219	cisco - multiple products	A vulnerability in the REST API of Cisco Slido could have allowed an authenticated, remote attacker to access the social profile data of other users or affect quiz and poll results. Cisco has addressed this vulnerability in Cisco Slido and no customer action is needed. This vulnerability existed because of the presence of an insecure direct object reference. Prior to this vulnerability being addressed, an attacker could have exploited this vulnerability by sending a crafted request to the vulnerable API endpoint. A successful exploit could have allowed the attacker to view the social profiles of other users or affect quiz and poll results.	2026-05-06	5.4
CVE-2026-7931	google - chrome	Insufficient validation of untrusted input in iOS in Google Chrome on iOS prior to 148.0.7778.96 allowed a remote attacker to perform UI spoofing via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7935	google - chrome	Inappropriate implementation in Speech in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform UI spoofing via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7939	google - chrome	Inappropriate implementation in SanitizerAPI in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to inject arbitrary scripts or HTML (UXSS) via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7950	google - chrome	Out of bounds read and write in GFX in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform arbitrary read/write via malicious network traffic. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7958	google - chrome	Inappropriate implementation in ServiceWorker in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to inject arbitrary scripts or HTML (UXSS) via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7962	google - chrome	Insufficient policy enforcement in DirectSockets in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform arbitrary read/write via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	5.4
CVE-2026-7998	google - chrome	Insufficient validation of untrusted input in Dialog in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform UI spoofing via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8003	google - chrome	Insufficient validation of untrusted input in TabGroups in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform UI spoofing via malicious network traffic. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8006	google - chrome	Insufficient policy enforcement in DevTools in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to perform UI spoofing via a crafted Chrome Extension. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8008	google - chrome	Inappropriate implementation in DevTools in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to perform UI spoofing via a crafted Chrome Extension. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8012	google - chrome	Inappropriate implementation in MHTML in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to inject arbitrary scripts or HTML (UXSS) via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8015	google - chrome	Inappropriate implementation in Media in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform UI spoofing via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-8019	google - chrome	Insufficient policy enforcement in WebApp in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform UI spoofing via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.4
CVE-2026-33857	apache - http_server	Out-of-bounds Read vulnerability in mod_proxy_ajp of Apache HTTP Server. This issue affects Apache HTTP Server: through 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-04	5.3
CVE-2026-34032	apache - http_server	Improper Null Termination, Out-of-bounds Read vulnerability in Apache HTTP Server. This issue affects Apache HTTP Server: through 2.4.66. Users are recommended to upgrade to version 2.4.67, which fixes the issue.	2026-05-04	5.3
CVE-2026-33007	apache - http_server	A NULL pointer dereference in the mod_authn_socache in Apache HTTP Server 2.4.66 and earlier allows an unauthenticated remote user to crash a child process in a caching forward proxy configuration. Users are recommended to upgrade to version 2.4.67, which fixes this issue.	2026-05-04	5.3
CVE-2026-43868	apache - thrift	Memory Allocation with Excessive Size Value 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.	2026-05-05	5.3
CVE-2026-20195	cisco - Cisco Identity Services Engine Software	A vulnerability in an identity management API endpoint of Cisco ISE could allow an unauthenticated, remote attacker to enumerate valid user accounts on an affected device. This vulnerability exists because error messages are observed when the affected API endpoint is called. An attacker could exploit this vulnerability by sending a series of crafted requests to the affected endpoint and analyzing the differentiated responses. A successful exploit could allow the attacker to compile a list of valid usernames on an affected system.	2026-05-06	5.3
CVE-2026-7955	google - chrome	Uninitialized Use in GPU in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	5.3
CVE-2026-7960	google - chrome	Race in Speech in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	5.3

CVE-2026-8020	google - chrome	Uninitialized Use in GPU in Google Chrome on Android prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.3
CVE-2026-23927	zabbix - Zabbix	A user able to connect to Agent 2 can inject an Oracle TNS connection string via the 'service' parameter. This can lead to Agent 2 connecting to an attacker-controlled server and leaking Oracle database credentials if they are saved in a named session.	2026-05-06	5.1
CVE-2026-8009	google - chrome	Inappropriate implementation in Cast in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass navigation restrictions via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	5.0
CVE-2026-33006	apache - http_server	A timing attack against mod_auth_digest in Apache HTTP Server 2.4.66 allows a bypass of Digest authentication by a remote attacker. Users are recommended to upgrade to version 2.4.67, which fixes this issue.	2026-05-04	4.8
CVE-2026-35253	oracle - macaron	Vulnerability in the Oracle Macoron Tool product of Oracle Open Source Projects. The supported versions that is affected is v0.22.0. Easily exploitable vulnerability allows unauthenticated attacker with network access via HTTP to compromise Oracle Macaron Tool. Successful attacks of this vulnerability can result in Oracle Macaron Tool failing host address validation.	2026-05-06	4.7
CVE-2025-71274	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rpmsg: core: fix race in driver_override_show() and use core helper The driver_override_show function reads the driver_override string without holding the device_lock. However, the store function modifies and frees the string while holding the device_lock. This creates a race condition where the string can be freed by the store function while being read by the show function, leading to a use-after-free. To fix this, replace the rpmsg_string_attr macro with explicit show and store functions. The new driver_override_store uses the standard driver_set_override helper. Since the introduction of driver_set_override, the comments in include/linux/rpmsg.h have stated that this helper must be used to set or clear driver_override, but the implementation was not updated until now. Because driver_set_override modifies and frees the string while holding the device_lock, the new driver_override_show now correctly holds the device_lock during the read operation to prevent the race. Additionally, since rpmsg_string_attr has only ever been used for driver_override, removing the macro simplifies the code.	2026-05-06	4.7

CVE-2026-43121	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		io_uring/zcrx: fix user_ref race between scrub and refill paths
		
		The io_zcrx_put_niov_uref() function uses a non-atomic check-then-decrement pattern (atomic_read followed by separate atomic_dec) to manipulate user_refs. This is serialized against other callers by rq_lock, but io_zcrx_scrub() modifies the same counter with atomic_xchg() WITHOUT holding rq_lock.
		
		On SMP systems, the following race exists:
		
		CPU0 (refill, holds rq_lock) CPU1 (scrub, no rq_lock)
		put_niov_uref:
		atomic_read(uref) - 1
		// window opens
		atomic_xchg(uref, 0) - 1
		return_niov_freelist(niov) [PUSH #1]
		// window closes
		atomic_dec(uref) - wraps to -1
		returns true
		return_niov(niov)
		return_niov_freelist(niov) [PUSH #2: DOUBLE-FREE]
		
		The same niov is pushed to the freelist twice, causing free_count to exceed nr_iovs. Subsequent freelist pushes then perform an out-of-bounds write (a u32 value) past the kvmalloc'd freelist array into the adjacent slab object.
		
		Fix this by replacing the non-atomic read-then-dec in io_zcrx_put_niov_uref() with an atomic_try_cmpxchg loop that atomically tests and decrements user_refs. This makes the operation safe against concurrent atomic_xchg from scrub without requiring scrub to acquire rq_lock.
		
		[pavel: removed a warning and a comment]
			2026-05-06	4.7

CVE-2026-43163	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: md/bitmap: fix GPF in write_page caused by resize race	2026-05-06	4.7

		A General Protection Fault occurs in write_page() during array resize: RIP: 0010:write_page+0x22b/0x3c0 [md_mod] This is a use-after-free race between bitmap_daemon_work() and __bitmap_resize(). The daemon iterates over `bitmap->storage.filemap` without locking, while the resize path frees that storage via md_bitmap_file_unmap(). `quiesce()` does not stop the md thread, allowing concurrent access to freed pages. Fix by holding `mddev->bitmap_info.mutex` during the bitmap update.		
CVE-2026-43275	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Flush exception handling work when RPM level is zero Ensure that the exception event handling work is explicitly flushed during suspend when the runtime power management level is set to UFS_PM_LVL_0. When the RPM level is zero, the device power mode and link state both remain active. Previously, the UFS core driver bypassed flushing exception event handling jobs in this configuration. This created a race condition where the driver could attempt to access the host controller to handle an exception after the system had already entered a deep power-down state, resulting in a system crash. Explicitly flush this work and disable auto BKOPs before the suspend callback proceeds. This guarantees that pending exception tasks complete and prevents illegal hardware access during the power-down sequence.	2026-05-06	4.7
CVE-2026-43342	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_rndis: Protect RNDIS options with mutex The class/subclass/protocol options are suspectible to race conditions as they can be accessed concurrently through configfs. Use existing mutex to protect these options. This issue was identified during code inspection.	2026-05-08	4.7
CVE-2026-7932	google - chrome	Insufficient policy enforcement in Downloads in Google Chrome prior to 148.0.7778.96 allowed a local attacker to bypass navigation restrictions via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.4
CVE-2026-7941	google - chrome	Insufficient validation of untrusted input in Mobile in Google Chrome on Android prior to 148.0.7778.96 allowed a local attacker to inject arbitrary scripts or HTML (UXSS) via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	4.4
CVE-2026-41004	vmware - multiple products	When enabling trace logging in Spring Cloud Config Server sensitive information was placed in plain text in the logs. Spring Cloud Config 3.1.x: affected from 3.1.0 through 3.1.13 (inclusive); upgrade to 3.1.14 or greater (Enterprise Support Only). Spring Cloud Config 4.1.x: affected from 4.1.0 through 4.1.9 (inclusive); upgrade to 4.1.10 or greater (Enterprise Support Only). Spring Cloud Config 4.2.x: affected from 4.2.0 through 4.2.6 (inclusive); upgrade to 4.2.7 or greater (Enterprise Support Only). Spring Cloud Config 4.3.x: affected from 4.3.0 through 4.3.2 (inclusive); upgrade to 4.3.3 or greater. Spring Cloud Config 5.0.x: affected from 5.0.0 through 5.0.2 (inclusive); upgrade to 5.0.3 or greater.	2026-05-07	4.4
CVE-2026-20172	cisco - Cisco Enterprise Chat and Email	A vulnerability in the Lite Agent feature of Cisco Enterprise Chat and Email (ECE) could allow an authenticated, remote attacker to conduct browser-based attacks. To exploit this vulnerability, the attacker must have valid credentials for a user account with at least the role of Agent. This vulnerability is due to inadequate validation of file contents during file upload operations. An attacker could exploit this vulnerability by uploading a file that contains malicious scripts or HTML code, which the application could make available to other users to access. A successful exploit could allow the attacker to execute the contents of that file in the browser of a user and conduct browser-based attacks.	2026-05-06	4.3
CVE-2026-20189	cisco - Cisco Prime Infrastructure	A vulnerability in the log file download functionality of Cisco Prime Infrastructure could allow anauthenticated, remote attacker to download arbitrary log files from the server. This vulnerability is due to insufficient authorization checks on the download service API. An attacker could exploit this vulnerability by submitting a crafted URL request to an affected device. A successful exploit could allow the attacker to download sensitive log files that they would otherwise not have authorization to access. To exploit this vulnerability, the attacker must have valid credentials to access the web-based management interface of the affected device.	2026-05-06	4.3
CVE-2026-20193	cisco - Cisco Identity Services Engine Software	A vulnerability in the RADIUS Policy API endpoints of Cisco ISE could allow anauthenticated, remote attacker with read-only Administrator privileges to gain unauthorized access to sensitive information on an affected device. This vulnerability is due to improper role-based access control (RBAC) permissions on the RADIUS Policy API endpoints. An attacker could exploit this vulnerability by bypassing the web-based management interface and directly calling an affected endpoint. A successful exploit could allow the attacker to gain unauthorizedread access to sensitive RADIUS Policy details that are restricted for their role.	2026-05-06	4.3
CVE-2026-7904	google - chrome	Out of bounds read in Fonts in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform an out of bounds memory read via a crafted HTML page. (Chromium security severity: High)	2026-05-06	4.3

CVE-2026-7915	google - chrome	Insufficient data validation in DevTools in Google Chrome on Android prior to 148.0.7778.96 allowed a remote attacker to bypass navigation restrictions via a crafted HTML page. (Chromium security severity: High)	2026-05-06	4.3
CVE-2026-7933	google - chrome	Out of bounds read in WebCodecs in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform an out of bounds memory read via a crafted video file. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7936	google - chrome	Object lifecycle issue in V8 in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to perform an out of bounds memory read via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7942	google - chrome	Integer overflow in ANGLE in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7946	google - chrome	Insufficient policy enforcement in WebUI in Google Chrome on Linux, Mac, Windows, ChromeOS prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7961	google - chrome	Insufficient validation of untrusted input in Permissions in Google Chrome prior to 148.0.7778.96 allowed an attacker on the local network segment to leak cross-origin data via malicious network traffic. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7969	google - chrome	Integer overflow in Network in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass same origin policy via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7972	google - chrome	Uninitialized Use in GPU in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7979	google - chrome	Inappropriate implementation in Media in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7983	google - chrome	Out of bounds read in Dawn in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7986	google - chrome	Insufficient policy enforcement in Autofill in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.3
CVE-2026-7999	google - chrome	Inappropriate implementation in V8 in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.3
CVE-2026-8004	google - chrome	Insufficient policy enforcement in DevTools in Google Chrome prior to 148.0.7778.96 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: Low)	2026-05-06	4.3
CVE-2026-8005	google - chrome	Insufficient validation of untrusted input in Cast in Google Chrome prior to 148.0.7778.96 allowed an attacker on the local network segment to bypass same origin policy via malicious network traffic. (Chromium security severity: Low)	2026-05-06	4.3
CVE-2026-8011	google - chrome	Insufficient policy enforcement in Search in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.3
CVE-2026-8013	google - chrome	Insufficient validation of untrusted input in FedCM in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.3
CVE-2026-8014	google - chrome	Inappropriate implementation in Preload in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.3
CVE-2026-7912	google - chrome	Integer overflow in GPU in Google Chrome on Android prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform arbitrary read/write via a crafted HTML page. (Chromium security severity: High)	2026-05-06	4.2
CVE-2026-7934	google - chrome	Insufficient validation of untrusted input in Popup Blocker in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass navigation restrictions via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7943	google - chrome	Insufficient validation of untrusted input in ANGLE in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform arbitrary read/write via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7947	google - chrome	Insufficient validation of untrusted input in Network in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform UI spoofing via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7952	google - chrome	Insufficient policy enforcement in Extensions in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass discretionary access control via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7964	google - chrome	Insufficient validation of untrusted input in FileSystem in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform arbitrary read/write via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7989	google - chrome	Insufficient data validation in DataTransfer in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform arbitrary read/write via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7993	google - chrome	Insufficient validation of untrusted input in Payments in Google Chrome on Android prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to spoof the contents of the Omnibox (URL bar) via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	4.2
CVE-2026-7996	google - chrome	Insufficient validation of untrusted input in SSL in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to perform UI spoofing via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.2
CVE-2026-8021	google - chrome	Script injection in UI in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who convinced a user to engage in specific UI gestures to inject arbitrary scripts or HTML (UXSS) via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	4.2
CVE-2026-32803	dell - multiple products	Dell PowerScale OneFS versions 9.5.0.0 through 9.5.1.6, 9.6.0.0 through 9.7.1.13, 9.8.0.0 through 9.10.1.5 and 9.11.0.0 through 9.12.0.1 contains an Insufficient Logging vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Information tampering.	2026-05-08	3.3

CVE-2026-7909	google - chrome	Inappropriate implementation in ServiceWorker in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	2026-05-06	3.1
CVE-2026-7937	google - chrome	Insufficient policy enforcement in DevTools in Google Chrome prior to 148.0.7778.96 allowed an attacker who convinced a user to install a malicious extension to bypass navigation restrictions via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7944	google - chrome	Insufficient validation of untrusted input in Persistent Cache in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7945	google - chrome	Insufficient validation of untrusted input in COOP in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7949	google - chrome	Out of bounds read in Skia in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to leak cross-origin data via a crafted Chrome Extension. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7954	google - chrome	Race in Shared Storage in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7959	google - chrome	Inappropriate implementation in Navigation in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7965	google - chrome	Insufficient validation of untrusted input in DevTools in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to leak cross-origin data via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7966	google - chrome	Insufficient validation of untrusted input in SiteIsolation in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-7968	google - chrome	Insufficient validation of untrusted input in CORS in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who had compromised the renderer process to bypass same origin policy via a crafted HTML page. (Chromium security severity: Medium)	2026-05-06	3.1
CVE-2026-8017	google - chrome	Side-channel information leakage in Media in Google Chrome prior to 148.0.7778.96 allowed a remote attacker to leak cross-origin data via a crafted HTML page. (Chromium security severity: Low)	2026-05-06	3.1
CVE-2026-8022	google - chrome	Inappropriate implementation in MHTML in Google Chrome prior to 148.0.7778.96 allowed a remote attacker who convinced a user to engage in specific UI gestures to leak cross-origin data via a crafted MHTML page. (Chromium security severity: Low)	2026-05-06	3.1
CVE-2026-20188	cisco - Cisco Crosswork Network Change Automation	Following the initial publication of the Security Advisory about a denial of service (DoS) condition in Cisco Crosswork Network Controller and Cisco Network Services Orchestrator (NSO), additional information has been made available to the Cisco Product Security Incident Response Team (PSIRT). Upon further analysis, the Cisco PSIRT has reclassified this issue as a customer-configurable, resource management issue rather than a security vulnerability.	2026-05-06	0

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.