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 21 يونيو إلى 27 يونيو. 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 21th of June to 27th of June. 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-2026-12537	google - multiple products	Improper Neutralization used in an OS Command in the container launcher in Google Gemini CLI (versions prior to 0.39.1) and run-gemini-cli GitHub Action (versions prior to 0.1.22) on headless CI platforms allows an unprivileged attacker to achieve pre-sandbox host-level code execution a maliciously crafted .gemini/.env file.	2026-06-24	10
CVE-2026-46752	apache software foundation - Apache Kvrocks	Redis Lua HEAP overflow in cjson library vulnerability in Apache Kvrocks. This issue affects Apache Kvrocks: from 2.0.4 through 2.15.0. Users are recommended to upgrade to version 2.16.0, which fixes the issue.	2026-06-25	10
CVE-2026-52914	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix fragment reassembly length accounting batman-adv keeps a running payload length for queued fragments and uses it to validate a fragment chain before reassembly. That accounting currently allows the accumulated fragment length to be truncated during updates. As a result, malformed fragment chains can bypass the intended validation and drive reassembly with inconsistent length state, leading to a local denial of service. Fix the accounting by storing the accumulated length in a length-typed field and rejecting update overflows before the existing validation logic runs. The fix was verified against the original reproducer and against valid fragment reassembly paths.	2026-06-24	9.8
CVE-2026-52924	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: purge outqueue on stale COOKIE-ECHO handling sctp_stream_update() is only invoked when the association is moved into COOKIE_WAIT during association setup/reconfiguration. In this path, the outbound stream scheduler state (stream->out_curr) is expected to be clean, since no user data should have been transmitted yet unless the state machine has already partially progressed. However, a corner case exists in sctp_sf_do_5_2_6_stale(): when a Stale Cookie ERROR is received, the association is rolled back from COOKIE_ECHOED to COOKIE_WAIT. In this scenario, user data may already have been queued and even bundled with the COOKIE-ECHO chunk. During the rollback, sctp_stream_update() frees the old stream table and installs a new one, but it does not invalidate stream->out_curr. As a result, out_curr may still point to a freed sctp_stream_out entry from the previous stream state.	2026-06-24	9.8

		Later, SCTP scheduler dequeue paths (FCFS, RR, PRIO, etc.) rely on stream->out_curr->ext, which can lead to use-after-free once the old stream state has been released via sctp_stream_free(). This results in crashes such as (reported by Yuqi): BUG: KASAN: slab-use-after-free in sctp_sched_fcfs_dequeue+0x13a/0x140 Read of size 8 at addr ff1100004d4d3208 by task mini_poc/9312 CPU: 1 UID: 1001 PID: 9312 Comm: mini_poc Not tainted 7.1.0-rc1-00305-gbd3a4795d574 #5 PREEMPT(full) sctp_sched_fcfs_dequeue+0x13a/0x140 sctp_outq_flush+0x1603/0x33e0 sctp_do_sm+0x31c9/0x5d30 sctp_assoc_bh_rcv+0x392/0x6f0 sctp_inq_push+0x1db/0x270 sctp_rcv+0x138d/0x3c10 Fix this by fully purging the association outqueue when handling the Stale Cookie case. This ensures all pending transmit and retransmit state is dropped, and any scheduler cached pointers are invalidated, making it safe to rebuild stream state during COOKIE_WAIT restart. Updating only stream->out_curr would be insufficient, since queued and retransmittable data would still reference the old stream state and trigger later use-after-free in dequeue paths.		
CVE-2026-52931	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: batman-adv: tp_meter: avoid use of uninit sender vars batadv_tp_recv_ack() and batadv_tp_stop() are only valid for tp_vars in the BATADV_TP_SENDER role. When called with a BATADV_TP_RECEIVER role, it proceeds to read sender-only members that were never initialized, leading to undefined behavior. This can be triggered when a node that is currently acting as a receiver in an ongoing tp_meter session receives a malicious ACK packet. Guard against this by checking tp_vars->role immediately after the lookup and bailing out if it is not BATADV_TP_SENDER, before any of those members are accessed.	2026-06-24	9.8
CVE-2026-52955	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Fix potential out-of-bounds access in crush_decode() A message of type CEPH_MSG_OSD_MAP containing a crush map with at least one bucket has two fields holding the bucket algorithm. If the values in these two fields differ, an out-of-bounds access can occur. This is the case because the first algorithm field (alg) is used to allocate the correct amount of memory for a bucket of this type, while the second algorithm field inside the bucket (b->alg) is used in the subsequent processing. This patch fixes the issue by adding a check that compares alg and b->alg and aborts the processing in case they differ. Furthermore, b->alg is set to 0 in this case, because the destruction of the crush map also uses this field to determine the bucket type, which can again result in an out-of-bounds access when trying to free the memory pointed to by the fields of the bucket. To correctly free the memory allocated for the bucket in such a case, the corresponding call to kfree is moved from the algorithm-specific crush_destroy_bucket functions to the generic crush_destroy_bucket().	2026-06-24	9.8
CVE-2026-52982	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: usb: rtl8150: fix use-after-free in rtl8150_start_xmit() syzbot reported a KASAN slab-use-after-free read in rtl8150_start_xmit() when accessing skb->len for tx statistics after usb_submit_urb() has been called: BUG: KASAN: slab-use-after-free in rtl8150_start_xmit+0x71f/0x760 drivers/net/usb/rtl8150.c:712 Read of size 4 at addr ffff88810eb7a930 by task kworker/0:4/5226 The URB completion handler write_bulk_callback() frees the skb via dev_kfree_skb_irq(dev->tx_skb). The URB may complete on another CPU in softirq context before usb_submit_urb() returns in the submitter, so by the time the submitter reads skb->len the skb has already been queued to the per-CPU completion_queue and freed by net_tx_action(): CPU A (xmit) CPU B (USB completion softirq) ---------- ----------	2026-06-24	9.8

		dev->tx_skb = skb; usb_submit_urb() --+ |-------> write_bulk_callback() | dev_kfree_skb_irq(dev->tx_skb) | net_tx_action() | napi_skb_cache_put() <-- free netdev->stats.tx_bytes | += skb->len; <-- UAF read Fix it by caching skb->len before submitting the URB and using the cached value when updating the tx_bytes counter. The pre-existing tx_bytes semantics are preserved: the counter tracks the original frame length (skb->len), not the ETH_ZLEN/USB-alignment padded "count" value that is handed to the device. Changing that would be a user-visible accounting change and is out of scope for this UAF fix.		
CVE-2026-52986	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_conntrack_sip: don't use simple_strtoul Replace unsafe port parsing in epaddr_len(), ct_sip_parse_header_uri(), and ct_sip_parse_request() with a new sip_parse_port() helper that validates each digit against the buffer limit, eliminating the use of simple_strtoul() which assumes NUL-terminated strings. The previous code dereferenced pointers without bounds checks after sip_parse_addr() and relied on simple_strtoul() on non-NUL-terminated skb data. A port that reaches the buffer limit without a trailing character is also rejected as malformed. Also get rid of all simple_strtoul() usage in conntrack, prefer a stricter version instead. There are intentional changes: - Bail out if number is > UINT_MAX and indicate a failure, same for too long sequences. While we do accept 05535 as port 5535, we will not accept e.g. 'sip:10.0.0.1:005060'. While its syntactically valid under RFC 3261, we should restrict this to not waste cycles when presented with malformed packets with 64k '0' characters. - Force base 10 in ct_sip_parse_numerical_param(). This is used to fetch 'expire=' and 'rports='; both are expected to use base-10. - In nf_nat_sip.c, only accept the parsed value if its within the 1k-64k range. - epaddr_len now returns 0 if the port is invalid, as it already does for invalid ip addresses. This is intentional. nf_conntrack_sip performs lots of guesswork to find the right parts of the message to parse. Being stricter could break existing setups. Connection tracking helpers are designed to allow traffic to pass, not to block it. Based on an earlier patch from Jenny Guanni Qu <qguanni@gmail.com>.	2026-06-24	9.8
CVE-2026-52989	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: propagate nvmet_tcp_build_pdu_iovec() errors to its callers Currently, when nvmet_tcp_build_pdu_iovec() detects an out-of-bounds PDU length or offset, it triggers nvmet_tcp_fatal_error(cmd->queue) and returns early. However, because the function returns void, the callers are entirely unaware that a fatal error has occurred and that the cmd->recv_msg.msg_iter was left uninitialized. Callers such as nvmet_tcp_handle_h2c_data_pdu() proceed to blindly overwrite the queue state with queue->rcv_state = NVMET_TCP_RECV_DATA Consequently, the socket receiving loop may attempt to read incoming network data into the uninitialized iterator. Fix this by shifting the error handling responsibility to the callers.	2026-06-24	9.8
CVE-2026-52993	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tipc: fix double-free in tipc_buf_append() tipc_msg_validate() can potentially reallocate the skb it is validating, freeing the old one. In tipc_buf_append(), it was being called with a pointer to a local variable which was a copy of the caller's skb pointer. If the skb was reallocated and validation subsequently failed, the error	2026-06-24	9.8

		handling path would free the original skb pointer, which had already been freed, leading to double-free. Fix this by checking if head now points to a newly allocated reassembled skb. If it does, reassign *headbuf for later freeing operations.		
CVE-2026-53002	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: remove sprintf usage Replace it with scnprintf, the buffer sizes are expected to be large enough to hold the result, no need for snprintf+overflow check. Increase buffer size in mangle_content_len() while at it. BUG: KASAN: stack-out-of-bounds in vsnprintf+0xea5/0x1270 Write of size 1 at addr [..] vsnprintf+0xea5/0x1270 sprintf+0xb1/0xe0 mangle_content_len+0x1ac/0x280 nf_nat_sdp_session+0x1cc/0x240 process_sdp+0x8f8/0xb80 process_invite_request+0x108/0x2b0 process_sip_msg+0x5da/0xf50 sip_help_tcp+0x45e/0x780 nf_confirm+0x34d/0x990 [..]	2026-06-24	9.8
CVE-2026-53006	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix possible UAF in icmpv6_rcv() Caching saddr and daddr before pskb_pull() is problematic since skb->head can change. Remove these temporary variables: - We only access &ipv6_hdr(skb)->saddr and &ipv6_hdr(skb)->daddr when net_dbg_ratelimited() is called in the slow path. - Avoid potential future misuse after pskb_pull() call.	2026-06-24	9.8
CVE-2026-53010	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb2_open during durable reconnect In smb2_open, the call to ksmbd_put_durable_fd(fp) drops the reference to the durable file descriptor early during the durable reconnect process. If an error occurs subsequently (eg, ksmbd_iov_pin_rsp fails) or a scavenger accesses the file, it leads to a use-after-free when accessing fp properties (eg fp->create_time). Move the single put to the end of the function below err_out2 so fp stays valid until smb2_open returns.	2026-06-24	9.8
CVE-2026-53045	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: memory: tegra124-emc: Fix dll_change check The code checking whether the specified memory timing enables DLL in the EMRS register was reversed. DLL is enabled if bit A0 is low. Fix the check.	2026-06-24	9.8
CVE-2026-53046	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free from async crypto on Qualcomm crypto engine ksmbd_crypt_message() sets a NULL completion callback on AEAD requests and does not handle the -EINPROGRESS return code from async hardware crypto engines like the Qualcomm Crypto Engine (QCE). When QCE returns -EINPROGRESS, ksmbd treats it as an error and immediately frees the request while the hardware DMA operation is still in flight. The DMA completion callback then dereferences freed memory, causing a NULL pointer crash: pc : qce_skcipher_done+0x24/0x174 lr : vchan_complete+0x230/0x27c ... el1h_64_irq+0x68/0x6c ksmbd_free_work_struct+0x20/0x118 [ksmbd] ksmbd_exit_file_cache+0x694/0xa4c [ksmbd] Use the standard crypto_wait_req() pattern with crypto_req_done() as the completion callback, matching the approach used by the SMB client in fs/smb/client/smb2ops.c. This properly handles both synchronous	2026-06-24	9.8

		engines (immediate return) and async engines (-EINPROGRESS followed by callback notification).		
CVE-2026-53049	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gfs2: add some missing log locking Function gfs2_logd() calls the log flushing functions gfs2_ail1_start(), gfs2_ail1_wait(), and gfs2_ail1_empty() without holding sdp->sd_log_flush_lock, but these functions require exclusion against concurrent transactions. To fix that, add a non-locking __gfs2_log_flush() function. Then, in gfs2_logd(), take sdp->sd_log_flush_lock before calling the above mentioned log flushing functions and __gfs2_log_flush().	2026-06-24	9.8
CVE-2026-53055	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: hisilicon/sec2 - prevent req used-after-free for sec During packet transmission, if the system is under heavy load, the hardware might complete processing the packet and free the request memory (req) before the transmission function finishes. If the software subsequently accesses this req, a use-after-free error will occur. The qp_ctx memory exists throughout the packet sending process, so replace the req with the qp_ctx.	2026-06-24	9.8
CVE-2026-53086	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: bcmgenet: fix racing timeout handler The bcmgenet_timeout handler tries to take down all tx queues when a single queue times out. This is over zealous and causes many race conditions with queues that are still chugging along. Instead lets only restart the timed out queue.	2026-06-24	9.8
CVE-2026-53088	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: bcmgenet: fix off-by-one in bcmgenet_put_txcb The write_ptr points to the next open tx_cb. We want to return the tx_cb that gets rewinded, so we must rewind the pointer first then return the tx_cb that it points to. That way the txcb can be correctly cleaned up.	2026-06-24	9.8
CVE-2026-39893	cacti - cacti	Cacti is an open source performance and fault management framework. In versions 1.2.30 and prior, the rfilter request variable was concatenated into a RLIKE SQL clause without sanitization. The endpoint does not require authentication (graph viewing supports guest access via the configured guest user), so the SQLi was reachable pre-auth on installs with guest viewing enabled. This issue was fixed in version 1.2.31.	2026-06-24	9.8
CVE-2026-39938	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have unauthenticated LFI through graph_theme and rrdtool IPC serialization hardening. This issue has been resolved in version 1.2.31.	2026-06-24	9.8
CVE-2026-39955	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have pre-authentication SQL Injection via unanchored FILTER_VALIDATE_REGEXP in graph_view.php. This issue has been fixed in version 1.2.31.	2026-06-24	9.8
CVE-2026-53151	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix the ACK parser to extract the SACK table for parsing Fix modification of the received skbuff in rxrpc_input_soft_acks() and a potential incorrect access of the buffer in a fragmented UDP packet (the packet would probably have to be deliberately pre-generated as fragmented) when AF_RXRPC tries to extract the contents of the SACK table by copying out the contents of the SACK table into a buffer before attempting to parse AF_RXRPC assumes that it can just call skb_condense() and then validly access the SACK table from skb->data and that it will be a flat buffer - but skb_condense() can silently fail to do anything under some circumstances. Note that whilst rxrpc_input_soft_acks() should be able to parse extended ACKs, the rest of AF_RXRPC doesn't currently support that. Further, there's then no need to call skb_condense() in rxrpc_input_ack(), so don't.	2026-06-25	9.8
CVE-2026-53175	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: inet: frags: fix use-after-free caused by the fqdir_pre_exit() flush On netns teardown, fqdir_pre_exit() walks the fqdir rhashtable and flushes every fragment queue that is not yet complete using inet_frag_queue_flush(). That helper frees all the skbs queued on the fragment queue but does not set INET_FRAG_COMPLETE, and leaves q->fragments_tail and q->last_run_head pointing at the freed skbs. The queue itself stays in the rhashtable.	2026-06-25	9.8

		fqdir_pre_exit() first lowers high_thresh to 0 to stop new queue lookups, but it cannot stop a fragment that already obtained the queue through inet_frag_find() earlier and stalled just before taking the queue lock. Once that fragment resumes after the flush and takes the queue lock, it passes the INET_FRAG_COMPLETE check and then dereferences the freed fragments_tail. inet_frag_queue_insert() reads FRAG_CB() and ->len of that pointer and, on the append path, writes ->next_frag, causing a slab use-after-free. IPv6, nf_conntrack_reasm6 and 6lowpan reassembly share the same flush path and are affected as well. Reset rb_fragments, fragments_tail and last_run_head in inet_frag_queue_flush() so a flushed queue no longer points at the freed skbs. A fragment that resumes after the flush and takes the queue lock then finds an empty queue and starts a new run instead of dereferencing the freed fragments_tail. ip_frag_reinit() already performed this reset after its own flush, so drop the now duplicate code there.		
CVE-2026-53176	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: IB/isert: Reject login PDUs shorter than ISER_HEADERS_LEN In drivers/infiniband/ulp/isert/ib_isert.c, isert_login_recv_done() computes the login request payload length as wc->byte_len minus ISER_HEADERS_LEN with no lower bound, and login_req_len is a signed int. A remote iSER initiator can post a login Send work request carrying fewer than ISER_HEADERS_LEN (76) bytes, so the subtraction underflows and login_req_len becomes negative. isert_rx_login_req() then reads that negative length back into a signed int, takes size = min(rx_buflen, MAX_KEY_VALUE_PAIRS), and because the min() is signed it keeps the negative value; the value is then passed as the memcpy() length and sign-extended to a multi-gigabyte size_t. The copy into the 8192-byte login->req_buf runs far out of bounds and faults, crashing the target node. The login phase precedes iSCSI authentication, so no credentials are required to reach this path. Reject any login PDU shorter than ISER_HEADERS_LEN before the subtraction, mirroring the existing early return on a failed work completion, so login_req_len can never go negative. The upper bound was already safe: a posted login buffer cannot deliver more than ISER_RX_PAYLOAD_SIZE, so the difference stays at or below MAX_KEY_VALUE_PAIRS and the existing min() clamps it; only the missing lower bound needs to be added.	2026-06-25	9.8
CVE-2026-53215	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mvpp2: refill RX buffers before XDP or skb use The RX error path returns the current descriptor buffer to the hardware BM pool. That is only valid while the driver still owns the buffer. mvpp2_rx_refill() can fail after the current buffer has been handed to XDP or attached to an skb. In those cases mvpp2_run_xdp() may have recycled, redirected, or queued the page for XDP_TX, and an skb free also retires the data buffer. Returning such a buffer to BM lets hardware DMA into memory that is no longer owned by the RX ring. Refill the BM pool before handing the current buffer to XDP or to the skb. If the allocation fails there, drop the packet and return the still-owned current buffer to BM, preserving the pool depth. Once the refill succeeds, later local drops retire/free the current buffer instead of returning it to BM.	2026-06-25	9.8
CVE-2026-53216	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mvpp2: limit XDP frame size to the RX buffer mvpp2 has short and long BM pools, and short pool buffers can be smaller than PAGE_SIZE. The XDP path nevertheless initializes every xdp_buff with PAGE_SIZE as frame size. XDP helpers use frame_sz to validate tail growth and to derive the hard end of the data area. Advertising PAGE_SIZE for short buffers can let bpf_xdp_adjust_tail() grow a packet past the real allocation, corrupting memory or later tripping skb tailroom checks. Initialize the XDP buffer with bm_pool->frag_size so XDP tailroom matches the actual buffer backing the packet.	2026-06-25	9.8
CVE-2026-53221	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ip6_vti: fix incorrect tunnel matching in vti6_tnl_lookup() In vti6_tnl_lookup(), when an exact match for a tunnel fails,	2026-06-25	9.8

		the code falls back to searching for wildcard tunnels: - Tunnels matching the packet's local address, with any remote address wildcard remote). - Tunnels matching the packet's remote address, with any local address (wildcard local). However, vti6 stores all these different types of tunnels in the same hash table (ip6n->tnls_r_l) prone to hash collisions. The bug is that the fallback search loops in vti6_tnl_lookup() were missing checks to ensure that the candidate tunnel actually has a wildcard address.		
CVE-2026-53228	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: sit: reload inner IPv6 header after GSO offloads ipip6_tunnel_xmit() caches the inner IPv6 header pointer at function entry and continues using it after iptunnel_handle_offloads(). For GSO skbs, iptunnel_handle_offloads() calls skb_header_unclone(). When the skb header is cloned, skb_header_unclone() can call pskb_expand_head(), which may move the skb head. The pskb_expand_head() contract requires pointers into the skb header to be reloaded after the call. If the later skb_realloc_headroom() branch is not taken, SIT uses the stale iph6 pointer to read the inner hop limit and DS field. That can read from a freed skb head after the old head's remaining clone is released. Reload iph6 after the offload helper succeeds and before subsequent reads from the inner IPv6 header. Keep the existing reload after skb_realloc_headroom(), since that branch can also replace the skb.	2026-06-25	9.8
CVE-2026-53246	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: validate cached peer INIT chunk length in COOKIE_ECHO processing When a listening SCTP server processes a COOKIE_ECHO chunk, the cached peer INIT chunk embedded after the cookie is parsed and its parameters are later walked by sctp_process_init() using sctp_walk_params(). However, the chunk header length of this cached INIT chunk was not validated against the remaining buffer in the COOKIE_ECHO payload. If the length field is inflated, the parameter walk can run beyond the actual received data, leading to out-of-bounds reads and potential memory corruption during later parameter handling (e.g. STATE_COOKIE processing and kmemdup() copies). Add a bounds check in sctp_unpack_cookie() to ensure the cached INIT chunk length does not exceed the available data in the COOKIE_ECHO buffer before it is used.	2026-06-25	9.8
CVE-2026-53247	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_eth_soc: Fix use-after-free in metadata dst teardown mtk_free_dev() calls metadata_dst_free() which frees the metadata_dst with kfree() immediately, bypassing the RCU grace period. In the RX path, skb_dst_set_noref() sets a non-refcounted pointer from the skb to the metadata_dst. This function requires RCU read-side protection and the dst must remain valid until all RCU readers complete. Since metadata_dst_free() calls kfree() directly, a use-after-free can occur if any skb still holds a noref pointer to the dst when the driver tears it down. Replace metadata_dst_free() with dst_release() which properly goes through the refcount path: when the refcount drops to zero, it schedules the actual free via call_rcu_hurry(), ensuring all RCU readers have completed before the memory is freed.	2026-06-25	9.8
CVE-2026-53260	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tcp: Add preempt_{disable,enable}_nested() in reqsk_queue_hash_req(). syzbot reported a weird reqsk->rsk_refcnt underflow in __inet_csk_reqsk_queue_drop(). The captured reqsk_put() in __inet_csk_reqsk_queue_drop() is called only when it successfully removes reqsk from ehash. Moreover, reqsk_timer_handler() calls another reqsk_put() after that.	2026-06-25	9.8

		This indicates that the reqsk was missing both refcnts for ehash and the timer itself. Since all the syzbot reports had PREEMPT_RT enabled, the only possible scenario is that reqsk_queue_hash_req() is preempted after mod_timer() and before refcount_set(), and then the timer triggered after 1s aborts the reqsk due to its listener's close(). Let's wrap mod_timer() and refcount_set() with preempt_disable_nested() and preempt_enable_nested(). Note that inet_ehash_insert() holds the normal spin_lock() (mutex in PREEMPT_RT), so it must be called outside of preempt_disable_nested(), but this is fine. The lookup path just ignores 0 sk_refcnt entries in ehash and tries to create another reqsk, but this will fail at inet_ehash_insert(). [0]: refcount_t: underflow; use-after-free. WARNING: lib/refcount.c:28 at refcount_warn_saturate+0xb2/0x110 lib/refcount.c:28, CPU#0: ktimers/0/16 Modules linked in: CPU: 0 UID: 0 PID: 16 Comm: ktimers/0 Tainted: G L syzkaller #0 PREEMPT_{RT,(full)} Tainted: [L]=SOFTLOCKUP Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/18/2026 RIP: 0010:refcount_warn_saturate+0xb2/0x110 lib/refcount.c:28 Code: e4 7d d1 0a 67 48 0f b9 3a eb 4a e8 38 3d 23 fd 48 8d 3d e1 7d d1 0a 67 48 0f b9 3a eb 37 e8 25 3d 23 fd 48 8d 3d de 7d d1 0a <67> 48 0f b9 3a eb 24 e8 12 3d 23 fd 48 8d 3d db 7d d1 0a 67 48 0f RSP: 0000:ffffc90000157948 EFLAGS: 00010246 RAX: ffffffff84a1301b RBX: 0000000000000003 RCX: ffff88801ca98000 RDX: 0000000000000100 RSI: 0000000000000000 RDI: ffffffff8f72ae00 RBP: ffffffff99ae3b01 R08: ffff88801ca98000 R09: 0000000000000005 R10: 0000000000000100 R11: 0000000000000004 R12: ffff8880425ef568 R13: ffff8880425ef4f8 R14: ffff8880425ef578 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff88812638600 0(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f7b46710e9c CR3: 000000000dbb6000 CR4: 00000000003526f0 Call Trace: <TASK> __refcount_sub_and_test include/linux/refcount.h:400 [inline] __refcount_dec_and_test include/linux/refcount.h:432 [inline] refcount_dec_and_test include/linux/refcount.h:450 [inline] reqsk_put include/net/request_sock.h:136 [inline] __inet_csk_reqsk_queue_drop+0x3ce/0x440 net/ipv4/inet_connection_sock.c:1007 reqsk_timer_handler+0x651/0xdf0 net/ipv4/inet_connection_sock.c:1137 call_timer_fn+0x192/0x5e0 kernel/time/timer.c:1748 expire_timers kernel/time/timer.c:1799 [inline] __run_timers kernel/time/timer.c:2374 [inline] __run_timer_base+0x6a3/0x9f0 kernel/time/timer.c:2386 run_timer_base kernel/time/timer.c:2395 [inline] run_timer_softirq+0x67/0x170 kernel/time/timer.c:2403 handle_softirqs+0x1de/0x6d0 kernel/softirq.c:622 __do_softirq kernel/softirq.c:656 [inline] run_ktimerd+0x69/0x100 kernel/softirq.c:1151 smpboot_thread_fn+0x541/0xa50 kernel/smpboot.c:160 kthread+0x388/0x470 kernel/kthread.c:436 ret_from_fork+0x514/0xb70 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK>		
CVE-2026-41120	dell - multiple products	Dell Wyse Management Suite, versions prior to WMS 5.5 HF1, contain an Acceptance of Extraneous Untrusted Data With Trusted Data vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Remote Code Execution.	2026-06-25	9.8
CVE-2026-53309	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2/dlm: fix off-by-one in dlm_match_regions() region comparison The local-vs-remote region comparison loop uses '<=' instead of '<', causing it to read one entry past the valid range of qr_regions. The other loops in the same function correctly use '<'. Fix the loop condition to use '<' for consistency and correctness.	2026-06-26	9.8
CVE-2026-28381	grafana - snowflake	The Snowflake datasource allows for GET/PUT commands, which can allow any user with access to run queries against the data source to read/write files between the local grafana server and the connected Snowflake host.	2026-06-22	9.6
CVE-2026-11807	red hat - multiple products	A missing authorization vulnerability was found in the Event-Driven Ansible (EDA) websocket API. The /api/eda/ws/ansible-rulebook endpoint does not verify user permissions when	2026-06-23	9.6

		processing Worker messages. Any authenticated user can send a forged message with an arbitrary activation_id to receive plaintext credentials associated with that activation, including OAuth tokens, vault passwords, and SSH keys.		
CVE-2026-13028	google - chrome	Use after free in WebGL in Google Chrome on Android prior to 149.0.7827.197 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Critical)	2026-06-24	9.6
CVE-2026-13032	google - chrome	Use after free in WebGL in Google Chrome on Android prior to 149.0.7827.197 allowed a remote attacker to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: Critical)	2026-06-24	9.6
CVE-2026-41566	apache software foundation - Apache Kvrocks	Improper Handling of Insufficient Permissions or Privileges vulnerability in Apache Kvrocks. This issue affects Apache Kvrocks: 2.8.0. Users are recommended to upgrade to version 2.16.0, which fixes the issue.	2026-06-25	9.4
CVE-2026-53131	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: require Ethernet MAC header before using eth_hdr() `ip6t_eui64`, `xt_mac`, the `bitmap:ip,mac`, `hash:ip,mac`, and `hash:mac` ipset types, and `nf_log_syslog` access `eth_hdr(skb)` after either assuming that the skb is associated with an Ethernet device or checking only that the `ETH_HLEN` bytes at `skb_mac_header(skb)` lie between `skb->head` and `skb->data`. Make these paths first verify that the skb is associated with an Ethernet device, that the MAC header was set, and that it spans at least a full Ethernet header before accessing `eth_hdr(skb)`.	2026-06-25	9.4
CVE-2026-39948	cacti - cacti	Cacti is an open source performance and fault management framework. In versions 1.2.30 and prior, the rfilter request parameter is retrieved via the raw accessor grv() (rather than gfrv() with FILTER_VALIDATE_IS_REGEX validation) and concatenated directly into RLIKE SQL clauses in lib/html_graph.php and lib/html_tree.php, which are reachable pre-authentication through graph_view.php on installations with guest graph viewing enabled. Because the unbalanced-quote payload bypasses the regex validation that would otherwise reject it, an unauthenticated attacker can inject arbitrary SQL to compromise the confidentiality, integrity, and availability of the database. This advisory is similar to GHSA-69gg-mjfm-jjpc. This issue has been fixed in version 1.2.31.	2026-06-24	9.3
CVE-2026-12628	ibm - storage_protect	IBM Storage Protect Client 8.1.0.0 through 8.2.1.0 and IBM Storage Protect Snapshot For Windows 8.1.0.0 through 8.2.1.0 could allow a remote attacker to bypass authentication due to the use of a hardcoded credential in the FlashCopy Manager (FCM) authentication mechanism. The application contains a static credential embedded in multiple authentication code paths, and does not properly validate authentication responses, which may allow an unauthenticated attacker to establish a trusted session and access protected services. This vulnerability affects client components across multiple versions and may allow an attacker to impersonate legitimate clients, potentially leading to unauthorized access to system resources.	2026-06-22	9.1
CVE-2026-52958	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Fix potential out-of-bounds access in osdmap_decode() When decoding osd_state and osd_weight from an incoming osdmap in osdmap_decode(), both are decoded for each osd, i.e., map->max_osd times. The ceph_decode_need() check only accounts for sizeof(*map->osd_weight) once. This can potentially result in an out-of-bounds memory access if the incoming message is corrupted such that the max_osd value exceeds the actual content of the osdmap message. This patch fixes the issue by changing the corresponding part in the ceph_decode_need() check to account for map->max_osd*sizeof(*map->osd_weight).	2026-06-24	9.1
CVE-2026-52999	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nfnetlink_osf: fix out-of-bounds read on option matching In nf_osf_match(), the nf_osf_hdr_ctx structure is initialized once and passed by reference to nf_osf_match_one() for each fingerprint checked. During TCP option parsing, nf_osf_match_one() advances the shared ctx->optp pointer. If a fingerprint perfectly matches, the function returns early without restoring ctx->optp to its initial state. If the user has configured NF_OSF_LOGLEVEL_ALL, the loop continues to the next fingerprint. However, because ctx->optp was not restored, the next call to nf_osf_match_one() starts parsing from the end of the options buffer. This causes subsequent matches to read garbage data and fail immediately, making it impossible to log more than one match or logging incorrect matches. Instead of using a shared ctx->optp pointer, pass the context as a constant pointer and use a local pointer (optp) for TCP option traversal. This makes nf_osf_match_one() strictly stateless from the caller's perspective, ensuring every fingerprint check starts at the correct option offset.	2026-06-24	9.1

CVE-2026-53043	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2/dlm: validate qr_numregions in dlm_match_regions() Patch series "ocfs2/dlm: fix two bugs in dlm_match_regions()". In dlm_match_regions(), the qr_numregions field from a DLM_QUERY_REGION network message is used to drive loops over the qr_regions buffer without sufficient validation. This series fixes two issues: - Patch 1 adds a bounds check to reject messages where qr_numregions exceeds O2NM_MAX_REGIONS. The o2net layer only validates message byte length; it does not constrain field values, so a crafted message can set qr_numregions up to 255 and trigger out-of-bounds reads past the 1024-byte qr_regions buffer. - Patch 2 fixes an off-by-one in the local-vs-remote comparison loop, which uses '<=' instead of '<', reading one entry past the valid range even when qr_numregions is within bounds. This patch (of 2): The qr_numregions field from a DLM_QUERY_REGION network message is used directly as loop bounds in dlm_match_regions() without checking against O2NM_MAX_REGIONS. Since qr_regions is sized for at most O2NM_MAX_REGIONS (32) entries, a crafted message with qr_numregions > 32 causes out-of-bounds reads past the qr_regions buffer. Add a bounds check for qr_numregions before entering the loops.	2026-06-24	9.1
CVE-2026-53186	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/srp: bound SRP_RSP sense copy by the received length srp_process_rsp() copies sense data from rsp->data + resp_data_len, where resp_data_len is the full 32-bit value supplied by the SRP target and is never checked against the number of bytes actually received (wc->byte_len). The copy length is bounded to SCSI_SENSE_BUFFERSIZE, so at most 96 bytes are copied, but the source offset is not bounded. A malicious or compromised SRP target on the InfiniBand/RoCE fabric that the initiator has logged into can return an SRP_RSP with SRP_RSP_FLAG_SNSVALID set and a large resp_data_len. The receive buffer is allocated at the target-chosen max_ti_iu_len, so the source of the sense copy lands past the bytes actually received; with resp_data_len near 0xFFFFFFFF it is gigabytes past the buffer and the read faults. Copy the sense data only if it has not been truncated, that is, only if the response header, the response data, and the sense region fit within the bytes actually received; otherwise drop the sense and log. The in-tree iSER and NVMe-RDMA receive paths already bound their parse by wc->byte_len; this brings ib_srp into line with them.	2026-06-25	9.1
CVE-2026-53224	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: validate embedded INIT chunk and address list lengths in cookie sctp_unpack_cookie() only checked that the embedded INIT chunk length did not exceed the remaining cookie payload, but did not ensure that the INIT chunk is large enough to contain a complete INIT header. A malformed COOKIE_ECHO can therefore carry a truncated INIT chunk whose length field is smaller than sizeof(struct sctp_init_chunk). Later, sctp_process_init() accesses INIT parameters unconditionally, which may lead to out-of-bounds reads. In addition, raw_addr_list_len is not fully validated against the remaining cookie payload. When cookie authentication is disabled, an attacker can supply an oversized raw_addr_list_len and cause sctp_raw_to_bind_addrs() to read beyond the end of the cookie. The address parser also lacks sufficient bounds checks for parameter headers and lengths, allowing malformed address parameters to trigger out-of-bounds reads. Fix this by: - requiring the embedded INIT chunk length to be at least sizeof(struct sctp_init_chunk); - validating that the INIT chunk and raw address list together fit within the cookie payload; - verifying sufficient data exists for each address parameter header and payload before parsing it.	2026-06-25	9.1

		Note that sctp_verify_init() must be called after sctp_unpack_cookie() and before sctp_process_init() when cookie authentication is disabled. This will be addressed in a separate patch.		
CVE-2026-53225	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: fix uninit-value in __sctp_rcv_asconf_lookup() __sctp_rcv_asconf_lookup() in net/sctp/input.c only checks that the ASCONF chunk can hold the ADDIP header and a parameter header, then calls af->from_addr_param(), which reads the full address (16 bytes for IPv6) trusting the parameter's declared length. An unauthenticated peer can send a truncated trailing ASCONF chunk that declares an IPv6 address parameter but stops after the 4-byte parameter header; reached from the no-association lookup path, from_addr_param() then reads uninitialized bytes past the parameter. Impact: an unauthenticated SCTP peer makes the receive path read up to 16 bytes of uninitialized memory past a truncated ASCONF address parameter. The sibling __sctp_rcv_init_lookup() bounds parameters with sctp_walk_params(); this path open-codes the fetch and omits the bound. Verify the whole address parameter lies within the chunk before from_addr_param() reads it, the same class of fix as commit 51e5ad549c43 ("net: sctp: fix KMSAN uninit-value in sctp_inq_pop").	2026-06-25	9.1
CVE-2025-55017	apache software foundation - Apache IoTDB	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') vulnerability in Apache IoTDB. This issue affects Apache IoTDB: from 2.0.0 before 2.0.6, from 1.0.0 before 1.3.6. Users are recommended to upgrade to version 1.3.6 and 2.0.6, which fixes the issue.	2026-06-26	9.1
CVE-2025-64152	apache software foundation - Apache IoTDB	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') vulnerability in Apache IoTDB. This issue affects Apache IoTDB: from 1.0.0 before 1.3.6, from 2.0.0 before 2.0.7. Users are recommended to upgrade to version 1.3.6 and 2.0.7, which fixes the issue.	2026-06-26	9.1
CVE-2026-11374	zohocorp - multiple products	In ManageEngine ADSelfService Plus, RecoveryManager Plus, M365 Manager Plus, and ADAudit Plus, the SSO tickets generated to authenticate that session could be predicted by an unauthenticated user, leading to account takeover.	2026-06-23	9
CVE-2026-12681	google - go-attestation	Improper Validation of Specified Index, Position, or Offset in Input vulnerability in Google go-attestation. parseEfiSignatureList() does not advance the buffer past vendor bytes before reading entries. For hashSHA256SigGUID lists, this allows attacker-controlled vendor header bytes to be appended to the trusted SHA256 hash list. A crafted TPM event log could inject arbitrary SHA256 hashes into the verifier's trusted measurement database, enabling a remote attestation verifier to accept a compromised boot state. This issue affects go-attestation: through 0.6.0.	2026-06-24	8.9
CVE-2026-52911	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: scope conn->binding slowpath to bound sessions only When the binding SESSION_SETUP sets conn->binding = true, the flag stays set after the call so that the global session lookup in ksmbd_session_lookup_all() can find the session, which was not added to conn->sessions. Because the flag is connection-wide, the global lookup path will also resolve any other session by id if asked. Tighten the global lookup so that the returned session must have this connection registered in its channel xarray (sess->ksmbd_chann_list). The channel entry is installed by the existing binding_session path in ntlm_authenticate()/krb5_authenticate() when a SESSION_SETUP completes successfully, so this condition is a strict equivalent of "this connection has been accepted as a channel of this session". Connections that have not bound to a given session cannot reach it via the global table. The existing conn->binding gate for entering the slowpath is preserved so that non-binding connections keep the fast-path-only behavior, and the session->state check is unchanged.	2026-06-21	8.8
CVE-2026-54099	red hat - multiple products	A flaw was found in the Windows Machine Config Operator (WMCO) for Red Hat OpenShift Container Platform. The WICD CSR auto-approver validates that a Certificate Signing Request contains the organization system:wicd-nodes but does not reject additional organization values such as system:masters. A compromised Windows worker node that holds WICD credentials can submit a CSR that is auto-approved and signed by the cluster, yielding a client certificate that grants cluster-administrator privileges and enabling full cluster takeover.	2026-06-22	8.8
CVE-2026-44272	dell - wyse_management_suite	Dell Wyse Management Suite (WMS), versions prior to WMS 2605, contain an Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Unauthorized access.	2026-06-22	8.8

CVE-2026-52918	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: serialize accept_q access bt_sock_poll() walks the accept queue without synchronization, while child teardown can unlink the same socket and drop its last reference. The unsynchronized accept queue walk has existed since the initial Bluetooth import. Protect accept_q with a dedicated lock for queue updates and polling. Also rework bt_accept_dequeue() to take temporary child references under the queue lock before dropping it and locking the child socket.	2026-06-24	8.8
CVE-2026-52934	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: batman-adv: tvlv: reject oversized TVLV packets batadv_tvlv_container_ogm_append() builds a TVLV packet section from the tvlv.container_list. The total size of this section is computed by batadv_tvlv_container_list_size(), which sums the sizes of all registered containers. The return type and accumulator in batadv_tvlv_container_list_size() were u16. If the accumulated size exceeds U16_MAX, the value wraps around, causing the subsequent allocation in batadv_tvlv_container_ogm_append() to be undersized. The memcpy-style copy that follows would then write beyond the end of the allocated buffer, corrupting kernel memory. Fix this by widening the return type of batadv_tvlv_container_list_size() to size_t. In batadv_tvlv_container_ogm_append(), check the computed length against U16_MAX before proceeding, and bail out as if the allocation had failed when the limit is exceeded.	2026-06-24	8.8
CVE-2026-57280	jenkins - script_security	Jenkins Script Security Plugin 1402.v94c9ce464861 and earlier does not intercept the implicit type casts applied to the elements of typed for-each loops in sandboxed Groovy scripts, allowing attackers able to provide such scripts to invoke arbitrary constructors and bypass the sandbox protection.	2026-06-24	8.8
CVE-2026-57301	jenkins - official_owasp_zap	Jenkins OWASP ZAP Plugin 1.0.7 and earlier performs build operations on the Jenkins controller rather than the assigned agent, allowing attackers with Item/Configure permission to execute arbitrary code on the Jenkins controller.	2026-06-24	8.8
CVE-2026-52952	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix WARN_ON in __iommu_group_set_domain_nofail() due to reset In __iommu_group_set_domain_internal(), concurrent domain attachments are rejected when any device in the group is recovering. This is necessary to fence concurrent attachments to a multi-device group where devices might share the same RID due to PCI DMA alias quirks, but triggers the WARN_ON in __iommu_group_set_domain_nofail(). Other IOMMU_SET_DOMAIN_MUST_SUCCEED callers in detach/teardown paths, such as __iommu_group_set_core_domain and __iommu_release_dma_ownership, should not be rejected, as the domain would be freed anyway in these nofail paths while group->domain is still pointing to it. So pci_dev_reset_iommu_done() could trigger a UAF when re-attaching group->domain. Honor the IOMMU_SET_DOMAIN_MUST_SUCCEED flag, allowing the callers through the group->recovery_cnt fence, so as to update the group->domain pointer. Instead add a gdev->blocked check in the device iteration loop, to prevent any concurrent per-device detachment.	2026-06-24	8.8
CVE-2026-53053	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix clone_alias() to use the original device's devid Currently clone_alias() assumes first argument (pdev) is always the original device pointer. This function is called by pci_for_each_dma_alias() which based on topology decides to send original or alias device details in first argument. This meant that the source devid used to look up and copy the DTE may be incorrect, leading to wrong or stale DTE entries being propagated to alias device. Fix this by passing the original pdev as the opaque data argument to both the direct clone_alias() call and pci_for_each_dma_alias(). Inside clone_alias(), retrieve the original device from data and compute devid from it.	2026-06-24	8.8
CVE-2026-53057	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/riscv: Add IOTINVAL after updating DDT/PDT entries Add riscv_iommu_iodir_iotinval() to perform required TLB and context cache	2026-06-24	8.8

		invalidations after updating DDT or PDT entries, as mandated by the RISC-V IOMMU specification (Section 6.3.1 and 6.3.2).		
CVE-2026-53071	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: l2cap: Add missing chan lock in l2cap_ecred_reconf_rsp l2cap_ecred_reconf_rsp() calls l2cap_chan_del() without holding l2cap_chan_lock(). Every other l2cap_chan_del() caller in the file acquires the lock first. A remote BLE device can send a crafted L2CAP ECRED reconfiguration response to corrupt the channel list while another thread is iterating it. Add l2cap_chan_hold() and l2cap_chan_lock() before l2cap_chan_del(), and l2cap_chan_unlock() and l2cap_chan_put() after, matching the pattern used in l2cap_ecred_conn_rsp() and l2cap_conn_del().	2026-06-24	8.8
CVE-2026-53072	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix locking in hci_conn_request_evt() with HCI_PROTO_DEFER When protocol sets HCI_PROTO_DEFER, hci_conn_request_evt() calls hci_connect_cfm(conn) without hdev->lock. Generally hci_connect_cfm() assumes it is held, and if conn is deleted concurrently -> UAF. Only SCO and ISO set HCI_PROTO_DEFER and only for defer setup listen, and HCI_EV_CONN_REQUEST is not generated for ISO. In the non-deferred listening socket code paths, hci_connect_cfm(conn) is called with hdev->lock held. Fix by holding the lock.	2026-06-24	8.8
CVE-2026-53075	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ppp: require CAP_NET_ADMIN in target netns for unattached ioctls /dev/ppp open is currently authorized against file->f_cred->user_ns, while unattached administrative ioctls operate on current->nsproxy->net_ns. As a result, a local unprivileged user can create a new user namespace with CLONE_NEWUSER, gain CAP_NET_ADMIN only in that new user namespace, and still issue PPPIOCNEWUNIT, PPPIOCATTACH, or PPPIOCATTCHAN against an inherited network namespace. Require CAP_NET_ADMIN in the user namespace that owns the target network namespace before handling unattached PPP administrative ioctls. This preserves normal pppd operation in the network namespace it is actually privileged in, while rejecting the userns-only inherited-netns case.	2026-06-24	8.8
CVE-2026-13026	google - chrome	Use after free in Digital Credentials in Google Chrome on Mac prior to 149.0.7827.197 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2026-06-24	8.8
CVE-2026-13027	google - chrome	Use after free in FileSystem in Google Chrome prior to 149.0.7827.197 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2026-06-24	8.8
CVE-2026-13031	google - chrome	Use after free in Blink in Google Chrome prior to 149.0.7827.197 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-06-24	8.8
CVE-2026-13033	google - chrome	Out of bounds read and write in Blink>InterestGroups in Google Chrome prior to 149.0.7827.197 allowed a remote attacker to execute arbitrary code via a crafted HTML page. (Chromium security severity: Critical)	2026-06-24	8.8
CVE-2026-13035	google - chrome	Use after free in Bluetooth in Google Chrome on Mac prior to 149.0.7827.197 allowed a remote attacker to execute arbitrary code via a malicious peripheral. (Chromium security severity: High)	2026-06-24	8.8
CVE-2026-13036	google - chrome	Use after free in Blink in Google Chrome prior to 149.0.7827.197 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-06-24	8.8
CVE-2026-13038	google - chrome	Use after free in Autofill in Google Chrome on Windows prior to 149.0.7827.197 allowed a remote attacker to execute arbitrary code via a crafted HTML page. (Chromium security severity: Critical)	2026-06-24	8.8
CVE-2026-9155	gnu - sed	OS Command Injection vulnerability in Rapid7 InsightConnect Sed Plugin on Linux allows authenticated attackers to execute arbitrary OS commands via the expression parameter due to insufficient input validation.	2026-06-25	8.8
CVE-2026-53170	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ethosu: reject DMA commands with uninitialized length cmd_state_init() initializes the command state with memset(0xff), leaving dma->len at U64_MAX to signal missing setup. The only setter is NPU_SET_DMA0_LEN; if userspace omits this command and issues NPU_OP_DMA_START, dma->len remains U64_MAX. In dma_length(), a positive stride added to U64_MAX wraps to a small	2026-06-25	8.8

		value. With size0 == 1, check_mul_overflow() does not trigger and dma_length() returns 0 instead of U64_MAX. The caller's U64_MAX check then passes, region_size[] stays 0, and the bounds check in ethosu_job.c is bypassed, allowing hardware to execute DMA with stale physical addresses. Fix by checking for U64_MAX at the start of dma_length() before any arithmetic, consistent with the sentinel value used throughout the driver to detect uninitialized fields.		
CVE-2026-53171	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ethosu: fix arithmetic issues in dma_length() dma_length() derives DMA region usage from command stream values and updates region_size[]: len = ((len + stride[0]) * size0 + stride[1]) * size1 region_size[region] = max(..., len + dma->offset) Several arithmetic issues can corrupt the derived region size: - signed stride values may underflow when added to len - intermediate multiplications may overflow - len + dma->offset may overflow during region_size updates - dma_length() error returns were not validated by the caller region_size[] is later used by ethosu_job.c to validate command stream accesses against GEM buffer sizes. Arithmetic wraparound can therefore under-report region usage and bypass the bounds validation. Fix by validating signed additions, using overflow helpers for multiplications and offset updates, and propagating dma_length() failures to the caller.	2026-06-25	8.8
CVE-2026-53188	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Validate the passed in fops for ib_get_ucaps() Sashiko pointed out it is not safe to rely only on the devt because char/block alias so if the user finds a block device with the same dev_t it can masquerade as a ucap cdev fd. Test the f_ops to only accept authentic cdevs.	2026-06-25	8.8
CVE-2026-53198	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free of a deferred file_lock on double SMB2_CANCEL A deferred byte-range lock (an SMB2_LOCK that blocks) registers an async work on conn->async_requests via setup_async_work(), with cancel_fn = smb2_remove_blocked_lock and cancel_argv[0] pointing at the struct file_lock. When the request is cancelled, the worker frees the file_lock with locks_free_lock() and takes the cancelled early-exit, which "goto out"s and never reaches release_async_work() -- the only site that unlinks the work from conn->async_requests and clears cancel_fn/cancel_argv. The work therefore stays matchable on async_requests with a live cancel_fn pointing at the freed file_lock, until connection teardown finally runs release_async_work(). smb2_cancel() fires cancel_fn unconditionally with no state guard, so a second SMB2_CANCEL for the same AsyncId, arriving in that window, re-runs smb2_remove_blocked_lock() on the freed file_lock -- a slab use-after-free: BUG: KASAN: slab-use-after-free in __locks_delete_block __locks_delete_block locks_delete_block ksmbd_vfs_posix_lock_unblock smb2_remove_blocked_lock smb2_cancel <- 2nd SMB2_CANCEL fires cancel_fn handle_ksmbd_work Allocated by ...: locks_alloc_lock <- smb2_lock Freed by ...: locks_free_lock <- smb2_lock (cancelled branch) ... cache file_lock_cache of size 192 Reproduced on mainline with KASAN by an authenticated SMB client. Skip a work whose state is already KSMBD_WORK_CANCELLED so its cancel callback cannot be fired a second time.	2026-06-25	8.8
CVE-2026-53200	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: nv: Fix handling of XN[0] when !FEAT_XNX XN has already been extracted from its bitfield position so using	2026-06-25	8.8

		FIELD_PREP() on the mask that clears XN[0] is completely broken, having the effect of unconditionally granting execute permissions... Fix the obvious mistake by manipulating the right bit.		
CVE-2026-53232	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: phy: clean the sfp upstream if phy probing fails Sashiko reported that we don't call sfp_bus_del_upstream() in the probe failure path, so let's add it, otherwise the sfp-bus is left with a dangling 'upstream' field, that may be used later on during SFP events. This issue existed before the generic phylib sfp support, back when drivers were calling phy_sfp_probe themselves.	2026-06-25	8.8
CVE-2026-53240	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: iptfs: fix use-after-free on first_skb in __input_process_payload __input_process_payload() stores first_skb into xtfs->ra_newskb under drop_lock when starting partial reassembly, then unlocks and breaks out of the processing loop. The post-loop check reads xtfs->ra_newskb without the lock to decide whether first_skb is still owned: if (first_skb && first_iplen && !defer && first_skb != xtfs->ra_newskb) Between spin_unlock and this read, a concurrent CPU running iptfs_reassem_cont() (or the drop_timer hrtimer) can complete reassembly, NULL xtfs->ra_newskb, and free the skb. The check then evaluates first_skb != NULL as true, and pskb_trim/ip_summed/consume_skb operate on the freed skb — a use-after-free in skbuff_head_cache. Replace the unlocked read with a local bool that records whether first_skb was handed to the reassembly state in the current call. The flag is set after the existing spin_unlock, before the break, using the pointer equality that is stable at that point (first_skb == skb iff first_skb was stored in ra_newskb).	2026-06-25	8.8
CVE-2026-53248	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: airoha: Fix use-after-free in metadata dst teardown airoha_metadata_dst_free() runs metadata_dst_free() which frees the metadata_dst with kfree() immediately, bypassing the RCU grace period. In the RX path, skb_dst_set_noref() sets a non-refcounted pointer from the skb to the metadata_dst. This function requires RCU read-side protection and the dst must remain valid until all RCU readers complete. Since metadata_dst_free() calls kfree() directly, an use-after-free can occur if any skb still holds a noref pointer to the dst when the driver tears it down. Replace metadata_dst_free() with dst_release() which properly goes through the refcount path: when the refcount drops to zero, it schedules the actual free via call_rcu_hurry(), ensuring all RCU readers have completed before the memory is freed.	2026-06-25	8.8
CVE-2026-53266	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: bridge: make ebt_snat ARP rewrite writable The ebtables SNAT target keeps the Ethernet source address rewrite behind skb_ensure_writable(skb, 0). This is intentional: at the bridge ebtables hooks the Ethernet header is addressed through skb_mac_header()/eth_hdr(), while skb->data points at the Ethernet payload. Asking skb_ensure_writable() for ETH_HLEN bytes would check the payload, not the Ethernet header, and would reintroduce the small packet regression fixed by commit 63137bc5882a. However, the optional ARP sender hardware address rewrite is different. It writes through skb_store_bits() at an offset relative to skb->data: skb_store_bits(skb, sizeof(struct arphdr), info->mac, ETH_ALEN) skb_header_pointer() only safely reads the ARP header; it does not make the later sender hardware address range writable. If that range is still held in a nonlinear skb fragment backed by a splice-imported file page, skb_store_bits() maps the frag page and copies the new MAC address directly into it. Ensure the ARP SHA range is writable before reading the ARP header and before calling skb_store_bits().	2026-06-25	8.8
CVE-2026-53275	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: Fix use-after-free when processing MLD queries	2026-06-25	8.8

		When processing an MLD query, a pointer to the multicast group address is retrieved when initially parsing the packet. This pointer is later dereferenced without being reloaded despite the fact that the skb header might have been reallocated following the pskb_may_pull() calls, leading to a use-after-free [1]. Fix by copying the multicast group address when the packet is initially parsed. [1] BUG: KASAN: slab-use-after-free in __mld_query_work (net/ipv6/mcast.c:1512) Read of size 8 at addr ffff8881154b8e90 by task kworker/4:1/118 Workqueue: mld mld_query_work Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:94 lib/dump_stack.c:120) print_address_description.constprop.0 (mm/kasan/report.c:378) print_report (mm/kasan/report.c:482) kasan_report (mm/kasan/report.c:595) __mld_query_work (net/ipv6/mcast.c:1512) mld_query_work (net/ipv6/mcast.c:1563) process_one_work (kernel/workqueue.c:3314) worker_thread (kernel/workqueue.c:3397 kernel/workqueue.c:3478) kthread (kernel/kthread.c:436) ret_from_fork (arch/x86/kernel/process.c:158) ret_from_fork_asm (arch/x86/entry/entry_64.S:245) </TASK> [...] Freed by task 118: kasan_save_stack (mm/kasan/common.c:57) kasan_save_track (mm/kasan/common.c:78) kasan_save_free_info (mm/kasan/generic.c:584) __kasan_slab_free (mm/kasan/common.c:253 mm/kasan/common.c:285) kfree (./include/linux/kasan.h:235 mm/slub.c:2689 mm/slub.c:6251 mm/slub.c:6566) pskb_expand_head (net/core/skbuff.c:2335) __pskb_pull_tail (net/core/skbuff.c:2878 (discriminator 4)) __mld_query_work (net/ipv6/mcast.c:1495 (discriminator 1)) mld_query_work (net/ipv6/mcast.c:1563) process_one_work (kernel/workqueue.c:3314) worker_thread (kernel/workqueue.c:3397 kernel/workqueue.c:3478) kthread (kernel/kthread.c:436) ret_from_fork (arch/x86/kernel/process.c:158) ret_from_fork_asm (arch/x86/entry/entry_64.S:245)		
CVE-2026-53277	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Take the SRCU lock for page table walks in fault injection and AT emulation walk_s1() and kvm_walk_nested_s2() expect to be called while holding kvm->srcu to guard against memslot changes. While this is generally the case, __kvm_at_s12() and __kvm_find_s1_desc_level() call into the respective walkers without taking kvm->srcu. Fix by acquiring kvm->srcu prior to the table walk in both instances.	2026-06-25	8.8
CVE-2026-53281	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Avoid NULL pointer dereference or refcount corruption Commit 60f030f7418d ("iommu/vt-d: Avoid use of NULL after WARN_ON_ONCE") fixed a NULL pointer dereference in an unlikely situation partly. If dev_pasid is not found in the dev_pasids list, it remains NULL. However, the teardown operations are executed unconditionally, this lead to a NULL pointer dereference or refcount corruption. If the domain was never attached to this IOMMU, info will be NULL, which would cause an immediate dereference when checking --info->refcnt. Even if info is not NULL, decrementing the refcount without having removed a valid PASID might unbalance the count. This could lead to premature dropping of the refcount to 0, potentially causing a use-after-free for the remaining active devices sharing the domain. Fix it by returning early if dev_pasid is NULL, before executing the teardown operations. Issue found by AI review and suggested by Kevin Tian. https://sashiko.dev/#/patchset/20260421031347.1408890-1-zhenzhong.duan%40intel.com	2026-06-26	8.8

CVE-2026-53322	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Clean up DMABUFs before disabling function On device shutdown, make vfio_pci_core_close_device() call vfio_pci_dma_buf_cleanup() before the function is disabled via vfio_pci_core_disable(). This ensures that all access via DMABUFs is revoked before the function's BARs become inaccessible. This fixes an issue where, if the function is disabled first, a tiny window exists in which the function's MSE is cleared and yet BARs could still be accessed via the DMABUF. The resources would also be freed and up for grabs by a different driver.	2026-06-26	8.8
CVE-2026-53230	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix slab-out-of-bounds in mlx5_query_nic_vport_mac_list mlx5_query_nic_vport_mac_list() sizes its firmware command buffer using the PF's log_max_current_uc/mc_list capabilities. When querying a VF vport with a larger configured max (via devlink), the firmware response can overflow this buffer: BUG: KASAN: slab-out-of-bounds in mlx5_query_nic_vport_mac_list+0x453/0x4c0 [mlx5_core] Read of size 4 at addr ff1100013ffc8a12 by task kworker/u96:2/385 CPU: 12 UID: 0 PID: 385 Comm: kworker/u96:2 Not tainted 7.0.0-rc6+ #1 PREEMPT Hardware name: QEMU Standard PC (Q35 + ICH9, 2009) Workqueue: mlx5_esw_wq esw_vport_change_handler [mlx5_core] Call Trace: <TASK> dump_stack_lvl+0x69/0xa0 print_report+0x176/0x4e4 kasan_report+0xc8/0x100 mlx5_query_nic_vport_mac_list+0x453/0x4c0 [mlx5_core] esw_update_vport_addr_list+0x2e3/0xda0 [mlx5_core] esw_vport_change_handle_locked+0xa1f/0x1060 [mlx5_core] esw_vport_change_handler+0x6a/0x90 [mlx5_core] process_one_work+0x87f/0x15e0 worker_thread+0x62b/0x1020 kthread+0x375/0x490 ret_from_fork+0x4dc/0x810 ret_from_fork_asm+0x11/0x20 </TASK> Fix by querying the vport's own HCA caps to size the buffer correctly. Refactor the function to allocate and return the MAC list internally, removing the caller's dependency on knowing the correct max.	2026-06-25	8.7
CVE-2026-9716	schneider-electric - powerlogic_p7_firmware	CWE-476 NULL Pointer Dereference vulnerability exists that could cause a denial-of-service condition, rendering the device’s HMI and configuration functionality unavailable when malformed requests are received over exposed network interfaces.	2026-06-25	8.7
CVE-2026-40079	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior are vulnerable to Command Injection due to lack of sanitization in the escape_command() function. The escape_command() function at lib/rrd.php is a no-op: it returns $command unchanged. The command line built by rrdtool_function_graph() is passed through this function and then to shell_exec($full_commandline). The risk is in __rrd_execute() where text_format values from graph templates (which may contain host variable substitutions) reach shell_exec without adequate escaping. This issue has been addressed in version 1.2.31.	2026-06-25	8.6
CVE-2026-53217	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mvpp2: sync RX data at the hardware packet offset mvpp2 programs the RX queue packet offset, so hardware writes received data at dma_addr + MVPP2_SKB_HEADROOM. The current CPU sync starts at dma_addr and only covers rx_bytes + MVPP2_MH_SIZE bytes, which syncs the unused headroom and misses the same number of bytes at the packet tail. On non-coherent DMA systems this can leave the CPU reading stale cache contents for the end of the received frame. Use dma_sync_single_range_for_cpu() with MVPP2_SKB_HEADROOM as the range offset so the sync covers the Marvell header and packet data actually written by hardware.	2026-06-25	8.6
CVE-2026-9717	schneider-electric - powerlogic_p7_firmware	CWE-78 Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability exists that could allow unauthorized execution of commands with elevated privileges, impacting system integrity, confidentiality, and availability when a privileged authenticated user interacts with a vulnerable network-exposed service.	2026-06-25	8.6
CVE-2026-12975	red hat - multiple products	A flaw was found in Apicurio Registry. The ContentTypeUtil.isParsableXml() method creates a SAXParserFactory without enabling secure processing features or disabling external entity resolution. An attacker with artifact-write permission (or unauthenticated when the registry	2026-06-25	8.5

		runs with default configuration) can upload a crafted XML document to trigger blind server-side request forgery (SSRF) via external DTD/entity fetch, or cause denial of service via entity expansion.		
CVE-2026-13325	red hat - multiple products	A flaw was found in KubeVirt's migration proxy. When spec.configuration.migrations.disableTLS is set to true on the KubeVirt custom resource, the target virt-handler binds a plain TCP listener on all interfaces (0.0.0.0/::) on a random port with no authentication, peer allow-list, or handshake token. This listener proxies directly into the target virt-launcher's virtqemud control socket. An attacker with a running pod on the cluster network can connect to this listener and issue unfiltered libvirt RPC commands against another tenant's virtual machine, including reading VM memory and configuration, modifying VM state via QMP, or destroying the VM. The bind address is unconditionally 0.0.0.0 — configuring a dedicated migration network via migrations.network only changes the advertised migration IP, not the listener bind address, so the port remains reachable on the pod network even when a dedicated migration network is configured. The API documentation describes disableTLS as removing "the additional layer of live migration encryption" without disclosing that it also removes all mutual authentication.	2026-06-26	8.5
CVE-2026-53091	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: pull headers in qdisc_pkt_len_segs_init() Most ndo_start_xmit() methods expects headers of gso packets to be already in skb->head. net/core/tso.c users are particularly at risk, because tso_build_hdr() does a memcpy(hdr, skb->data, hdr_len); qdisc_pkt_len_segs_init() already does a dissection of gso packets. Use pskb_may_pull() instead of skb_header_pointer() to make sure drivers do not have to reimplement this. Some malicious packets could be fed, detect them so that we can drop them sooner with a new SKB_DROP_REASON_SKB_BAD_GSO drop_reason.	2026-06-24	8.4
CVE-2026-54100	red hat - multiple products	A flaw was found in the Windows Machine Config Operator (WMCO) for Red Hat OpenShift Container Platform. WMCO establishes SSH connections to Windows worker nodes without verifying the remote server host key. An adjacent-network attacker who can intercept or redirect WMCO's SSH session can capture WICD and kubelet bootstrap credentials transferred during node configuration, enabling compromise of Windows node identities in the cluster.	2026-06-22	8.3
CVE-2026-52920	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: xt_policy: fix strict mode inbound policy matching match_policy_in() walks sec_path entries from the last transform to the first one, but strict policy matching needs to consume info->pol[] in the same forward order as the rule layout. Derive the strict-match policy position from the number of transforms already consumed so that multi-element inbound rules are matched consistently.	2026-06-24	8.3
CVE-2026-13025	google - chrome	Race in DevTools in Google Chrome prior to 149.0.7827.197 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)	2026-06-24	8.3
CVE-2026-13281	google - chrome	Integer overflow in Mojo in Google Chrome prior to 149.0.7827.201 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a malicious file. (Chromium security severity: High)	2026-06-25	8.3
CVE-2026-2053	wso2 - multiple products	The WSO2 API Manager's message flow component, when processing WS-Addressing headers, does not sufficiently validate or restrict user-controlled input within these headers. This omission allows an attacker to manipulate WS-Addressing headers to specify arbitrary destinations for server-initiated requests. Successful exploitation allows an unauthenticated attacker to control the destination of server-initiated requests originating from the WSO2 API Manager. This direct control can enable unauthorized access to internal network resources or services that would typically be inaccessible from external networks.	2026-06-26	8.3
CVE-2026-11878	microfocus - multiple products	Improper neutralization of input during web page generation ('cross-site scripting') vulnerability in OpenText Access Manager allows Cross-Site Scripting (XSS). This issue affects Access Manager: from 5.1 through 5.1.2.	2026-06-24	8.2
CVE-2026-53268	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack_irc: fix possible out-of-bounds read When parsing fails after we've matched the command string we should bail out instead of trying to match a different command. This helper should be deprecated, given prevalence of TLS I doubt it has any relevance in 2026.	2026-06-25	8.2
CVE-2026-56091	apache software foundation - Apache Shiro	When using Apache Shiro with the shiro-guice module in a web servlet context, a specially crafted HTTP request may cause an authentication bypass. This vulnerability is similar to https://www.cve.org/CVERecord?id=CVE-2020-1957 https://www.cve.org/CVERecord , except that it affects the `shiro-guice` module instead of the	2026-06-25	8.2

		`shiro-spring` module. This issue affects all Apache Shiro versions through 2.x, and 3.0.0-alpha-1 only when using `shiro-guice` module in a web servlet context. Upgrade to version 3.0.0 or later, which fixes the issue.		
CVE-2025-66336	apache - doris_mcp_server	Apache Doris MCP Server contains a SQL injection vulnerability in a metadata query path. A user-controlled database name is directly interpolated into a SQL query, and the query is executed without passing the caller's authorization context. This may allow an authenticated attacker, or an anonymous attacker if authentication is disabled, to bypass SQL security validation and access metadata outside the intended database scope. Affected users are recommended to upgrade to Doris version 0.6.1 or later, which fixes the issue.	2026-06-22	8.1
CVE-2026-9072	ibm - i	IBM WebSphere Application Server and IBM WebSphere Application Server Liberty - when using Intelligent Management with the WebSphere WebServer Plug-in component - are vulnerable to remote code execution and denial of service. This vulnerability can be exploited when an attacker impersonates backend servers and sends crafted responses to the plug-in.	2026-06-22	8.1
CVE-2026-44271	dell - wyse_management_suite	Dell Wyse Management Suite (WMS), versions prior to WMS 2605, contain an Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability. A low privileged attacker with remote access could potentially exploit this vulnerability, leading to Unauthorized access.	2026-06-22	8.1
CVE-2026-52967	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb/client: fix possible infinite loop and oob read in symlink_data() On 32-bit architectures, the infinite loop is as follows: len = p->ErrorDataLength == 0xffffffff8 u8 *next = p->ErrorContextData + len next == p On 32-bit architectures, the out-of-bounds read is as follows: len = p->ErrorDataLength == 0xfffffff0 u8 *next = p->ErrorContextData + len next == (u8 *)p - 8	2026-06-24	8.1
CVE-2026-53147	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Validate XDomain request packet size before type cast tb_xdp_handle_request() casts the received packet buffer to protocol-specific structs without verifying that the allocation is large enough for the target type. A peer can send a minimal XDomain packet that passes the generic header length check but is shorter than the struct accessed after the cast, causing out-of-bounds reads from the kmemdup allocation. Plumb the packet length through xdomain_request_work and validate it against the expected struct size before each cast.	2026-06-25	8.1
CVE-2026-53178	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: staging: rtl8723bs: rtw_mlme: add bounds checks before ie_length subtraction Add guards to ensure ie_length is large enough before subtracting fixed IE offsets to prevent unsigned integer underflow.	2026-06-25	8.1
CVE-2026-53254	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: validate skb length in MCC handlers The RFCOMM MCC handlers cast skb->data to protocol-specific structs without validating skb->len first. A malicious remote device can send truncated MCC frames and trigger out-of-bounds reads in these handlers. Fix this by using skb_pull_data() to validate and access the required data before dereferencing it. rfcomm_recv_rpn() requires special handling since ETSI TS 07.10 allows 1-byte RPN requests. Handle this by validating only the DLCI byte first, and validating the full struct only when len > 1.	2026-06-25	8.1
CVE-2026-9800	redhat - multiple products	A flaw was found in Keycloak Policy Enforcer. This vulnerability allows any authenticated user to bypass all authorization policies, including role, scope, and User-Managed Access (UMA) permission checks. By including the configured access-denied page path within a request URL, either as a path segment or a query parameter, an attacker can gain unauthorized access to protected resources.	2026-06-25	8.1
CVE-2026-11800	redhat - build_of_keycloak	A flaw was found in Keycloak. This JWT algorithm confusion vulnerability in the JWT Authorization Grant flow allows an attacker with valid client credentials to bypass signature verification. By forging an assertion, the attacker can create unauthorized access tokens. This enables the attacker to impersonate any federated user linked to the affected Identity Provider, leading to unauthorized access and potential privilege escalation.	2026-06-25	8.1

CVE-2026-53256	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: hold listener socket in rfcomm_connect_ind() rfcomm_get_sock_by_channel() scans rfcomm_sk_list under the list lock, but returns the selected listener after dropping that lock without taking a reference. rfcomm_connect_ind() then locks the listener, queues a child socket on it, and may notify it after unlocking it. The buggy scenario involves two paths, with each column showing the order within that path: rfcomm_connect_ind(): listener close: 1. Find parent in rfcomm_get_sock_by_channel() 1. close() enters rfcomm_sock_release(). 2. Drop rfcomm_sk_list.lock without pinning parent. 2. rfcomm_sock_shutdown() closes the listener. 3. Call lock_sock(parent) and bt_accept_enqueue(parent, sk, true). 3. rfcomm_sock_kill() unlinks and puts parent. 4. Read parent flags and may call sk_state_change(). 4. parent can be freed. If close wins the race, parent can be freed before rfcomm_connect_ind() reaches lock_sock(), bt_accept_enqueue(), or the deferred-setup callback. Take a reference on the listener before leaving rfcomm_sk_list.lock. After lock_sock() succeeds, recheck that it is still in BT_LISTEN before queueing a child, cache the deferred-setup bit while the parent is locked, and drop the reference after the last parent use. KASAN reported a slab-use-after-free in lock_sock_nested() from rfcomm_connect_ind(), with the freeing stack going through rfcomm_sock_kill() and rfcomm_sock_release().	2026-06-25	8
CVE-2026-40711	dell - Container Storage Modules	Dell Dell Container Storage Modules, version(s) csi-powerstore v2.16.0, csi-unity v2.16.0, csi-powerflex v2.16.0, csi-powermax v2.16.0, contain(s) an Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Command execution.	2026-06-26	8
CVE-2026-44274	dell - wyse_management_suite	Dell Wyse Management Suite (WMS), versions prior to WMS 2605, contain an Improper Link Resolution Before File Access vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Unauthorized access.	2026-06-22	7.8
CVE-2020-9695	adobe - multiple products	Acrobat Reader versions 2020.009.20074, 2020.001.30002, 2017.011.30171, 2015.006.30523 and earlier are affected by an out-of-bounds write vulnerability that could result in arbitrary code execution in the context of the current user. Exploitation of this issue requires user interaction in that a victim must open a malicious file.	2026-06-23	7.8
CVE-2026-12112	redhat - multiple products	A flaw was found in the foreman-mcp-server. A session management vulnerability in the MCP Server allows unauthenticated attackers to hijack active administrative sessions due to an improper cache of authenticated client connections, by trusting a non-secret session ID without re-validating authentication tokens and by logging all newly created session IDs to standard logs. This issue can result in privilege escalation and infrastructure-wide code execution.	2026-06-23	7.8
CVE-2026-52912	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_queue: hold bridge skb->dev while queued br_pass_frame_up() rewrites skb->dev from the ingress port to the bridge master before queueing bridge LOCAL_IN packets. NFQUEUE only holds references on state.in/out and bridge physdevs, so a queued bridge packet can retain a freed bridge master in skb->dev until reinjection. When the verdict is reinjected later, br_netif_receive_skb() re-enters the receive path with skb->dev still pointing at the freed bridge master, triggering a use-after-free. Store skb->dev in the queue entry, hold a reference on it for the queue lifetime, and use the saved device when dropping queued packets during NETDEV_DOWN handling.	2026-06-24	7.8
CVE-2026-52919	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix tp_meter counter underflow during shutdown batadv_tp_sender_shutdown() unconditionally decrements the "sending" atomic counter. If multiple paths (e.g. timeout, user cancel, and normal finish) call this function, the counter can underflow to -1. Since the sender logic treats any non-zero value as "still sending", a negative value causes the sender kthread to loop indefinitely. This leads to a use-after-free when the interface is removed while the zombie thread is still active.	2026-06-24	7.8

		Fix this by using atomic_xchg() to ensure the counter only transitions from 1 to 0 once. [sven: added missing change in batadv_tp_send]		
CVE-2026-52923	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipc: limit next_id allocation to the valid ID range The checkpoint/restore sysctl path can request the next SysV IPC id through ids->next_id. ipc_idr_alloc() currently forwards that request to idr_alloc() with an open-ended upper bound. If the valid tail of the SysV IPC id space is full, the allocation can spill beyond ipc_mni. The returned SysV IPC id still uses the normal index encoding, so later lookup and removal can target the wrong slot. This leaves the real IDR entry behind and breaks the IDR state for the object. The bug is in ipc_idr_alloc() in the checkpoint/restore path. 1. ids->next_id is passed to: idr_alloc(&ids->ipcs_idr, new, ipcid_to_idx(next_id), 0, ...) 2. The zero upper bound makes the allocation effectively open-ended. Once the valid SysV IPC tail is occupied, idr_alloc() can spill past ipc_mni and allocate an entry beyond the valid IPC id range. 3. The new object id is still encoded with the narrower SysV IPC index width: new->id = (new->seq << ipcmni_seq_shift()) + idx 4. Later removal goes through ipc_rmid(), which uses: ipcid_to_idx(ipcp->id) That truncates the real IDR index. An object actually stored at a high index can then be removed as if it lived at a low in-range index. 5. For shared memory, shm_destroy() frees the current object anyway, but the real high IDR slot is left behind as a dangling pointer. 6. A subsequent walk of /proc/sysvipc/shm reaches the stale IDR entry and dereferences freed memory. Prevent this by bounding the requested allocation to ipc_mni so the checkpoint/restore path fails once the valid range is exhausted.	2026-06-24	7.8
CVE-2026-52927	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: ebtables: fix OOB read in compat_mtw_from_user Luxiao Xu says: The function compat_mtw_from_user() converts ebtables extensions from 32-bit user structures to kernel native structures. However, it lacks proper validation of the user-supplied match_size/target_size. When certain extensions are processed, the kernel-side translation logic may perform memory accesses based on the extension's expected size. If the user provides a size smaller than what the extension requires, it results in an out-of-bounds read as reported by KASAN. This fix introduces a check to ensure match_size is at least as large as the extension's required compatsize. This covers matches, watchers, and targets, while maintaining compatibility with standard targets. AFAIU this is relevant for matches that need to go though match->compat_from_user() call. Those that use plain memcpy with the user-provided size are ok because the caller checks that size vs the start of the next rule entry offset (which itself is checked vs. total size copied from userspace). The ->compat_from_user() callbacks assume they can read compatsize bytes, so they need this extra check. Based on an earlier patch from Luxiao Xu.	2026-06-24	7.8
CVE-2026-52933	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-06-24	7.8

		io_uring/poll: fix signed comparison in io_poll_get_ownership() io_poll_get_ownership() uses a signed comparison to check whether poll_refs has reached the threshold for the slowpath: if (unlikely(atomic_read(&req->poll_refs) >= IO_POLL_REF_BIAS)) atomic_read() returns int (signed). When IO_POLL_CANCEL_FLAG (BIT(31)) is set in poll_refs, the value becomes negative in signed arithmetic, so the >= 128 comparison always evaluates to false and the slowpath is never taken. Fix this by casting the atomic_read() result to unsigned int before the comparison, so that the cancel flag is treated as a large positive value and correctly triggers the slowpath.		
CVE-2026-52935	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: espintcp: do not reuse an in-progress partial send espintcp keeps a single in-flight transmit in ctx->partial. Before building a new sk_msg, espintcp_sendmsg() first tries to flush that state through espintcp_push_msgs(). For blocking callers, espintcp_push_msgs() may return success even when the previous partial send is still pending. espintcp_sendmsg() would then reinitialize emsg->skmsg and reuse ctx->partial while the old transfer still owns that state. Do not rebuild the send message when ctx->partial is still in progress. If espintcp_push_msgs() returns with emsg->len still set, fail the new send instead of overwriting the live partial state. This is a memory-safety fix: reusing the live partial-send state can leave a stale offset attached to a new sk_msg and lead to an out-of- bounds read in the send path. tcp_sendmsg_locked() already handles waiting for send buffer memory, so the fix here is just to preserve espintcp's one-message-at-a-time transmit state.	2026-06-24	7.8
CVE-2026-52943	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: skbuff: fix missing zerocopy reference in pskb_carve helpers pskb_carve_inside_header() and pskb_carve_inside_nonlinear() both copy the old skb_shared_info header into a new buffer via memcpy(), which includes the destructor_arg pointer (uarg) for MSG_ZEROCOPY skbs. Neither function calls net_zcopy_get() for the new shinfo, creating an unaccounted holder: every skb_shared_info with destructor_arg set will call skb_zcopy_clear() once when freed, but the corresponding net_zcopy_get() was never called for the new copy. Repeated calls drive uarg->refcnt to zero prematurely, freeing ubuf_info_msgzc while TX skbs still hold live destructor_arg pointers. KASAN reports use-after-free on a freed ubuf_info_msgzc: BUG: KASAN: slab-use-after-free in skb_release_data+0x77b/0x810 Read of size 8 at addr ffff88801574d3e8 by task poc/220 Call Trace: skb_release_data+0x77b/0x810 kfree_skb_list_reason+0x13e/0x610 skb_release_data+0x4cd/0x810 sk_skb_reason_drop+0xf3/0x340 skb_queue_purge_reason+0x282/0x440 rds_tcp_inc_free+0x1e/0x30 rds_recvmsg+0x354/0x1780 __sys_recvmsg+0xdf/0x180 Allocated by task 219: msg_zerocopy_realloc+0x157/0x7b0 tcp_sendmsg_locked+0x2892/0x3ba0 Freed by task 219: ip_recv_error+0x74a/0xb10 tcp_recvmsg+0x475/0x530 The skb consuming the late access still referenced the same uarg via shinfo->destructor_arg copied by pskb_carve_inside_nonlinear() without a refcount bump. This has been verified to be reliably exploitable: a working proof-of-concept achieves full root privilege escalation from an unprivileged local user on a default kernel configuration.	2026-06-24	7.8

		The fix follows the pattern of pskb_expand_head() which has the same memcpy/cloned structure. For pskb_carve_inside_header(), net_zcopy_get() is placed after skb_orphan_frags() succeeds, so the orphan error path needs no cleanup. For pskb_carve_inside_nonlinear(), net_zcopy_get() is placed after all failure points and just before skb_release_data(), so no error path needs cleanup at all -- matching pskb_expand_head() more closely and avoiding the need for a balancing net_zcopy_put().		
CVE-2026-52947	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: fix refcount saturation and potential UAF in qrtr_port_remove In qrtr_port_remove(), the socket reference count is decremented via __sock_put() before the port is removed from the qrtr_ports XArray and before the RCU grace period elapses. This breaks the fundamental RCU update paradigm. It exposes a race window where a concurrent RCU reader (such as qrtr_reset_ports() or qrtr_port_lookup()) can obtain a pointer to the socket from the XArray, and attempt to call sock_hold() on a socket whose reference count has already dropped to zero. This exact race condition was hit during syzkaller fuzzing, leading to the following refcount saturation warning and a potential Use-After-Free: refcount_t: saturated; leaking memory. WARNING: CPU: 3 PID: 1273 at lib/refcount.c:22 refcount_warn_saturate+0xae/0x1d0 Modules linked in: qrtr(+) bochs drm_shmem_helper ... Call Trace: <TASK> qrtr_reset_ports net/qrtr/af_qrtr.c:768 [inline] [qrtr] __qrtr_bind.isra.0+0x48b/0x570 net/qrtr/af_qrtr.c:805 [qrtr] qrtr_bind+0x17d/0x210 net/qrtr/af_qrtr.c:901 [qrtr] kernel_bind+0xe4/0x120 net/socket.c:3592 qrtr_ns_init+0x1a6/0x380 net/qrtr/ns.c:715 [qrtr] qrtr_proto_init+0x3b/0xff0 net/qrtr/af_qrtr.c:169 [qrtr] do_one_initcall+0xf5/0x5e0 init/main.c:1283 ... </TASK> Fix this by deferring the reference count decrement until after the xa_erase() and the synchronize_rcu() complete. (Note: The v1 of this patch incorrectly replaced __sock_put() with sock_put(). As Simon Horman pointed out, the callers of qrtr_port_remove() still hold a reference to the socket, so freeing the socket memory here would lead to a subsequent UAF in the caller. Thus, the __sock_put() is kept, but only repositioned to close the RCU race.)	2026-06-24	7.8
CVE-2026-52950	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/dma-buf: fix UAF with retry loop Retry doesn't work here, since bo will be freed on error, leading to UAF. However, now that we do the alloc & init before the attach, we can now combine this as one unit and have the init do the alloc for us. This should make the retry safe. Reported by Sashiko. v2: Fix up the error unwind (CI) (cherry picked from commit 479669418253e0f27f8cf5db01a731352ea592e7)	2026-06-24	7.8
CVE-2026-52951	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/dma-buf: handle empty bo and UAF races There look to be some nasty races here when triggering the invalidate_mappings hook: 1) We do xe_bo_alloc() followed by the attach, before the actual full bo init step in xe_dma_buf_init_obj(). However the bo is visible on the attachments list after the attach. This is bad since exporter driver, say amdgpu, can at any time call back into our invalidate_mappings hook, with an empty/bogus bo, leading to potential bugs/crashes. 2) Similar to 1) but here we get a UAF, when the invalidate_mappings hook is triggered. For example, we get as far as xe_bo_init_locked() but this fails in some way. But here the bo will be freed on error, but we still have it attached from dma-buf pov, so if the invalidate_mappings is now triggered then the bo we access is gone and we trigger UAF and more bugs/crashes.	2026-06-24	7.8

		To fix this, move the attach step until after we actually have a fully set up buffer object. Note that the bo is not published to userspace until later, so not sure what the comment "Don't publish the bo until we have a valid attachment", is referring to. We have at least two different customers reporting hitting a NULL ptr deref in evict_flags when importing something from amdgpu, followed by triggering the evict flow. Hit rate is also pretty low, which would hint at some kind of race, so something like 1) or 2) might explain this. v2: - Shuffle the order of the ops slightly (no functional change) - Improve the comment to better explain the ordering (Matt B) (cherry picked from commit af1f2ad0c59fe4e2f924c526f66e968289d77971)		
CVE-2026-52959	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: virt: sev-guest: Do not use host-controlled page order in cleanup path When issuing an extended guest request (SVM_VMGEXIT_EXT_GUEST_REQUEST), get_ext_report() allocates a buffer to retrieve a certificate blob from the host, keeping track of its size in report_req->certs_len. However, the host may return SNP_GUEST_VMM_ERR_INVALID_LEN, indicating an invalid buffer size, as well as the expected length of such buffer. get_ext_report() subsequently updates report_req->certs_len with the host-controlled value, and cleans up the buffer by computing a page order from such value. This is incorrect, as the host-provided length may not match the page order of the original allocation, potentially resulting in corruption in the page allocator. Fix this by using alloc_pages_exact() instead, and reusing @npages to compute the size passed to free_pages_exact(). For consistency, also use @npages to compute the size when allocating the pages, even though this last change has no functional effect.	2026-06-24	7.8
CVE-2026-52971	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ena: PHC: Fix potential use-after-free in get_timestamp Move the phc->active check and resp pointer assignment to after acquiring the spinlock. Previously, phc->active was checked without holding the lock, and resp was cached from ena_dev->phc.virt_addr before the lock was acquired. If ena_com_phc_destroy() runs between the lockless active check and the lock acquisition, it sets active=false, releases the lock, frees the DMA memory, and sets virt_addr=NULL. The get_timestamp path would then read a NULL virt_addr and dereference it. With both the active check and the pointer read under the lock, destroy cannot free the memory while get_timestamp is using it.	2026-06-24	7.8
CVE-2026-52973	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: futex: Drop CLONE_THREAD requirement for private default hash alloc Currently need_futex_hash_allocate_default() depends on strict pthread semantics, abusing CLONE_THREAD. This breaks the non-concurrency assumptions when doing the mm->futex_ref pcpu allocations, leading to bugs[0] when sharing the mm in other ways; ie: BUG: KASAN: slab-use-after-free in futex_hash_put ... where the +1 bias can end up on a percpu counter that mm->futex_ref no longer points at. Loosen the check to cover any CLONE_VM clone, except vfork(). Excluding vfork keeps the existing paths untouched (no overhead), and we can't race in the first place: either the parent is suspended and the child runs alone, or mm->futex_ref is already allocated from an earlier CLONE_VM.	2026-06-24	7.8
CVE-2026-52975	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bonding: 3ad: implement proper RCU rules for port->aggregator syzbot found a data-race in bond_3ad_get_active_agg_info / bond_3ad_state_machine_handler [1] which hints at lack of proper RCU implementation. Add __rcu qualifier to port->aggregator, and add proper RCU API.	2026-06-24	7.8

		[1] BUG: KCSAN: data-race in bond_3ad_get_active_agg_info / bond_3ad_state_machine_handler write to 0xffff88813cf5c4b0 of 8 bytes by task 36 on cpu 0: ad_port_selection_logic drivers/net/bonding/bond_3ad.c:1659 [inline] bond_3ad_state_machine_handler+0x9d5/0x2d60 drivers/net/bonding/bond_3ad.c:2569 process_one_work kernel/workqueue.c:3302 [inline] process_scheduled_works+0x4f0/0x9c0 kernel/workqueue.c:3385 worker_thread+0x58a/0x780 kernel/workqueue.c:3466 kthread+0x22a/0x280 kernel/kthread.c:436 ret_from_fork+0x146/0x330 arch/x86/kernel/process.c:158 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 read to 0xffff88813cf5c4b0 of 8 bytes by task 22063 on cpu 1: __bond_3ad_get_active_agg_info drivers/net/bonding/bond_3ad.c:2858 [inline] bond_3ad_get_active_agg_info+0x8c/0x230 drivers/net/bonding/bond_3ad.c:2881 bond_fill_info+0xe0f/0x10f0 drivers/net/bonding/bond_netlink.c:853 rtnl_link_info_fill net/core/rtnetlink.c:906 [inline] rtnl_link_fill+0x1d7/0x4e0 net/core/rtnetlink.c:927 rtnl_fill_ifinfo+0xf8e/0x1380 net/core/rtnetlink.c:2168 rtmsg_ifinfo_build_skb+0x11c/0x1b0 net/core/rtnetlink.c:4453 rtmsg_ifinfo_event net/core/rtnetlink.c:4486 [inline] rtmsg_ifinfo+0x6d/0x110 net/core/rtnetlink.c:4495 __dev_notify_flags+0x76/0x390 net/core/dev.c:9790 netif_change_flags+0xac/0xd0 net/core/dev.c:9823 do_setlink+0x905/0x2950 net/core/rtnetlink.c:3180 rtnl_group_changelink net/core/rtnetlink.c:3813 [inline] __rtnl_newlink net/core/rtnetlink.c:3981 [inline] rtnl_newlink+0xf55/0x1400 net/core/rtnetlink.c:4109 rtnetlink_rcv_msg+0x64b/0x720 net/core/rtnetlink.c:6995 netlink_rcv_skb+0x123/0x220 net/netlink/af_netlink.c:2550 rtnetlink_rcv+0x1c/0x30 net/core/rtnetlink.c:7022 netlink_unicast_kernel net/netlink/af_netlink.c:1318 [inline] netlink_unicast+0x5a8/0x680 net/netlink/af_netlink.c:1344 netlink_sendmsg+0x5c8/0x6f0 net/netlink/af_netlink.c:1894 sock_sendmsg_nosec net/socket.c:787 [inline] __sock_sendmsg net/socket.c:802 [inline] ____sys_sendmsg+0x563/0x5b0 net/socket.c:2698 ___sys_sendmsg+0x195/0x1e0 net/socket.c:2752 __sys_sendmsg net/socket.c:2784 [inline] __do_sys_sendmsg net/socket.c:2789 [inline] __se_sys_sendmsg net/socket.c:2787 [inline] __x64_sys_sendmsg+0xd4/0x160 net/socket.c:2787 x64_sys_call+0x194c/0x3020 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0x12c/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f value changed: 0x0000000000000000 -> 0xffff88813cf5c400 Reported by Kernel Concurrency Sanitizer on: CPU: 1 UID: 0 PID: 22063 Comm: syz.0.31122 Tainted: G W syzkaller #0 PREEMPT(full) Tainted: [W]=WARN Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/18/2026		
CVE-2026-52976	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix error cleanup in xe_exec_queue_create_ioctl() Two error handling issues exist in xe_exec_queue_create_ioctl(): 1. When xe_hw_engine_group_add_exec_queue() fails, the error path jumps to put_exec_queue which skips xe_exec_queue_kill(). If the VM is in preempt fence mode, xe_vm_add_compute_exec_queue() has already added the queue to the VM's compute exec queue list. Skipping the kill leaves the queue on that list, leading to a dangling pointer after the queue is freed. 2. When xa_alloc() fails after xe_hw_engine_group_add_exec_queue() has succeeded, the error path does not call xe_hw_engine_group_del_exec_queue() to remove the queue from the hw engine group list. The queue is then freed while still linked into the hw engine group, causing a use-after-free. Fix both by: - Changing the xe_hw_engine_group_add_exec_queue() failure path to jump to kill_exec_queue so that xe_exec_queue_kill() properly removes the queue from the VM's compute list.	2026-06-24	7.8

		- Adding a del_hw_engine_group label before kill_exec_queue for the xa_alloc() failure path, which removes the queue from the hw engine group before proceeding with the rest of the cleanup. (cherry picked from commit 37c831f401746a45d510b312b0ed7a77b1e06ec8)		
CVE-2026-52987	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: avoid double drm_exec_fini() in userq validate When new_addition is true, amdgpu_userq_vm_validate() calls drm_exec_fini(&exec) before iterating over the collected HMM ranges and calling amdgpu_ttm_tt_get_user_pages(). If amdgpu_ttm_tt_get_user_pages() fails in that path, the code jumps to unlock_all and calls drm_exec_fini(&exec) a second time on the same exec object. drm_exec_fini() is not idempotent: it frees exec->objects and may also drop exec->contended and finalize the ww acquire context. Route that error path directly to the range cleanup once exec has already been finalized. Issue found using a prototype static analysis tool and confirmed by code review. (cherry picked from commit 2802952e4a07306da6ebe813ff1acacc5691851a)	2026-06-24	7.8

CVE-2026-52991	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		sched/psi: fix race between file release and pressure write
		
		A potential race condition exists between pressure write and cgroup file
		release regarding the priv member of struct kernfs_open_file, which
		triggers the uaf reported in [1].
		
		Consider the following scenario involving execution on two separate CPUs:
		
		CPU0 CPU1
		==== ====
		vfs_rmdir()
		kernfs_iop_rmdir()
		cgroup_rmdir()
		cgroup_kn_lock_live()
		cgroup_destroy_locked()
		cgroup_addrm_files()
		cgroup_rm_file()
		kernfs_remove_by_name()
		kernfs_remove_by_name_ns()
		vfs_write() __kernfs_remove()
		new_sync_write() kernfs_drain()
		kernfs_fop_write_iter() kernfs_drain_open_files()
		cgroup_file_write() kernfs_release_file()
		pressure_write() cgroup_file_release()
		ctx = of->priv;
		kfree(ctx);
		of->priv = NULL;
		cgroup_kn_unlock()
		cgroup_kn_lock_live()
		cgroup_get(cgrp)
		cgroup_kn_unlock()
		if (ctx->psi.trigger) // here, trigger uaf for ctx, that is of->priv
		
		The cgroup_rmdir() is protected by the cgroup_mutex, it also safeguards
		the memory deallocation of of->priv performed within cgroup_file_release().
		However, the operations involving of->priv executed within pressure_write()
		are not entirely covered by the protection of cgroup_mutex. Consequently,
		if the code in pressure_write(), specifically the section handling the
		ctx variable executes after cgroup_file_release() has completed, a uaf
		vulnerability involving of->priv is triggered.
		
		Therefore, the issue can be resolved by extending the scope of the
		cgroup_mutex lock within pressure_write() to encompass all code paths
		involving of->priv, thereby properly synchronizing the race condition
		occurring between cgroup_file_release() and pressure_write().
		
		And, if an live kn lock can be successfully acquired while executing
		the pressure write operation, it indicates that the cgroup deletion
		process has not yet reached its final stage; consequently, the priv
		pointer within open_file cannot be NULL. Therefore, the operation to
		retrieve the ctx value must be moved to a point *after* the live kn
		lock has been successfully acquired.
		
		In another situation, specifically after entering cgroup_kn_lock_live()
			2026-06-24	7.8

		but before acquiring cgroup_mutex, there exists a different class of
		race condition:
		
		CPU0: write memory.pressure CPU1: write cgroup.pressure=0
		========== ==========
		
		kernfs_fop_write_iter()
		kernfs_get_active_of(of)
		pressure_write()
		cgroup_kn_lock_live(memory.pressure)
		cgroup_tryget(cgrp)
		kernfs_break_active_protection(kn)
		... blocks on cgroup_mutex
		
		cgroup_pressure_write()
		cgroup_kn_lock_live(cgroup.pressure)
		cgroup_file_show(memory.pressure, false)
		kernfs_show(false)
		kernfs_drain_open_files()
		cgroup_file_release(of)
		kfree(ctx)
		of->priv = NULL
		cgroup_kn_unlock()
		
		... acquires cgroup_mutex
		ctx = of->priv; // may now be NULL
		if (ctx->psi.trigger) // NULL dereference
		
		Consequently, there is a possibility that of->priv is NULL, the pressure
		write needs to check for this.
		
		Now that the scope of the cgroup_mutex has been expanded, the original
		explicit cgroup_get/put operations are no longer necessary, this is
		because acquiring/releasing the live kn lock inherently executes a
		cgroup get/put operation.
		
		[1]
		BUG: KASAN: slab-use-after-free in pressure_write+0xa4/0x210
		kernel/cgroup/cgroup.c:4011
		Call Trace:
		pressure_write+0xa4/0x210 kernel/cgroup/cgroup.c:4011
		cgroup_file_write+0x36f/0x790 kernel/cgroup/cgroup.c:43
		---truncated---		

CVE-2026-53000	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nat: use kfree_rcu to release ops Florian Westphal says: "Historically this is not an issue, even for normal base hooks: the data path doesn't use the original nf_hook_ops that are used to register the callbacks. However, in v5.14 I added the ability to dump the active netfilter hooks from userspace. This code will peek back into the nf_hook_ops that are available at the tail of the pointer-array blob used by the datapath. The nat hooks are special, because they are called indirectly from the central nat dispatcher hook. They are currently invisible to the nfnl hook dump subsystem though. But once that changes the nat ops structures have to be deferred too." Update nf_nat_register_fn() to deal with partial exposition of the hooks from error path which can be also an issue for nfnetlink_hook.	2026-06-24	7.8
CVE-2026-53005	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: af_unix: Drop all SCM attributes for SOCKMAP. SOCKMAP can hide inflight fd from AF_UNIX GC. When a socket in SOCKMAP receives skb with inflight fd, sk_psock_verdict_data_ready() looks up the mapped socket and enqueue skb to its psock->ingress_skb. Since neither the old nor the new GC can inspect the psock queue, the hidden skb leaks the inflight sockets. Note that this cannot be detected via kmemleak because inflight sockets are linked to a global list.	2026-06-24	7.8

		In addition, SOCKMAP redirect breaks the Tarjan-based GC's assumption that unix_edge.successor is always alive, which is no longer true once skb is redirected, resulting in use-after-free below. [0] Moreover, SOCKMAP does not call scm_stat_del() properly, so unix_show_fdinfo() could report an incorrect fd count. sk_msg_recvmsg() does not support any SCM attributes in the first place. Let's drop all SCM attributes before passing skb to the SOCKMAP layer. [0]: BUG: KASAN: slab-use-after-free in unix_del_edges (net/unix/garbage.c:118 net/unix/garbage.c:181 net/unix/garbage.c:251) Read of size 8 at addr ffff888125362670 by task kworker/56:1/496 CPU: 56 UID: 0 PID: 496 Comm: kworker/56:1 Not tainted 7.0.0-rc7-00263-gb9d8b856689d #3 PREEMPT(lazy) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-debian-1.17.0-1 04/01/2014 Workqueue: events sk_psock_backlog Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:122) print_report (mm/kasan/report.c:379) kasan_report (mm/kasan/report.c:597) unix_del_edges (net/unix/garbage.c:118 net/unix/garbage.c:181 net/unix/garbage.c:251) unix_destroy_fpl (net/unix/garbage.c:317) unix_destruct_scm (./include/net/scm.h:80 ./include/net/scm.h:86 net/unix/af_unix.c:1976) sk_psock_backlog (./include/linux/skbuff.h:?) process_scheduled_works (kernel/workqueue.c:?) worker_thread (kernel/workqueue.c:?) kthread (kernel/kthread.c:438) ret_from_fork (arch/x86/kernel/process.c:164) ret_from_fork_asm (arch/x86/entry/entry_64.S:258) </TASK> Allocated by task 955: kasan_save_track (mm/kasan/common.c:58 mm/kasan/common.c:78) __kasan_slab_alloc (mm/kasan/common.c:369) kmem_cache_alloc_noprof (mm/slub.c:4539) sk_prot_alloc (net/core/sock.c:2240) sk_alloc (net/core/sock.c:2301) unix_create1 (net/unix/af_unix.c:1099) unix_create (net/unix/af_unix.c:1169) __sock_create (net/socket.c:1606) __sys_socketpair (net/socket.c:1811) __x64_sys_socketpair (net/socket.c:1863 net/socket.c:1860 net/socket.c:1860) do_syscall_64 (arch/x86/entry/syscall_64.c:?) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Freed by task 496: kasan_save_track (mm/kasan/common.c:58 mm/kasan/common.c:78) kasan_save_free_info (mm/kasan/generic.c:587) __kasan_slab_free (mm/kasan/common.c:287) kmem_cache_free (mm/slub.c:6165) __sk_destruct (net/core/sock.c:2282 net/core/sock.c:2384) sk_psock_destroy (./include/net/sock.h:?) process_scheduled_works (kernel/workqueue.c:?) worker_thread (kernel/workqueue.c:?) kthread (kernel/kthread.c:438) ret_from_fork (arch/x86/kernel/process.c:164) ret_from_fork_asm (arch/x86/entry/entry_64.S:258)		
CVE-2026-53009	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ice: fix double-free of tx_buf skb If ice_tso() or ice_tx_csum() fail, the error path in ice_xmit_frame_ring() frees the skb, but the 'first' tx_buf still points to it and is marked as valid (ICE_TX_BUF_SKB). 'next_to_use' remains unchanged, so the potential problem will likely fix itself when the next packet is transmitted and the tx_buf gets overwritten. But if there is no next packet and the interface is brought down instead, ice_clean_tx_ring() -> ice_unmap_and_free_tx_buf() will find the tx_buf and free the skb for the second time. The fix is to reset the tx_buf type to ICE_TX_BUF_EMPTY in the error	2026-06-24	7.8

		path, so that ice_unmap_and_free_tx_buf(). Move the initialization of 'first' up, to ensure it's already valid in case we hit the linearization error path. The bug was spotted by AI while I had it looking for something else. It also proposed an initial version of the patch. I reproduced the bug and tested the fix by adding code to inject failures, on a build with KASAN. I looked for similar bugs in related Intel drivers and did not find any.		
CVE-2026-53011	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/sched: taprio: fix use-after-free in advance_sched() on schedule switch In advance_sched(), when should_change_schedules() returns true, switch_schedules() is called to promote the admin schedule to oper. switch_schedules() queues the old oper schedule for RCU freeing via call_rcu(), but 'next' still points into an entry of the old oper schedule. The subsequent 'next->end_time = end_time' and rcu_assign_pointer(q->current_entry, next) are use-after-free. Fix this by selecting 'next' from the new oper schedule immediately after switch_schedules(), and using its pre-calculated end_time. setup_first_end_time() sets the first entry's end_time to base_time + interval when the schedule is installed, so the value is already correct. The deleted 'end_time = sched_base_time(admin)' assignment was also harmful independently: it would overwrite the new first entry's pre-calculated end_time with just base_time.	2026-06-24	7.8
CVE-2026-53016	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - copy IV using skcipher ivsize AF_ALG rfc3686-ctr-aes-ccp requests pass an 8-byte IV to the driver. ccp_aes_complete() restores AES_BLOCK_SIZE bytes into the caller's IV buffer while RFC3686 skciphers expose an 8-byte IV, so the restore overruns the provided buffer. Use crypto_skcipher_ivsize() to copy only the algorithm's IV length.	2026-06-24	7.8
CVE-2026-53020	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: um: Fix potential race condition in TLB sync During the TLB sync, we need to traverse and modify the page table, so we should hold the page table lock. Since full SMP support for threads within the same process is still missing, let's disable the split page table lock for simplicity.	2026-06-24	7.8
CVE-2026-53024	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: greybus: raw: fix use-after-free if write is called after disconnect If a user writes to the chardev after disconnect has been called, the kernel panics with the following trace (with CONFIG_INIT_ON_FREE_DEFAULT_ON=y): BUG: kernel NULL pointer dereference, address: 0000000000000218 ... Call Trace: <TASK> gb_operation_create_common+0x61/0x180 gb_operation_create_flags+0x28/0xa0 gb_operation_sync_timeout+0x6f/0x100 raw_write+0x7b/0xc7 [gb_raw] vfs_write+0xcf/0x420 ? task_mm_cid_work+0x136/0x220 ksys_write+0x63/0xe0 do_syscall_64+0xa4/0x290 entry_SYSCALL_64_after_hwframe+0x77/0x7f Disconnect calls gb_connection_destroy, which ends up freeing the connection object. When gb_operation_sync is called in the write file operations, its gets a freed connection as parameter and the kernel panics. The gb_connection_destroy cannot be moved out of the disconnect function, as the Greybus subsystem expect all connections belonging to a bundle to be destroyed when disconnect returns.	2026-06-24	7.8

		To prevent this bug, use a rw lock to synchronize access between write and disconnect. This guarantees that the write function doesn't try to use a disconnected connection.		
CVE-2026-53025	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: greybus: raw: fix use-after-free on cdev close This addresses a use-after-free bug when a raw bundle is disconnected but its chardev is still opened by an application. When the application releases the cdev, it causes the following panic when init on free is enabled (CONFIG_INIT_ON_FREE_DEFAULT_ON=y): refcount_t: underflow; use-after-free. WARNING: CPU: 0 PID: 139 at lib/refcount.c:28 refcount_warn_saturate+0xd0/0x130 ... Call Trace: <TASK> cdev_put+0x18/0x30 __fput+0x255/0x2a0 __x64_sys_close+0x3d/0x80 do_syscall_64+0xa4/0x290 entry_SYSCALL_64_after_hwframe+0x77/0x7f The cdev is contained in the "gb_raw" structure, which is freed in the disconnect operation. When the cdev is released at a later time, cdev_put gets an address that points to freed memory. To fix this use-after-free, convert the struct device from a pointer to being embedded, that makes the lifetime of the cdev and of this device the same. Then, use cdev_device_add, which guarantees that the device won't be released until all references to the cdev have been released. Finally, delegate the freeing of the structure to the device release function, instead of freeing immediately in the disconnect callback.	2026-06-24	7.8
CVE-2026-53031	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Validate node_id in arena_alloc_pages() arena_alloc_pages() accepts a plain int node_id and forwards it through the entire allocation chain without any bounds checking. Validate node_id before passing it down the allocation chain in arena_alloc_pages().	2026-06-24	7.8

CVE-2026-53033	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		bpf, sockmap: Take state lock for af_unix iter
		
		When a BPF iterator program updates a sockmap, there is a race condition in
		unix_stream_bpf_update_proto() where the `peer` pointer can become stale[1]
		during a state transition TCP_ESTABLISHED -> TCP_CLOSE.
		
		CPU0 bpf CPU1 close
		-------- ----------
		// unix_stream_bpf_update_proto()
		sk_pair = unix_peer(sk)
		if (unlikely(!sk_pair))
		return -EINVAL;
		// unix_release_sock()
		skpair = unix_peer(sk);
		unix_peer(sk) = NULL;
		sock_put(skpair)
		sock_hold(sk_pair) // UaF
		
		More practically, this fix guarantees that the iterator program is
		consistently provided with a unix socket that remains stable during
		iterator execution.
		
		[1]:
		BUG: KASAN: slab-use-after-free in unix_stream_bpf_update_proto+0x155/0x490
		Write of size 4 at addr ffff8881178c9a00 by task test_progs/2231
		Call Trace:
		dump_stack_lvl+0x5d/0x80
		print_report+0x170/0x4f3
		kasan_report+0xe4/0x1c0
		kasan_check_range+0x125/0x200
		unix_stream_bpf_update_proto+0x155/0x490
		sock_map_link+0x71c/0xec0
		sock_map_update_common+0xbc/0x600
		sock_map_update_elem+0x19a/0x1f0
		bpf_prog_bbbf56096cdd4f01_selective_dump_unix+0x20c/0x217
		bpf_iter_run_prog+0x21e/0xae0
		bpf_iter_unix_seq_show+0x1e0/0x2a0
			2026-06-24	7.8

		bpf_seq_read+0x42c/0x10d0 vfs_read+0x171/0xb20 ksys_read+0xff/0x200 do_syscall_64+0xf7/0x5e0 entry_SYSCALL_64_after_hwframe+0x76/0x7e Allocated by task 2236: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 __kasan_slab_alloc+0x63/0x80 kmem_cache_alloc_noprof+0x1d5/0x680 sk_prot_alloc+0x59/0x210 sk_alloc+0x34/0x470 unix_create1+0x86/0x8a0 unix_stream_connect+0x318/0x15b0 __sys_connect+0xfd/0x130 __x64_sys_connect+0x72/0xd0 do_syscall_64+0xf7/0x5e0 entry_SYSCALL_64_after_hwframe+0x76/0x7e Freed by task 2236: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x70 __kasan_slab_free+0x47/0x70 kmem_cache_free+0x11c/0x590 __sk_destruct+0x432/0x6e0 unix_release_sock+0x9b3/0xf60 unix_release+0x8a/0xf0 __sock_release+0xb0/0x270 sock_close+0x18/0x20 __fput+0x36e/0xac0 fput_close_sync+0xe5/0x1a0 __x64_sys_close+0x7d/0xd0 do_syscall_64+0xf7/0x5e0 entry_SYSCALL_64_after_hwframe+0x76/0x7e		
CVE-2026-53036	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf, arm64: Fix off-by-one in check_imm signed range check check_imm(bits, imm) is used in the arm64 BPF JIT to verify that a branch displacement (in arm64 instruction units) fits into the signed N-bit immediate field of a B, B.cond or CBZ/CBNZ encoding before it is handed to the encoder. The macro currently tests for (imm > 0 && imm >> bits) || (imm < 0 && ~imm >> bits) which admits values in [-2^N, 2^N) — effectively a signed (N+1)-bit range. A signed N-bit field only holds [-2^(N-1), 2^(N-1)), so the check admits one extra bit of range on each side. In particular, for check_imm19(), values in [2^18, 2^19) slip past the check but do not fit into the 19-bit signed imm19 field of B.cond. aarch64_insn_encode_immediate() then masks the raw value into the 19-bit field, setting bit 18 (the sign bit) and flipping a forward branch into a backward one. Same class of issue exists for check_imm26() and the B/BL encoding. Shift by (bits - 1) instead of bits so the actual signed N-bit range is enforced.	2026-06-24	7.8

CVE-2026-53050	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		quota: Fix race of dquot_scan_active() with quota deactivation
		
		dquot_scan_active() can race with quota deactivation in
		quota_release_workfn() like:
		
		CPU0 (quota_release_workfn) CPU1 (dquot_scan_active)
		============================ ============================
		spin_lock(&dq_list_lock);
		list_replace_init(
		&releasing_dquots, &rls_head);
		/* dquot X on rls_head,
		dq_count == 0,
		DQ_ACTIVE_B still set */
		spin_unlock(&dq_list_lock);
		synchronize_srcu(&dquot_srcu);
		spin_lock(&dq_list_lock);
		list_for_each_entry(dquot,
		&inuse_list, dq_inuse) {
		/* finds dquot X */
		dquot_active(X) -> true
		atomic_inc(&X->dq_count);
		}
		spin_unlock(&dq_list_lock);
			2026-06-24	7.8

		spin_lock(&dq_list_lock);
		dquot = list_first_entry(&rls_head);
		WARN_ON_ONCE(atomic_read(&dquot->dq_count));
		
		The problem is not only a cosmetic one as under memory pressure the
		caller of dquot_scan_active() can end up working on freed dquot.
		
		Fix the problem by making sure the dquot is removed from releasing list
		when we acquire a reference to it.		

CVE-2026-53054	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Fix VM_BIND UNMAP locking Wrong argument meant that the objs involved in UNMAP ops were not always getting locked. Since _NO_SHARE objs share a common resv with the VM (which is always locked) this would only show up with non-_NO_SHARE BOs. Patchwork: https://patchwork.freedesktop.org/patch/713898/	2026-06-24	7.8
CVE-2026-53062	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm cache policy smq: fix missing locks in invalidating cache blocks In passthrough mode, the policy invalidate_mapping operation is called simultaneously from multiple workers, thus it should be protected by a lock. Otherwise, we might end up with data races on the allocated blocks counter, or even use-after-free issues with internal data structures when doing concurrent writes. Note that the existing FIXME in smq_invalidate_mapping() doesn't affect passthrough mode since migration tasks don't exist there, but would need attention if supporting fast device shrinking via suspend/resume without target reloading. Reproduce steps: 1. Create a cache device consisting of 1024 cache entries dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" dmsetup create cdata --table "0 131072 linear /dev/sdc 8192" dmsetup create corig --table "0 262144 linear /dev/sdc 262144" dd if=/dev/zero of=/dev/mapper/cmeta bs=4k count=1 oflag=direct dmsetup create cache --table "0 262144 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 writethrough smq 0" 2. Populate the cache, and record the number of cached blocks fio --name=populate --filename=/dev/mapper/cache --rw=randwrite --bs=4k \ --size=64m --direct=1 nr_cached=$(dmsetup status cache | awk '{split($7, a, "/"); print a[1]}') 3. Reload the cache into passthrough mode dmsetup suspend cache dmsetup reload cache --table "0 262144 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 passthrough smq 0" dmsetup resume cache 4. Write to the passthrough cache. By setting multiple jobs with I/O size equal to the cache block size, cache blocks are invalidated concurrently from different workers. fio --filename=/dev/mapper/cache --name=test --rw=randwrite --bs=64k \ --direct=1 --numjobs=2 --randrepeat=0 --size=64m 5. Check if demoted matches cached block count. These numbers should match but may differ due to the data race. nr_demoted=$(dmsetup status cache | awk '{print $12}') echo "$nr_cached, $nr_demoted"	2026-06-24	7.8
CVE-2026-53077	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/rds: Restrict use of RDS/IB to the initial network namespace Prevent using RDS/IB in network namespaces other than the initial one. The existing RDS/IB code will not work properly in non-initial network namespaces.	2026-06-24	7.8
CVE-2026-53078	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix same-register dst/src OOB read and pointer leak in sock_ops	2026-06-24	7.8

		When a BPF sock_ops program accesses ctx fields with dst_reg == src_reg, the SOCK_OPS_GET_SK() and SOCK_OPS_GET_FIELD() macros fail to zero the destination register in the !fullsock / !locked_tcp_sock path. Both macros borrow a temporary register to check is_fullsock / is_locked_tcp_sock when dst_reg == src_reg, because dst_reg holds the ctx pointer. When the check is false (e.g., TCP_NEW_SYN_RECV state with a request_sock), dst_reg should be zeroed but is not, leaving the stale ctx pointer: - SOCK_OPS_GET_SK: dst_reg retains the ctx pointer, passes NULL checks as PTR_TO_SOCKET_OR_NULL, and can be used as a bogus socket pointer, leading to stack-out-of-bounds access in helpers like bpf_skc_to_tcp6_sock(). - SOCK_OPS_GET_FIELD: dst_reg retains the ctx pointer which the verifier believes is a SCALAR_VALUE, leaking a kernel pointer. Fix both macros by: - Changing JMP_A(1) to JMP_A(2) in the fullsock path to skip the added instruction. - Adding BPF_MOV64_IMM(si->dst_reg, 0) after the temp register restore in the !fullsock path, placed after the restore because dst_reg == src_reg means we need src_reg intact to read ctx->temp.		
CVE-2026-53081	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Enforce regsafe base id consistency for BPF_ADD_CONST scalars When regsafe() compares two scalar registers that both carry BPF_ADD_CONST, check_scalar_ids() maps their full compound id (aka base | BPF_ADD_CONST flag) as one idmap entry. However, it never verifies that the underlying base ids, that is, with the flag stripped are consistent with existing idmap mappings. This allows construction of two verifier states where the old state has R3 = R2 + 10 (both sharing base id A) while the current state has R3 = R4 + 10 (base id C, unrelated to R2). The idmap creates two independent entries: A->B (for R2) and A|flag->C|flag (for R3), without catching that A->C conflicts with A->B. State pruning then incorrectly succeeds. Fix this by additionally verifying base ID mapping consistency whenever BPF_ADD_CONST is set: after mapping the compound ids, also invoke check_ids() on the base IDs (flag bits stripped). This ensures that if A was already mapped to B from comparing the source register, any ADD_CONST derivative must also derive from B, not an unrelated C.	2026-06-24	7.8
CVE-2026-53085	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: fix mm lifecycle in open-coded task_vma iterator The open-coded task_vma iterator reads task->mm locklessly and acquires mmap_read_trylock() but never calls mmget(). If the task exits concurrently, the mm_struct can be freed as it is not SLAB_TYPESAFE_BY_RCU, resulting in a use-after-free. Safely read task->mm with a trylock on alloc_lock and acquire an mm reference. Drop the reference via bpf_iter_mmput_async() in _destroy() and error paths. bpf_iter_mmput_async() is a local wrapper around mmput_async() with a fallback to mmput() on !CONFIG_MMU. Reject irqs-disabled contexts (including NMI) up front. Operations used by _next() and _destroy() (mmap_read_unlock, bpf_iter_mmput_async) take spinlocks with IRQs disabled (pool->lock, pi_lock). Running from NMI or from a tracepoint that fires with those locks held could deadlock. A trylock on alloc_lock is used instead of the blocking task_lock() (get_task_mm) to avoid a deadlock when a softirq BPF program iterates a task that already holds its alloc_lock on the same CPU.	2026-06-24	7.8
CVE-2026-53090	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix ld_{abs,ind} failure path analysis in subprogs Usage of ld_{abs,ind} instructions got extended into subprogs some time ago via commit 09b28d76eac4 ("bpf: Add abnormal return checks."). These are only allowed in subprograms when the latter are BTF annotated and have scalar return types. The code generator in bpf_gen_ld_abs() has an abnormal exit path (r0=0 +	2026-06-24	7.8

		exit) from legacy cBPF times. While the enforcement is on scalar return types, the verifier must also simulate the path of abnormal exit if the packet data load via ld_{abs,ind} failed. This is currently not the case. Fix it by having the verifier simulate both success and failure paths, and extend it in similar ways as we do for tail calls. The success path (r0=unknown, continue to next insn) is pushed onto stack for later validation and the r0=0 and return to the caller is done on the fall-through side.		
CVE-2026-53092	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix linked reg delta tracking when src_reg == dst_reg Consider the case of rX += rX where src_reg and dst_reg are pointers to the same bpf_reg_state in adjust_reg_min_max_vals(). The latter first modifies the dst_reg in-place, and later in the delta tracking, the subsequent is_reg_const(src_reg)/reg_const_value(src_reg) reads the post-{add,sub} value instead of the original source. This is problematic since it sets an incorrect delta, which sync_linked_regs() then propagates to linked registers, thus creating a verifier-vs-runtime mismatch. Fix it by just skipping this corner case.	2026-06-24	7.8
CVE-2026-53094	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix stale offload->prog pointer after constant blinding When a dev-bound-only BPF program (BPF_F_XDP_DEV_BOUND_ONLY) undergoes JIT compilation with constant blinding enabled (bpf_jit_harden >= 2), bpf_jit_blind_constants() clones the program. The original prog is then freed in bpf_jit_prog_release_other(), which updates aux->prog to point to the surviving clone, but fails to update offload->prog. This leaves offload->prog pointing to the freed original program. When the network namespace is subsequently destroyed, cleanup_net() triggers bpf_dev_bound_netdev_unregister(), which iterates ondev->progs and calls __bpf_prog_offload_destroy(offload->prog). Accessing the freed prog causes a page fault: BUG: unable to handle page fault for address: ffffc900085f1038 Workqueue: netns cleanup_net RIP: 0010:__bpf_prog_offload_destroy+0xc/0x80 Call Trace: __bpf_offload_dev_netdev_unregister+0x257/0x350 bpf_dev_bound_netdev_unregister+0x4a/0x90 unregister_netdevice_many_notify+0x2a2/0x660 ... cleanup_net+0x21a/0x320 The test sequence that triggers this reliably is: 1. Set net.core.bpf_jit_harden=2 (echo 2 > /proc/sys/net/core/bpf_jit_harden) 2. Run xdp_metadata selftest, which creates a dev-bound-only XDP program on a veth inside a netns (./test_progs -t xdp_metadata) 3. cleanup_net -> page fault in __bpf_prog_offload_destroy Dev-bound-only programs are unique in that they have an offload structure but go through the normal JIT path instead of bpf_prog_offload_compile(). This means they are subject to constant blinding's prog clone-and-replace, while also having offload->prog that must stay in sync. Fix this by updating offload->prog in bpf_jit_prog_release_other(), alongside the existing aux->prog update. Both are back-pointers to the prog that must be kept in sync when the prog is replaced.	2026-06-24	7.8
CVE-2026-53096	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Use RCU-safe iteration in dev_map_redirect_multi() SKB path The DEVMAP_HASH branch in dev_map_redirect_multi() uses hlist_for_each_entry_safe() to iterate hash buckets, but this function runs under RCU protection (called from xdp_do_generic_redirect_map() in softirq context). Concurrent writers (__dev_map_hash_update_elem, dev_map_hash_delete_elem) modify the list using RCU primitives (hlist_add_head_rcu, hlist_del_rcu). hlist_for_each_entry_safe() performs plain pointer dereferences without rcu_dereference(), missing the acquire barrier needed to pair with writers' rcu_assign_pointer(). On weakly-ordered architectures (ARM64, POWER), a reader can observe a partially-constructed node. It also defeats CONFIG_PROVE_RCU lockdep validation and KCSAN data-race detection.	2026-06-24	7.8

		Replace with hlist_for_each_entry_rcu() using rcu_read_lock_bh_held() as the lockdep condition, consistent with the rcu_dereference_check() used in the DEVMAP (non-hash) branch of the same functions. Also fix the same incorrect lockdep_is_held(&dtab->index_lock) condition in dev_map_enqueue_multi(), where the lock is not held either.		
CVE-2026-53110	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: s390/bpf: Zero-extend bpf prog return values and kfunc arguments s390x ABI requires callers to zero-extend unsigned arguments and sign-extend signed arguments, and callees to zero-extend unsigned return values and sign-extend signed return values. s390 BPF JIT currently implements only sign extension. Fix this omission and implement zero extension too.	2026-06-24	7.8
CVE-2026-53130	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fs/omfs: reject s_sys_blocksize smaller than OMFS_DIR_START omfs_fill_super() rejects oversized s_sys_blocksize values (> PAGE_SIZE), but it does not reject values smaller than OMFS_DIR_START (0x1b8 = 440). Later, omfs_make_empty() uses sbi->s_sys_blocksize - OMFS_DIR_START as the length argument to memset(). Since s_sys_blocksize is u32, a crafted filesystem image with s_sys_blocksize < OMFS_DIR_START causes an unsigned underflow there, wrapping to a value near 2^32. That drives a ~4 GiB memset() from bh->b_data + OMFS_DIR_START and overwrites kernel memory far beyond the backing block buffer. Add the corresponding lower-bound check alongside the existing upper-bound check in omfs_fill_super(), so that malformed images are rejected during superblock validation before any filesystem data is processed.	2026-06-24	7.8
CVE-2026-13037	google - chrome	Use after free in WebView in Google Chrome on Android prior to 149.0.7827.197 allowed a local attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: High)	2026-06-24	7.8
CVE-2026-53133	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/umem: Fix truncation for block sizes >= 4G When the iommu is used the linearization of the mapping can give a single block that is very large split across multiple SG entries. When __rdma_block_iter_next() reassembles the split SG entries it is overflowing the 32 bit stack values and computed the wrong DMA addresses for blocks after the truncation. Use the right types to hold DMA addresses.	2026-06-25	7.8
CVE-2026-53145	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/gem: Try to fix change_handle ioctl, attempt 4 [airlied: just added some comments on how to reenable] On-list because the cat is out of the bag and we're clearly not good enough to figure this out in private. The story thus far: 5e28b7b94408 ("drm: Set old handle to NULL before prime swap in change_handle") tried to fix a race condition between the gem_close and gem_change_handle ioctls, but got a few things wrong: - There's a confusion with the local variable handle, which is actually the new handle, and so the two-stage trick was actually applied to the wrong idr slot. 7164d78559b0 ("drm/gem: fix race between change_handle and handle_delete") tried to fix that by adding yet another code block, but forgot to add the error handling. Which meant we now have two paths, both kinda wrong. - dc366607c41c ("drm: Replace old pointer to new idr") tried to apply another fix, but inconsistently, again because of the handle confusion - this would be the right fix (kinda, somewhat, it's a mess) if we'd do the two-stage approach for the new handle. Except that wasn't the intent of the original fix. We also didn't have an igt merged for the original ioctl, which is a big no-go. This was attempted to address off-list in the original bugfix, and amd QA people claimed the bug was fixed now. Very clearly that's not the case. Here's my attempt to sort this out: - Rename the local variable to new_handle, the old aliasing with	2026-06-25	7.8

		args->handle is just too dangerously confusing. - Merge the gem obj lookup with the two-stage idr_replace so that we avoid getting ourselves confused there. - This means we don't have a surplus temporary reference anymore, only an inherited from the idr. A concurrent gem_close on the new_handle could steal that. Fix that with the same two-stage approach create_tail uses. This is a bit overkill as documented in the comment, but I also don't trust my ability to understand this all correctly, so go with the established pattern we have from other ioctls instead for maximum paranoia. - Adjust error paths. I've tried to make the error and success paths common, because they are identical except for which handle is removed and on which we call idr_replace to (re)install the object again. But that made things messier to read, so I've left it at the more verbose version, which unfortunately hides the symmetry in the entire code flow a bit. - While at it, also replace the 7 space indent with 1 tab. And finally, because I flat out don't trust my abilities here at all anymore: - Disable the ioctl until we have the igt situation and everything else sorted out on-list and with full consensus. v2: Sashiko noticed that I didn't handle the error path for idr_replace correctly, it must be checked with IS_ERR_OR_NULL like in gem_handle_delete. So yeah, definitely should just the existing paths 1:1 because this is endless amounts of tricky. Also add the Fixes: line for the original ioctl, I forgot that too.		
CVE-2026-53153	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/list_lru: drain before clearing xarray entry on reparent memcg_reparent_list_lrus() clears the dying memcg's xarray entry with xas_store(&xas, NULL) before reparenting its per-node lists into the parent. This opens a window where a concurrent list_lru_del() arriving for the dying memcg sees xa_load() == NULL, walks to the parent in lock_list_lru_of_memcg(), takes the parent's per-node lock, and calls list_del_init() on an item still physically linked on the dying memcg's list. If another in-flight thread holds the dying memcg's per-node lock at the same moment (another list_lru_del, or a list_lru_walk_one running an isolate callback), both threads modify ->next/->prev pointers on the same physical list under different locks. Adjacent items can corrupt each other's links. Fix it by reversing the order: reparent each per-node list and mark the child's list lru dead and then clear the xarray entry. Any concurrent list_lru op that finds the still-set xarray entry either takes the dying memcg's per-node lock (synchronizing with the drain) or sees LONG_MIN and walks to the parent, where the items now live.	2026-06-25	7.8
CVE-2026-53160	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: fix use-after-free race in fastrpc_map_create fastrpc_map_lookup returns a raw pointer after releasing fl->lock. The caller fastrpc_map_create then calls fastrpc_map_get (kref_get_unless_zero) on this unprotected pointer. A concurrent MEM_UNMAP can free the map between the lock release and the kref operation, resulting in a use-after-free on the freed slab object. Restore the take_ref parameter to fastrpc_map_lookup so the reference is acquired atomically under fl->lock before the pointer is exposed to the caller.	2026-06-25	7.8
CVE-2026-53161	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: fix use-after-free of fastrpc_user in workqueue context There is a race between fastrpc_device_release() and the workqueue that processes DSP responses. When the user closes the file descriptor, fastrpc_device_release() frees the fastrpc_user structure. Concurrently, an in-flight DSP invocation can complete and fastrpc_rpmsg_callback() schedules context cleanup via schedule_work(&ctx->put_work). If the	2026-06-25	7.8

		workqueue runs fastrpc_context_free() in parallel with or after fastrpc_device_release() has freed the user structure, it dereferences the freed fastrpc_user. Depending on the state of the context at the time of the race, any one of the following accesses can be hit: 1. fastrpc_buf_free() calls fastrpc_ipa_to_dma_addr(buf->fl->cctx, ...) to strip the SID bits from the stored IOVA before passing the physical address to dma_free_coherent(). 2. fastrpc_free_map() reads map->fl->cctx->vmperms[0].vmid to reconstruct the source permission bitmask needed for the qcom_scm_assign_mem() call that returns memory from the DSP VM back to HLOS. 3. fastrpc_free_map() acquires map->fl->lock to safely remove the map node from the fl->maps list. The resulting use-after-free manifests as: pc : fastrpc_buf_free+0x38/0x80 [fastrpc] lr : fastrpc_context_free+0xa8/0x1b0 [fastrpc] fastrpc_context_free+0xa8/0x1b0 [fastrpc] fastrpc_context_put_wq+0x78/0xa0 [fastrpc] process_one_work+0x180/0x450 worker_thread+0x26c/0x388 Add kref-based reference counting to fastrpc_user. Have each invoke context take a reference on the user at allocation time and release it when the context is freed. Release the initial reference in fastrpc_device_release() at file close. Move the teardown of the user structure — freeing pending contexts, maps, mmaps, and the channel context reference — into the kref release callback fastrpc_user_free(), so that it runs only when the last reference is dropped, regardless of whether that happens at device close or after the final in-flight context completes.		
CVE-2026-53162	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: memcg: use round-robin victim selection in refill_stock Harry Yoo reported that get_random_u32_below() is not safe to call in the nmi context and memcg charge draining can happen in nmi context. More specifically get_random_u32_below() is neither reentrant- nor NMI-safe: it acquires a per-cpu local_lock via local_lock_irqsave() on the batched_entropy_u32 state. An NMI that lands on a CPU mid-update of the ChaCha batch state and recurses into the random subsystem would corrupt that state. The memcg_stock local_trylock prevents re-entry on the percpu stock itself, but cannot protect an unrelated subsystem's per-cpu lock. Replace the random pick with a per-cpu round-robin counter stored in memcg_stock_pcp and serialized by the same local_trylock that already guards cached[] and nr_pages[]. No atomics, no random calls, no extra locks needed.	2026-06-25	7.8
CVE-2026-53172	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ethosu: fix IFM region index out-of-bounds in command stream parser NPU_SET_IFM_REGION extracts the region index with param & 0x7f, giving a maximum value of 127. However region_size[] and output_region[] in struct ethosu_validated_cmdstream_info are both sized to NPU_BASEP_REGION_MAX (8), giving valid indices [0..7]. Every other region assignment in the same switch uses param & 0x7: NPU_SET_OFM_REGION: st.ofm.region = param & 0x7; NPU_SET_IFM2_REGION: st.ifm2.region = param & 0x7; NPU_SET_WEIGHT_REGION: st.weight[0].region = param & 0x7; NPU_SET_SCALE_REGION: st.scale[0].region = param & 0x7; The 0x7f mask on IFM is inconsistent and appears to be a typo. feat_matrix_length() and calc_sizes() use the region index directly as an array subscript into the kzalloc'd info struct: info->region_size[fm->region] = max(...); A userspace caller supplying NPU_SET_IFM_REGION with param > 7 causes a write up to 127*8 = 1016 bytes past the start of region_size[], corrupting adjacent kernel heap data. Fix by applying the same & 0x7 mask used by all other region assignments.	2026-06-25	7.8

CVE-2026-53173	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		accel/ethosu: fix OOB write in ethosu_gem_cmdstream_copy_and_validate()
		
		The command stream parsing loop increments the index variable a second time when a 64-bit command word is encountered (bit 14 set), but does not re-check the loop bound before writing the second word:
		
		`for (i = 0; i < size / 4; i++) {`
		`  bocmds[i] = cmds[0];`
		`  if (cmd & 0x4000) {`
		`  i++;`
		`  bocmds[i] = cmds[1];  /* unchecked */`
		`  }`
		`}`
		
		The buffer bocmds is backed by a DMA allocation of exactly size bytes from drm_gem_dma_create(ddev, size), giving valid indices [0, size/4-1].
		
		When i == size/4 - 1 on entry to an iteration and bit 14 of cmds[0] is set, bocmds[size/4-1] is written in bounds, i is then incremented to size/4, and bocmds[size/4] writes four bytes past the end of the allocation.
		
		Userspace controls both the buffer contents and the size argument via the ioctl, making this a userspace-triggerable heap out-of-bounds write.
		
		Fix by checking the incremented index against the buffer bound before the second write and returning -EINVAL if the buffer is too small to contain the extended command.
			2026-06-25	7.8

CVE-2026-53174	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ovl: keep err zero after successful ovl_cache_get()
		
		ovl_iterate_merged() stores PTR_ERR(cache) in err before checking IS_ERR(cache). On success err holds the truncated cache pointer and can be returned as a bogus non-zero error.
		
		The syzbot reproducer reaches this through overlay-on-overlay readdir:
		
		`getdents64`
		`  iterate_dir(outer overlay file)`
		`  ovl_iterate_merged()`
		`  ovl_cache_get()`
		`  ovl_dir_read_merged()`
		`  ovl_dir_read()`
		`  iterate_dir(inner overlay file)`
		`  ovl_iterate_merged()`
		
		Only compute PTR_ERR(cache) on the error path.
			2026-06-25	7.8

CVE-2026-53182	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: reject oversized EMA RNR lists nl80211_parse_rnr_elems() stores the parsed element count in a u8-backed cfg80211_rnr_elems::cnt field and uses that count to size the flexible array allocation. Reject nested NL80211_ATTR_EMA_RNR_ELEMS input once the count reaches 255, before incrementing it again. This keeps the parser aligned with the data structure it fills and matches the existing bound check used by nl80211_parse_mbssid_elems().	2026-06-25	7.8
CVE-2026-53185	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: zram: fix use-after-free in zram_bvec_write_partial() zram_read_page() picks the sync or async backing device read path based on whether the parent bio is NULL. zram_bvec_write_partial() passes its parent bio down, so for ZRAM_WB slots the read is dispatched asynchronously and zram_read_page() returns 0 while the bio is still in flight. The caller then runs memcpy_from_bvec(), zram_write_page() and __free_page() on the buffer, leaving the async read to write into a freed page. zram_bvec_read_partial() was switched to NULL in commit 4e3c87b9421d ("zram: fix synchronous reads") for the same reason; the write_partial counterpart was missed.	2026-06-25	7.8
CVE-2026-53189	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: update file PMD counter before folio_put()	2026-06-25	7.8

		__split_huge_pmd_locked() updates the file/shmem RSS counter after dropping the PMD mapping's folio reference. If folio_put() drops the last reference, mm_counter_file() can later read freed folio state via folio_test_swapbacked(). Move the counter update before folio_put().		
CVE-2026-53191	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring/net: inherit IORING_CQE_F_BUF_MORE across bundle recv retries When a bundle recv retries inside io_recv_finish(), the merge logic OR the saved cflags from the previous iteration with the cflags returned by the new iteration: cflags = req->cqe.flags | (cflags & CQE_F_MASK); Bits listed in CQE_F_MASK are inherited from the new iteration, and all other bits (notably IORING_CQE_F_BUFFER and the buffer ID) come from the saved cflags. Before this change CQE_F_MASK covered only IORING_CQE_F_SOCK_NONEMPTY and IORING_CQE_F_MORE. When using provided buffer rings (IOU_PBUF_RING_INC) with incremental mode, and bundle recv, io_kbuf_inc_commit() can leave the head ring entry partially consumed, __io_put_kbufs() then sets IORING_CQE_F_BUF_MORE on the returned cflags so userspace knows the buffer ID will be reused for subsequent completions. Because IORING_CQE_F_BUF_MORE was not in CQE_F_MASK, the merge above silently dropped it whenever the final retry iteration partially consumed the buffer, and the subsequent req->cqe.flags = cflags & ~CQE_F_MASK save would have left a stale IORING_CQE_F_BUF_MORE in the carried-over cflags had one been present. Userspace would then wrongfully advance it ring head past an entry the kernel still uses. Add IORING_CQE_F_BUF_MORE to CQE_F_MASK so it is both inherited from the new iteration into the user-visible CQE and stripped from the saved cflags between iterations.	2026-06-25	7.8
CVE-2026-53192	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: timer: Fix UAF at snd_timer_user_params() At releasing a timer object, e.g. when a userspace timer (CONFIG_SND_UTIMER) gets closed and snd_timer_free() is called, it tries to detach the timer instances and release the resources. However, it's still possible that other in-flight tasks are holding the timer instance where the to-be-deleted timer object is associated, and this may lead to racy accesses. Fortunately, most of ioctls dealing with the timer instance list already have the protection with register_mutex, and this also avoids such races. But, SNDRV_TIMER_IOCTL_PARAMS isn't protected, hence the concurrent ioctl may lead to use-after-free. This patch just adds the guard with register_mutex to protect snd_timer_user_params() for covering the code path as a quick workaround. It's no hot-path but rather a rarely issued ioctl, so the performance penalty doesn't matter.	2026-06-25	7.8
CVE-2026-53193	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: timer: Forcibly close timer instances at closing When snd_timer object is freed via snd_timer_free() and still pending snd_timer_instance objects are assigned to the timer object, it tries to unlink all instances and just set NULL to each ti->timer, then releases the resources immediately. The problem is, however, when there are slave timer instances that are associated with a master instance linked to this timer: namely, those slave instances still point to the freed timer object although the master instance is unlinked, which may lead to user-after-free. The bug can be easily triggered particularly when a new userspace-driven timers (CONFIG_SND_UTIMER) is involved, since it can create and delete the timer object via a simple file open/close, while the other applications may keep accessing to that timer. This patch is an attempt to paper over the problem above: now instead of just unlinking, call snd_timer_close[_locked]() forcibly for each pending timer instance, so that all assigned slave timer instances are properly detached, too. Since snd_timer_close() might be called later by the driver that created that instance, the check of SNDRV_TIMER_IFLG_DEAD is added at the beginning, too.	2026-06-25	7.8
CVE-2026-53194	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2026-06-25	7.8

		USB: serial: kl5kusb105: fix bulk-out buffer overflow klsi_105_prepare_write_buffer() is called by the generic write path with the bulk-out buffer and its size (bulk_out_size, 64 bytes). It stores a two-byte length header at the start of the buffer and copies the payload from the write fifo starting at buf + KLSI_HDR_LEN, but passes the full buffer size as the number of bytes to copy: count = kfifo_out_locked(&port->write_fifo, buf + KLSI_HDR_LEN, size, &port->lock); When the fifo holds at least size bytes, size bytes are copied starting two bytes into the size-byte buffer, writing KLSI_HDR_LEN bytes past its end. Copy at most size - KLSI_HDR_LEN bytes instead, leaving room for the header as safe_serial already does. Writing bulk_out_size or more bytes to the tty triggers a slab out-of-bounds write, observed with KASAN by emulating the device with dummy_hcd and raw-gadget: BUG: KASAN: slab-out-of-bounds in kfifo_copy_out+0x83/0xc0 Write of size 64 at addr ffff888112c62202 by task python3 kfifo_copy_out klsi_105_prepare_write_buffer [kl5kusb105] usb_serial_generic_write_start [usbserial] Allocated by task 139: usb_serial_probe [usbserial] The buggy address is located 2 bytes inside of allocated 64-byte region The out-of-bounds write no longer occurs with this change applied.		
CVE-2026-53201	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/xe: Skip exec queue schedule toggle if queue is idle during suspend" This reverts commit 8533051ce92015e9cc6f75e0d52119b9d91610b6. The idle-skip optimization bypasses GuC suspend, so the GPU may not perform the context switch that flushes TLB entries for invalidated userptr VMAs. In LR/preempt-fence VM mode, this can lead to missed TLB invalidation and page faults during userptr invalidation tests. Restore unconditional schedule toggling on suspend so the context-switch TLB flush is always performed. This optimization will be reintroduced with a fix that does not skip suspend in LR/preempt-fence VM mode. (cherry picked from commit 6a1e7934d9a6cf46aecae00a99c2603d1295e170)	2026-06-25	7.8
CVE-2026-53202	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix signed integer truncation in IPC receive Fix potential buffer overflow where firmware-supplied data_size is cast to signed int before being used in min_t(). Large unsigned values (>= 0x80000000) become negative, causing unsigned wraparound and oversized memcpy operations that can overflow the stack buffer. Change min_t(int, ...) to min() as both values are unsigned and can be handled by min() without explicit cast.	2026-06-25	7.8
CVE-2026-53209	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: reject oversized Broadcast Announcement prepend Existing advertising instances can already hold the maximum extended advertising payload. When hci_adv_bcast_annoucement() prepends the Broadcast Announcement service data to that payload, the combined data may no longer fit in the temporary buffer used to rebuild the advertising data. Reject that case before copying the existing payload and report the failure through the device log. This keeps the existing advertising data intact and avoids overrunning the temporary buffer.	2026-06-25	7.8
CVE-2026-53212	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_tunnel: fix use-after-free on object destroy nft_tunnel_obj_destroy() calls metadata_dst_free() which directly kfree()s the metadata_dst, ignoring the dst_entry refcount. Packets that took a reference via dst_hold() in nft_tunnel_obj_eval() and are still queued (e.g. in a netem qdisc) are left with a dangling pointer. When these packets are eventually dequeued, dst_release()	2026-06-25	7.8

		operates on freed memory. Replace metadata_dst_free() with dst_release() so the metadata_dst is freed only after all references are dropped. The dst subsystem already handles metadata_dst cleanup in dst_destroy() when DST_METADATA is set.		

CVE-2026-53239	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		xfrm: policy: fix use-after-free on inexact bin in xfrm_policy_bysel_ctx()
		
		Fix the race by pruning the bin while still holding xfrm_policy_lock, before dropping it. Use __xfrm_policy_inexact_prune_bin() directly since the lock is already held. The wrapper xfrm_policy_inexact_prune_bin() becomes unused and is removed.
		
		Race:
		
		CPU0 (XFRM_MSG_DELPOLICY) CPU1 (XFRM_MSG_NEWSPDINFO)
		========== ==========
		xfrm_policy_bysel_ctx():
		spin_lock_bh(xfrm_policy_lock)
		bin = xfrm_policy_inexact_lookup()
		__xfrm_policy_unlink(pol)
		spin_unlock_bh(xfrm_policy_lock)
		xfrm_policy_kill(ret)
		// wide window, lock not held
		xfrm_hash_rebuild():
		spin_lock_bh(xfrm_policy_lock)
		__xfrm_policy_inexact_flush():
		kfree_rcu(bin) // bin freed
		spin_unlock_bh(xfrm_policy_lock)
		xfrm_policy_inexact_prune_bin(bin)
		// UAF: bin is freed
			2026-06-25	7.8

CVE-2026-53242	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: PCM: Fix wait queue list corruption in snd_pcm_drain() on linked streams snd_pcm_drain() uses init_waitqueue_entry which does not clear entry.prev/next, and add_wait_queue with a conditional remove_wait_queue that is skipped when to_check is no longer in the group after concurrent UNLINK. The orphaned wait entry remains on the unlinked substream sleep queue. On the next drain iteration, add_wait_queue adds the entry to a new queue while still linked on the old one, corrupting both lists. A subsequent wake_up dereferences NULL at the func pointer (mapped from the spinlock at offset 0 of the misinterpreted wait_queue_head_t), causing a kernel panic. Replace init_waitqueue_entry/add_wait_queue/conditional remove_wait_queue with init_wait_entry/prepare_to_wait/ finish_wait. init_wait_entry clears prev/next via INIT_LIST_HEAD on each iteration and sets autoremove_wake_function which auto-removes the entry on wake-up. finish_wait safely handles both the already-removed and still-queued cases.	2026-06-25	7.8
CVE-2026-53250	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xsk: cache csum_start/csum_offset to fix TOCTOU in xsk_skb_metadata() The TX metadata area resides in the UMEM buffer which is memory-mapped and concurrently writable by userspace. In xsk_skb_metadata(), csum_start and csum_offset are read from shared memory for bounds validation, then read again for skb assignment. A malicious userspace application can race to overwrite these values between the two reads, bypassing the bounds check and causing out-of-bounds memory access during checksum computation in the transmit path. Fix this by reading csum_start and csum_offset into local variables once, then using the local copies for both validation and assignment. Note that other metadata fields (flags, launch_time) and the cached csum fields may be mutually inconsistent due to concurrent userspace writes, but this is benign: the only security-critical invariant is that each field's validated value is the same one used, which local caching guarantees.	2026-06-25	7.8
CVE-2026-53259	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: anycast: insert aca into global hash under idev->lock syzbot reported a splat [1]: a slab-use-after-free in ipv6_chk_acast_addr(), which walks the global inet6_acaddr_lst[] hash	2026-06-25	7.8

		under RCU and dereferences a struct ifacaddr6 that has already been freed while still linked in the hash, so a later reader walks into a dangling node. In __ipv6_dev_ac_inc() the aca is allocated with refcount 1, then aca_get() bumps it to 2 to keep it alive across the unlocked region. It is published to idev->ac_list under idev->lock, but ipv6_add_acaddr_hash() runs after write_unlock_bh(). A concurrent teardown (ipv6_ac_destroy_dev() from addrconf_ifdown(), under RTNL) can slip into that window: CPU0 __ipv6_dev_ac_inc CPU1 ipv6_ac_destroy_dev (RTNL) ---------- ---------- aca_alloc() refcnt 1 aca_get() refcnt 2 write_lock_bh(idev->lock) add aca to ac_list write_unlock_bh(idev->lock) write_lock_bh(idev->lock) pull aca off ac_list write_unlock_bh(idev->lock) ipv6_del_acaddr_hash(aca) hlist_del_init_rcu() is a no-op, aca is not in the hash yet aca_put() refcnt 2->1 ipv6_add_acaddr_hash(aca) aca now inserted into the hash aca_put() refcnt 1->0 call_rcu(aca_free_rcu) -> kfree(aca) The hash removal becomes a no-op because the insertion has not happened yet, so once CPU0 inserts and drops the last reference, the aca is freed while still linked in inet6_acaddr_lst[], and readers dereference freed memory after the slab slot is reused. This window opened once RTNL stopped serializing the join path against device teardown. Move ipv6_add_acaddr_hash() inside the idev->lock section so the ac_list and hash insertions are atomic with respect to teardown: a racing remover now either misses the aca entirely or finds it in both lists. acaddr_hash_lock is now nested under idev->lock, which is acquired in softirq context, so switch all acaddr_hash_lock sites to spin_lock_bh() to avoid the irq lock inversion reported in [2]. [1] https://syzkaller.appspot.com/bug?extid=a01df04303c131efbf3a [2] https://lore.kernel.org/netdev/6a194ef7.ba3b1513.1890b4.0000.GAE@google.com/		
CVE-2026-53262	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: l2tp: pppol2tp: hold reference to session in pppol2tp_ioctl() pppol2tp_ioctl() read sock->sk->sk_user_data directly without any locks or reference counting. If a controllable sleep was induced during copy_from_user() (e.g. via a userfaultfd page fault sleep), a concurrent socket close could trigger pppol2tp_session_close() asynchronously. This frees the l2tp_session structure via the l2tp_session_del_work workqueue. Upon resuming, the ioctl thread dereferences the stale session pointer, resulting in a Use-After-Free (UAF). Fix this by securely fetching the session reference using the RCU-safe, refcounted helper pppol2tp_sock_to_session(sk) on entry. This locks the session's refcount across the sleep. We structured the function to exit via standard err breaks, guaranteeing that l2tp_session_put() is cleanly called on all return paths to drop the reference. To preserve existing behavior we validate the session and its magic signature only for the specific L2TP commands that require it. This ensures that generic/unknown ioctls called on an unconnected socket still return -ENOIOCTLCMD and correctly fall back to generic handlers (e.g. in sock_do_ioctl()).	2026-06-25	7.8
CVE-2026-53264	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_api: use RCU with deferred freeing for action lifecycle When NEWTFILTER and DELFILTER are run concurrently it is possible to create a race with an associated action. Let's illustrate with CPU0 running NEWTFILTER and CPU1 running DELFILTER: 0: mutex_lock() <-- holds the idr lock 0: rcu_read_lock()	2026-06-25	7.8

		0: p = idr_find(idr, index) <-- action p is valid (RCU protects IDR) 0: mutex_unlock() <-- releases the idr lock 1: refcount_dec_and_mutex_lock() <-- refcnt 1->0, mutex held 1: idr_remove(idr, index) <-- Action removed from IDR 1: mutex_unlock() <-- mutex released allowing us to delete the action 1: tcf_action_cleanup(p); kfree(p) <-- Kfrees p immediately, no deferral 0: refcount_inc_not_zero(&p->tcfa_refcnt) <-- ouch, UAF p points to freed memory This patch fixes the race condition between NEWTFILTER and DELFILTER by adding struct rcu_head to tc_action used in the deferral and introducing a call_rcu() in the delete path to defer the final kfree(). Note: this is a revert of commit d7fb60b9cafb ("net_sched: get rid of tcfa_rcu") but also modernization/simplification to directly use kfree_rcu(). Let's illustrate the new restored code path: 0: rcu_read_lock() 1: refcount_dec_and_mutex_lock() <-- refcnt 1->0, mutex held 1: idr_remove(idr, index) 1: mutex_unlock() 1: call_rcu(&p->tcfa_rcu, tcf_action_rcu_free) <-- defer kfree after grace period 0: p = idr_find(idr, index) 0: refcount_inc_not_zero(&p->tcfa_refcnt) <-- fails, refcnt already 0 1: rcu_read_unlock() <-- release so freeing can run after grace period After CPU1 calls idr_remove(), the object is no longer reachable through the IDR. CPU0's subsequent idr_find() will return NULL, and even if it still held a stale pointer, the immediate kfree() is now deferred until after the RCU grace period, so no UAF can occur.		
CVE-2026-53265	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm cache policy smq: check allocation under invalidate lock commit 2d1f7b65f5de ("dm cache policy smq: fix missing locks in invalidating cache blocks") added mq->lock around the destructive part of smq_invalidate_mapping(), but left the e->allocated check outside the critical section. That leaves a check-then-act race. Two concurrent invalidators can both observe e->allocated as true before either of them takes mq->lock. The first invalidator that acquires the lock removes the entry from the queues and hash table and then calls free_entry(), which clears e->allocated and puts the entry back on the free list. The second invalidator can then acquire mq->lock and continue with the stale result of the unlocked check. This can corrupt the SMQ queues or hash table by deleting an entry that is no longer on those structures. It can also hit the allocation check in free_entry() when the same entry is freed again. Move the allocation check under mq->lock so the predicate and the destructive operations are serialized by the same lock.	2026-06-25	7.8

CVE-2026-53267	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		netfilter: nft_ct: bail out on template ct in get eval
		
		I noticed this issue while looking at a historic syzbot report [1].
		
		A rule like the one below is enough to trigger the bug:
		
		table ip t {
		chain pre {
		type filter hook prerouting priority raw;
		ct zone set 1
		ct original saddr 1.2.3.4 accept
		}
		}
		
		The first expression attaches a per-cpu template ct via
		nft_ct_set_zone_eval() (nf_ct_tmpl_alloc -> kzalloc, tuple is all
		zero, nf_ct_l3num(ct) == 0). The next expression then calls
		nft_ct_get_eval() on the same skb, treats the template as a real ct
		and hits the 16-byte memcpy path. With dreg at NFT_REG32_15 this
		overflows past struct nft_regs on the kernel stack; with smaller
		dreg values it silently clobbers adjacent registers.
		
		Reject template ct at the eval entry and in nft_ct_get_fast_eval(),
		mirroring the check nft_ct_set_eval() already has. Additionally,
		bound the address copy in NFT_CT_SRC / NFT_CT_DST by priv->len
		instead of by nf_ct_l3num(ct): nf_ct_get_tuple() zeroes the tuple
			2026-06-25	7.8

		before pkt_to_tuple() fills in only the protocol-relevant leading bytes, so the trailing bytes of tuple->{src,dst}.u3.all are well-defined zero. priv->len is validated at rule load, so the copy size is now bounded by the destination register rather than by an untrusted field on the conntrack. [1]: https://syzkaller.appspot.com/bug?id=389cf09cb72926114fce90dc85a2c3231dcb647c		
CVE-2026-53270	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipvs: clear the svc scheduler ptr early on edit ip_vs_edit_service() while unbinding the old scheduler clears the svc->scheduler ptr after the scheduler module initiates RCU callbacks. This can cause packets to use the old scheduler at the time when svc->sched_data is already freed after RCU grace period. Fix it by clearing the ptr early in ip_vs_unbind_scheduler(), before the done_service method schedules any RCU callbacks. Also, if the new scheduler fails to initialize when replacing the old scheduler, try to restore the old scheduler while still returning the error code.	2026-06-25	7.8
CVE-2026-53273	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tee: optee: prevent use-after-free when the client exits before the supplicant Commit 70b0d6b0a199 ("tee: optee: Fix supplicant wait loop") made the client wait as killable so it can be interrupted during shutdown or after a supplicant crash. This changes the original lifetime expectations: the client task can now terminate while the supplicant is still processing its request. If the client exits first it removes the request from its queue and kfree()s it, while the request ID remains in supp->idr. A subsequent lookup on the supplicant path then dereferences freed memory, leading to a use-after-free. Serialise access to the request with supp->mutex: * Hold supp->mutex in optee_supp_recv() and optee_supp_send() while looking up and touching the request. * Let optee_supp_thrd_req() notice that the client has terminated and signal optee_supp_send() accordingly. With these changes the request cannot be freed while the supplicant still has a reference, eliminating the race.	2026-06-25	7.8

CVE-2026-53276	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		Bluetooth: ISO: Fix a use-after-free of the hci_conn pointer
		
		In iso_sock_rebind_bc(), the bis pointer is cached, then the socket lock is dropped:
		bis = iso_pi(sk)->conn->hcon;
		/* Release the socket before lookups since that requires hci_dev_lock
		* which shall not be acquired while holding sock_lock for proper
		* ordering.
		*/
		release_sock(sk);
		hci_dev_lock(bis->hdev);
		
		During the unlocked window, could a concurrent close() destroy the connection and free the bis structure, causing hci_dev_lock(bis->hdev) to access memory after it is freed, fix this by using the hdev reference which was safely acquired via iso_conn_get_hdev().
			2026-06-25	7.8

CVE-2026-46733	dell - display_and_peripheral_manager	Dell Display and Peripheral Manager (DDPM Windows), versions prior to 2.3, contain an Improper Access Control vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Code execution.	2026-06-25	7.8
CVE-2026-46735	dell - Display and Peripheral Manager	Dell Display and Peripheral Manager (DDPM Mac), versions prior to 2.3, contain an Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Command execution.	2026-06-25	7.8
CVE-2026-53290	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/eustall: Fix drm_dev_put called before stream disable in close In xe_eu_stall_stream_close(), drm_dev_put() is called before the stream is disabled and its resources are freed. If this drops the last reference, the device structures could be freed while the subsequent cleanup code still accesses them, leading to a use-after-free.	2026-06-26	7.8

		Fix this by moving drm_dev_put() after all device accesses are complete. This matches the ordering in xe_oa_release(). (cherry picked from commit 35aff528f7297e949e5e19c9cd7fd748cf1cf21c)		
CVE-2026-53300	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: enetc: fix NTMP DMA use-after-free issue The AI-generated review reported a potential DMA use-after-free issue [1]. If netc_xmit_ntmp_cmd() times out and returns an error, the pending command is not explicitly aborted, while ntmp_free_data_mem() unconditionally frees the DMA buffer. If the buffer has already been reallocated elsewhere, this may lead to silent memory corruption. Because the hardware eventually processes the pending command and perform a DMA write of the response to the physical address of the freed buffer. To resolve this issue, this patch does the following modifications: 1. Convert cbdr->ring_lock from a spinlock to a mutex The lock was originally a spinlock in case NTMP operations might be invoked from atomic context. After downstream support for all NTMP tables, no such usage has materialized. A mutex lock is now required because the driver now needs to reclaim used BDs and release associated DMA memory within the lock's context, while dma_free_coherent() might sleep. 2. Introduce software command BD (struct netc_swcbd) The hardware write-back overwrites the addr and len fields of the BD, so the driver cannot rely on the hardware BD to free the associated DMA memory. The driver now maintains a software shadow BD storing the DMA buffer pointer, DMA address, and size. And netc_xmit_ntmp_cmd() only reclaims older BDs when the number of used BDs reaches NETC_CBDR_CLEAN_WORK (16). The software BD enables correct DMA memory release. With this, struct ntmp_dma_buf and ntmp_free_data_mem() are no longer needed and are removed. 3. Require callers to hold ring_lock across netc_xmit_ntmp_cmd() netc_xmit_ntmp_cmd() releases the ring_lock before the caller finishes consuming the response. At this point, if a concurrent thread submits a new command, it may trigger ntmp_clean_cbdr() and free the DMA buffer while it is still in use. Move ring_lock ownership to the caller to ensure the response buffer cannot be reclaimed prematurely. So the helpers ntmp_select_and_lock_cbdr() and ntmp_unlock_cbdr() are added. These changes eliminate the DMA use-after-free condition and ensure safe and consistent BD reclamation and DMA buffer lifecycle management.	2026-06-26	7.8
CVE-2026-42129	grafana - loki_datasource	The Loki datasource plugin's callResource handler contains a path traversal vulnerability. An authenticated Viewer-role user can escape the plugin's resource sandbox and access administrative Loki endpoints (e.g. /config, /services, /ready) to extract sensitive backend configuration and internal service information.	2026-06-22	7.7
CVE-2026-9099	redhat - multiple products	A flaw was found in Keycloak. A missing authorization check in the GroupResource.addChild() endpoint within the Admin REST API allows an authenticated user with limited administrative privileges to reparent any existing group. When Fine-Grained Admin Permissions v2 (FGAPv2) is enabled, an attacker with management rights over a single low-privilege group can reparent a highly privileged group (such as one possessing the realm-admin role) under their managed group. Because group permissions follow a hierarchical structure, this action unauthorizedly grants the attacker management and password-reset capabilities over the members of the targeted privileged group. An attacker can exploit this to reset an administrator's password, compromise the account, and achieve a full realm takeover, leading to a complete compromise of confidentiality, integrity, and availability.	2026-06-25	7.7
CVE-2026-11998	google - AngularJS	A flaw in AngularJS' Strict Contextual Escaping (SCE) logic allows bypassing certain SCE policies for resource URLs and can lead to arbitrary JavaScript execution within the context of the victim's browser session. SCE's purpose is to ensure that only trusted or safe values are used in certain security-sensitive contexts, such as resource URLs, including URLs that define executable JavaScript scripts, '<iframe>' documents, route templates, etc. A flaw in the logic that tries to match entire URLs against regular expression matchers can result in partial matches for certain types of regular expressions, effectively bypassing the policies and allowing the use of unsafe values as resource URLs. This issue affects AngularJS versions greater than or equal to 1.2.0-rc.3.	2026-06-24	7.6

		Note: The AngularJS project was already End-of-Life when this CVE was published and will not receive any updates to address this issue. For more information see the End-of-Life announcement https://docs.angularjs.org/misc/version-support-status .		
CVE-2026-39951	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have a Stored SQL Injection vulnerability through graph_name_regexp in the Reports feature. This issue has been fixed in version 1.2.31.	2026-06-25	7.6
CVE-2026-44914	apache - nifi	Apache NiFi 1.12.0 through 2.9.0 are missing authorization when replacing Process Groups that include extension components with specific Required Permissions based on the Restricted annotation. The Restricted annotation indicates additional privileges required, but framework authorization did not check restricted status when handling requests to replace Process Groups. The missing authorization permits a user with general write access to add components with Restricted status. Apache NiFi installations that do not implement specific authorization for Restricted components are not subject to this vulnerability because the framework enforces write permissions as the security boundary. Upgrading to Apache NiFi 2.9.0 is the recommended mitigation, which removes the implementation of Restricted status authorization from the framework.	2026-06-22	7.5
CVE-2025-66389	microsoft - github_copilot	GitHub Copilot 1.372.0 allows filesystem access outside of a workspace folder (without user approval) via a file-handler URI parameter to fetch_webpage. Therefore, exfiltration could occur if there is indirect prompt injection.	2026-06-22	7.5
CVE-2026-8858	ibm - i	IBM WebSphere Application Server and IBM WebSphere Application Server Liberty are vulnerable to remote code execution and denial of service in the WebSphere Web Server Plug-in component. This vulnerability can be exploited when an attacker impersonates the application server and sends crafted responses to the plug-in.	2026-06-22	7.5
CVE-2026-9071	ibm - multiple products	IBM WebSphere Application Server 9.0, and 8.5 and IBM WebSphere Application Server - Liberty 17.0.0.3 through 26.0.0.6 are vulnerable to a denial of service, caused by sending a specially-crafted request. A remote attacker could exploit this vulnerability to cause the server to consume memory resources.	2026-06-22	7.5
CVE-2026-42127	grafana - multiple products	The public dashboard query endpoint does not limit request body size before processing, allowing unauthenticated attackers to trigger excessive memory allocation by sending arbitrarily large JSON payloads. This can lead to denial of service through memory exhaustion. No valid dashboard access token or authentication is required to exploit this vulnerability.	2026-06-22	7.5
CVE-2026-52922	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: batman-adv: dat: handle forward allocation error batadv_dat_forward_data() calls pskb_copy_for_clone() to duplicate an skb for each DHT candidate, but does not check the return value before passing it to batadv_send_skb_prepare_unicast_4addr(). That function dereferences the skb unconditionally, so a failed allocation triggers a NULL pointer dereference. Skip forwarding to the current DHT candidate on allocation failure.	2026-06-24	7.5
CVE-2026-52929	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: stream: fully roll back denied add-stream state When ADD_OUT_STREAMS is denied, SCTP only shrinks the queued chunks and then lowers outcnt. That leaves removed stream metadata behind, so a later re-add can reuse a stale ext and hit a null-pointer dereference in the scheduler get path. Fix the rollback by tearing down the removed stream state the same way other stream resizes do. Unschedule the current scheduler state, drop the removed stream ext state with sctp_stream_outq_migrate(), and then reschedule the remaining streams. This keeps scheduler-private RR/FC/PRIO lists consistent while fully rolling back denied outgoing stream additions.	2026-06-24	7.5
CVE-2026-52932	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xfrm: ipcomp: Free destination pages on acomp errors Move the out_free_req label up by a couple of lines so that the allocated dst SG list gets freed on error as well as success.	2026-06-24	7.5
CVE-2026-57281	jenkins - script_security	Jenkins Script Security Plugin 1402.v94c9ce464861 and earlier does not reject Groovy AST transformation annotations carrying an extensions member, allowing attackers able to run sandboxed Groovy scripts to execute code outside the sandbox if a suitable script is present on the classpath of the component that evaluates the script.	2026-06-24	7.5
CVE-2026-52945	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Revert "wireguard: device: enable threaded NAPI" This reverts commit 933466fc50a8e4eb167acbd0d8ec96a078462e9c which is commit db9ae3b6b43c79b1ba87eea849fd65efa05b4b2e upstream. We have had three independent production user reports in combination with Cilium utilizing WireGuard as encryption underneath that k8s Pod E/W traffic to certain peer nodes fully stalled. The situation appears	2026-06-24	7.5

		as follows: - Occurs very rarely but at random times under heavy networking load. - Once the issue triggers the decryption side stops working completely for that WireGuard peer, other peers keep working fine. The stall happens also for newly initiated connections towards that particular WireGuard peer. - Only the decryption side is affected, never the encryption side. - Once it triggers, it never recovers and remains in this state, the CPU/mem on that node looks normal, no leak, busy loop or crash. - bpftrace on the affected system shows that wg_prev_queue_enqueue fails, thus the MAX_QUEUED_PACKETS (1024 skbs!) for the peer's rx_queue is reached. - Also, bpftrace shows that wg_packet_rx_poll for that peer is never called again after reaching this state for that peer. For other peers wg_packet_rx_poll does get called normally. - Commit db9ae3b ("wireguard: device: enable threaded NAPI") switched WireGuard to threaded NAPI by default. The default has not been changed for triggering the issue, neither did CPU hotplugging occur (i.e. 5bd8de2 ("wireguard: queueing: always return valid online CPU in wg_cpumask_choose_online()")). - The issue has been observed with stable kernels of v5.15 as well as v6.1. It was reported to us that v5.10 stable is working fine, and no report on v6.6 stable either (somewhat related discussion in [0] though). - In the WireGuard driver the only material difference between v5.10 stable and v5.15 stable is the switch to threaded NAPI by default. [0] https://lore.kernel.org/netdev/CA+wXwBTT74RErDGAnj98PqS=wvdh8eM1pi4q6tTdExtjnokKqA@mail.gmail.com/ Breakdown of the problem: 1) skbs arriving for decryption are enqueued to the peer->rx_queue in wg_packet_consume_data via wg_queue_enqueue_per_device_and_peer. 2) The latter only moves the skb into the MPSC peer queue if it does not surpass MAX_QUEUED_PACKETS (1024) which is kept track in an atomic counter via wg_prev_queue_enqueue. 3) In case enqueueing was successful, the skb is also queued up in the device queue, round-robin picks a next online CPU, and schedules the decryption worker. 4) The wg_packet_decrypt_worker, once scheduled, picks these up from the queue, decrypts the packets and once done calls into wg_queue_enqueue_per_peer_rx. 5) The latter updates the state to PACKET_STATE_CRYPTED on success and calls napi_schedule on the per peer->napi instance. 6) NAPI then polls via wg_packet_rx_poll. wg_prev_queue_peek checks on the peer->rx_queue. It will wg_prev_queue_dequeue if the queue->peeked skb was not cached yet, or just return the latter otherwise. (wg_prev_queue_drop_peeked later clears the cache.) 7) From an ordering perspective, the peer->rx_queue has skbs in order while the device queue with the per-CPU worker threads from a global ordering PoV can finish the decryption and signal the skb PACKET_STATE_CRYPTED out of order. 8) A situation can be observed that the first packet coming in will be stuck waiting for the decryption worker to be scheduled for a longer time when the system is under pressure. 9) While this is the case, the other CPUs in the meantime finish decryption and call into napi_schedule. 10) Now in wg_packet_rx_poll it picks up the first in-order skb from the peer->rx_queue and sees that its state is still PACKET_STATE_UNCRYPTED. The NAPI poll routine then exits e ---truncated---		
CVE-2026-52946	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fs/fcntl: fix SOFTIRQ-unsafe lock order in fasync signaling A SOFTIRQ-safe to SOFTIRQ-unsafe lock order deadlock can occur in send_sigio() and send_sigurg() when a process group receives a signal. When FASYNC is configured for a process group (PIDTYPE_PGID), both functions use read_lock(&tasklist_lock) to traverse the task list. However, they are frequently called from softirq context: - send_sigio() via input_inject_event -> kill_fasync - send_sigurg() via tcp_check_urg -> sk_send_sigurg (NET_RX_SOFTIRQ) The deadlock is caused by the rwlock writer fairness mechanism: 1. CPU 0 (process context) holds read_lock(&tasklist_lock) in do_wait(). 2. CPU 1 (process context) attempts write_lock(&tasklist_lock) in fork() or exit() and spins, which blocks all new readers.	2026-06-24	7.5

		3. CPU 0 is interrupted by a softirq (e.g., TCP URG packet reception).
		4. The softirq calls send_sigurg() and attempts to acquire
		read_lock(&tasklist_lock), deadlocking because CPU 1 is waiting.
		
		Since PID hashing and do_each_pid_task() traversals are already
		RCU-protected, the read_lock on tasklist_lock is no longer strictly
		required for safe traversal. Fix this by replacing tasklist_lock with
		rcu_read_lock(), aligning the process group signaling path with the
		single-PID path. This also mitigates a potential remote denial of
		service vector via TCP URG packets.
		
		Lockdep splat:
		===
		WARNING: SOFTIRQ-safe -> SOFTIRQ-unsafe lock order detected
		[...]
		Chain exists of:
		&dev->event_lock --> &f_owner->lock --> tasklist_lock
		
		Possible interrupt unsafe locking scenario:
		CPU0 CPU1
		---- ----
		lock(tasklist_lock);
		local_irq_disable();
		lock(&dev->event_lock);
		lock(&f_owner->lock);
		<Interrupt>
		lock(&dev->event_lock);
		
		*** DEADLOCK ***		

CVE-2026-52954	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: handle rbtree insertion error in decode_choose_args() A message of type CEPH_MSG_OSD_MAP contains an OSD map that itself contains a CRUSH map. The received CRUSH map may optionally contain choose_args that get decoded in decode_choose_args(). In this function, num_choose_arg_maps is read from the message, and a corresponding number of crush_choose_arg_maps gets decoded afterwards. Each crush_choose_arg_map has a choose_args_index, which serves as the key when inserting it into the choose_args rbtree of the decoded crush_map. If a (potentially corrupted) message contains two crush_choose_arg_maps with the same index, the assertion in insert_choose_arg_map() triggers a kernel BUG when trying to insert the second crush_choose_arg_map. This patch fixes the issue by switching to the non-asserting rbtree insertion function and rejecting the message if the insertion fails. [idryomov: changelog]	2026-06-24	7.5
CVE-2026-52956	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Fix potential out-of-bounds access in __ceph_x_decrypt() In __ceph_x_decrypt(), a part of the buffer p is interpreted as a ceph_x_encrypt_header, and the magic field of this struct is accessed. This happens without any guarantee that the buffer is large enough to hold this struct. The function parameter ciphertext_len represents the length of the ciphertext to decrypt and is guaranteed to be at most the remaining size of the allocated buffer p. However, this value is not necessarily greater than sizeof(ceph_x_encrypt_header). E.g., a message frame of type FRAME_TAG_AUTH_REPLY_MORE, that is just as long to hold the ciphertext at its end with a ciphertext_len of 8 or less, can trigger an out-of-bounds memory access when accessing hdr->magic. This patch fixes the issue by adding a check to ensure that the decrypted plaintext in the buffer is large enough to represent at least the ceph_x_encrypt_header.	2026-06-24	7.5
CVE-2026-52957	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libceph: Fix potential null-ptr-deref in decode_choose_args() A message of type CEPH_MSG_OSD_MAP contains an OSD map that itself contains a CRUSH map. When decoding this CRUSH map in crush_decode(), an array of max_buckets CRUSH buckets is decoded, where some indices may not refer to actual buckets and are therefore set to NULL. The received CRUSH map may optionally contain choose_args that get decoded in decode_choose_args(). When decoding a crush_choose_arg_map, a series of choose_args for different buckets is decoded, with the bucket_index being read from the incoming message. It is only checked that the bucket index does not exceed max_buckets, but not that it doesn't point to an index with a NULL bucket. If a (potentially corrupted) message contains a crush_choose_arg_map including such a bucket_index, a null pointer	2026-06-24	7.5

		dereference may occur in the subsequent processing when attempting to access the bucket with the given index. This patch fixes the issue by extending the affected check. Now, it is only attempted to access the bucket if it is not NULL.		
CVE-2026-52960	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ceph: put folios not suitable for writeback The batch holds references to the folios (see `filemap_get_folios`, `folio_batch_release`), so we need to `folio_put` the folios we remove. Tested on v6.18.	2026-06-24	7.5
CVE-2026-52974	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: tls: fix strparser anchor skb leak on offload RX setup failure When tls_set_device_offload_rx() fails at tls_dev_add(), the error path calls tls_sw_free_resources_rx() to clean up the SW context that was initialized by tls_set_sw_offload(). This function calls tls_sw_release_resources_rx() (which stops the strparser via tls_strp_stop()) and tls_sw_free_ctx_rx() (which kfrees the context), but never frees the anchor skb that was allocated by alloc_skb(0) in tls_strp_init(). Note that tls_sw_free_resources_rx() is exclusively used for this "failed to start offload" code path, there's no other caller. The leak did not exist before commit 84c61fe1a75b ("tls: rx: do not use the standard strparser"), because the standard strparser doesn't try to pre-allocate an skb. The normal close path in tls_sk_proto_close() handles cleanup by calling tls_sw_strparser_done() (which calls tls_strp_done()) after dropping the socket lock, because tls_strp_done() does cancel_work_sync() and the strparser work handler takes the socket lock.	2026-06-24	7.5
CVE-2026-52981	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: neigh: let neigh_xmit take skb ownership neigh_xmit always releases the skb, except when no neighbour table is found. But even the first added user of neigh_xmit (mpls) relied on neigh_xmit to release the skb (or queue it for tx). sashiko reported: If neigh_xmit() is called with an uninitialized neighbor table (for example, NEIGH_ND_TABLE when IPv6 is disabled), it returns -EAFNOSUPPORT and bypasses its internal out_kfree_skb error path. Because the return value of neigh_xmit() is ignored here, does this leak the SKB? Assume full ownership and remove the last code path that doesn't xmit or free skb.	2026-06-24	7.5
CVE-2026-52983	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: airoha: fix BQL imbalance in TX path Fix a possible BQL imbalance in airoha_dev_xmit(), where inflight packets are accounted only for the AIROHA_NUM_TX_RING netdev TX queues. The queue index is computed as: qid = skb_get_queue_mapping(skb) % ARRAY_SIZE(qdma->q_tx) txq = netdev_get_tx_queue(dev, qid); However, airoha_qdma_tx_napi_poll() accounts completions across all netdev TX queues (num_tx_queues), leading to inconsistent BQL accounting. Also reset all netdev TX queues in the ndo_stop callback.	2026-06-24	7.5
CVE-2026-52998	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nfnetlink_osf: fix potential NULL dereference in ttl check The nf_osf_ttl() function accessed skb->dev to perform a local interface address lookup without verifying that the device pointer was valid. Additionally, the implementation utilized an in_dev_for_each_ifa_rcu loop to match the packet source address against local interface addresses. It assumed that packets from the same subnet should not see a decrement on the initial TTL. A packet might appear it is from the same subnet but it actually isn't especially in modern environments with containers and virtual switching.	2026-06-24	7.5

		Remove the device dereference and interface loop. Replace the logic with a switch statement that evaluates the TTL according to the ttl_check.		
CVE-2026-53003	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: pppoe: drop PFC frames RFC 2516 Section 7 states that Protocol Field Compression (PFC) is NOT RECOMMENDED for PPPoE. In practice, pppd does not support negotiating PFC for PPPoE sessions, and the current PPPoE driver assumes an uncompressed (2-byte) protocol field. However, the generic PPP layer function ppp_input() is not aware of the negotiation result, and still accepts PFC frames. If a peer with a broken implementation or an attacker sends a frame with a compressed (1-byte) protocol field, the subsequent PPP payload is shifted by one byte. This causes the network header to be 4-byte misaligned, which may trigger unaligned access exceptions on some architectures. To reduce the attack surface, drop PPPoE PFC frames. Introduce ppp_skb_is_compressed_proto() helper function to be used in both ppp_generic.c and pppoe.c to avoid open-coding.	2026-06-24	7.5
CVE-2026-53026	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix nfs4_file access extra count in nfsd4_add_rdaccess_to_wrdeleg In nfsd4_add_rdaccess_to_wrdeleg, if fp->fi_fds[O_RDONLY] is already set by another thread, __nfs4_file_get_access should not be called to increment the nfs4_file access count since that was already done by the thread that added READ access to the file. The extra fi_access count in nfs4_file can prevent the corresponding nfsd_file from being freed. When stopping nfs-server service, these extra access counts trigger a BUG in kmem_cache_destroy() that shows nfsd_file object remaining on __kmem_cache_shutdown. This problem can be reproduced by running the Git project's test suite over NFS.	2026-06-24	7.5
CVE-2026-53069	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net, bpf: fix null-ptr-deref in xdp_master_redirect() for down master syzkaller reported a kernel panic in bond_rr_gen_slave_id() reached via xdp_master_redirect(). Full decoded trace: https://syzkaller.appspot.com/bug?extid=80e046b8da2820b6ba73 bond_rr_gen_slave_id() dereferences bond->rr_tx_counter, a per-CPU counter that bonding only allocates in bond_open() when the mode is round-robin. If the bond device was never brought up, rr_tx_counter stays NULL. The XDP redirect path can still reach that code on a bond that was never opened: bpf_master_redirect_enabled_key is a global static key, so as soon as any bond device has native XDP attached, the XDP_TX -> xdp_master_redirect() interception is enabled for every slave system-wide. The path xdp_master_redirect() -> bond_xdp_get_xmit_slave() -> bond_xdp_xmit_roundrobin_slave_get() -> bond_rr_gen_slave_id() then runs against a bond that has no rr_tx_counter and crashes. Fix this in the generic xdp_master_redirect() by refusing to call into the master's ->ndo_xdp_get_xmit_slave() when the master device is not up. IFF_UP is only set after ->ndo_open() has successfully returned, so this reliably excludes masters whose XDP state has not been fully initialized. Drop the frame with XDP_ABORTED so the exception is visible via trace_xdp_exception() rather than silently falling through. This is not specific to bonding: any current or future master that defers XDP state allocation to ->ndo_open() is protected.	2026-06-24	7.5
CVE-2026-53070	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: disable BH before calling udp_tunnel_xmit_skb() udp_tunnel_xmit_skb() / udp_tunnel6_xmit_skb() are expected to run with BH disabled. After commit 6f1a9140ecda ("add xmit recursion limit to tunnel xmit functions"), on the path: udp(6)_tunnel_xmit_skb() -> ip(6)tunnel_xmit()	2026-06-24	7.5

		dev_xmit_recursion_inc()/dec() must stay balanced on the same CPU. Without local_bh_disable(), the context may move between CPUs, which can break the inc/dec pairing. This may lead to incorrect recursion level detection and cause packets to be dropped in ip(6)_tunnel_xmit() or __dev_queue_xmit(). Fix it by disabling BH around both IPv4 and IPv6 SCTP UDP xmit paths. In my testing, after enabling the SCTP over UDP: # ip net exec ha sysctl -w net.sctp.udp_port=9899 # ip net exec ha sysctl -w net.sctp.encap_port=9899 # ip net exec hb sysctl -w net.sctp.udp_port=9899 # ip net exec hb sysctl -w net.sctp.encap_port=9899 # ip net exec ha iperf3 -s - without this patch: # ip net exec hb iperf3 -c 192.168.0.1 --sctp [5] 0.00-10.00 sec 37.2 MBytes 31.2 Mbits/sec sender [5] 0.00-10.00 sec 37.1 MBytes 31.1 Mbits/sec receiver - with this patch: # ip net exec hb iperf3 -c 192.168.0.1 --sctp [5] 0.00-10.00 sec 3.14 GBytes 2.69 Gbits/sec sender [5] 0.00-10.00 sec 3.14 GBytes 2.69 Gbits/sec receiver		
CVE-2026-53087	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: bcmgenet: fix leaking free_bds While reclaiming the tx queue we fast forward the write pointer to drop any data in flight. These dropped frames are not added back to the pool of free bds. We also need to tell the netdev that we are dropping said data.	2026-06-24	7.5
CVE-2026-13029	google - chrome	Use after free in Web Authentication in Google Chrome prior to 149.0.7827.197 allowed an attacker who convinced a user to install a malicious extension to potentially exploit heap corruption via a crafted Chrome Extension. (Chromium security severity: High)	2026-06-24	7.5
CVE-2026-53165	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iomap: avoid potential null folio->mapping deref during error reporting When a buffered read fails, iomap_finish_folio_read() reports the error with fserror_report_io(folio->mapping->host, ...). This is called after ifs->read_bytes_pending has been decremented by the bytes attempted to be read. For a folio split across multiple read completions, the folio is only guaranteed to stay locked while read_bytes_pending > 0. Once iomap_finish_folio_read() decrements read_bytes_pending, another in-flight read can complete and end the read on the folio, which unlocks it. This allows truncate logic to run and detach the folio (set folio->mapping to NULL). The error reporting path then can dereference a NULL folio->mapping. As reported by Sam Sun, this is the race that can occur: CPU0: failed completion CPU1: final completion CPU2: truncate ---------- ---------- ---------- read_bytes_pending -= len finished = false /* preempted before fserror_report_io() */ read_bytes_pending -= len finished = true folio_end_read() truncate clears folio->mapping fserror_report_io(folio->mapping->host, ...) ^ NULL deref Fix this by reporting the error first before decrementing ifs->read_bytes_pending.	2026-06-25	7.5
CVE-2026-53180	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: timers/migration: Fix livelock in tmigr_handle_remote_up() tmigr_handle_remote_cpu() skips timer_expire_remote() when cpu == smp_processor_id(), assuming the local softirq path already handled this	2026-06-25	7.5

		CPU's timers. This assumption is wrong because jiffies can advance after the handling of the CPU's global timers in run_timer_base(BASE_GLOBAL) and before tmigr_handle_remote() evaluates the expiry times. As a consequence a timer which expires after the CPU local timer wheel advanced and becomes expired in the remote handling is ignored and the callback is never invoked and removed from the timer wheel. What's worse is that fetch_next_timer_interrupt_remote() keeps reporting it as expired, and the event is re-queued with expires == now on each iteration. The goto-again loop spins indefinitely. Fix this by calling timer_expire_remote() unconditionally. That's minimal overhead for the common case as __run_timer_base() returns immediately if there is nothing to expire in the local wheel. [tglx: Amend change log and add a comment]		
CVE-2026-53183	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mptcp: allow subflow rcv wnd to shrink In MPTCP connection, the `window` field in the TCP header refers to the MPTCP-level rcv_nxt and it's right edge should not move backward. Such constraint is enforced at DSS option generation time. At the same time, the TCP stack ensures independently that the TCP-level rcv wnd right's edge does not move backward. That in turn causes artificial inflating of the MPTCP rcv window when the incoming data is acked at the TCP level and is OoO in the MPTCP sequence space (or lands in the backlog). As a consequence, the incoming traffic can exceed the receiver rcvbuf size even when the sender is not misbehaving. Prevent such scenario forcibly allowing the TCP subflow to shrink the TCP-level rcv wnd regardless of the current netns setting.	2026-06-25	7.5

CVE-2026-53184	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		udp: clear skb->dev before running a sockmap verdict
		
		On the UDP receive path skb->dev is repurposed as dev_scratch (the truesize/state cache set by udp_set_dev_scratch()), through the union { struct net_device *dev; unsigned long dev_scratch; } in sk_buff.
		
		When a UDP socket is in a sockmap, sk_data_ready is sk_psock_verdict_data_ready(), which calls udp_read_skb() -> recv_actor() (sk_psock_verdict_recv) to run the attached SK_SKB verdict program in softirq. If that program calls a socket-lookup helper (bpf_sk_lookup_tcp/udp, bpf_skc_lookup_tcp), bpf_skc_lookup() does:
		
		if (skb->dev)
		caller_net = dev_net(skb->dev);
		
		skb->dev still holds the dev_scratch value (a non-NULL integer), so dev_net() dereferences it as a struct net_device * and the kernel takes a general protection fault on a non-canonical address in softirq:
		
		Oops: general protection fault, probably for non-canonical address 0x1010000800004a0
		CPU: 1 UID: 0 PID: 1406 Comm: syz.2.19 Not tainted 7.1.0-rc6 #1 PREEMPT(full)
		RIP: 0010:bpf_skc_lookup net/core/filter.c:7033 [inline]
		RIP: 0010:bpf_sk_lookup+0x45/0x160 net/core/filter.c:7047
		Call Trace:
		<IRQ>
		bpf_prog_4675cb904b7071f8+0x12e/0x14e
		bpf_prog_run_pin_on_cpu+0xc6/0x1f0
		sk_psock_verdict_recv+0x1ba/0x350
		udp_read_skb+0x31a/0x370
		sk_psock_verdict_data_ready+0x2e3/0x600
		__udp_enqueue_schedule_skb+0x4c8/0x650
		udpv6_queue_rcv_one_skb+0x3ec/0x740
		udp6_unicast_rcv_skb+0x11d/0x140
		ip6_protocol_deliver_rcu+0x61e/0x950
		ip6_input_finish+0xa9/0x150
		NF_HOOK+0x286/0x2f0
		ip6_input+0x117/0x220
		NF_HOOK+0x286/0x2f0
		__netif_receive_skb+0x85/0x200
		process_backlog+0x374/0x9a0
		__napi_poll+0x4f/0x1c0
		net_rx_action+0x3b0/0x770
			2026-06-25	7.5

		handle_softirqs+0x15a/0x460 do_softirq+0x57/0x80 </IRQ> The rmem charge that dev_scratch accounted for is released by skb_recv_udp() on dequeue, just above, so the scratch is dead by the time recv_actor() runs. Clear skb->dev so bpf_skc_lookup() falls back to sock_net(skb->sk), which skb_set_owner_sk_safe() set just above.		
CVE-2026-53199	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hv_netvsc: use kmap_local_page in netvsc_copy_to_send_buf netvsc_copy_to_send_buf() copies page buffer entries into the VMBus send buffer using phys_to_virt() on the entry PFN. Entries for the RNDIS header and the skb linear data come from kmalloc'd memory and are always in the kernel direct map, but entries for skb fragments reference page cache or user pages, which on 32-bit x86 with CONFIG_HIGHMEM=y can live above the LOWMEM boundary. For such a page phys_to_virt() returns an address outside the direct map and the subsequent memcpy() faults on the transmit softirq path, which is fatal. Map the pages with kmap_local_page() instead, handling two properties of the page buffer entries: - pb[i].pfn is a Hyper-V PFN at HV_HYP_PAGE_SIZE (4K) granularity, not a native PFN. Reconstruct the physical address first and derive the native page from it, so the mapping stays correct where PAGE_SIZE > HV_HYP_PAGE_SIZE (e.g. arm64 with 64K pages). - Since commit 41a6328b2c55 ("hv_netvsc: Preserve contiguous PFN grouping in the page buffer array"), an entry describes a full physically contiguous fragment and pb[i].len can exceed PAGE_SIZE, while kmap_local_page() maps a single page. Copy page by page, splitting at native page boundaries. The copy path only handles packets smaller than the send section size (6144 bytes by default); larger packets take the cp_partial path where only the RNDIS header is copied. So entries here are bounded by the section size and a copy is split at most once on 4K-page systems. On !CONFIG_HIGHMEM configs kmap_local_page() folds to page_address() and no mapping work is added.	2026-06-25	7.5
CVE-2026-53229	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: xsk: Fix DMA and xdp_frame leak on XDP_TX xmit failure In the XSK branch of mlx5e_xmit_xdp_buff(), when sq->xmit_xdp_frame() returns false (e.g. XDPSQ is full), the function returns without unmapping the DMA address or freeing the xdp_frame allocated by xdp_convert_zc_to_xdp_frame(). The xdpi_fifo push only happens on success, so the completion path cannot recover these entries. With CONFIG_DMA_API_DEBUG=y, the leak surfaces on driver unbind: DMA-API: pci 0000:08:00.0: device driver has pending DMA allocations while released from device [count=1116] One of leaked entries details: [device address=0x000000010ffd7028] [size=1534 bytes] [mapped with DMA_TO_DEVICE] [mapped as phy] WARNING: kernel/dma/debug.c:881 at dma_debug_device_change+0x127/0x180 ... DMA-API: Mapped at: debug_dma_map_phys+0x4b/0xd0 dma_map_phys+0xfd/0x2d0 mlx5e_xdp_handle+0x5ae/0xac0 [mlx5_core] mlx5e_xsk_skb_from_cqe_mpwrq_linear+0xc4/0x170 [mlx5_core] mlx5e_handle_rx_cqe_mpwrq+0xc1/0x290 [mlx5_core] Add the missing unmap + xdp_return_frame, matching the cleanup already done in mlx5e_xdp_xmit(). has_frags is rejected earlier in this branch, so no per-frag unmap is needed.	2026-06-25	7.5
CVE-2026-53235	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: add pskb_may_pull() to skb_gro_receive_list() skb_gro_receive_list() calls skb_pull(skb, skb_gro_offset(skb)) without first ensuring the data is in the linear area via pskb_may_pull(). When the skb arrives via napi_gro_frags(), skb_headlen can be 0 (all data in page fragments) while skb_gro_offset is non-zero (after IP+TCP header parsing). The skb_pull() then decrements skb->len by skb_gro_offset but skb->data_len stays unchanged, hitting BUG_ON(skb->len < skb->data_len) in __skb_pull().	2026-06-25	7.5

		The UDP fraglist GRO path already contains this guard at udp_offload.c:749. Adding it to skb_gro_receive_list() itself provides centralized protection for all callers (TCP, UDP, and any future protocols), and ensures the precondition of skb_pull() is satisfied before it is called. On pskb_may_pull() failure, set NAPI_GRO_CB(skb)->flush = 1 so the skb is not held as a new GRO head and is instead delivered through the normal receive path, matching the UDP handling.		
CVE-2026-53244	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: VFS: fix possible failure to unlock in nfsd4_create_file() atomic_create() in fs/namei.c drops the reference to the dentry when it returns an error. This behaviour was imported into dentry_create() so that it will drop the reference if an error is returned from atomic_create(), though not if vfs_create() returns an error (in the case where ->atomic_create is not supported). The caller - nfsd4_create_file() - is made aware of this by checking path->dentry, which will either be a counted reference to a dentry, or an error pointer. However the change to use start_creating()/end_creating() (which landed shortly before the dentry_create() change landed, though was likely developed around the same time) means that nfsd4_create_file() *needs* a valid dentry so that it can unlock the parent. The net result is that if NFSD exports a filesystem which uses ->atomic_create, and if a call to ->atomic_create returns an error, then nfsd4_create_file() will pass an error pointer to end_creating() and the parent will not be unlocked. Fix this by changing dentry_create() to make sure path->dentry is always a valid dentry, never an error-pointer. The actual error is already returned a different way. Note that if ->atomic_create() returns a different dentry (which may not be possible in practice) we are guaranteed (because it is only ever provided by d_spliace_alias()) that it will have the same d_parent and so it will have the same effect when passed to end_creating().	2026-06-25	7.5
CVE-2026-13283	google - chrome	Use after free in AdFilter in Google Chrome on Android prior to 149.0.7827.201 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)	2026-06-25	7.5
CVE-2026-49486	apache - apache-airflow-providers-ftp	The Apache Airflow FTP provider's `FTPSHook.get_conn()` created an `ftplib.FTP_TLS` connection but never called `prot_p()`, so although the control channel was TLS-protected the data channel was transmitted in cleartext. Any deployment using `FTPSHook` or `FTPSFileTransmitOperator` to move files over FTPS exposed file contents and credentials-in-transit to a network attacker able to observe the data connection. Upgrade apache-airflow-providers-ftp to `3.15.1` or later, which issues `PROT P` to encrypt the data channel.	2026-06-26	7.5
CVE-2026-53284	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: only release the dirty pages io tree after successful writes [WARNING] With extra warning on dirty extent buffers at umount (aka, the next patch in the series), test case generic/388 can trigger the following warning about dirty extent buffers at unmount time: BTRFS critical (device dm-2 state E): emergency shutdown BTRFS error (device dm-2 state E): error while writing out transaction: -30 BTRFS warning (device dm-2 state E): Skipping commit of aborted transaction. BTRFS error (device dm-2 state EA): Transaction 9 aborted (error -30) BTRFS: error (device dm-2 state EA) in cleanup_transaction:2068: errno=-30 Readonly filesystem BTRFS info (device dm-2 state EA): forced readonly BTRFS info (device dm-2 state EA): last unmount of filesystem 4fbf2e15-f941-49a0-bc7c-716315d2777c ------------[cut here]------------ WARNING: disk-io.c:3311 at invalidate_and_check_btree_folios+0xfd/0x1ca [btrfs], CPU#8: umount/914368 CPU: 8 UID: 0 PID: 914368 Comm: umount Tainted: G OE 7.1.0-rc1-custom+ #372 PREEMPT(full) 2de38db8d1deae71fde295430a0ff3ab98ccf596 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS unknown 02/02/2022 RIP: 0010:invalidate_and_check_btree_folios+0xfd/0x1ca [btrfs] Call Trace: <TASK> close_ctree+0x52e/0x574 [btrfs d2f0b1cd330d1287e7a9919d112eadfc0e914efd] generic_shutdown_super+0x89/0x1a0	2026-06-26	7.5

		kill_anon_super+0x16/0x40
		btrfs_kill_super+0x16/0x20 [btrfs d2f0b1cd330d1287e7a9919d112eadfc0e914efd]
		deactivate_locked_super+0x2d/0xb0
		cleanup_mnt+0xdc/0x140
		task_work_run+0x5a/0xa0
		exit_to_user_mode_loop+0x123/0x4b0
		do_syscall_64+0x243/0x7c0
		entry_SYSCALL_64_after_hwframe+0x4b/0x53
		</TASK>
		---[end trace 0000000000000000]---
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30539776 owner 9 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30621696 owner 257 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30638080 owner 258 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30654464 owner 7 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30703616 owner 2 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30720000 owner 10 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30736384 owner 4 gen 9 refs 2 flags 0x7
		BTRFS warning (device dm-2 state EA): unable to release extent buffer 30752768 owner 11 gen 9 refs 2 flags 0x7
		
		I'm using a stripped down version, which seems to trigger the warning more reliably:
		
		_fsstress_pid=""
		workload()
		{
		dmesg -C
		mkfs.btrfs -f -K $dev > /dev/null
		echo 1 > /sys/kernel/debug/clear_warn_once
		mount $dev $mnt
		$fsstress -w -n 1024 -p 4 -d $mnt &
		_fsstress_pid=$!
		sleep 0
		$godown $mnt
		pkill --echo -PIPE fsstress > /dev/null
		wait $_fsstress_pid
		unset _fsstress_pid
		umount $mnt
		
		if dmesg | grep -q "WARNING"; then
		fail
		fi
		}
		
		for ((i = 0; i < $runtime; i++)); do
		echo "=== $i/$runtime ==="
		workload
		done
		
		[CAUSE]
		Inside btrfs_write_and_wait_transaction(), we first try to write all dirty ebs, then wait for them to finish.
		
		After that we call btrfs_extent_io_tree_release() to free all extent states from dirty_pages io tree.
		
		However if we hit an error from btrfs_write_marked_extent(), then we still call btrfs_extent_io_tree_release() to clear that dirty_pages io tree, which may contain dirty records that we haven't yet submitted.
		
		Furthermore, the later transaction cleanup path will utilize that dirty_pages io tree to properly cleanup those dirty ebs, but since it's already empty, no dirty ebs are properly cleaned up, thus will later trigger the warnings inside invalidate_btree_folios().
		---truncated---		

CVE-2026-8646	ibm - multiple products	IBM WebSphere Application Server 9.0 and 8.5 and IBM WebSphere Application Server - Liberty 17.0.0.3 through 26.0.0.6 are vulnerable to HTTP request smuggling. A remote attacker could smuggle a specially crafted request to the application server thereby allowing the attacker to bypass security controls, spoof identity, escalate privilege, and expose sensitive information.	2026-06-22	7.4
CVE-2026-9006	ibm - multiple products	IBM WebSphere Application Server 9.0, and 8.5 is vulnerable to server-side request forgery (SSRF) with the Ajax Proxy configured. This may allow an attacker to send unauthorized requests from the system, resulting in a security bypass or information disclosure.	2026-06-22	7.4

CVE-2026-57589	openbsd - OpenBSD	sys/kern/sysv_sem.c in OpenBSD through 7.9 has a use-after-free allowing local privilege escalation to root. This is a context switch use-after-free after tsleep in sys_semget().	2026-06-25	7.4
CVE-2026-12992	red hat - multiple products	A flaw was found in Apicurio Registry. The WSDLReaderAccessor creates a wsdl4j WSDLReader without disabling the javax.wsdl.importDocuments feature. When the VALIDITY rule is set to FULL, an attacker with Developer-role access can upload a WSDL document containing attacker-controlled import locations, causing the registry to issue HTTP requests to arbitrary internal URLs (server-side request forgery).	2026-06-25	7.4
CVE-2026-9029	grafana - grafana	The geomap panel's XYZ tile layer has a sanitize-then-interpolate ordering bug. sanitizeTextPanelContent() runs on the raw template string before getTemplateSrv().replace() substitutes the variable value, which uses the glob format with no HTML escaping. The result is passed to OpenLayers via element.innerHTML. An Editor can set a textbox variable's default value to an XSS payload that executes for every user who opens the dashboard. This is a bypass of the CVE-2023-0507 fix	2026-06-22	7.3
CVE-2026-10845	ibm - multiple products	IBM WebSphere Application Server 8.5 and 9.0 could allow a remote attacker to bypass authentication and gain unauthorized access to JAX-WS applications.	2026-06-22	7.3
CVE-2026-13201	red hat - multiple products	A flaw was found in KubeVirt's safepath package used by virt-handler. The OpenAtNoFollow function uses O_PATH|O_NOFOLLOW to obtain a file descriptor to a path leaf, but downstream operations resolve the path via /proc/self/fd/N using link-following syscalls. When the leaf is a symlink, the kernel dereferences it, defeating the intended no-follow protection. An attacker with access to a virt-launcher pod can exploit this to redirect virt-handler's IPC socket connections, including the notify socket used for VM domain lifecycle events. By hijacking this socket, the attacker can inject arbitrary domain events into virt-handler, causing it to take incorrect lifecycle actions, corrupt VM state in the Kubernetes API, or crash — resulting in sustained denial of VM management services for all virtual machines on the affected node. Additionally, the same symlink following flaw allows virt-handler to apply file ownership or permission changes to unintended host paths.	2026-06-24	7.3
CVE-2026-46734	dell - display_and_peripheral_manager	Dell Display and Peripheral Manager (DDPM Mac), versions prior to 2.3, contain an Improper Certificate Validation vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Protection mechanism bypass.	2026-06-25	7.3
CVE-2026-9086	redhat - multiple products	A flaw was found in Keycloak. A remote attacker with administrative privileges, specifically those with `manage-client` permission or access to client registration endpoints, could bypass client Uniform Resource Identifier (URI) validation. This is achieved by registering a malicious client with a specially crafted redirect URI using a case-insensitive `javascript:` or `data:` scheme. This Cross-Site Scripting (XSS) vulnerability allows for arbitrary code execution in the Keycloak origin when a victim clicks the crafted link, such as in the logout flow or the Admin Console.	2026-06-25	7.3
CVE-2026-57915	apache software foundation - Apache Kerby	It is possible to bypass the Kerberos pre-authentication check in Apache Kerby by sending a PA-DATA with an unrecognized or unsupported type. Users are recommended to upgrade to version 2.1.2, which fixes this issue.	2026-06-26	7.3
CVE-2026-49506	dell - multiple products	Dell Wyse Management Suite, versions prior to WMS 5.5 HF1, contain an Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') vulnerability. A high privileged attacker with remote access could potentially exploit this vulnerability, leading to Remote Code Execution.	2026-06-25	7.2
CVE-2026-40083	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have SQL Injection through unsanitized unserialize+implode in managers.php. At line 756 of managers.php, the application assigns $selected_items by calling cacti_unserialize(stripslashes(gnrv('selected_graphs_array'))). The cacti_unserialize() function calls unserialize() with allowed_classes set to false, which prevents object injection but still allows arbitrary string arrays to be deserialized. Then, at lines 760 to 766, the deserialized array values are passed directly into db_execute('DELETE FROM snmpagent_managers WHERE id IN (' . implode(',', $selected_items) . ')'), where they are imploded into the SQL statement without any integer validation, resulting in SQL Injection when using SNMP agent management permissions. This issue has been fixed in version 1.2.31.	2026-06-25	7.2
CVE-2026-52915	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: ip6t_hbh: reject oversized option lists struct ip6t_opts stores at most IP6T_OPTS_OPTSNR option descriptors, but hbh_mt6_check() does not reject larger optsnr values supplied from userspace. Validate optsnr in the rule setup path so only match data that fits the fixed-size opts array can be installed. This follows the existing xtables pattern of rejecting invalid user-provided counts in checkentry() and keeps the packet matching path unchanged. `struct ip6t_opts` has a fixed `opts[IP6T_OPTS_OPTSNR]` array, where `IP6T_OPTS_OPTSNR` is 16, then off-by-one array access is possible: [137.924693][T8692] UBSAN: array-index-out-of-bounds in ../net/ipv6/netfilter/ip6t_hbh.c:110:29 [137.926167][T8692] index 16 is out of range for type '__u16 [16]'	2026-06-24	7.1
CVE-2026-52917	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: diag: reject stale associations in dump_one path The SCTP exact sock_diag lookup can hold a transport reference, block on lock_sock(sk), and then resume after sctp_association_free() has marked the association dead and freed its bind address list.	2026-06-24	7.1

		When that happens, inet_assoc_attr_size() and inet_diag_msg_sctpasoc_fill() can still dereference association state that is no longer valid for reporting. In particular, inet_diag_msg_sctpasoc_fill() may read an empty bind-address list as a real sctp_sockaddr_entry and trigger an out-of-bounds read from unrelated association memory. Reject the association after taking the socket lock if it has been reaped or detached from the endpoint, and report the lookup as stale. This keeps the exact dump-one path from formatting torn association state.		
CVE-2026-52942	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_log: validate MAC header was set before dumping it The fallback path of dump_mac_header() guards the MAC header access only with "skb->mac_header != skb->network_header", without checking skb_mac_header_was_set(). When the MAC header is unset, mac_header is 0xffff, so the test passes and skb_mac_header(skb) returns skb->head + 0xffff, ~64 KiB past the buffer; the loop then reads dev->hard_header_len bytes out of bounds into the kernel log. This is reachable via the netdev logger: nf_log_unknown_packet() calls dump_mac_header() unconditionally, and an skb sent through AF_PACKET with PACKET_QDISC_BYPASS reaches the egress hook with mac_header still unset (__dev_queue_xmit(), which would reset it, is bypassed). Add the skb_mac_header_was_set() check the ARPHRD_ETHER path already uses, and replace the open-coded MAC header length test with skb_mac_header_len(). Only skbs with an unset MAC header are affected; valid ones are dumped as before. BUG: KASAN: slab-out-of-bounds in dump_mac_header (net/netfilter/nf_log_syslog.c:831) Read of size 1 at addr ffff88800ea49d3f by task exploit/148 Call Trace: kasan_report (mm/kasan/report.c:595) dump_mac_header (net/netfilter/nf_log_syslog.c:831) nf_log_netdev_packet (net/netfilter/nf_log_syslog.c:938 net/netfilter/nf_log_syslog.c:963) nf_log_packet (net/netfilter/nf_log.c:260) nft_log_eval (net/netfilter/nft_log.c:60) nft_do_chain (net/netfilter/nf_tables_core.c:285) nft_do_chain_netdev (net/netfilter/nft_chain_filter.c:307) nf_hook_slow (net/netfilter/core.c:619) nf_hook_direct_egress (net/packet/af_packet.c:257) packet_xmit (net/packet/af_packet.c:280) packet_sendmsg (net/packet/af_packet.c:3114) __sys_sendto (net/socket.c:2265)	2026-06-24	7.1
CVE-2026-57303	jenkins - assembla	Jenkins Assembla Plugin 1.4 and earlier does not configure its XML parser to prevent XML external entity (XXE) attacks, allowing attackers able to control the responses of the configured Assembla server to extract secrets from the Jenkins controller or perform server-side request forgery.	2026-06-24	7.1
CVE-2026-52953	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix oops due to out of scope access Below oops triggers when kill QEMU process: Oops: general protection fault, probably for non-canonical address 0x7fffffff844eaaa7: 0000 [#1] SMP NOPTI Call Trace: <TASK> do_raw_spin_lock+0xaa/0xc0 _raw_spin_lock_irqsave+0x21/0x40 domain_remove_dev_pasid+0x52/0x160 intel_nested_set_dev_pasid+0x1b9/0x1e0 __iommu_set_group_pasid+0x56/0x120 pci_dev_reset_iommu_done+0xe3/0x180 pcie_flr+0x65/0x160 __pci_reset_function_locked+0x5b/0x120 vfio_pci_core_close_device+0x63/0xe0 [vfio_pci_core] vfio_df_close+0x4f/0xa0 vfio_df_unbind_iommufd+0x2d/0x60 vfio_device_fops_release+0x3e/0x40 __fput+0xe5/0x2c0 task_work_run+0x58/0xa0 do_exit+0x2c8/0x600 do_group_exit+0x2f/0xa0 get_signal+0x863/0x8c0 arch_do_signal_or_restart+0x24/0x100 exit_to_user_mode_loop+0x87/0x380 do_syscall_64+0x2ff/0x11e0	2026-06-24	7.1

		entry_SYSCALL_64_after_hwframe+0x76/0x7e The global static blocked domain is a dummy domain without corresponding dmar_domain structure, accessing beyond iommu_domain structure triggers oops easily. Fix it by return early in domain_remove_dev_pasid() like identity domain.		
CVE-2026-52988	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: join hook list via splice_list_rcu() in commit phase Publish new hooks in the list into the basechain/flowtable using splice_list_rcu() to ensure netlink dump list traversal via rcu is safe while concurrent ruleset update is going on.	2026-06-24	7.1
CVE-2026-53040	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate bg_bits during freefrag scan [BUG] A crafted filesystem can trigger an out-of-bounds bitmap walk when OCFS2_IOC_INFO is issued with OCFS2_INFO_FL_NON_COHERENT. BUG: KASAN: use-after-free in instrument_atomic_read include/linux/instrumented.h:68 [inline] BUG: KASAN: use-after-free in _test_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline] BUG: KASAN: use-after-free in test_bit_le include/asm-generic/bitops/le.h:21 [inline] BUG: KASAN: use-after-free in ocfs2_info_freefrag_scan_chain fs/ocfs2/ioctl.c:495 [inline] BUG: KASAN: use-after-free in ocfs2_info_freefrag_scan_bitmap fs/ocfs2/ioctl.c:588 [inline] BUG: KASAN: use-after-free in ocfs2_info_handle_freefrag fs/ocfs2/ioctl.c:662 [inline] BUG: KASAN: use-after-free in ocfs2_info_handle_request+0x1c66/0x3370 fs/ocfs2/ioctl.c:754 Read of size 8 at addr ffff888031bce000 by task syz.0.636/1435 Call Trace: __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0xbe/0x130 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xd1/0x650 mm/kasan/report.c:482 kasan_report+0xfb/0x140 mm/kasan/report.c:595 check_region_inline mm/kasan/generic.c:186 [inline] kasan_check_range+0x11c/0x200 mm/kasan/generic.c:200 __kasan_check_read+0x11/0x20 mm/kasan/shadow.c:31 instrument_atomic_read include/linux/instrumented.h:68 [inline] _test_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline] test_bit_le include/asm-generic/bitops/le.h:21 [inline] ocfs2_info_freefrag_scan_chain fs/ocfs2/ioctl.c:495 [inline] ocfs2_info_freefrag_scan_bitmap fs/ocfs2/ioctl.c:588 [inline] ocfs2_info_handle_freefrag fs/ocfs2/ioctl.c:662 [inline] ocfs2_info_handle_request+0x1c66/0x3370 fs/ocfs2/ioctl.c:754 ocfs2_info_handle+0x18d/0x2a0 fs/ocfs2/ioctl.c:828 ocfs2_ioctl+0x632/0x6e0 fs/ocfs2/ioctl.c:913 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:597 [inline] __se_sys_ioctl fs/ioctl.c:583 [inline] __x64_sys_ioctl+0x197/0x1e0 fs/ioctl.c:583 ... [CAUSE] ocfs2_info_freefrag_scan_chain() uses on-disk bg_bits directly as the bitmap scan limit. The coherent path reads group descriptors through ocfs2_read_group_descriptor(), which validates the descriptor before use. The non-coherent path uses ocfs2_read_blocks_sync() instead and skips that validation, so an impossible bg_bits value can drive the bitmap walk past the end of the block. [FIX] Compute the bitmap capacity from the filesystem format with ocfs2_group_bitmap_size(), report descriptors whose bg_bits exceeds that limit, and clamp the scan to the computed capacity. This keeps the freefrag report going while avoiding reads beyond the buffer.	2026-06-24	7.1
CVE-2026-53041	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix listxattr handling when the buffer is full [BUG] If an OCFS2 inode has both inline and block-based xattrs, listxattr() can return a size larger than the caller's buffer when the inline names consume that buffer exactly. kernel BUG at mm/usercopy.c:102! Oops: invalid opcode: 0000 [#1] SMP KASAN NOPTI RIP: 0010:usercopy_abort+0xb7/0xd0 mm/usercopy.c:102	2026-06-24	7.1

		Call Trace: __check_heap_object+0xe3/0x120 mm/slub.c:8243 check_heap_object mm/usercopy.c:196 [inline] __check_object_size mm/usercopy.c:250 [inline] __check_object_size+0x5c5/0x780 mm/usercopy.c:215 check_object_size include/linux/ucopysize.h:22 [inline] check_copy_size include/linux/ucopysize.h:59 [inline] copy_to_user include/linux/uaccess.h:219 [inline] listxattr+0xb0/0x170 fs/xattr.c:926 filename_listxattr fs/xattr.c:958 [inline] path_listxattrat+0x137/0x320 fs/xattr.c:988 __do_sys_listxattr fs/xattr.c:1001 [inline] __se_sys_listxattr fs/xattr.c:998 [inline] __x64_sys_listxattr+0x7f/0xd0 fs/xattr.c:998 ... [CAUSE] Commit 936b8834366e ("ocfs2: Refactor xattr list and remove ocfs2_xattr_handler().") replaced the old per-handler list accounting with ocfs2_xattr_list_entry(), but it kept using size == 0 to detect probe mode. That assumption stops being true once ocfs2_listxattr() finishes the inline-xattr pass. If the inline names fill the caller buffer exactly, the block-xattr pass runs with a non-NULL buffer and a remaining size of zero. ocfs2_xattr_list_entry() then skips the bounds check, keeps counting block names, and returns a positive size larger than the supplied buffer. [FIX] Detect probe mode by testing whether the destination buffer pointer is NULL instead of whether the remaining size is zero. That restores the pre-refactor behavior and matches the OCFS2 getxattr helpers. Once the remaining buffer reaches zero while more names are left, the block-xattr pass now returns -ERANGE instead of reporting a size larger than the allocated list buffer.		
CVE-2026-53044	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: soc/tegra: cbb: Fix incorrect ARRAY_SIZE in fabric lookup tables Fix incorrect ARRAY_SIZE usage in fabric lookup tables which could cause out-of-bounds access during target timeout lookup.	2026-06-24	7.1
CVE-2026-53068	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/komeda: fix integer overflow in AFBC framebuffer size check The AFBC framebuffer size validation calculates the minimum required buffer size by adding the AFBC payload size to the framebuffer offset. This addition is performed without checking for integer overflow. If the addition oveflows, the size check may incorrectly succed and allow userspace to provide an undersized drm_gem_object, potentially leading to out-of-bounds memory access. Add usage of check_add_overflow() to safely compute the minimum required size and reject the framebuffer if an overflow is detected. This makes the AFBC size validation more robust against malformed. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2026-06-24	7.1
CVE-2026-53076	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix OOB in pcpu_init_value An out-of-bounds read occurs when copying element from a BPF_MAP_TYPE_CGROUP_STORAGE map to another pcpu map with the same value_size that is not rounded up to 8 bytes. The issue happens when: 1. A CGROUP_STORAGE map is created with value_size not aligned to 8 bytes (e.g., 4 bytes) 2. A pcpu map is created with the same value_size (e.g., 4 bytes) 3. Update element in 2 with data in 1 pcpu_init_value assumes that all sources are rounded up to 8 bytes, and invokes copy_map_value_long to make a data copy, However, the assumption doesn't stand since there are some cases where the source may not be rounded up to 8 bytes, e.g., CGROUP_STORAGE, skb->data. the verifier verifies exactly the size that the source claims, not the size rounded up to 8 bytes by kernel, an OOB happens when the source has only 4 bytes while the copy size(4) is rounded up to 8.	2026-06-24	7.1

CVE-2026-12760	tp-link - multiple products	A denial-of-service (DoS) vulnerability has been identified in Tapo C200 v3 in the network packet handling logic due to improper handling of IPv4 fragmented packets. An unauthenticated adjacent attacker can send crafted packets to cause excessive resource consumption, leading to instability of the device.Successful exploitation can remotely trigger a temporary denial-of-service condition, causing the camera to become unresponsive and resulting in intermittent loss of video monitoring and recording.	2026-06-24	7.1
CVE-2026-9154	gnu - sed	Arbitrary File Write vulnerability in Rapid7 InsightConnect Sed Plugin on Linux allows authenticated attackers to write attacker-controlled content to arbitrary file paths via the expression parameter.	2026-06-25	7.1
CVE-2026-53132	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: fix potential unbounded skb queue virtio_transport_inc_rx_pkt() checks vvs->rx_bytes + len > vvs->buf_alloc. virtio_transport_recv_enqueue() skips coalescing for packets with VIRTIO_VSOCK_SEQ_EOM. If fed with packets with len == 0 and VIRTIO_VSOCK_SEQ_EOM, a very large number of packets can be queued because vvs->rx_bytes stays at 0. Fix this by estimating the skb metadata size: (Number of skbs in the queue) * SKB_TRUESIZE(0)	2026-06-25	7.1
CVE-2026-53146	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Limit XDomain response copy to actual frame size tb_xdomain_copy() copies req->response_size bytes from the received packet buffer regardless of the actual frame size. When a short response arrives, this reads past the valid frame data in the DMA pool buffer into stale contents from previous transactions. Use the minimum of frame size and expected response size for the copy length.	2026-06-25	7.1
CVE-2026-53187	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Validate cpu_id against nr_cpu_ids in DMAH alloc The cpu_id attribute supplied by user space through UVERBS_ATTR_ALLOC_DMAH_CPU_ID is passed directly to cpumask_test_cpu() without first verifying that the value is within the valid CPU range. Passing such untrusted data to cpumask_test_cpu() may lead to an out-of-bounds read of the underlying cpumask bitmap: the helper expands to a test_bit() that indexes the bitmap by cpu_id / BITS_PER_LONG with no bound check. In addition, on kernels built with CONFIG_DEBUG_PER_CPU_MAPS it trips the WARN_ON_ONCE() in cpumask_check(); combined with panic_on_warn this turns a bad user input into a machine reboot. Reject any cpu_id that is not smaller than nr_cpu_ids with -EINVAL before it is used. Reported by Smatch.	2026-06-25	7.1
CVE-2026-53203	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Add buffer overflow check in MS get_info_ioctl Add validation that the info size returned from the metric stream info query is not exceeded when checked against the allocated buffer size. If the firmware returns a size larger than the buffer, reject the operation with -EOVERFLOW instead of proceeding with an incorrect buffer copy.	2026-06-25	7.1
CVE-2026-53205	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Add bounds checks for firmware log indices Add validation that read and write indices in the firmware log buffer are within valid bounds (< data_size) before using them. If out-of-bounds indices are encountered (from firmware), clamp them to safe values instead of proceeding with invalid offsets. This prevents potential out-of-bounds buffer access when firmware supplies invalid log indices.	2026-06-25	7.1
CVE-2026-53223	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: guard timestamp cmsgs to real error queue skbs	2026-06-25	7.1

		skb_is_err_queue() treats PACKET_OUTGOING as the sole marker for an skb from sk_error_queue. That assumption is not true for AF_PACKET sockets: outgoing packet taps are also delivered to packet sockets with skb->pkt_type == PACKET_OUTGOING, but their skb->cb is owned by AF_PACKET instead of struct sock_exterr_skb. If such an skb is received with timestamping enabled, the generic timestamp cmsg path can read AF_PACKET control-buffer state as sock_exterr_skb::opt_stats. With SO_RXQ_OVFL enabled, the packet drop counter overlaps opt_stats. An odd drop count makes the path emit SCM_TIMESTAMPING_OPT_STATS with skb->len and skb->data. For non-linear skbs this copies past the linear head and can trigger hardened usercopy or disclose adjacent heap contents. Keep skb_is_err_queue() local to net/socket.c, but make it verify that the PACKET_OUTGOING marker is paired with the sock_rmem_free destructor installed by sock_queue_err_skb(). AF_PACKET receive skbs use normal receive ownership and no longer pass as error-queue skbs, while legitimate sk_error_queue entries keep the PACKET_OUTGOING marker and sock_rmem_free ownership.		
CVE-2026-53253	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: bnep: reject short frames before parsing A BNEP peer can send a short BNEP SDU. bnep_rx_frame() reads the packet type byte immediately and, for control packets, reads the control opcode and setup UUID-size byte before proving that those bytes are present. bnep_rx_control() also dereferences the control opcode without rejecting an empty control payload. Use skb_pull_data() for the fixed fields in bnep_rx_frame() so a NULL return gates each dereference. Split the control handler so the frame path can pass an opcode that has already been pulled, and keep the byte-buffer wrapper for extension control payloads. For BNEP_SETUP_CONN_REQ, name the UUID-size byte before pulling the setup payload. struct bnep_setup_conn_req carries destination and source service UUIDs after that byte, each uuid_size bytes, so the parser now documents that tuple explicitly instead of leaving the pull length as an opaque multiplication. Validation reproduced this kernel report: KASAN slab-out-of-bounds in bnep_rx_frame.isra.0+0x130c/0x1790 The buggy address belongs to the object at ffff88800c0f7908 which belongs to the cache kmalloc-8 of size 8 The buggy address is located 0 bytes to the right of allocated 1-byte region [ffff88800c0f7908, ffff88800c0f7909) Read of size 1 Call trace: dump_stack_lvl+0xb3/0x140 (?:?) print_address_description+0x57/0x3a0 (?:?) bnep_rx_frame+0x130c/0x1790 (net/bluetooth/bnep/core.c:306) print_report+0xb9/0x2b0 (?:?) __virt_addr_valid+0x1ba/0x3a0 (?:?) srso_alias_return_thunk+0x5/0xfbef5 (?:?) kasan_addr_to_slab+0x21/0x60 (?:?) kasan_report+0xe0/0x110 (?:?) process_one_work+0xfce/0x17e0 (kernel/workqueue.c:3200) worker_thread+0x65c/0xe40 (?:?) __kthread_parkme+0x184/0x230 (?:?) kthread+0x35e/0x470 (?:?) _raw_spin_unlock_irq+0x28/0x50 (?:?) ret_from_fork+0x586/0x870 (?:?) __switch_to+0x74f/0xdc0 (?:?) ret_from_fork_asm+0x1a/0x30 (?:?)	2026-06-25	7.1
CVE-2026-40941	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have a package import signature validation bypass allows which allows self-signed packages. This issue has been fixed in version 1.2.31.	2026-06-25	7.1
CVE-2025-7958	trellix - Trellix Network Security NX, EX, FX, AX, and CMS	A Code Injection vulnerability existed in Trellix Network Security CM and NX. A locally authenticated admin user can execute arbitrary code using the web interface and Alert artifact details.	2026-06-26	7.1
CVE-2026-52969	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KVM: Reject wrapped offset in kvm_reset_dirty_gfn() kvm_reset_dirty_gfn() guards the gfn range with if (!memslot || (offset + __fls(mask)) >= memslot->npages) return;	2026-06-24	7

		but offset is u64 and the addition is unchecked. The check can be silently bypassed by a u64 wrap. The dirty ring backing those entries is MAP_SHARED at KVM_DIRTY_LOG_PAGE_OFFSET of the vcpu fd, so the VMM can rewrite the slot and offset fields of any entry between when the kernel pushes them and when KVM_RESET_DIRTY_RINGS consumes them. On reset, kvm_dirty_ring_reset() re-reads the values via READ_ONCE() and feeds them straight back into this check; only the flags handshake is treated as the handover, the slot/offset payload is taken on trust. Crafting two entries entry[i].offset = 0xfffffffffffffc1 entry[i+1].offset = 0 makes the coalescing loop in kvm_dirty_ring_reset() compute delta = (s64)(0 - 0xfffffffffffffc1) = 63 which falls in [0, BITS_PER_LONG), so it folds entry[i+1] into the existing mask by setting bit 63. The trailing kvm_reset_dirty_gfn() call then sees offset = 0xfffffffffffffc1 and __fls(mask) = 63; the sum is 0 in u64 and the bounds check passes. That offset propagates into kvm_arch_mmu_enable_log_dirty_pt_masked() unchanged. On the legacy MMU path -- kvm_memslots_have_rmaps() == true, i.e. shadow paging, any VM that has allocated shadow roots, or a write-tracked slot -- it reaches gfn_to_rmap(), which indexes slot->arch.rmap[0][] with a near-U64_MAX gfn. That is an out-of-bounds load of a kvm_rmap_head, followed by a conditional clear of PT_WRITABLE_MASK in whatever the loaded pointer points at. The path is reachable from any process holding /dev/kvm. Range-check offset on its own first, so the addition cannot wrap. memslot->npages is bounded well below U64_MAX, so once offset < npages holds, offset + __fls(mask) (with __fls(mask) < BITS_PER_LONG) stays in range.		
CVE-2026-52972	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: af_alg - Cap AEAD AD length to 0x80000000 In order to prevent arithmetic overflows when checking the TX buffer size, cap the associated data length to 0x80000000.	2026-06-24	7
CVE-2026-53143	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix buffer overflow in SDMA queue checkpoint/restore on GFX11 The v11 MQD manager incorrectly assigned the CP-compute variants of checkpoint_mqd/restore_mqd for KFD_MQD_TYPE_SDMA queues. These functions use sizeof(struct v11_compute_mqd) (2048 bytes) instead of sizeof(struct v11_sdma_mqd) (512 bytes), causing a 1536-byte overflow. During CRIU checkpoint of an SDMA queue on Navi3x: - checkpoint_mqd() reads 2048 bytes from a 512-byte SDMA MQD buffer, leaking 1536 bytes of adjacent GTT memory to userspace During CRIU restore: - restore_mqd() writes 2048 bytes into a 512-byte SDMA MQD buffer, corrupting 1536 bytes of adjacent GTT memory (often the ring buffer or neighboring MQDs) This is a copy-paste regression unique to v11. All other ASIC backends (cik, vi, v9, v10, v12) correctly use the SDMA-specific variants. Add checkpoint_mqd_sdma() and restore_mqd_sdma() functions that properly handle the smaller v11_sdma_mqd structure, matching the pattern used in other MQD managers. (cherry picked from commit 6fa41db7ffdec97d62433adf03b7b9b759af8c2c)	2026-06-25	7
CVE-2026-53148	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Clamp XDomain response data copy to allocation size tb_xdp_properties_request() derives the per-packet copy length from the response header without checking that it fits in the previously allocated data buffer. A malicious peer can set its length field larger than the declared data_length, causing memcpy to write past the kcalloc allocation.	2026-06-25	7

		Clamp the per-packet copy length so that the cumulative offset never exceeds data_len.		
CVE-2026-39899	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior are vulnerable to Path Traversal via filename parameter in package_import.php. This issue has been fixed in version 1.2.31.	2026-06-24	6.9
CVE-2026-9718	schneider-electric - powerlogic_p7_firmware	CWE-617 Reachable Assertion vulnerability exists that could allow an authenticated attacker to trigger a denial-of-service condition, impacting system availability when a specially crafted request is sent to a vulnerable network-exposed service.	2026-06-25	6.9
CVE-2026-13083	red hat - multiple products	A flaw was found in the Pen Drive report generator. Cluster-sourced data is rendered into HTML reports without proper escaping or sanitization. An attacker with cluster administrator privileges can inject a stored cross-site scripting (XSS) payload into cluster objects (such as ClusterVersion spec.channel) that executes in the browser of any user who opens the generated HTML report.	2026-06-26	6.9
CVE-2026-10609	red hat - Logging Subsystem for Red Hat OpenShift	A missing authorization flaw was found in the OpenShift Cluster Logging Operator. The operator creates and forwards ServiceAccount tokens to output destinations without verifying that the ClusterLogForwarder creator has permission to use those credentials, allowing a delegated editor to exfiltrate SA tokens and escalate privileges.	2026-06-23	6.8
CVE-2026-53196	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: USB: serial: io_ti: fix heap overflow in get_manuf_info() get_manuf_info() reads le16_to_cpu(rom_desc->Size) bytes from the device I2C EEPROM into a buffer allocated with kmalloc_obj(), which is sizeof(struct edge_ti_manuf_descriptor) = 10 bytes. The Size field comes from the device and is only validated (in check_i2c_image()) to make sure the descriptor fits within TI_MAX_I2C_SIZE (16384 bytes), not against the destination buffer size. A malicious USB device can therefore set Size to any value up to 16377, causing a heap overflow of up to 16367 bytes when plugged into a host running this driver. valid_csum() is called after read_rom() and also iterates buffer[0..Size-1], compounding the out-of-bounds access. Fix by rejecting descriptors with unexpected length before calling read_rom(). [johan: amend commit message; also check for short descriptors]	2026-06-25	6.8
CVE-2026-13282	google - chrome	Use after free in Payments in Google Chrome on Android prior to 149.0.7827.201 allowed a local attacker to potentially exploit heap corruption via physical access to the device. (Chromium security severity: High)	2026-06-25	6.8
CVE-2026-46732	dell - display_and_peripheral_manager	Dell Display and Peripheral Manager (DDPM Mac), versions prior to 2.3, contain a Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Elevation of Privileges.	2026-06-25	6.7
CVE-2024-54178	ibm - multiple products	IBM Db2 on Cloud Pak for Data and Db2 Warehouse on Cloud Pak for Data versions 4.8,5.0,5.1,5.2,5.3 could allow an authenticated user to cause a denial of service when creating new databases due to improper allocation of resources.	2026-06-22	6.5
CVE-2024-51454	ibm - multiple products	IBM Engineering Workflow Management 7.0.2 through 7.0.2 Interim Fix 035, 7.0.3 through 7.0.3 Interim Fix 017, and 7.1 through 7.1 Interim Fix 004 is vulnerable to HTTP header injection, caused by improper validation of input by the HOST headers. This could allow an attacker to conduct various attacks against the vulnerable system, including cross-site scripting, cache poisoning or session hijacking.	2026-06-22	6.5
CVE-2026-11820	redhat - multiple products	A flaw was found in the community.general Ansible collection's nexmo module. The module constructs HTTP requests to the Vonage/Nexmo SMS API by encoding API credentials (api_key and api_secret) into URL query parameters and sending them via GET requests. This causes credentials to be exposed in web server access logs, proxy logs, HTTP Referer headers, and network monitoring tools, despite the Ansible argument specification marking these parameters as no_log. An attacker with access to any of these logging or monitoring points can obtain the full API credentials and gain unauthorized access to the victim's Vonage/Nexmo account.	2026-06-23	6.5
CVE-2026-13022	google - chrome	Inappropriate implementation in Autofill in Google Chrome prior to 149.0.7827.197 allowed a remote attacker who had compromised the renderer process to leak cross-origin data via a crafted HTML page. (Chromium security severity: High)	2026-06-24	6.5
CVE-2026-13208	red hat - multiple products	A flaw was found in KubeVirt's virt-handler domain notify server. The gRPC handlers for HandleDomainEvent and HandleK8SEvent derive the VMI identity (namespace/name) solely from the request body without validating it against the connection's origin. Each virt-launcher pod connects through a per-VMI pipe socket, but no identity tag is propagated from the pipe path to the server handlers. This allows a compromised virt-launcher process to send forged domain lifecycle events for any other VMI scheduled on the same node, causing virt-handler to erroneously update that VMI's state and disrupt its lifecycle management.	2026-06-24	6.5
CVE-2026-9153	gnu - sed	Arbitrary File Read vulnerability in Rapid7 InsightConnect Sed Plugin on Linux allows authenticated attackers to read arbitrary files via the expression parameter due to insufficient input validation.	2026-06-25	6.5
CVE-2026-9705	redhat - multiple products	A flaw was found in Keycloak's client registration service. A remote attacker, possessing a previously issued Registration Access Token (RAT), could exploit this vulnerability to re-enable a client that an administrator had explicitly disabled. This bypasses security controls, allowing the attacker to reset the client's secret and potentially regain privileged API access. The	2026-06-25	6.5

		primary impact includes unauthorized information disclosure and potential integrity compromise.		
CVE-2026-40084	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior are vulnerable to Path Traversal through the Report format_file Parameter, causing arbitrary file read. This vulnerability occurs in two stages. In the first stage (stored injection), lib/html_reports.php at line 283 stores $save['format_file'] = $post['format_file'] directly into the database without any validation. In the second stage (file read), lib/reports.php at line 667 concatenates CACTI_PATH_FORMATS . '/' . $format_file, and line 670 then calls file($format_file), reading arbitrary files from the filesystem. This issue has been fixed in version 1.2.31.	2026-06-25	6.5
CVE-2026-12993	red hat - multiple products	A flaw was found in Apicurio Registry. The DocumentBuilderAccessor correctly blocks external DTD and schema access but does not disable DOCTYPE declarations or enable FEATURE_SECURE_PROCESSING. An attacker with artifact-write permission can upload XML documents with internal entity-expansion payloads (billion-laughs variant) that cause CPU and heap exhaustion, partially mitigated by the JAXP default 64,000 entity-expansion limit.	2026-06-26	6.5
CVE-2026-57914	apache software foundation - Apache Kerby	By sending a deeply nested ASN1 structure to a Apache Kerby client or service, it's possible to trigger a StackOverFlow Exception which can lead to denial of service issues. Users are recommended to upgrade to version 2.1.2, which fixes this issue.	2026-06-26	6.5
CVE-2026-54226	apache software foundation - Apache Kvrocks	A vulnerability in Apache Kvrocks. This issue affects Apache Kvrocks: from 2.6.0 through 2.15.0. Users are recommended to upgrade to version 2.16.0, which fixes the issue.	2026-06-25	6.4
CVE-2026-13318	red hat - multiple products	A server-side request forgery (SSRF) flaw was found in KubeVirt's virt-api port-forward handler. When processing a port-forward request to a VirtualMachineInstance (VMI), virt-api reads the target IP from vmi.Status.Interfaces[0].IP and passes it directly to net.Dial() without validation. For VMIs using non-masquerade network bindings (bridge or secondary-only), this IP is reported by the QEMU guest agent running inside the VM and is fully controllable by the VM owner. An attacker with kubevirt.io:edit permissions can create a VM with a modified guest agent that reports an arbitrary IP address, then request port-forward to establish a bidirectional TCP tunnel from virt-api's cluster-internal network position to any routable destination, bypassing NetworkPolicy isolation.	2026-06-26	6.4
CVE-2026-54665	apache - nifi	Apache NiFi 0.0.1 through 2.9.0 support building qualified URLs from one of several HTTP request headers that provide an alternative to the standard Host header without validating the values provided. Apache NiFi 1.6.0 introduced a configurable application property to restrict values provided in the HTTP Host header, but did not apply the validation to alternative Proxy and Forwarded headers. The absence of proxy host header validation allowed a client to instruct Apache NiFi web services to construct invalid qualified URLs for redirection or data references. Upgrading to Apache NiFi 2.10.0 is the recommended mitigation, which implements validation for the X-ProxyHost and X-Forwarded-Host HTTP request headers based on the nifi.web.proxy.host property. Enabling header validation requires configuring the application with HTTPS. Reverse proxy servers in front of Apache NiFi are responsible for filtering input request headers and providing allowed values to the application.	2026-06-22	6.3
CVE-2026-11877	microfocus - multiple products	An unauthorized user can modify configuration through API calls that affects the OpenText Access Manager. This issue affects Access Manager before 5.1.3.	2026-06-24	6.3
CVE-2026-53059	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dm log: fix out-of-bounds write due to region_count overflow The local variable region_count in create_log_context() is declared as unsigned int (32-bit), but dm_sector_div_up() returns sector_t (64-bit). When a device-mapper target has a sufficiently large ti->len with a small region_size, the division result can exceed UINT_MAX. The truncated value is then used to calculate bitset_size, causing clean_bits, sync_bits, and recovering_bits to be allocated far smaller than needed for the actual number of regions. Subsequent log operations (log_set_bit, log_clear_bit, log_test_bit) use region indices derived from the full untruncated region space, causing out-of-bounds writes to kernel heap memory allocated by vmalloc. This can be reproduced by creating a mirror target whose region_count overflows 32 bits: dmsetup create bigzero --table '0 8589934594 zero' dmsetup create mymirror --table '0 8589934594 mirror \ core 2 2 nosync 2 /dev/mapper/bigzero 0 \ /dev/mapper/bigzero 0' The status output confirms the truncation (sync_count=1 instead of 4294967297, because 0x100000001 was truncated to 1): $ dmsetup status mymirror 0 8589934594 mirror 2 254:1 254:1 1/4294967297 ... This leads to a kernel crash in core_in_sync: BUG: scheduling while atomic: (udev-worker)/9150/0x00000000 RIP: 0010:core_in_sync+0x14/0x30 [dm_log] CR2: 0000000000000008	2026-06-24	6.3

		Fixing recursive fault but reboot is needed! Fix by widening the local region_count to sector_t and adding an explicit overflow check before the value is assigned to lc->region_count.		
CVE-2026-9073	red hat - Red Hat Satellite 6.19	A flaw was found in foreman-mcp-server. This component utilizes two distinct logging mechanisms that can expose sensitive session and authentication data. One mechanism logs session identifiers, which are treated as authentication credentials, at an informational level. The other, when debug logging is enabled, incompletely sanitizes HTTP request headers, leading to the cleartext logging of sensitive information such as authorization tokens and API keys. This vulnerability can result in a confidentiality breach, as sensitive authentication data is persisted in plain text within container logs, increasing the risk if logs are forwarded to a centralized platform.	2026-06-23	6.2
CVE-2026-8059	ibm - multiple products	IBM Datacap 9.1.7, 9.1.8, and 9.1.9 and IBM Datacap Navigator 9.1.7, 9.1.8, and 9.1.9 is vulnerable to cross-site scripting. This vulnerability allows an unauthenticated attacker to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session.	2026-06-22	6.1
CVE-2026-53765	google - chrome-devtools-mcp	Chrome DevTools for agents (chrome-devtools-mcp) lets your coding agent control and inspect a live Chrome browser. From 0.20.0 until 1.1.0, The chrome-devtools-mcp daemon writes its PID file with fs.writeFileSync() to a deterministic runtime path. On typical macOS environments, and on Linux sessions where $XDG_RUNTIME_DIR is unset, that runtime path falls back to /tmp/chrome-devtools-mcp-<uid>/daemon.pid. Because the write does not use O_NOFOLLOW, a local low-privilege user on the same POSIX host can pre-create /tmp/chrome-devtools-mcp-<victim_uid>/daemon.pid as a symlink to a file writable by the victim. When the victim later starts daemon mode, fs.writeFileSync() follows the symlink and truncates the target file to the daemon PID string. This vulnerability is fixed in 1.1.0.	2026-06-24	6.1
CVE-2026-53766	google - chrome-devtools-mcp	Chrome DevTools for agents (chrome-devtools-mcp) lets your coding agent control and inspect a live Chrome browser. From 0.24.0 until 1.1.0, McpContext.validatePath() enforces workspace roots by checking whether path.resolve(filePath) textually falls under one of the configured root paths. path.resolve() does not canonicalize symbolic links. As a result, a symlink inside a configured workspace root can point to a file outside that root, pass validation, and then be followed by downstream file read/write operations. This bypass applies even when the MCP client correctly declares the roots capability with a non-empty list. It is separate from the documented legacy behavior where missing roots capability allows all paths. The practical impact is a workspace-boundary bypass. In the write direction, filePath-writing tools can overwrite out-of-root files through an in-root symlink. In the read direction, upload_file can read through the symlink and send the file to the currently selected web page. This vulnerability is fixed in 1.1.0.	2026-06-24	6.1
CVE-2026-40080	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior are vulnerable to Open Redirect through a substring check rather than a host check at str_contains($referer, CACTI_PATH_URL). When the user's login_opts == '1' (redirect to referer after login), the function used $_SERVER['HTTP_REFERER'] directly. An attacker could craft a referer such as https://evil.com/cacti/. Where CACTI_PATH_URL is /cacti/, the substring matches and the user is redirected to evil.com after login. The pre-existing validate_redirect_url() helper at lib/html_utility.php performed proper validation but was not invoked from auth_login_redirect(). This issue has been fixed in version 1.2.31.	2026-06-25	6.1
CVE-2025-2669	ibm - multiple products	IBM Db2 on Cloud Pak for Data and Db2 Warehouse on Cloud Pak for Data versions 4.8, 5.0, 5.1, 5.2, 5.3 could allow a privileged user to perform operations and obtain sensitive information outside of their authority due to improper token validation.	2026-06-22	6
CVE-2026-44273	dell - wyse_management_suite	Dell Wyse Management Suite (WMS), versions prior to WMS 2605, contain a Use of Default Credentials vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Information Disclosure.	2026-06-22	6
CVE-2026-12725	red hat - multiple products	A heap-based buffer overflow was found in dnsmasq. When DNSSEC validation and query logging are both enabled, logging of DS or DNSKEY replies containing unsupported algorithm or digest types can cause dnsmasq to write past the end of an internal logging buffer. A remote attacker able to supply such a DNS response may crash the dnsmasq process, resulting in denial of service.	2026-06-22	5.9
CVE-2026-9320	ibm - multiple products	IBM WebSphere Application Server 9.0, and 8.5 and IBM WebSphere Application Server - Liberty 17.0.0.3 through 26.0.0.6 are vulnerable to a denial of service, caused by sending a specially-crafted request. A remote attacker could exploit this vulnerability to cause the server to consume memory resources.	2026-06-22	5.9
CVE-2026-10852	ibm - multiple products	IBM WebSphere Application Server and IBM WebSphere Application Server Liberty are vulnerable to denial of service in the WebSphere WebServer Plug-in component when an attacker can pass crafted requests to the web server.	2026-06-22	5.9
CVE-2026-8636	ibm - multiple products	IBM Datacap 9.1.7, 9.1.8, and 9.1.9 and IBM Datacap Navigator 9.1.7, 9.1.8, and 9.1.9 allows an attacker to retrieve user passwords and cryptographic keys from memory. Attacker can use the same keys to decrypt password, gain access to the application and access sensitive data in the database.	2026-06-22	5.5
CVE-2020-9711	adobe - multiple products	Acrobat Reader versions 2020.009.20074, 2020.001.30002, 2017.011.30171, 2015.006.30523 and earlier are affected by an out-of-bounds read vulnerability that could lead to disclosure of sensitive memory. An attacker could leverage this vulnerability to disclose sensitive information. Exploitation of this issue requires user interaction in that a victim must open a malicious file.	2026-06-23	5.5
CVE-2020-9713	adobe - multiple products	Adobe Acrobat and Reader versions 2020.009.20074 and earlier, 2020.001.30002, 2017.011.30171 and earlier, and 2015.006.30523 and earlier are affected by an out-of-bounds read vulnerability that could lead to disclosure of sensitive memory. An attacker could leverage this vulnerability to disclose sensitive information. Exploitation of this issue requires user interaction in that a victim must open a malicious file.	2026-06-23	5.5
CVE-2026-11819	red hat - multiple products	Module: plugins/modules/keyring_info.py CVSS 3.1: 5.5 MEDIUM — AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	2026-06-23	5.5

		Issue: The module retrieves a passphrase from the OS native keyring (GNOME Keyring, macOS Keychain, Windows Credential Manager) and places it directly into result["passphrase"] with no output suppression, no no_log protection, and no documentation warning.
		
		Root Cause:
		
		Line 105 (protected): keyring_password=dict(type="str", required=True, no_log=True)
		Line 127 (NOT protected): result["passphrase"] = passphrase
		
		Observed Output:
		
		{
		"changed": false,
		"passphrase": "MyMasterP@ssw0rd!SSH_Key_Secret"
		}
		Visible via register + debug:
		{
		"keyring_result": {
		"changed": false,
		"passphrase": "MyMasterP@ssw0rd!SSH_Key_Secret"
		}
		}
		
		Impact:
		
		Master passwords, SSH key passphrases and service credentials appear in all Ansible output
		
		register: keyring_result followed by debug: var=keyring_result prints passphrase in full
		
		Ansible fact caching backends (Redis, JSON file, memcached) may persist the passphrase
		
		AWX/Tower job logs silently store the live credential
		
		Fix:
		
		module.exit_json(changed=False, passphrase=passphrase, _ansible_no_log=True)
		
		Also add a documentation warning requiring callers to use no_log: true at the task level.
		
		PoCs
		
		Fig 1: PoC execution showing passphrase in plaintext output
		
		Fig 2: Source code showing no_log=True on input (line 105) vs unprotected output (line 127)		

CVE-2026-46751	apache software foundation - Apache Kvrocks	A vulnerability in Apache Kvrocks. This issue affects Apache Kvrocks: from 2.2.0 through 2.15.0. Users are recommended to upgrade to version 2.16.0, which fixes the issue.	2026-06-25	5.5
CVE-2026-53204	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: firmware: stratix10-rsu: Fix NULL deref on rsu_send_msg() timeout in probe rsu_send_msg() can return -ETIMEDOUT when wait_for_completion_interruptible_timeout() fires while the SMC call is still pending. In stratix10_rsu_probe(), the error paths for COMMAND_RSU_DCMF_VERSION, COMMAND_RSU_DCMF_STATUS, COMMAND_RSU_MAX_RETRY and COMMAND_RSU_GET_SPT_TABLE call stratix10_svc_free_channel() - which sets chan->scl to NULL - but then fall through and queue the next request on the same channel. The next svc kthread that runs will dereference pdata->chan->scl in its receive callback path, triggering a NULL pointer dereference identical to the one fixed by commit c45f7263100c ("firmware: stratix10-rsu: Fix NULL pointer dereference when RSU is disabled") for the COMMAND_RSU_STATUS path. Apply the same cleanup pattern to the remaining failure paths: remove the async client, free the channel, and return early so no further messages are queued on a channel whose scl has been cleared. While at it, clean up stratix10_rsu_probe() in two ways without changing behavior: - Drop redundant zero-initialization of fields already cleared by devm_kzalloc(): client.receive_cb, status.* and spt0/1_address (INVALID_SPT_ADDRESS is 0x0). - Replace five identical 3-line error-cleanup blocks (stratix10_svc_remove_async_client() + stratix10_svc_free_channel() + return ret) with goto labels (remove_async_client, free_channel),	2026-06-25	5.5

		matching the standard kernel resource-unwinding pattern and making it easier to extend the probe sequence without forgetting matching cleanup. Also move init_completion() next to mutex_init() so sync-primitive initialization is grouped before anything that could trigger a callback. --- v2: Add a minor clean-up of the function stratix10_rsu_probe() to have a centralize exit for all the rsu_send_async_msg() and rsu_send_msg().		
CVE-2026-53206	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Add bounds check for firmware runtime memory Validate that the firmware runtime memory specified in the image header is properly aligned and sized to hold the firmware image. This prevents errors during memory allocation and image transfer.	2026-06-25	5.5
CVE-2026-53207	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/memory-failure: fix hugetlb_lock AA deadlock in get_huge_page_for_hwpoison Two concurrent madvise(MADV_HWPOISON) calls on the same hugetlb page can trigger a recursive spinlock self-deadlock (AA deadlock) on hugetlb_lock when racing with a concurrent unmap: thread#0 thread#1 -------- -------- madvise(folio, MADV_HWPOISON) -> poisons the folio successfully madvise(folio, MADV_HWPOISON) unmap(folio) try_memory_failure_hugetlb get_huge_page_for_hwpoison spin_lock_irq(&hugetlb_lock) <- held __get_huge_page_for_hwpoison hugetlb_update_hwpoison() -> MF_HUGETLB_FOLIO_PRE_POISONED goto out: folio_put() refcount: 1 -> 0 free_huge_folio() spin_lock_irqsave(&hugetlb_lock) -> AA DEADLOCK! The out: path in __get_huge_page_for_hwpoison() calls folio_put() to drop the GUP reference while the hugetlb_lock is still held by the hugetlb.c wrapper get_huge_page_for_hwpoison(). If concurrent unmap has released the page table mapping reference, folio_put() drops the folio refcount to zero, triggering free_huge_folio() which attempts to re-acquire the non-recursive hugetlb_lock. Fix this by moving hugetlb_lock acquisition from the hugetlb.c wrapper into get_huge_page_for_hwpoison(). Place spin_unlock_irq() before the folio_put() at the out: label so the folio is always released outside the lock. [akpm@linux-foundation.org: fix race, rename label per Miaohe]	2026-06-25	5.5
CVE-2026-53208	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: reject BR/EDR signaling packets over MTUsig net/bluetooth/l2cap_core.c:l2cap_sig_channel() accepts BR/EDR signaling packets up to the channel MTU and dispatches each command without enforcing the signaling MTU (MTUsig). A Bluetooth BR/EDR peer within radio range can send a fixed-channel CID 0x0001 packet that is larger than MTUsig and contains many L2CAP_ECHO_REQ commands before pairing. In a real-radio stock-kernel run, one 681-byte signaling packet containing 168 zero-length ECHO_REQ commands made the target transmit 168 ECHO_RSP frames over about 220 ms. Impact: a Bluetooth BR/EDR peer within radio range, before pairing, can force 168 ECHO_RSP frames from one 681-byte fixed-channel signaling packet containing packed ECHO_REQ commands. Define Linux's BR/EDR signaling MTU as the spec minimum of 48 bytes and reject any larger signaling packet with one L2CAP_COMMAND_REJECT_RSP carrying L2CAP_REJ_MTU_EXCEEDED before any command is dispatched. The Bluetooth Core spec wording for MTUExceeded says the reject identifier shall match the first request command in the packet, and that packets containing only responses shall be silently discarded.	2026-06-25	5.5

		Linux intentionally deviates from that prescription: silently discarding desynchronizes the peer because the remote stack never learns its responses were dropped, and locating the first request command requires walking command headers past MTUsig, i.e. processing bytes from a packet we have already decided is too large to process. We therefore always emit one reject and use the identifier from the first command header, a single fixed-offset byte read. The unrestricted BR/EDR signaling parser and ECHO_REQ response path both trace to the initial git import; no later introducing commit is available for a Fixes tag.		
CVE-2026-53210	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tee: shm: fix shm leak in register_shm_helper() register_shm_helper() allocates shm before calling iov_iter_npages(). If iov_iter_npages() returns 0, the function jumps to err_ctx_put and leaks shm. This can be triggered by TEE_IOC_SHM_REGISTER with struct tee_ioctl_shm_register_data where length is 0. Jump to err_free_shm instead.	2026-06-25	5.5
CVE-2026-53211	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_meta_bridge: fix stale stack leak via IIFHWADDR register NFT_META_BRI_IIFHWADDR declares its destination register with len = ETH_ALEN (6 bytes), which the register-init tracking rounds up to two 32-bit registers (8 bytes). nft_meta_bridge_get_eval() then does memcpy(dest, br_dev->dev_addr, ETH_ALEN), writing only 6 bytes and leaving the upper 2 bytes of the second register as uninitialised nft_do_chain() stack. A downstream load of that register span leaks those stale bytes to userspace. Zero the second register before the memcpy so the full declared span is written.	2026-06-25	5.5
CVE-2026-53213	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: fix krealloc() memory leak Don't just overwrite the original pointer passed to krealloc() with its return value without checking latter: MEM = krealloc(MEM, SZ, GFP); If krealloc() returns NULL, that erases the pointer to the still allocated memory, hence leaks this memory. Instead, use a temporary variable, check it's not NULL and only then assign it to the original pointer: TMP = krealloc(MEM, SZ, GFP); if (!TMP) return; MEM = TMP; While on it, use krealloc_array().	2026-06-25	5.5
CVE-2026-53214	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix a potential NPD in cleanup_prefix_route() addrconf_get_prefix_route() can return the fib6_null_entry sentinel entry which has a NULL fib6_table pointer. Therefore, before setting the route's expiration time, check that we are not working with this entry, as otherwise a NPD will be triggered [1]. Note that the other callers of addrconf_get_prefix_route() are not susceptible to this bug: 1. addrconf_prefix_rcv(): Requests a route with the 'RTF_ADDRCONF | RTF_PREFIX_RT' flags which are not set on fib6_null_entry. 2. modify_prefix_route(): Fixed by commit a747e02430df ("ipv6: avoid possible NULL deref in modify_prefix_route()"). 3. __ipv6_ifa_notify(): Calls ip6_del_rt() which specifically checks for fib6_null_entry and returns an error. [1] Oops: general protection fault, probably for non-canonical address 0xdffffc0000000006: 0000 [#1] SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000030-0x0000000000000037]	2026-06-25	5.5

		[...] Call Trace: <TASK> __kasan_check_byte (mm/kasan/common.c:573) lock_acquire.part.0 (kernel/locking/lockdep.c:5842 (discriminator 1)) _raw_spin_lock_bh (kernel/locking/spinlock.c:182 (discriminator 1)) cleanup_prefix_route (net/ipv6/addrconf.c:1280) ipv6_del_addr (net/ipv6/addrconf.c:1342) inet6_addr_del.isra.0 (net/ipv6/addrconf.c:3119) inet6_rtm_deladdr (net/ipv6/addrconf.c:4812) rtnetlink_rcv_msg (net/core/rtnetlink.c:6997) netlink_rcv_skb (net/netlink/af_netlink.c:2555) netlink_unicast (net/netlink/af_netlink.c:1344) netlink_sendmsg (net/netlink/af_netlink.c:1899) __sock_sendmsg (net/socket.c:802 (discriminator 4)) ____sys_sendmsg (net/socket.c:2698) ___sys_sendmsg (net/socket.c:2752) __sys_sendmsg (net/socket.c:2784) do_syscall_64 (arch/x86/entry/syscall_64.c:63 arch/x86/entry/syscall_64.c:94) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:121)		
CVE-2026-53218	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_exthdr: fix register tracking for F_PRESENT flag nft_exthdr_init() passes user-controlled priv->len to nft_parse_register_store(), which marks that many bytes in the register bitmap as initialized. However, when NFT_EXTHDR_F_PRESENT is set, the eval paths write only 1 byte (nft_reg_store8) or 4 bytes (*dest = 0 on TCP/DCCP error path). When len > 4, registers beyond the first are never written, retaining uninitialized stack data from nft_regs. Bail out if userspace requests too much data when F_PRESENT is set.	2026-06-25	5.5
CVE-2026-53219	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: avoid leaking percpu counter pointers The native and compat get-entries paths copy the fixed rule entry header from the kernelized rule blob to userspace before overwriting the entry's counter fields with a sanitized counter snapshot. On SMP kernels, entry->counters.pcnt contains the percpu allocation address used by x_tables rule counters. A caller can provide a userspace buffer that faults during the initial fixed-header copy after pcnt has been copied but before the later sanitized counter copy runs. The syscall then returns -EFAULT while leaving the raw percpu pointer in userspace. Copy only the fixed entry prefix before counters from the kernelized rule blob, then copy the sanitized counter snapshot into the counter field. Apply this ordering to the IPv4, IPv6, and ARP native and compat get-entries implementations so a fault cannot expose the internal percpu counter pointer.	2026-06-25	5.5
CVE-2026-53220	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: revalidate bridge ports ebt_redirect_tg() dereferences br_port_get_rcu() return without a NULL check, causing a kernel panic when the bridge port has been removed between the original hook invocation and an NFQUEUE reinject. A mere NULL check isn't sufficient, however. As sashiko review points out userspace can not only remove the port from the bridge, it could also place the device in a different virtual device, e.g. macvlan. If this happens, we must drop the packet, there is no way for us to reinject it into the bridge path. Switch to _upper API, we don't need the bridge port structure. Also, this fix keeps another bug intact: Both nfnetlink_log and nfnetlink_queue use CONFIG_BRIDGE_NETFILTER too aggressive, which prevents certain logging features when queueing in bridge family: NETFILTER_FAMILY_BRIDGE can be enabled while the old CONFIG_BRIDGE_NETFILTER cruft is off. Fixes tag is a common ancestor, this was always broken.	2026-06-25	5.5
CVE-2026-53222	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ptp: ocp: fix resource freeing order	2026-06-25	5.5

		Commit a60fc3294a37 ("ptp: rework ptp_clock_unregister() to disable events") added a call to ptp_disable_all_events() which changes the configuration of pins if they support EXTTS events. In ptp_ocp_detach() pins resources are freed before ptp_clock_unregister() and it leads to use-after-free during driver removal. Fix it by changing the order of free/unregister calls. To avoid irq handler running on the other core while ptp device unregistering, call synchronize_irq() after HW is configured to stop producing irqs and no irqs are in-flight.		
CVE-2026-53226	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio: rockchip: fix generic IRQ chip leak on remove The driver allocates domain generic chips using irq_alloc_domain_generic_chips() during probe. However, on driver remove/teardown, the generic chips are not automatically freed when the IRQ domain is removed because the domain flags do not include IRQ_DOMAIN_FLAG_DESTROY_GC. This causes both the domain generic chips structure and the associated generic chips to be leaked. Additionally, the generic chips remain on the global gc_list and may later be visited by generic IRQ chip suspend, resume, or shutdown callbacks after the GPIO bank has been removed, potentially resulting in a use-after-free and kernel crash. Fix the resource leak by explicitly calling irq_domain_remove_generic_chips() before removing the IRQ domain in rockchip_gpio_remove().	2026-06-25	5.5
CVE-2026-53227	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix possible kfree_skb of ERR_PTR After the patch in the "Fixes" tag, the allocation of the "reply" skb can happen either before or after locking the ovs_mutex. However, error cleanups still follow the classical reversed order, assuming "reply" is allocated before locking: it is freed after unlocking. If "reply" allocation happens after locking the mutex and it fails, "reply" is left with an ERR_PTR, and execution jumps to the correspondent cleanup stage which will try to free an invalid pointer. Fix this by setting the pointer to NULL after having saved its error value.	2026-06-25	5.5
CVE-2026-53231	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: phy: don't try to setup PHY-driven SFP cages when using genphy We don't have support for PHY-driver SFP cages with the genphy code. On top of that, it was found by sashiko that running sfp_bus_add_upstream() for genphy deadlocks, as for genphy the PHY probing runs under RTNL, which isn't the case for non-genphy drivers. This problem was reproduced, and does lead to a deadlock on RTNL. Before the blamed commit, the phy_sfp_probe() call was made by individual PHY drivers, so there was no way to get to the SFP probing path when using genphy. Let's therefore only run phy_sfp_probe when not using genphy.	2026-06-25	5.5
CVE-2025-62198	apache - atlas	An authenticated user can perform XSS. This issue affects Apache Atlas versions 2.4.0 and earlier. Users are recommended to upgrade to version 2.5.0, which fixes the issue.	2026-06-22	5.4
CVE-2025-33128	ibm - multiple products	IBM Engineering Workflow Management 7.0.3 through 7.0.3 Interim Fix 020, and 7.1 through 7.1 Interim Fix 007 is vulnerable to cross-site scripting. This vulnerability allows an authenticated user to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session.	2026-06-22	5.4
CVE-2026-10601	grafana - grafana	The Tempo and Loki datasource plugins construct backend HTTP requests by interpolating user-supplied input into URL paths without sanitization, enabling path traversal. A Viewer-role user can: (1) capture admin-configured datasource credentials (secureJsonData custom headers) by traversing to an attacker-controlled endpoint, (2) invoke state-changing admin endpoints on Tempo (e.g. /flush, /shutdown), and (3) exfiltrate internal service data via Loki's CallResource which returns full HTTP response bodies.	2026-06-22	5.4
CVE-2026-11372	ibm - multiple products	IBM TRIRIGA Application Platform 5.0.2 through 5.0.3 is vulnerable to cross-site scripting. This vulnerability allows an authenticated user to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session.	2026-06-22	5.4

CVE-2026-57294	jenkins - ec2_fleet	A missing permission check in Jenkins EC2 Fleet Plugin 4.2.3.539.v8fedff2a_81c3 and earlier allows attackers with Overall/Read permission to connect to an attacker-specified URL using attacker-specified credentials IDs obtained through another method, capturing AWS credentials stored in Jenkins.	2026-06-24	5.4
CVE-2026-57295	jenkins - ec2_fleet	A cross-site request forgery (CSRF) vulnerability in Jenkins EC2 Fleet Plugin 4.2.3.539.v8fedff2a_81c3 and earlier allows attackers to connect to an attacker-specified URL using attacker-specified credentials IDs obtained through another method, capturing AWS credentials stored in Jenkins.	2026-06-24	5.4
CVE-2026-57304	jenkins - assembla	A missing permission check in Jenkins Assembla Plugin 1.4 and earlier allows attackers with Overall/Read permission to connect to an attacker-specified URL using an attacker-specified username and password.	2026-06-24	5.4
CVE-2026-57305	jenkins - assembla	A cross-site request forgery (CSRF) vulnerability in Jenkins Assembla Plugin 1.4 and earlier allows attackers to connect to an attacker-specified URL using an attacker-specified username and password.	2026-06-24	5.4
CVE-2026-40082	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior have missing session_regenerate_id() after login, leading to Session Fixation. session_regenerate_id() is NOT called after successful login. The login flow at auth_login.php:203-207 directly sets $_SESSION[SESS_USER_ID] without rotating the session ID. The session cookie configuration is otherwise good (httponly=true, samesite=Strict, secure=true for HTTPS at include/global.php:513-537), but these do not prevent session fixation via same-site vectors. This issue has been fixed in version 1.2.31.	2026-06-25	5.4
CVE-2023-33854	ibm - multiple products	IBM Db2 on Cloud Pak for Data and Db2 Warehouse on Cloud Pak for Data versions 4.8, 5.0, 5.1, 5.2, and 5.3 could allow an authenticated user to bypass client-side validation and manipulate input data using man in the middle techniques.	2026-06-22	5.3
CVE-2026-7253	ibm - multiple products	IBM Watson Speech Services Cartridge is vulnerable to Server-Side Request Forgery (SSRF) in Sterling File Gateway, due to a flaw which may allow an authenticated attacker to send unauthorized requests from the system, potentially leading to network enumeration or facilitating other attacks [GHSA-rr7j-v2q5-chgv] [CVE-2026-7253]. IBM Sterling File Gateway is used in our speech runtimes. This vulnerabilitiy has been addressed. Please read the details for remediation below.	2026-06-22	5.3
CVE-2026-12969	red hat - multiple products	An out-of-bounds read vulnerability exists in dnsmasq's find_soa() function in src/rfc1035.c. When parsing NS section records, extract_name() is called with extrabytes=0, failing to validate that 10 additional bytes exist for fixed-length DNS record fields. A remote attacker controlling a DNS zone can exploit this via a crafted NXDOMAIN response to cause a 10-byte heap out-of-bounds read, potentially accessing stale data from prior transactions.	2026-06-23	5.3
CVE-2026-13023	google - chrome	Uninitialized Use in GPU in Google Chrome prior to 149.0.7827.197 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: High)	2026-06-24	5.3
CVE-2026-13030	google - chrome	Uninitialized Use in GPU in Google Chrome on Android prior to 149.0.7827.197 allowed a remote attacker to obtain potentially sensitive information from process memory via a crafted HTML page. (Chromium security severity: High)	2026-06-24	5.3
CVE-2026-39897	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and below contain a Reflected XSS vulnerability in the html_auth_footer. This issue has been fixed in version 1.2.31.	2026-06-24	5.3
CVE-2026-39900	cacti - cacti	Cacti is an open source performance and fault management framework. Versions 1.2.30 and prior are vulnerable to Reflected XSS via tab parameter in the auth_profile.php JavaScript context. This issue has been fixed in version 1.2.31.	2026-06-24	5.3
CVE-2026-28898	apple - swiftnio_http\/2	swift-nio-http2's HTTP/2-to-HTTP/1.1 codec did not validate pseudo-header values for control characters before placing them into the translated HTTP/1.1 message. swift-nio-http2 1.44.1 adds validation of all pseudo-header values (:path, :authority, :scheme, :method, and :status) at both the HPACK header validation layer and the HTTP/2-to-HTTP/1.1 translation layer. Requests or responses containing CR, LF, or NUL bytes in any pseudo-header value are now rejected with a connection error. This issue is fixed in swift-nio-http2 1.44.1.	2026-06-25	5.3
CVE-2026-44913	apache - nifi	Improper escaping of database table names in the CaptureChangeMySQL Processor included with Apache NiFi 1.2.0 through 2.9.0 allows for injecting SQL commands using crafted naming. Manual quoted boundaries added in Apache NiFi 1.8.0 narrowed the scope of potential injection options, but did not cover additional strategies. Apache NiFi installations that do not use the CaptureChangeMySQL Processor are not subject to this vulnerability. Upgrading to Apache NiFi 2.10.0 is the recommended mitigation, which incorporates more robust identifier escaping.	2026-06-22	5.2
CVE-2026-55655	openbsd - multiple products	A flaw was found in OpenSSH. A local unprivileged attacker on a Linux client host can hijack client-side X11 forwarding connections. This is possible by pre-binding the preferred abstract X socket name when X11 forwarding is enabled and a local UNIX-domain X socket is used. A successful attack can compromise the confidentiality of forwarded X11 traffic, including sensitive window contents and input, and may allow some manipulation of the forwarded session.	2026-06-23	5
CVE-2026-57282	jenkins - git_client	Jenkins Git client Plugin 6.6.0 and earlier does not correctly escape the workspace directory name when it is embedded into a generated SSH wrapper script, allowing attackers able to control the name of a build's working directory to execute arbitrary operating system commands on the agent.	2026-06-24	5
CVE-2026-9083	redhat - multiple products	A flaw was found in Keycloak. A realm administrator with the "manage-realm" role can exploit this vulnerability by submitting an arbitrary filesystem path as a keystore parameter when creating a key provider component. This allows the administrator to probe arbitrary filesystem paths, determining which files exist and are readable by the Keycloak process. This information disclosure could be used to identify high-value targets for follow-on attacks.	2026-06-25	4.9
CVE-2026-13434	red hat - Red Hat OpenShift Virtualization 4	A flaw was found in KubeVirt's network annotation generator. When a tenant creates a VirtualMachineInstance with a Multus network configuration, the supplied networkName value is written verbatim into the launcher pod's v1.multus-cni.io/default-network annotation without format validation or sanitization. The only admission check rejects empty strings; no DNS-1123 format validation, JSON detection, or special character rejection is performed. When the ExternalNetResourceInjection Beta feature gate is enabled (off by default, cluster-	2026-06-26	4.9

		admin only), the NAD lookup that would otherwise catch malformed names is skipped by design. A tenant with kubevirt.io:edit permissions can inject a JSON-formatted NetworkSelectionElement array specifying an arbitrary namespace, NAD name, static IP address, and MAC address. Multus on the node parses this JSON and attaches the launcher pod to the specified network attachment in any namespace, enabling cross-namespace network access and IP/MAC impersonation on network segments normally segregated from tenant workloads. The ExternalNetResourceInjection feature gate was introduced in KubeVirt v1.8.0 (first shipped in OpenShift Virtualization 4.21).		
CVE-2026-12549	red hat - multiple products	The fix for CVE-2026-2443 was regressed by a subsequent rework commit that replaced specific overflow checks with a general signed comparison. When a client sends a Range request with a suffix length exceeding the content size, the resulting negative start value is not properly clamped, leading to malformed HTTP 206 responses and log flooding.	2026-06-22	4.8
CVE-2026-57289	jenkins - bitbucket_push_and_pull_request	Jenkins Bitbucket Push and Pull Request Plugin 3.3.8 and earlier unconditionally disables SSL/TLS certificate and hostname validation for connections sending Bearer token authenticated requests to the configured Bitbucket Server endpoint, allowing attackers able to intercept network traffic to capture the token.	2026-06-24	4.8
CVE-2026-13034	google - chrome	Inappropriate implementation in Passwords in Google Chrome prior to 149.0.7827.197 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	2026-06-24	4.7
CVE-2026-9799	redhat - multiple products	A flaw was found in org.keycloak.authorization. An authenticated user with a granted User-Managed Access (UMA) permission ticket for one resource can exploit this by using a specific permission request prefix to bypass per-resource access control. This allows the user to gain unauthorized access to all resources of that type within the same resource server, even if they do not have a ticket for those specific resources. This vulnerability requires the resource server to be configured in PERMISSIVE policy enforcement mode and affects typed resources with ownerManagedAccess enabled, where no explicit policy protects the resource type. The primary consequence is unauthorized information disclosure or modification of resources.	2026-06-25	4.6
CVE-2026-12892	red hat - multiple products	A flaw was found in GStreamer's gst-plugins-bad package. When processing a specially crafted H.264 video file containing malformed MVC or SVC extension slice NAL units, a 1-byte heap out-of-bounds read can occur during parsing. This happens when the parser attempts to check slice boundary information without first verifying that the NAL unit contains enough data beyond the extension header. An attacker could exploit this by tricking a user into opening a malicious H.264 video file, potentially causing the application to crash or leak a single byte of heap memory.	2026-06-23	4.4
CVE-2026-55653	openbsd - multiple products	A flaw was found in OpenSSH. A malicious SSH server can exploit a double free vulnerability in the Diffie-Hellman Group Exchange (DH-GEX) client path. This occurs during FIPS (Federal Information Processing Standards) mode known-group validation when the client processes attacker-controlled DH-GEX group parameters. Successful exploitation leads to client-side process termination, resulting in a Denial of Service (DoS).	2026-06-23	4.3
CVE-2026-57283	jenkins - pipeline\	A cross-site request forgery (CSRF) vulnerability in Jenkins Pipeline: Groovy Plugin 4331.v9d06ed4658ff and earlier allows attackers to instantiate types related to job or system configuration other than Pipeline steps through the Pipeline Snippet Generator.	2026-06-24	4.3
CVE-2026-57284	jenkins - pipeline\	Jenkins Pipeline: Groovy Plugin 4331.v9d06ed4658ff and earlier does not restrict the types that can be instantiated through the Pipeline Snippet Generator, allowing attackers to instantiate types related to job or system configuration other than Pipeline steps.	2026-06-24	4.3
CVE-2026-57285	jenkins - github_branch_source	A missing permission check in Jenkins GitHub Branch Source Plugin 1967.1969.v205fd594c821 and earlier allows attackers with Overall/Read permission to obtain the URLs of GitHub Enterprise servers configured in the global plugin configuration.	2026-06-24	4.3
CVE-2026-57286	jenkins - git_parameter	A missing permission check in Jenkins Git Parameter Plugin 462.vdcf3df2ed2ca_ and earlier allows attackers with Item/Read permission to obtain information about the SCM repository used by a job, such as branch names, tag names, and revision metadata.	2026-06-24	4.3
CVE-2026-57287	jenkins - job_configuration_history	Jenkins Job Configuration History Plugin 1356.ve360da_6c523a_ and earlier does not redact the encrypted values of secrets when displaying historical job and agent configurations, allowing attackers with Extended Read permission to view encrypted secret values that would otherwise be redacted.	2026-06-24	4.3
CVE-2026-57290	jenkins - priority_sorter	A cross-site request forgery (CSRF) vulnerability in Jenkins Priority Sorter Plugin 936.v2c01c6b_84449 and earlier allows attackers to overwrite the global job priority configuration.	2026-06-24	4.3
CVE-2026-57297	jenkins - contrast_continuous_application_security	A missing permission check in Jenkins Contrast Continuous Application Security Plugin 3.11 and earlier allows attackers with Overall/Read permission to connect to an attacker-specified URL using an attacker-specified username, API key, and service key.	2026-06-24	4.3
CVE-2026-57299	jenkins - contrast_continuous_application_security	Missing permission checks in Jenkins Contrast Continuous Application Security Plugin 3.11 and earlier allow attackers with Overall/Read permission to enumerate the names of configured Contrast metadata.	2026-06-24	4.3
CVE-2026-57300	jenkins - mcp_server	A missing permission check in Jenkins MCP Server Plugin 0.177.v629fdb_2557fe and earlier allows attackers with Item/Read permission to read the Pipeline replay scripts of jobs they can access.	2026-06-24	4.3
CVE-2026-57302	jenkins - fitnesse	Jenkins FitNesse Plugin 1.36 and earlier stores passwords unencrypted in job config.xml files on the Jenkins controller, where they can be viewed by users with Extended Read permission or access to the Jenkins controller file system.	2026-06-24	4.3
CVE-2026-13021	google - chrome	Inappropriate implementation in DeviceBoundSessionCredentials in Google Chrome prior to 149.0.7827.197 allowed a remote attacker to bypass same origin policy via a crafted HTML page. (Chromium security severity: High)	2026-06-24	4.3
CVE-2026-57306	jenkins - zowe_zdevops	A cross-site request forgery (CSRF) vulnerability in Jenkins Zowe zDevOps Plugin 1.1.3.50.ve350c9b_450b_1 and earlier allows attackers to connect to an attacker-specified URL using attacker-specified credentials IDs obtained through another method, capturing credentials stored in Jenkins.	2026-06-24	4.2
CVE-2026-57307	jenkins - zowe_zdevops	A missing permission check in Jenkins Zowe zDevOps Plugin 1.1.3.50.ve350c9b_450b_1 and earlier allows attackers with Overall/Read permission to connect to an attacker-specified URL	2026-06-24	4.2

		using attacker-specified credentials IDs obtained through another method, capturing credentials stored in Jenkins.		
CVE-2026-13024	google - chrome	Insufficient validation of untrusted input in Navigation in Google Chrome prior to 149.0.7827.197 allowed a remote attacker who had compromised the renderer process to bypass site isolation via a crafted HTML page. (Chromium security severity: High)	2026-06-24	4.2
CVE-2026-13218	red hat - multiple products	A flaw was found in KubeVirt's virt-handler network cache handling. The WriteToCachedFile function writes data to a launcher-rooted path using os.WriteFile and os.Chown without symlink protection. A user with access to the virt-launcher container can plant a symlink at the cache file path, causing virt-handler to follow it and overwrite an arbitrary host file with JSON content and change its ownership.	2026-06-26	4.2
CVE-2026-57053	gnu - libidn	GNU libidn before 1.44 is prone to out-of-bounds reads of uninitialized memory in the ToUnicode APIs because of mishandling in idna_to_unicode_internal. The affected code is not present in libidn2.	2026-06-23	4
CVE-2026-13322	red hat - multiple products	A flaw was found in KubeVirt's downward metrics virtio-serial server. The server reads guest requests using textproto.Reader.ReadLine(), which buffers input indefinitely until a newline character is received, with no length limit or read deadline. A user with access to a VM guest that has the downward metrics virtio-serial device configured can write a continuous byte stream to the device, causing unbounded memory allocation in the virt-handler process until it is OOM-killed.	2026-06-26	3.8
CVE-2026-55654	openbsd - multiple products	A flaw was found in OpenSSH. This vulnerability, a heap out-of-bounds read, occurs during the cleanup of GSSAPI (Generic Security Service Application Programming Interface) indicators when a trailing NULL termination is missing in the auth-indicators array. A remote attacker, under specific configurations involving GSSAPI authentication and a Kerberos environment, could exploit this to cause the SSH authentication path to crash or abort. This leads to a denial of service (DoS), impacting the availability of the SSH service.	2026-06-23	3.7
CVE-2026-56968	gnu - sasl	GNU SASL before 2.2.4 lacks sanitization of a short challenge in _gsasl_ntlm_client_step in the NTLM client, which could result in memory disclosure via a crafted server.	2026-06-23	3.7
CVE-2026-57288	jenkins - active_directory	Jenkins Active Directory Plugin 2.41.1 and earlier does not escape the user name before building the LDAP search filter in the Windows native (ADSI) authentication path, allowing unauthenticated attackers to inject LDAP wildcard characters to enumerate directory entries and to authenticate as a matching user whose password they know without knowing their exact user name.	2026-06-24	3.7
CVE-2026-39894	cacti - cacti	Cacti is an open source performance and fault management framework. In versions 1.2.30 and below, the locale-dependent decimal formatting in rrdtool_function_update() can corrupt RRDtool metric values. The rrdtool_function_update() function checks metric values with is_numeric() and concatenates them into the RRDtool update command via PHP string interpolation. PHP's string cast of floats is locale-sensitive: if LC_NUMERIC uses comma as decimal separator (e.g., de_DE), a value of 1.5 becomes "1,5". RRDtool expects . as decimal separator, causing metric data to shift into wrong columns or be silently dropped. No setlocale() reset is present in the update path. This causes a data integrity issue, but is not remotely exploitable; it requires server locale misconfiguration. The issue has been fixed in version 1.2.31.	2026-06-24	2.9
CVE-2026-45188	apache software foundation - Apache Kvrocks	Relative Path Traversal vulnerability in Apache Kvrocks. This issue affects Apache Kvrocks: from 1.0.0 through 2.15.0. Users are recommended to upgrade to version 2.16.0, which fixes the issue.	2026-06-25	2.4
CVE-2026-44911	apache - nifi	Authorization handling for component configuration verification requests in Apache NiFi 1.15.0 through 2.9.0 allows clients with read access to submit proposed configuration properties. The proposed properties override current configuration, enabling users with read access to invoke predefined verification methods with alternative settings. Apache NiFi installations that do not implement different levels of authorization for viewing and modifying component configuration are not subject to this vulnerability. Upgrading to Apache NiFi 2.10.0 is the recommended mitigation, requiring write access to submit configuration verification requests.	2026-06-22	2.3
CVE-2026-9610	ibm - multiple products	IBM Datacap 9.1.7, 9.1.8, and 9.1.9 and IBM Datacap Navigator 9.1.7, 9.1.8, and 9.1.9 exposes resources or functionality that isn't linked in the UI but is accessible by directly requesting the URL, bypassing intended access controls.	2026-06-22	2.3
CVE-2026-56130	apache software foundation - Apache Shiro	"Remember me" cookie age is not verified on the server. This potentially allows an attacker to intercept a valid cookie and reuse it indefinitely, even after the configured expiration time has passed. This issue affects all Apache Shiro versions from 1.2.4 through 2.x, and 3.0.0-alpha-1, only when RememberMe functionality is enabled. Upgrade to version 3.0.0 or later, which fixes the issue.	2026-06-25	2

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

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