Security Bulletin

Bulletin ID: RITTAL-SA-C-2026-0002
Project: RITTAL ReOS1 - IOT
Version: v10.1.2
Compared against: v10.0.2
Generated on: 07.05.2026
Classification: TLP:CLEAR

Summary

This bulletin documents 130 security vulnerabilities (5 Critical, 25 High, 99 Medium, 1 Low) that were fixed in RITTAL ReOS1 - IOT version v10.1.2 compared to version v10.0.2. Rittal recommends updating to the latest firmware version.

Table of Contents

Affected Products (Article Numbers)

Article Number	Description
3312.800	Bundle Blue e+ IT, SK Dachaufbau-Kühlgerät 1,6 kW, RAL 7035
3312.810	Blue e+ für IT, SK Wandkühlgerät 3 kW mit IoT Interface, BHT: 450x1600x294 mm
3399.022	N/A
3412.800	Bundle Blue e+ IT, SK Dachaufbau-Kühlgerät 1,6 kW, R-1234yf
3412.810	Blue e+ für IT, SK Wandkühlgerät 3 kW mit IoT Interface, R-1234yf

Fixed Vulnerabilities Overview

All	Critical	High	Medium	Low
130	5	25	99	1

Fixed Vulnerabilities

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2020-13910	CRITICAL	9.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H	barebox 1.0	Pengutronix Barebox through v2020.05.0 has an out-of-bounds read in nfs_read_reply in net/nfs.c because a field of an incoming network packet is directly used as a length field without any bounds check.
CVE-2019-15937	CRITICAL	9.8	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	barebox 1.0	Pengutronix barebox through 2019.08.1 has a remote buffer overflow in nfs_readlink_reply in net/nfs.c because a length field is directly used for a memcpy.
CVE-2019-15938	CRITICAL	9.8	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	barebox 1.0	Pengutronix barebox through 2019.08.1 has a remote buffer overflow in nfs_readlink_req in fs/nfs.c because a length field is directly used for a memcpy.
CVE-2025-57052	CRITICAL	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	cjson 1.7.18	cJSON 1.5.0 through 1.7.18 allows out-of-bounds access via the decode_array_index_from_pointer function in cJSON_Utils.c, allowing remote attackers to bypass array bounds checking and access restricted data via malformed JSON pointer strings containing alphanumeric characters.
CVE-2023-26793	CRITICAL	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	libmodbus 3.1.10	libmodbus v3.1.10 has a heap-based buffer overflow vulnerability in read_io_status function in src/modbus.c.
CVE-2021-37847	HIGH	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	barebox 1.0	crypto/digest.c in Pengutronix barebox through 2021.07.0 leaks timing information because memcmp is used during digest verification.
CVE-2021-37848	HIGH	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	barebox 1.0	common/password.c in Pengutronix barebox through 2021.07.0 leaks timing information because strncmp is used during hash comparison.
CVE-2025-32988	HIGH	8.2	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:H	gnutls 3.8.4	A flaw was found in GnuTLS. A double-free vulnerability exists in GnuTLS due to incorrect ownership handling in the export logic of Subject Alternative Name (SAN) entries containing an otherName. If the type-id OID is invalid or malformed, GnuTLS will call asn1_delete_structure() on an ASN.1 node it does not own, leading to a double-free condition when the parent function or caller later attempts to free the same structure. This vulnerability can be triggered using only public GnuTLS APIs and may result in denial of service or memory corruption, depending on allocator behavior.
CVE-2024-34244	HIGH	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	libmodbus 3.1.10	libmodbus v3.1.10 is vulnerable to Buffer Overflow via the modbus_write_bits function. This issue can be triggered when the function is fed with specially crafted input, which leads to out-of-bounds read and can potentially cause a crash or other unintended behaviors.
CVE-2024-26954	HIGH	7.1	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-out-of-bounds in smb_strndup_from_utf16() If ->NameOffset of smb2_create_req is smaller than Buffer offset of smb2_create_req, slab-out-of-bounds read can happen from smb2_open. This patch set the minimum value of the name offset to the buffer offset to validate name length of smb2_create_req().

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-57982	HIGH	7.1	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: xfrm: state: fix out-of-bounds read during lookup lookup and resize can run in parallel. The xfrm_state_hash_generation seqlock ensures a retry, but the hash functions can observe a hmask value that is too large for the new hlist array. rehash does: rcu_assign_pointer(net->xfrm.state_bydst, ndst) [..] net->xfrm.state_hmask = nhashmask; While state lookup does: h = xfrm_dst_hash(net, daddr, saddr, tmpl->reqid, encap_family); hlist_for_each_entry_rcu(x, net->xfrm.state_bydst + h, bydst) { This is only safe in case the update to state_bydst is larger than net->xfrm.xfrm_state_hmask (or if the lookup function gets serialized via state spinlock again). Fix this by prefetching state_hmask and the associated pointers. The xfrm_state_hash_generation seqlock retry will ensure that the pointer and the hmask will be consistent. The existing helpers, like xfrm_dst_hash(), are now unsafe for RCU side, add lockdep assertions to document that they are only safe for insert side. xfrm_state_lookup_byaddr() uses the spinlock rather than RCU. AFAICS this is an oversight from back when state lookup was converted to RCU, this lock should be replaced with RCU in a future patch.
CVE-2024-53147	HIGH	7.1	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: exfat: fix out-of-bounds access of directory entries In the case of the directory size is greater than or equal to the cluster size, if start_clu becomes an EOF cluster(an invalid cluster) due to file system corruption, then the directory entry where ei->hint_femp.eidx hint is outside the directory, resulting in an out-of-bounds access, which may cause further file system corruption. This commit adds a check for start_clu, if it is an invalid cluster, the file or directory will be treated as empty.
CVE-2024-42118	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Do not return negative stream id for array [WHY] resource_stream_to_stream_idx returns an array index and it return -1 when not found; however, -1 is not a valid array index number. [HOW] When this happens, call ASSERT(), and return a zero instead. This fixes an OVERRUN and an NEGATIVE_RETURNS issues reported by Coverity.
CVE-2024-26914	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix incorrect mpc_combine array size [why] MAX_SURFACES is per stream, while MAX_PLANES is per asic. The mpc_combine is an array that records all the planes per asic. Therefore MAX_PLANES should be used as the array size. Using MAX_SURFACES causes array overflow when there are more than 3 planes. [how] Use the MAX_PLANES for the mpc_combine array size.
CVE-2025-40014	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: objtool, spi: amd: Fix out-of-bounds stack access in amd_set_spi_freq() If speed_hz < AMD_SPI_MIN_HZ, amd_set_spi_freq() iterates over the entire amd_spi_freq array without breaking out early, causing 'i' to go beyond the array bounds. Fix that by stopping the loop when it gets to the last entry, so the low speed_hz value gets clamped up to AMD_SPI_MIN_HZ. Fixes the following warning with an UBSAN kernel: drivers/spi/spi-amd.o: error: objtool: amd_set_spi_freq() falls through to next function amd_spi_set_opcode()

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-53203	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: usb: typec: fix potential array underflow in ucsi_ccg_sync_control() The "command" variable can be controlled by the user via debugfs. The worry is that if con_index is zero then "&uc->ucsi->connector[con_index - 1]" would be an array underflow.
CVE-2024-46729	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix incorrect size calculation for loop [WHY] fe_clk_en has size of 5 but sizeof(fe_clk_en) has byte size 20 which is lager than the array size. [HOW] Divide byte size 20 by its element size. This fixes 2 OVERRUN issues reported by Coverity.
CVE-2024-53170	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: block: fix uaf for flush rq while iterating tags blk_mq_clear_flush_rq_mapping() is not called during scsi probe, by checking blk_queue_init_done(). However, QUEUE_FLAG_INIT_DONE is cleared in del_gendisk by commit aec89dc5d421 ("block: keep q_usage_counter in atomic mode after del_gendisk"), hence for disk like scsi, following blk_mq_destroy_queue() will not clear flush rq from tags->rqs[] as well, cause following uaf that is found by our syzkaller for v6.6: == BUG: KASAN: slab-use-after-free in blk_mq_find_and_get_req+0x16e/0x1a0 block/blk-mq-tag.c:261 Read of size 4 at addr ffff88811c969c20 by task kworker/1:2H/224909 CPU: 1 PID: 224909 Comm: kworker/1:2H Not tainted 6.6.0-ga836a5060850 #32 Workqueue: kblockd blk_mq_timeout_work Call Trace: __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x91/0xf0 lib/dump_stack.c:106 print_address_description.constprop.0+0x66/0x300 mm/kasan/report.c:364 print_report+0x3e/0x70 mm/kasan/report.c:475 kasan_report+0xb8/0xf0 mm/kasan/report.c:588 blk_mq_find_and_get_req+0x16e/0x1a0 block/blk-mq-tag.c:261 bt_iter block/blk-mq-tag.c:288 [inline] __sbitmap_for_each_set include/linux/sbitmap.h:295 [inline] sbitmap_for_each_set include/linux/sbitmap.h:316 [inline] bt_for_each+0x455/0x790 block/blk-mq-tag.c:325 blk_mq_queue_tag_busy_iter+0x320/0x740 block/blk-mq-tag.c:534 blk_mq_timeout_work+0x1a3/0x7b0 block/blk-mq.c:1673 process_one_work+0x7c4/0x1450 kernel/workqueue.c:2631 process_scheduled_works kernel/workqueue.c:2704 [inline] worker_thread+0x804/0xe40 kernel/workqueue.c:2785 kthread+0x346/0x450 kernel/kthread.c:388 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1b/0x30 arch/x86/entry/entry_64.S:293 Allocated by task 942: kasan_save_stack+0x22/0x50 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 ____kasan_kmalloc mm/kasan/common.c:374 [inline] __kasan_kmalloc mm/kasan/common.c:383 [inline] __kasan_kmalloc+0xaa/0xb0 mm/kasan/common.c:380 kasan_kmalloc include/linux/kasan.h:198 [inline] __do_kmalloc_node mm/slab_common.c:1007 [inline] __kmalloc_node+0x69/0x170 mm/slab_common.c:1014 kmalloc_node include/linux/slab.h:620 [inline] kzalloc_node include/linux/slab.h:732 [inline] blk_alloc_flush_queue+0x144/0x2f0 block/blk-flush.c:499 blk_mq_alloc_hctx+0x601/0x940 block/blk-mq.c:3788 blk_mq_alloc_and_init_hctx+0x27f/0x330 block/blk-mq.c:4261 blk_mq_realloc_hw_ctxs+0x488/0x5e0 block/blk-mq.c:4294 blk_mq_init_allocated_queue+0x188/0x860

CVE	Severity	CVSS	CVSS Vector	Component	Description
					block/blk-mq.c:4350 blk_mq_init_queue_data block/blk-mq.c:4166 [inline] blk_mq_init_queue+0x8d/0x100 block/blk-mq.c:4176 scsi_alloc_sdev+0x843/0xd50 drivers/scsi/scsi_scan.c:335 scsi_probe_and_add_lun+0x77c/0xde0 drivers/scsi/scsi_scan.c:1189 __scsi_scan_target+0x1fc/0x5a0 drivers/scsi/scsi_scan.c:1727 scsi_scan_channel drivers/scsi/scsi_scan.c:1815 [inline] scsi_scan_channel+0x14b/0x1e0 drivers/scsi/scsi_scan.c:1791 scsi_scan_host_selected+0x2fe/0x400 drivers/scsi/scsi_scan.c:1844 scsi_scan+0x3a0/0x3f0 drivers/scsi/scsi_sysfs.c:151 store_scan+0x2a/0x60 drivers/scsi/scsi_sysfs.c:191 dev_attr_store+0x5c/0x90 drivers/base/core.c:2388 sysfs_kf_write+0x11c/0x170 fs/sysfs/file.c:136 kernfs_fop_write_iter+0x3fc/0x610 fs/kernfs/file.c:338 call_write_iter include/linux/fs.h:2083 [inline] new_sync_write+0x1b4/0x2d0 fs/read_write.c:493 vfs_write+0x76c/0xb00 fs/read_write.c:586 ksys_write+0x127/0x250 fs/read_write.c:639 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x70/0x120 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x78/0xe2 Freed by task 244687: kasan_save_stack+0x22/0x50 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 kasan_save_free_info+0x2b/0x50 mm/kasan/generic.c:522 ____kasan_slab_free mm/kasan/common.c:236 [inline] __kasan_slab_free+0x12a/0x1b0 mm/kasan/common.c:244 kasan_slab_free include/linux/kasan.h:164 [in ---truncated---
CVE-2024-41045	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Defer work in bpf_timer_cancel_and_free Currently, the same case as previous patch (two timer callbacks trying to cancel each other) can be invoked through bpf_map_update_elem as well, or more precisely, freeing map elements containing timers. Since this relies on hrtimer_cancel as well, it is prone to the same deadlock situation as the previous patch. It would be sufficient to use hrtimer_try_to_cancel to fix this problem, as the timer cannot be enqueued after async_cancel_and_free. Once async_cancel_and_free has been done, the timer must be reinitialized before it can be armed again. The callback running in parallel trying to arm the timer will fail, and freeing bpf_hrtimer without waiting is sufficient (given kfree_rcu), and bpf_timer_cb will return HRTIMER_NORESTART, preventing the timer from being rearmed again. However, there exists a UAF scenario where the callback arms the timer before entering this function, such that if cancellation fails (due to timer callback invoking this routine, or the target timer callback running concurrently). In such a case, if the timer expiration is significantly far in the future, the RCU grace period expiration happening before it will free the bpf_hrtimer state and along with it the struct hrtimer, that is enqueued. Hence, it is clear cancellation needs to occur after async_cancel_and_free, and yet it cannot be done inline due to deadlock issues. We thus modify bpf_timer_cancel_and_free to defer work to the global workqueue, adding a work_struct alongside rcu_head (both used at _different_ points of time, so can share space). Update existing code comments to reflect the new state of affairs.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-21751	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: HWS, change error flow on matcher disconnect Currently, when firmware failure occurs during matcher disconnect flow, the error flow of the function reconnects the matcher back and returns an error, which continues running the calling function and eventually frees the matcher that is being disconnected. This leads to a case where we have a freed matcher on the matchers list, which in turn leads to use-after-free and eventual crash. This patch fixes that by not trying to reconnect the matcher back when some FW command fails during disconnect. Note that we're dealing here with FW error. We can't overcome this problem. This might lead to bad steering state (e.g. wrong connection between matchers), and will also lead to resource leakage, as it is the case with any other error handling during resource destruction. However, the goal here is to allow the driver to continue and not crash the machine with use-after-free error.
CVE-2025-21786	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: workqueue: Put the pwq after detaching the rescuer from the pool The commit 68f83057b913("workqueue: Reap workers via kthread_stop() and remove detach_completion") adds code to reap the normal workers but mistakenly does not handle the rescuer and also removes the code waiting for the rescuer in put_unbound_pool(), which caused a use-after-free bug reported by Cheung Wall. To avoid the use-after-free bug, the pool's reference must be held until the detachment is complete. Therefore, move the code that puts the pwq after detaching the rescuer from the pool.
CVE-2025-21927	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix potential memory corruption in nvme_tcp_recv_pdu() nvme_tcp_recv_pdu() doesn't check the validity of the header length. When header digests are enabled, a target might send a packet with an invalid header length (e.g. 255), causing nvme_tcp_verify_hdgst() to access memory outside the allocated area and cause memory corruptions by overwriting it with the calculated digest. Fix this by rejecting packets with an unexpected header length.
CVE-2024-26945	HIGH	8.4	CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: crypto: iaa - Fix nr_cpus < nr_iaa case If nr_cpus < nr_iaa, the calculated cpus_per_iaa will be 0, which causes a divide-by-0 in rebalance_wq_table(). Make sure cpus_per_iaa is 1 in that case, and also in the nr_iaa == 0 case, even though cpus_per_iaa is never used if nr_iaa == 0, for paranoia.
CVE-2024-48208	HIGH	8.6	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H	pure-ftpd 1.0.51	pure-ftpd before 1.0.52 is vulnerable to Buffer Overflow. There is an out of bounds read in the domlsd() function of the ls.c file.
CVE-2020-17163	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	python3 3.12.11	Visual Studio Code Python Extension Remote Code Execution Vulnerability
CVE-2025-49714	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	python3 3.12.11	Trust boundary violation in Visual Studio Code - Python extension allows an unauthorized attacker to execute code locally.
CVE-2024-49050	HIGH	8.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	python3 3.12.11	Visual Studio Code Python Extension Remote Code Execution Vulnerability
CVE-2020-1192	HIGH	7.8	CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	python3 3.12.11	A remote code execution vulnerability exists in Visual Studio Code when the Python extension loads workspace settings from a notebook file, aka 'Visual Studio Code Python Extension Remote Code Execution Vulnerability'. This CVE ID is unique from CVE-2020-1171.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2020-1171	HIGH	8.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	python3 3.12.11	A remote code execution vulnerability exists in Visual Studio Code when the Python extension loads configuration files after opening a project, aka 'Visual Studio Code Python Extension Remote Code Execution Vulnerability'. This CVE ID is unique from CVE-2020-1192.
CVE-2025-6965	HIGH	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	sqlite3 3.45.3	There exists a vulnerability in SQLite versions before 3.50.2 where the number of aggregate terms could exceed the number of columns available. This could lead to a memory corruption issue. We recommend upgrading to version 3.50.2 or above.
CVE-2024-49934	MEDIUM	4.6	CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: fs/inode: Prevent dump_mapping() accessing invalid dentry.d_name.name It's observed that a crash occurs during hot-remove a memory device, in which user is accessing the hugetlb. See calltrace as following: ----------[cut here]---------- WARNING: CPU: 1 PID: 14045 at arch/x86/mm/fault.c:1278 do_user_addr_fault+0x2a0/0x790 Modules linked in: kmem device_dax cxl_mem cxl_pmem cxl_port cxl_pci dax_hmem dax_pmem nd_pmem cxl_acpi nd_btt cxl_core crc32c_intel nvme virtiofs fuse nvme_core nfit libnvdimm dm_multipath scsi_dh_rdac scsi_dh_emc s mirror dm_region_hash dm_log dm_mod CPU: 1 PID: 14045 Comm: daxctl Not tainted 6.10.0-rc2-lizhijian+ #492 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:do_user_addr_fault+0x2a0/0x790 Code: 48 8b 00 a8 04 0f 84 b5 fe ff ff e9 1c ff ff ff 4c 89 e9 4c 89 e2 be 01 00 00 00 bf 02 00 00 00 e8 b5 ef 24 00 e9 42 fe ff ff <0f> 0b 48 83 c4 08 4c 89 ea 48 89 ee 4c 89 e7 5b 5d 41 5c 41 5d 41 RSP: 0000:ffffc90000a575f0 EFLAGS: 00010046 RAX: ffff88800c303600 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000001000 RSI: ffffffff82504162 RDI: ffffffff824b2c36 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: ffffc90000a57658 R13: 0000000000001000 R14: ffff88800bc2e040 R15: 0000000000000000 FS: 00007f51cb57d880(0000) GS:ffff88807fd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000001000 CR3: 00000000072e2004 CR4: 00000000001706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: ? __warn+0x8d/0x190 ? do_user_addr_fault+0x2a0/0x790 ? report_bug+0x1c3/0x1d0 ? handle_bug+0x3c/0x70 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? do_user_addr_fault+0x2a0/0x790 ? exc_page_fault+0x31/0x200 exc_page_fault+0x68/0x200 <...snip...> BUG: unable to handle page fault for address: 0000000000001000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 800000000ad92067 P4D 800000000ad92067 PUD 7677067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI ---[end trace 0000000000000000]--- BUG: unable to handle page fault for address: 0000000000001000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 800000000ad92067 P4D 800000000ad92067 PUD 7677067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 PID: 14045 Comm: daxctl Kdump: loaded Tainted: G W 6.10.0-rc2-lizhijian+ #492 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009),

CVE	Severity	CVSS	CVSS Vector	Component	Description
					BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:dentry_name+0x1f4/0x440 <...snip...> ? dentry_name+0x2fa/0x440 vsnprintf+0x1f3/0x4f0 vprintk_store+0x23a/0x540 vprintk_emit+0x6d/0x330 _printk+0x58/0x80 dump_mapping+0x10b/0x1a0 ? __pfx_free_object_rcu+0x10/0x10 __dump_page+0x26b/0x3e0 ? vprintk_emit+0xe0/0x330 ? _printk+0x58/0x80 ? dump_page+0x17/0x50 dump_page+0x17/0x50 do_migrate_range+0x2f7/0x7f0 ? do_migrate_range+0x42/0x7f0 ? offline_pages+0x2f4/0x8c0 offline_pages+0x60a/0x8c0 memory_subsys_offline+0x9f/0x1c0 ? lockdep_hardirqs_on+0x77/0x100 ? _raw_spin_unlock_irqrestore+0x38/0x60 device_offline+0xe3/0x110 state_store+0x6e/0xc0 kernfs_fop_write_iter+0x143/0x200 vfs_write+0x39f/0x560 ksys_write+0x65/0xf0 do_syscall_64+0x62/0x130 Previously, some sanity check have been done in dump_mapping() before the print facility parsing '%pd' though, it's still possible to run into an invalid dentry.d_name.name. Since dump_mapping() only needs to dump the filename only, retrieve it by itself in a safer way to prevent an unnecessary crash. Note that either retrieving the filename with '%pd' or strncpy_from_kernel_nofault(), the filename could be unreliable.
CVE-2024-50010	MEDIUM	4.7	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: exec: don't WARN for racy path_noexec check Both i_mode and noexec checks wrapped in WARN_ON stem from an artifact of the previous implementation. They used to legitimately check for the condition, but that got moved up in two commits: 633fb6ac3980 ("exec: move S_ISREG() check earlier") 0fd338b2d2cd ("exec: move path_noexec() check earlier") Instead of being removed said checks are WARN_ON'ed instead, which has some debug value. However, the spurious path_noexec check is racy, resulting in unwarranted warnings should someone race with setting the noexec flag. One can note there is more to perm-checking whether execve is allowed and none of the conditions are guaranteed to still hold after they were tested for. Additionally this does not validate whether the code path did any perm checking to begin with -- it will pass if the inode happens to be regular. Keep the redundant path_noexec() check even though it's mindless nonsense checking for guarantee that isn't given so drop the WARN. Reword the commentary and do small tidy ups while here. [brauner: keep redundant path_noexec() check]
CVE-2024-36024	MEDIUM	4.7	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Disable idle reallow as part of command/gpint execution [Why] Workaroud for a race condition where DMCUB is in the process of committing to IPS1 during the handshake causing us to miss the transition into IPS2 and touch the INBOX1 RPTR causing a HW hang. [How] Disable the reallow to ensure that we have enough of a gap between entry and exit and we're not seeing back-to-back wake_and_executes.
CVE-2024-53124	MEDIUM	4.7	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net: fix data-races around sk->sk_forward_alloc Syzkaller reported this warning: ------------[cut here]------------ WARNING: CPU: 0 PID: 16 at net/ipv4/af_inet.c:156 inet_sock_destruct+0x1c5/0x1e0 Modules linked in: CPU: 0 UID: 0 PID: 16 Comm: ksoftirqd/0 Not tainted 6.12.0-rc5 #26 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:inet_sock_destruct+0x1c5/0x1e0 Code: 24 12 4c 89 e2 5b 48 c7 c7 98 ec bb 82 41 5c e9 d1 18 17 ff 4c 89 e6 5b 48 c7 c7 d0 ec bb 82 41 5c e9 bf 18 17 ff 0f 0b eb 83 <0f> 0b eb 97 0f 0b eb

CVE	Severity	CVSS	CVSS Vector	Component	Description
					87 0f 0b e9 68 ff ff ff 66 66 2e 0f 1f 84 00 RSP: 0018:ffffc9000008bd90 EFLAGS: 00010206 RAX: 0000000000000300 RBX: ffff88810b172a90 RCX: 0000000000000007 RDX: 0000000000000002 RSI: 0000000000000300 RDI: ffff88810b172a00 RBP: ffff88810b172a00 R08: ffff888104273c00 R09: 0000000000100007 R10: 0000000000020000 R11: 0000000000000006 R12: ffff88810b172a00 R13: 0000000000000004 R14: 0000000000000000 R15: ffff888237c31f78 FS: 0000000000000000(0000) GS:ffff888237c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffc63fecac8 CR3: 000000000342e000 CR4: 00000000000006f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: ? __warn+0x88/0x130 ? inet_sock_destruct+0x1c5/0x1e0 ? report_bug+0x18e/0x1a0 ? handle_bug+0x53/0x90 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? inet_sock_destruct+0x1c5/0x1e0 __sk_destruct+0x2a/0x200 rcu_do_batch+0x1aa/0x530 ? rcu_do_batch+0x13b/0x530 rcu_core+0x159/0x2f0 handle_softirqs+0xd3/0x2b0 ? __pfx_smpboot_thread_fn+0x10/0x10 run_ksoftirqd+0x25/0x30 smpboot_thread_fn+0xdd/0x1d0 kthread+0xd3/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x34/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 ---[end trace 0000000000000000]--- Its possible that two threads call tcp_v6_do_rcv()/sk_forward_alloc_add() concurrently when sk->sk_state == TCP_LISTEN with sk->sk_lock unlocked, which triggers a data-race around sk->sk_forward_alloc: tcp_v6_rcv tcp_v6_do_rcv skb_clone_and_charge_r sk_rmem_schedule __sk_mem_schedule sk_forward_alloc_add() skb_set_owner_r sk_mem_charge sk_forward_alloc_add() __kfree_skb skb_release_all skb_release_head_state sock_rfree sk_mem_uncharge sk_forward_alloc_add() sk_mem_reclaim // set local var reclaimable __sk_mem_reclaim sk_forward_alloc_add() In this syzkaller testcase, two threads call tcp_v6_do_rcv() with skb->truesize=768, the sk_forward_alloc changes like this: (cpu 1)	(cpu 2)	sk_forward_alloc ...	...	0 __sk_mem_schedule()		+4096 = 4096	__sk_mem_schedule()	+4096 = 8192 sk_mem_charge()		-768 = 7424	sk_mem_charge()	-768 = 6656 ...	...	sk_mem_uncharge()		+768 = 7424 reclaimable=7424			sk_mem_uncharge()	+768 = 8192	reclaimable=8192	__sk_mem_reclaim()		-4096 = 4096	__sk_mem_reclaim()	-8192 = -4096 != 0 The skb_clone_and_charge_r() should not be called in tcp_v6_do_rcv() when sk->sk_state is TCP_LISTEN, it happens later in tcp_v6_syn_recv_sock(). Fix the same issue in dccp_v6_do_rcv().
CVE-2024-24864	MEDIUM	4.7	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	A race condition was found in the Linux kernel's media/dvb-core in dvbdmx_write() function. This can result in a null pointer dereference issue, possibly leading to a kernel panic or denial of service issue.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-49998	MEDIUM	4.7	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net: dsa: improve shutdown sequence Alexander Sverdlin presents 2 problems during shutdown with the lan9303 driver. One is specific to lan9303 and the other just happens to reproduce there. The first problem is that lan9303 is unique among DSA drivers in that it calls dev_get_drvdata() at "arbitrary runtime" (not probe, not shutdown, not remove): phy_state_machine() -> ... -> dsa_user_phy_read() -> ds->ops->phy_read() -> lan9303_phy_read() -> chip->ops->phy_read() -> lan9303_mdio_phy_read() -> dev_get_drvdata() But we never stop the phy_state_machine(), so it may continue to run after dsa_switch_shutdown(). Our common pattern in all DSA drivers is to set drvdata to NULL to suppress the remove() method that may come afterwards. But in this case it will result in an NPD. The second problem is that the way in which we set dp->conduit->dsa_ptr = NULL; is concurrent with receive packet processing. dsa_switch_rcv() checks once whether dev->dsa_ptr is NULL, but afterwards, rather than continuing to use that non-NULL value, dev->dsa_ptr is dereferenced again and again without NULL checks: dsa_conduit_find_user() and many other places. In between dereferences, there is no locking to ensure that what was valid once continues to be valid. Both problems have the common aspect that closing the conduit interface solves them. In the first case, dev_close(conduit) triggers the NETDEV_GOING_DOWN event in dsa_user_netdevice_event() which closes user ports as well. dsa_port_disable_rt() calls phylink_stop(), which synchronously stops the phylink state machine, and ds->ops->phy_read() will thus no longer call into the driver after this point. In the second case, dev_close(conduit) should do this, as per Documentation/networking/driver.rst: | Quiescence | ---------- | | After the ndo_stop routine has been called, the hardware must | not receive or transmit any data. All in flight packets must | be aborted. If necessary, poll or wait for completion of | any reset commands. So it should be sufficient to ensure that later, when we zeroize conduit->dsa_ptr, there will be no concurrent dsa_switch_rcv() call on this conduit. The addition of the netif_device_detach() function is to ensure that ioctls, rtnetlinks and ethtool requests on the user ports no longer propagate down to the driver - we're no longer prepared to handle them. The race condition actually did not exist when commit 0650bf52b31f ("net: dsa: be compatible with masters which unregister on shutdown") first introduced dsa_switch_shutdown(). It was created later, when we stopped unregistering the user interfaces from a bad spot, and we just replaced that sequence with a racy zeroization of conduit->dsa_ptr (one which doesn't ensure that the interfaces aren't up).
CVE-2024-24859	MEDIUM	4.8	CVSS:3.1/AV:A/AC:H/PR:N/UI:R/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	A race condition was found in the Linux kernel's net/bluetooth in sniff_{min,max}_interval_set() function. This can result in a bluetooth sniffing exception issue, possibly leading denial of service.
CVE-2022-4543	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	linux-rittal 6.6.96+git	A flaw named "EntryBleed" was found in the Linux Kernel Page Table Isolation (KPTI). This issue could allow a local attacker to leak KASLR base via prefetch side-channels based on TLB timing for Intel systems.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-27017	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: walk over current view on netlink dump The generation mask can be updated while netlink dump is in progress. The pipapo set backend walk iterator cannot rely on it to infer what view of the datastructure is to be used. Add notation to specify if user wants to read/update the set. Based on patch from Florian Westphal.
CVE-2024-56591	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: Use disable_delayed_work_sync This makes use of disable_delayed_work_sync instead cancel_delayed_work_sync as it not only cancel the ongoing work but also disables new submit which is disarable since the object holding the work is about to be freed.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-53089	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Mark hrtimer to expire in hard interrupt context Like commit 2c0d278f3293f ("KVM: LAPIC: Mark hrtimer to expire in hard interrupt context") and commit 9090825fa9974 ("KVM: arm/arm64: Let the timer expire in hardirq context on RT"), On PREEMPT_RT enabled kernels unmarked hrtimers are moved into soft interrupt expiry mode by default. Then the timers are canceled from an preempt-notifier which is invoked with disabled preemption which is not allowed on PREEMPT_RT. The timer callback is short so in could be invoked in hard-IRQ context. So let the timer expire on hard-IRQ context even on -RT. This fix a "scheduling while atomic" bug for PREEMPT_RT enabled kernels: BUG: scheduling while atomic: qemu-system-loo/1011/0x00000002 Modules linked in: amdgpu rfkill nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat ns CPU: 1 UID: 0 PID: 1011 Comm: qemu-system-loo Tainted: G W 6.12.0-rc2+ #1774 Tainted: [W]=WARN Hardware name: Loongson Loongson-3A5000-7A1000-1w-CRB/Loongson-LS3A5000-7A1000-1w-CRB, BIOS vUDK2018-LoongArch-V2.0.0-prebeta9 10/21/2022 Stack : ffffffffffffffff 0000000000000000 9000000004e3ea38 9000000116744000 90000001167475a0 0000000000000000 90000001167475a8 9000000005644830 90000000058dc000 90000000058dbff8 9000000116747420 0000000000000001 0000000000000001 6a613fc938313980 000000000790c000 90000001001c1140 00000000000003fe 0000000000000001 000000000000000d 0000000000000003 0000000000000030 00000000000003f3 000000000790c000 9000000116747830 90000000057ef000 0000000000000000 9000000005644830 0000000000000004 0000000000000000 90000000057f4b58 0000000000000001 9000000116747868 900000000451b600 9000000005644830 9000000003a13998 0000000010000020 00000000000000b0 0000000000000004 0000000000000000 0000000000071c1d ... Call Trace: [<9000000003a13998>] show_stack+0x38/0x180 [<9000000004e3ea34>] dump_stack_lvl+0x84/0xc0 [<9000000003a71708>] __schedule_bug+0x48/0x60 [<9000000004e45734>] __schedule+0x1114/0x1660 [<9000000004e46040>] schedule_rtlock+0x20/0x60 [<9000000004e4e330>] rtlock_slowlock_locked+0x3f0/0x10a0 [<9000000004e4f038>] rt_spin_lock+0x58/0x80 [<9000000003b02d68>] hrtimer_cancel_wait_running+0x68/0xc0 [<9000000003b02e30>] hrtimer_cancel+0x70/0x80 [] kvm_restore_timer+0x50/0x1a0 [kvm] [] kvm_arch_vcpu_load+0x68/0x2a0 [kvm] [] kvm_sched_in+0x34/0x60 [kvm] [<9000000003a749a0>] finish_task_switch.isra.0+0x140/0x2e0 [<9000000004e44a70>] __schedule+0x450/0x1660 [<9000000004e45cb0>] schedule+0x30/0x180 [] kvm_vcpu_block+0x70/0x120 [kvm] [] kvm_vcpu_halt+0x60/0x3e0 [kvm] [] kvm_handle_gspr+0x3f4/0x4e0 [kvm] [] kvm_handle_exit+0x1c8/0x260 [kvm]

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-21949	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Set hugetlb mmap base address aligned with pmd size With ltp test case "testcases/bin/hugefork02", there is a dmesg error report message such as: kernel BUG at mm/hugetlb.c:5550! Oops - BUG[#1]: CPU: 0 UID: 0 PID: 1517 Comm: hugefork02 Not tainted 6.14.0-rc2+ #241 Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 2/2/2022 pc 90000000004eaf1c ra 9000000000485538 tp 900000010edbc000 sp 900000010edbf940 a0 900000010edbfb00 a1 9000000108d20280 a2 00007fffe9474000 a3 00007ffff3474000 a4 0000000000000000 a5 0000000000000003 a6 00000000003cadd3 a7 0000000000000000 t0 0000000001ffffff t1 0000000001474000 t2 900000010ecd7900 t3 00007fffe9474000 t4 00007fffe9474000 t5 0000000000000040 t6 900000010edbfb00 t7 0000000000000001 t8 0000000000000005 u0 90000000004849d0 s9 900000010edbfa00 s0 9000000108d20280 s1 00007fffe9474000 s2 0000000002000000 s3 9000000108d20280 s4 9000000002b38b10 s5 900000010edbfb00 s6 00007ffff3474000 s7 0000000000000406 s8 900000010edbfa08 ra: 9000000000485538 unmap_vmas+0x130/0x218 ERA: 90000000004eaf1c __unmap_hugepage_range+0x6f4/0x7d0 PRMD: 00000004 (PPLV0 +PIE -PWE) EUEN: 00000007 (+FPE +SXE +ASXE -BTE) ECFG: 00071c1d (LIE=0,2-4,10-12 VS=7) ESTAT: 000c0000 [BRK] (IS= ECode=12 EsubCode=0) PRID: 0014c010 (Loongson-64bit, Loongson-3A5000) Process hugefork02 (pid: 1517, threadinfo=00000000a670eaf4, task=000000007a95fc64) Call Trace: [<90000000004eaf1c>] __unmap_hugepage_range+0x6f4/0x7d0 [<9000000000485534>] unmap_vmas+0x12c/0x218 [<9000000000494068>] exit_mmap+0xe0/0x308 [<900000000025fdc4>] mmput+0x74/0x180 [<900000000026a284>] do_exit+0x294/0x898 [<900000000026aa30>] do_group_exit+0x30/0x98 [<900000000027bed4>] get_signal+0x83c/0x868 [<90000000002457b4>] arch_do_signal_or_restart+0x54/0xfa0 [<90000000015795e8>] irqentry_exit_to_user_mode+0xb8/0x138 [<90000000002572d0>] tlb_do_page_fault_1+0x114/0x1b4 The problem is that base address allocated from hugetlbfs is not aligned with pmd size. Here add a checking for hugetlbfs and align base address with pmd size. After this patch the test case "testcases/bin/hugefork02" passes to run. This is similar to the commit 7f24cbc9c4d42db8a3c8484d1 ("mm/mmap: teach generic_get_unmapped_area{_topdown} to handle hugetlb mappings").
CVE-2024-47703	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf, lsm: Add check for BPF LSM return value A bpf prog returning a positive number attached to file_alloc_security hook makes kernel panic. This happens because file system can not filter out the positive number returned by the LSM prog using IS_ERR, and misinterprets this positive number as a file pointer. Given that hook file_alloc_security never returned positive number before the introduction of BPF LSM, and other BPF LSM hooks may encounter similar issues, this patch adds LSM return value check in verifier, to ensure no unexpected value is returned.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-47702	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Fail verification for sign-extension of packet data/data_end/data_meta syzbot reported a kernel crash due to commit 1f1e864b6555 ("bpf: Handle sign-extenstin ctx member accesses"). The reason is due to sign-extension of 32-bit load for packet data/data_end/data_meta uapi field. The original code looks like: r2 = *(s32 *)(r1 + 76) /* load __sk_buff->data */ r3 = *(u32 *)(r1 + 80) /* load __sk_buff->data_end */ r0 = r2 r0 += 8 if r3 > r0 goto +1 ... Note that __sk_buff->data load has 32-bit sign extension. After verification and convert_ctx_accesses(), the final asm code looks like: r2 = *(u64 *)(r1 +208) r2 = (s32)r2 r3 = *(u64 *)(r1 +80) r0 = r2 r0 += 8 if r3 > r0 goto pc+1 ... Note that 'r2 = (s32)r2' may make the kernel __sk_buff->data address invalid which may cause runtime failure. Currently, in C code, typically we have void *data = (void *)(long)skb->data; void *data_end = (void *)(long)skb->data_end; ... and it will generate r2 = *(u64 *)(r1 +208) r3 = *(u64 *)(r1 +80) r0 = r2 r0 += 8 if r3 > r0 goto pc+1 If we allow sign-extension, void *data = (void *)(long)(int)skb->data; void *data_end = (void *)(long)skb->data_end; ... the generated code looks like r2 = *(u64 *)(r1 +208) r2 <<= 32 r2 s>>= 32 r3 = *(u64 *)(r1 +80) r0 = r2 r0 += 8 if r3 > r0 goto pc+1 and this will cause verification failure since "r2 <<= 32" is not allowed as "r2" is a packet pointer. To fix this issue for case r2 = *(s32 *)(r1 + 76) /* load __sk_buff->data */ this patch added additional checking in is_valid_access() callback function for packet data/data_end/data_meta access. If those accesses are with sign-extenstion, the verification will fail. [1] https://lore.kernel.org/bpf/000000000000c90eee061d236d37@google.com/
CVE-2024-50138	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Use raw_spinlock_t in ringbuf The function __bpf_ringbuf_reserve is invoked from a tracepoint, which disables preemption. Using spinlock_t in this context can lead to a "sleep in atomic" warning in the RT variant. This issue is illustrated in the example below: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 556208, name: test_progs preempt_count: 1, expected: 0 RCU nest depth: 1, expected: 1 INFO: lockdep is turned off. Preemption disabled at: [] migrate_enable+0xc0/0x39c CPU: 7 PID: 556208 Comm: test_progs Tainted: G Hardware name: Qualcomm SA8775P Ride (DT) Call trace: dump_backtrace+0xac/0x130 show_stack+0x1c/0x30 dump_stack_lvl+0xac/0xe8 dump_stack+0x18/0x30 __might_resched+0x3bc/0x4fc rt_spin_lock+0x8c/0x1a4 __bpf_ringbuf_reserve+0xc4/0x254 bpf_ringbuf_reserve_dynptr+0x5c/0xdc bpf_prog_ac3d15160d62622a_test_read_write+0x104/0x238 trace_call_bpf+0x238/0x774 perf_call_bpf_enter.isra.0+0x104/0x194 perf_syscall_enter+0x2f8/0x510 trace_sys_enter+0x39c/0x564 syscall_trace_enter+0x220/0x3c0 do_el0_svc+0x138/0x1dc el0_svc+0x54/0x130 el0t_64_sync_handler+0x134/0x150 el0t_64_sync+0x17c/0x180 Switch the spinlock to raw_spinlock_t to avoid this error.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-50178	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: cpufreq: loongson3: Use raw_smp_processor_id() in do_service_request() Use raw_smp_processor_id() instead of plain smp_processor_id() in do_service_request(), otherwise we may get some errors with the driver enabled: BUG: using smp_processor_id() in preemptible [00000000] code: (udev-worker)/208 caller is loongson3_cpufreq_probe+0x5c/0x250 [loongson3_cpufreq]
CVE-2024-35794	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: dm-raid: really frozen sync_thread during suspend 1) commit f52f5c71f3d4 ("md: fix stopping sync thread") remove MD_RECOVERY_FROZEN from __md_stop_writes() and doesn't realize that dm-raid relies on __md_stop_writes() to frozen sync_thread indirectly. Fix this problem by adding MD_RECOVERY_FROZEN in md_stop_writes(), and since stop_sync_thread() is only used for dm-raid in this case, also move stop_sync_thread() to md_stop_writes(). 2) The flag MD_RECOVERY_FROZEN doesn't mean that sync thread is frozen, it only prevent new sync_thread to start, and it can't stop the running sync thread; In order to frozen sync_thread, after seting the flag, stop_sync_thread() should be used. 3) The flag MD_RECOVERY_FROZEN doesn't mean that writes are stopped, use it as condition for md_stop_writes() in raid_postsuspend() doesn't look correct. Consider that reentrant stop_sync_thread() do nothing, always call md_stop_writes() in raid_postsuspend(). 4) raid_message can set/clear the flag MD_RECOVERY_FROZEN at anytime, and if MD_RECOVERY_FROZEN is cleared while the array is suspended, new sync_thread can start unexpected. Fix this by disallow raid_message() to change sync_thread status during suspend. Note that after commit f52f5c71f3d4 ("md: fix stopping sync thread"), the test shell/lvconvert-raid-reshape.sh start to hang in stop_sync_thread(), and with previous fixes, the test won't hang there anymore, however, the test will still fail and complain that ext4 is corrupted. And with this patch, the test won't hang due to stop_sync_thread() or fail due to ext4 is corrupted anymore. However, there is still a deadlock related to dm-raid456 that will be fixed in following patches.
CVE-2024-35931	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Skip do PCI error slot reset during RAS recovery Why: The PCI error slot reset maybe triggered after inject ue to UMC multi times, this caused system hang. [557.371857] amdgpu 0000:af:00.0: amdgpu: GPU reset succeeded, trying to resume [557.373718] [drm] PCIE GART of 512M enabled. [557.373722] [drm] PTB located at 0x0000031FED700000 [557.373788] [drm] VRAM is lost due to GPU reset! [557.373789] [drm] PSP is resuming... [557.547012] mlx5_core 0000:55:00.0: mlx5_pci_err_detected Device state = 1 pci_status: 0. Exit, result = 3, need reset [557.547067] [drm] PCI error: detected callback, state(1)!! [557.547069] [drm] No support for XGMI hive yet... [557.548125] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 0. Enter [557.607763] mlx5_core 0000:55:00.0: wait vital counter value 0x16b5b after 1 iterations [557.607777] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 1. Exit, err = 0, result = 5, recovered [557.610492] [drm] PCI error: slot reset callback!! ... [560.689382] amdgpu 0000:3f:00.0: amdgpu: GPU reset(2) succeeded! [560.689546] amdgpu 0000:5a:00.0: amdgpu: GPU reset(2) succeeded! [560.689562] general protection fault, probably for non-canonical address 0x5f080b54534f611f: 0000 [#1] SMP NOPTI [560.701008] CPU: 16 PID: 2361 Comm: kworker/u448:9 Tainted: G OE

CVE	Severity	CVSS	CVSS Vector	Component	Description
					5.15.0-91-generic #101-Ubuntu [560.712057] Hardware name: Microsoft C278A/C278A, BIOS C2789.5.BS.1C11.AG.1 11/08/2023 [560.720959] Workqueue: amdgpu-reset-hive amdgpu_ras_do_recovery [amdgpu] [560.728887] RIP: 0010:amdgpu_device_gpu_recover.cold+0xbf1/0xcf5 [amdgpu] [560.736891] Code: ff 41 89 c6 e9 1b ff ff ff 44 0f b6 45 b0 e9 4f ff ff ff be 01 00 00 00 4c 89 e7 e8 76 c9 8b ff 44 0f b6 45 b0 e9 3c fd ff ff <48> 83 ba 18 02 00 00 00 0f 84 6a f8 ff ff 48 8d 7a 78 be 01 00 00 [560.757967] RSP: 0018:ffa0000032e53d80 EFLAGS: 00010202 [560.763848] RAX: ffa00000001dfd10 RBX: ffa0000000197090 RCX: ffa0000032e53db0 [560.771856] RDX: 5f080b54534f5f07 RSI: 0000000000000000 RDI: ff11000128100010 [560.779867] RBP: ffa0000032e53df0 R08: 0000000000000000 R09: ffffffffffe77f08 [560.787879] R10: 0000000000ffff0a R11: 0000000000000001 R12: 0000000000000000 [560.795889] R13: ffa0000032e53e00 R14: 0000000000000000 R15: 0000000000000000 [560.803889] FS: 0000000000000000(0000) GS:ff11007e7e800000(0000) knlGS:0000000000000000 [560.812973] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [560.819422] CR2: 000055a04c118e68 CR3: 0000000007410005 CR4: 0000000000771ee0 [560.827433] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [560.835433] DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7: 0000000000000400 [560.843444] PKRU: 55555554 [560.846480] Call Trace: [560.849225] [560.851580] ? show_trace_log_lvl+0x1d6/0x2ea [560.856488] ? show_trace_log_lvl+0x1d6/0x2ea [560.861379] ? amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu] [560.867778] ? show_regs.part.0+0x23/0x29 [560.872293] ? __die_body.cold+0x8/0xd [560.876502] ? die_addr+0x3e/0x60 [560.880238] ? exc_general_protection+0x1c5/0x410 [560.885532] ? asm_exc_general_protection+0x27/0x30 [560.891025] ? amdgpu_device_gpu_recover.cold+0xbf1/0xcf5 [amdgpu] [560.898323] amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu] [560.904520] process_one_work+0x228/0x3d0 How: In RAS recovery, mode-1 reset is issued from RAS fatal error handling and expected all the nodes in a hive to be reset. no need to issue another mode-1 during this procedure.
CVE-2024-41008	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: change vm->task_info handling This patch changes the handling and lifecycle of vm->task_info object. The major changes are: - vm->task_info is a dynamically allocated ptr now, and its uasge is reference counted. - introducing two new helper funcs for task_info lifecycle management - amdgpu_vm_get_task_info: reference counts up task_info before returning this info - amdgpu_vm_put_task_info: reference counts down task_info - last put to task_info() frees task_info from the vm. This patch also does logistical changes required for existing usage of vm->task_info. V2: Do not block all the prints when task_info not found (Felix) V3: Fixed review comments from Felix - Fix wrong indentation - No debug message for -ENOMEM - Add NULL check for task_info - Do not duplicate the debug messages (ti vs no ti) - Get first reference of task_info in vm_init(), put last in vm_fini() V4: Fixed review comments from Felix - fix double reference increment in create_task_info - change amdgpu_vm_get_task_info_pasid - additional changes in amdgpu_gem.c while porting

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-21961	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: fix truesize for mb-xdp-pass case When mb-xdp is set and return is XDP_PASS, packet is converted from xdp_buff to sk_buff with xdp_update_skb_shared_info() in bnxt_xdp_build_skb(). bnxt_xdp_build_skb() passes incorrect truesize argument to xdp_update_skb_shared_info(). The truesize is calculated as BNXT_RX_PAGE_SIZE * sinfo->nr_frags but the skb_shared_info was wiped by napi_build_skb() before. So it stores sinfo->nr_frags before bnxt_xdp_build_skb() and use it instead of getting skb_shared_info from xdp_get_shared_info_from_buff(). Splat looks like: ------------[cut here]------------ WARNING: CPU: 2 PID: 0 at net/core/skbuff.c:6072 skb_try_coalesce+0x504/0x590 Modules linked in: xt_nat xt_tcpudp veth af_packet xt_conntrack nft_chain_nat xt_MASQUERADE nf_conntrack_netlink xfrm_user xt_addrtype nft_coms CPU: 2 UID: 0 PID: 0 Comm: swapper/2 Not tainted 6.14.0-rc2+ #3 RIP: 0010:skb_try_coalesce+0x504/0x590 Code: 4b fd ff ff 49 8b 34 24 40 80 e6 40 0f 84 3d fd ff ff 49 8b 74 24 48 40 f6 c6 01 0f 84 2e fd ff ff 48 8d 4e ff e9 25 fd ff ff <0f> 0b e99 RSP: 0018:ffffb62c4120caa8 EFLAGS: 00010287 RAX: 0000000000000003 RBX: ffffb62c4120cb14 RCX: 0000000000000ec0 RDX: 0000000000001000 RSI: ffffa06e5d7dc000 RDI: 0000000000000003 RBP: ffffa06e5d7ddec0 R08: ffffa06e6120a800 R09: ffffa06e7a119900 R10: 0000000000002310 R11: ffffa06e5d7dcec0 R12: ffffe4360575f740 R13: ffffe43600000000 R14: 0000000000000002 R15: 0000000000000002 FS: 0000000000000000(0000) GS:ffffa0755f700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f147b76b0f8 CR3: 00000001615d4000 CR4: 00000000007506f0 PKRU: 55555554 Call Trace: ? __warn+0x84/0x130 ? skb_try_coalesce+0x504/0x590 ? report_bug+0x18a/0x1a0 ? handle_bug+0x53/0x90 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? skb_try_coalesce+0x504/0x590 inet_frag_reasm_finish+0x11f/0x2e0 ip_defrag+0x37a/0x900 ip_local_deliver+0x51/0x120 ip_sublist_rcv_finish+0x64/0x70 ip_sublist_rcv+0x179/0x210 ip_list_rcv+0xf9/0x130 How to reproduce: ip link set $interface1 xdp obj xdp_pass.o ip link set $interface1 mtu 9000 up ip a a 10.0.0.1/24 dev $interface1 ip link set $interfac2 mtu 9000 up ip a a 10.0.0.2/24 dev $interface2 ping 10.0.0.1 -s 65000 Following ping.py patch adds xdp-mb-pass case. so ping.py is going to be able to reproduce this issue.
CVE-2024-46834	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ethtool: fail closed if we can't get max channel used in indirection tables Commit 0d1b7d6c9274 ("bnxt: fix crashes when reducing ring count with active RSS contexts") proves that allowing indirection table to contain channels with out of bounds IDs may lead to crashes. Currently the max channel check in the core gets skipped if driver can't fetch the indirection table or when we can't allocate memory. Both of those conditions should be extremely rare but if they do happen we should try to be safe and fail the channel change.
CVE-2024-49968	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ext4: filesystems without casefold feature cannot be mounted with siphash When mounting the ext4 filesystem, if the default hash version is set to DX_HASH_SIPHASH but the casefold feature is not set, exit the mounting.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-50304	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ipv4: ip_tunnel: Fix suspicious RCU usage warning in ip_tunnel_find() The per-netns IP tunnel hash table is protected by the RTNL mutex and ip_tunnel_find() is only called from the control path where the mutex is taken. Add a lockdep expression to hlist_for_each_entry_rcu() in ip_tunnel_find() in order to validate that the mutex is held and to silence the suspicious RCU usage warning [1]. [1] WARNING: suspicious RCU usage 6.12.0-rc3-custom-gd95d9a31aceb #139 Not tainted ----------------------------- net/ipv4/ip_tunnel.c:221 RCU-list traversed in non-reader section!! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by ip/362: #0: ffffffff86fc7cb0 (rtnl_mutex) {+.+.}-{3:3}, at: rtnetlink_rcv_msg+0x377/0xf60 stack backtrace: CPU: 12 UID: 0 PID: 362 Comm: ip Not tainted 6.12.0-rc3-custom-gd95d9a31aceb #139 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Call Trace: dump_stack_lvl+0xba/0x110 lockdep_rcu_suspicious.cold+0x4f/0xd6 ip_tunnel_find+0x435/0x4d0 ip_tunnel_newlink+0x517/0x7a0 ipgre_newlink+0x14c/0x170 __rtnl_newlink+0x1173/0x19c0 rtnl_newlink+0x6c/0xa0 rtnetlink_rcv_msg+0x3cc/0xf60 netlink_rcv_skb+0x171/0x450 netlink_unicast+0x539/0x7f0 netlink_sendmsg+0x8c1/0xd80 ____sys_sendmsg+0x8f9/0xc20 ___sys_sendmsg+0x197/0x1e0 __sys_sendmsg+0x122/0x1f0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-37925	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: jfs: reject on-disk inodes of an unsupported type Syzbot has reported the following BUG: kernel BUG at fs/inode.c:668! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 3 UID: 0 PID: 139 Comm: jfsCommit Not tainted 6.12.0-rc4-syzkaller-00085-g4e46774408d9 #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014 RIP: 0010:clear_inode+0x168/0x190 Code: 4c 89 f7 e8 ba fe e5 ff e9 61 ff ff ff 44 89 f1 80 e1 07 80 c1 03 38 c1 7c c1 4c 89 f7 e8 90 ff e5 ff eb b7 0b e8 01 5d 7f ff 90 0f 0b e8 f9 5c 7f ff 90 0f 0b e8 f1 5c 7f RSP: 0018:ffffc900027dfae8 EFLAGS: 00010093 RAX: ffffffff82157a87 RBX: 0000000000000001 RCX: ffff888104d4b980 RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000000 RBP: ffffc900027dfc90 R08: ffffffff82157977 R09: fffff520004fbf38 R10: dfffffc0000000000 R11: fffff520004fbf38 R12: dfffffc0000000000 R13: ffff88811315bc00 R14: ffff88811315bda8 R15: ffff88811315bb80 FS: 0000000000000000(0000) GS:ffff888135f00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005565222e0578 CR3: 0000000026ef0000 CR4: 00000000000006f0 Call Trace: ? __die_body+0x5f/0xb0 ? die+0x9e/0xc0 ? do_trap+0x15a/0x3a0 ? clear_inode+0x168/0x190 ? do_error_trap+0x1dc/0x2c0 ? clear_inode+0x168/0x190 ? __pfx_do_error_trap+0x10/0x10 ? report_bug+0x3cd/0x500 ? handle_invalid_op+0x34/0x40 ? clear_inode+0x168/0x190 ? exc_invalid_op+0x38/0x50 ? asm_exc_invalid_op+0x1a/0x20 ? clear_inode+0x57/0x190 ? clear_inode+0x167/0x190 ? clear_inode+0x168/0x190 ? clear_inode+0x167/0x190 jfs_evict_inode+0xb5/0x440 ? __pfx_jfs_evict_inode+0x10/0x10 evict+0x4ea/0x9b0 ? __pfx_evict+0x10/0x10 ? iput+0x713/0xa50 txUpdateMap+0x931/0xb10 ? __pfx_txUpdateMap+0x10/0x10 jfs_lazycommit+0x49a/0xb80 ? _raw_spin_unlock_irqrestore+0x8f/0x140 ? lockdep_hardirqs_on+0x99/0x150 ? __pfx_jfs_lazycommit+0x10/0x10 ? __pfx_default_wake_function+0x10/0x10 ? __kthread_parkme+0x169/0x1d0 ? __pfx_jfs_lazycommit+0x10/0x10 kthread+0x2f2/0x390 ? __pfx_jfs_lazycommit+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x4d/0x80 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 This happens when 'clear_inode()' makes an attempt to finalize an underlying JFS inode of unknown type. According to JFS layout description from https://jfs.sourceforge.net/project/pub/jfslayout.pdf, inode types from 5 to 15 are reserved for future extensions and should not be encountered on a valid filesystem. So add an extra check for valid inode type in 'copy_from_dinode()'.
CVE-2024-46823	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: kunit/overflow: Fix UB in overflow_allocation_test The 'device_name' array doesn't exist out of the 'overflow_allocation_test' function scope. However, it is being used as a driver name when calling 'kunit_driver_create' from 'kunit_device_register'. It produces the kernel panic with KASAN enabled. Since this variable is used in one place only, remove it and pass the device name into kunit_device_register directly as an ascii string.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-50289	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: media: av7110: fix a spectre vulnerability As warned by smatch: drivers/staging/media/av7110/av7110_ca.c:270 dvb_ca_ioctl() warn: potential spectre issue 'av7110->ci_slot' [w] (local cap) There is a spectre-related vulnerability at the code. Fix it.
CVE-2024-49885	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: mm, slub: avoid zeroing kmalloc redzone Since commit 946fa0dbf2d8 ("mm/slub: extend redzone check to extra allocated kmalloc space than requested"), setting orig_size treats the wasted space (object_size - orig_size) as a redzone. However with init_on_free=1 we clear the full object->size, including the redzone. Additionally we clear the object metadata, including the stored orig_size, making it zero, which makes check_object() treat the whole object as a redzone. These issues lead to the following BUG report with "slub_debug=FUZ init_on_free=1": [0.000000] ========== ========== = [0.000000] BUG kmalloc-8 (Not tainted): kmalloc Redzone overwritten [0.000000] ---------- [0.000000] [0.000000] 0xffff000010032858-0xffff00001003285f @offset=2136. First byte 0x0 instead of 0xcc [0.000000] FIX kmalloc-8: Restoring kmalloc Redzone 0xffff000010032858-0xffff00001003285f=0xcc [0.000000] Slab 0xfffffdffc0400c80 objects=36 used=23 fp=0xffff000010032a18 flags=0x3fffe0000000200(workingset|node=0|zone=0|lastcpupid=0x1ffff) [0.000000] Object 0xffff000010032858 @offset=2136 fp=0xffff0000100328c8 [0.000000] [0.000000] Redzone ffff000010032850: cc cc cc cc cc cc cc cc ..... [0.000000] Object ffff000010032858: cc cc cc cc cc cc cc cc ..... [0.000000] Redzone ffff000010032860: cc cc cc cc cc cc cc cc ..... [0.000000] Padding ffff0000100328b4: 00 00 00 00 00 00 00 00 00 00 00 00 ..... [0.000000] CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.11.0-rc3-next-20240814-00004-g61844c55c3f4 #144 [0.000000] Hardware name: NXP i.MX95 19X19 board (DT) [0.000000] Call trace: [0.000000] dump_backtrace+0x90/0xe8 [0.000000] show_stack+0x18/0x24 [0.000000] dump_stack_lvl+0x74/0x8c [0.000000] dump_stack+0x18/0x24 [0.000000] print_trailer+0x150/0x218 [0.000000] check_object+0xe4/0x454 [0.000000] free_to_partial_list+0x2f8/0x5ec To address the issue, use orig_size to clear the used area. And restore the value of orig_size after clear the remaining area. When CONFIG_SLUB_DEBUG not defined, (get_orig_size()' directly returns s->object_size. So when using memset to init the area, the size can simply be orig_size, as orig_size returns object_size when CONFIG_SLUB_DEBUG not enabled. And orig_size can never be bigger than object_size.
CVE-2024-42317	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: avoid PMD-size page cache if needed xarray can't support arbitrary page cache size. the largest and supported page cache size is defined as MAX_PAGECACHE_ORDER by commit 099d90642a71 ("mm/filemap: make MAX_PAGECACHE_ORDER acceptable to xarray"). However, it's possible to have 512MB page cache in the huge memory's collapsing path on ARM64 system whose base page size is 64KB. 512MB page cache is breaking the limitation and a warning is raised when the xarray entry is split as shown in the following example. [root@dhcp-10-26-1-207 ~]# cat /proc/1/smaps | grep KernelPageSize KernelPageSize: 64 kB [root@dhcp-10-26-1-207 ~]# cat /tmp/test.c : int main(int argc, char **argv) { const char *filename = TEST_XFS_FILENAME; int

CVE	Severity	CVSS	CVSS Vector	Component	Description
					fd = 0; void *buf = (void *)-1, *p; int pgsize = getpagesize(); int ret = 0; if (pgsize != 0x10000) { fprintf(stdout, "System with 64KB base page size is required!\n"); return -EPERM; } system("echo 0 > /sys/devices/virtual/bdi/253:0/read_ahead_kb"); system("echo 1 > /proc/sys/vm/drop_caches"); /* Open the xfs file */ fd = open(filename, O_RDONLY); assert(fd > 0); /* Create VMA */ buf = mmap(NULL, TEST_MEM_SIZE, PROT_READ, MAP_SHARED, fd, 0); assert(buf != (void *)-1); fprintf(stdout, "mapped buffer at 0x%p\n", buf); /* Populate VMA */ ret = madvise(buf, TEST_MEM_SIZE, MADV_NOHUGEPAGE); assert(ret == 0); ret = madvise(buf, TEST_MEM_SIZE, MADV_POPULATE_READ); assert(ret == 0); /* Collapse VMA */ ret = madvise(buf, TEST_MEM_SIZE, MADV_HUGEPAGE); assert(ret == 0); ret = madvise(buf, TEST_MEM_SIZE, MADV_COLLAPSE); if (ret) { fprintf(stdout, "Error %d to madvise(MADV_COLLAPSE)\n", errno); goto out; } /* Split xarray entry. Write permission is needed */ munmap(buf, TEST_MEM_SIZE); buf = (void *)-1; close(fd); fd = open(filename, O_RDWR); assert(fd > 0); fallocate(fd, FALLOC_FL_KEEP_SIZE | FALLOC_FL_PUNCH_HOLE, TEST_MEM_SIZE - pgsize, pgsize); out: if (buf != (void *)-1) munmap(buf, TEST_MEM_SIZE); if (fd > 0) close(fd); return ret; } [root@dhcp-10-26-1-207 ~]# gcc /tmp/test.c -o /tmp/test [root@dhcp-10-26-1-207 ~]# /tmp/test ------------[cut here]------------ WARNING: CPU: 25 PID: 7560 at lib/xarray.c:1025 xas_split_alloc+0xf8/0x128 Modules linked in: nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib \ nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct \ nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 \ ip_set rfkill nf_tables nfnetlink vfat fat virtio_balloon drm fuse \ xfs libcrc32c crct10dif_ce ghash_ce sha2_ce sha256_arm64 virtio_net \ sha1_ce net_failover virtio_blk virtio_console failover dimlib virtio_mmio CPU: 25 PID: 7560 Comm: test Kdump: loaded Not tainted 6.10.0-rc7-gavin+ #9 Hardware name: QEMU KVM Virtual Machine, BIOS edk2-20240524-1.el9 05/24/2024 pstate: 83400005 (Nzcv daif +PAN -UAO +TCO +DIT -SSBS BTYPE=--) pc : xas_split_alloc+0xf8/0x128 lr : split_huge_page_to_list_to_order+0x1c4/0x780 sp : ffff8000ac32f660 x29: ffff8000ac32f660 x28: ffff0000e0969eb0 x27: ffff8000ac32f6c0 x26: 0000000000000c40 x25: ffff0000e0969eb0 x24: 000000000000000d x23: ffff8000ac32f6c0 x22: fffffffdfc0700000 x21: 0000000000000000 x20: 0000000000000000 x19: fffffffdfc0700000 x18: 0000000000000000 x17: 0000000000000000 x16: ffffd5f3708ffc70 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: ffffffffffffffc0 x10: 0000000000000040 x9 : ffffd5f3708e692c x8 : 0000000000000003 x7 : 0000000000000000 x6 : ffff0000e0969eb8 x5 : ffffd5f37289e378 x4 : 0000000000000000 x3 : 0000000000000c40 x2 : 000000000000000d x1 : 000000000000000c x0 : 0000000000000000 Call trace: xas_split_alloc+0xf8/0x128 split_huge_page_to_list_to_order+0x1c4/0x780 truncate_inode_partial_folio+0xdc/0x160 truncate_inode_pages_range+0x1b4/0x4a8 truncate_pagecache_range+0x84/0xa ---truncated---

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-21696	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: mm: clear uffd-wp PTE/PMD state on mremap() When mremap()ing a memory region previously registered with userfaultfd as write-protected but without UFFD_FEATURE_EVENT_REMAP, an inconsistency in flag clearing leads to a mismatch between the vma flags (which have uffd-wp cleared) and the pte/pmd flags (which do not have uffd-wp cleared). This mismatch causes a subsequent mprotect(PROT_WRITE) to trigger a warning in page_table_check_pte_flags() due to setting the pte to writable while uffd-wp is still set. Fix this by always explicitly clearing the uffd-wp pte/pmd flags on any such mremap() so that the values are consistent with the existing clearing of VM_UFFD_WP. Be careful to clear the logical flag regardless of its physical form; a PTE bit, a swap PTE bit, or a PTE marker. Cover PTE, huge PMD and hugetlb paths.
CVE-2024-56647	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net: Fix icmp host relookup triggering ip_rt_bug arp link failure may trigger ip_rt_bug while xfrm enabled, call trace is: WARNING: CPU: 0 PID: 0 at net/ipv4/route.c:1241 ip_rt_bug+0x14/0x20 Modules linked in: CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.0-rc6-00077-g2e1b3cc9d7f7 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:ip_rt_bug+0x14/0x20 Call Trace: ip_send_skb+0x14/0x40 __icmp_send+0x42d/0x6a0 ipv4_link_failure+0xe2/0x1d0 arp_error_report+0x3c/0x50 neigh_invalidate+0x8d/0x100 neigh_timer_handler+0x2e1/0x330 call_timer_fn+0x21/0x120 __run_timer_base.part.0+0x1c9/0x270 run_timer_softirq+0x4c/0x80 handle_softirqs+0xac/0x280 irq_exit_rcu+0x62/0x80 sysvec_apic_timer_interrupt+0x77/0x90 The script below reproduces this scenario: ip xfrm policy add src 0.0.0.0/0 dst 0.0.0.0/0 \ dir out priority 0 ptype main flag localok icmp ip l a veth1 type veth ip a a 192.168.141.111/24 dev veth0 ip l s veth0 up ping 192.168.141.155 -c 1 icmp_route_lookup() create input routes for locally generated packets while xfrm relookup ICMP traffic.Then it will set input route (dst->out = ip_rt_bug) to skb for DESTUNREACH. For ICMP err triggered by locally generated packets, dst->dev of output route is loopback. Generally, xfrm relookup verification is not required on loopback interfaces (net.ipv4.conf.lo.disable_xfrm = 1). Skip icmp relookup for locally generated packets to fix it.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-26596	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net: dsa: fix netdev_priv() dereference before check on non-DSA netdevice events After the blamed commit, we started doing this dereference for every NETDEV_CHANGEUPPER and NETDEV_PRECHANGEUPPER event in the system. static inline struct dsa_port *dsa_user_to_port(const struct net_device *dev) { struct dsa_user_priv *p = netdev_priv(dev); return p->dp; } Which is obviously bogus, because not all net_devices have a netdev_priv() of type struct dsa_user_priv. But struct dsa_user_priv is fairly small, and p->dp means dereferencing 8 bytes starting with offset 16. Most drivers allocate that much private memory anyway, making our access not fault, and we discard the bogus data quickly afterwards, so this wasn't caught. But the dummy interface is somewhat special in that it calls alloc_netdev() with a priv size of 0. So every netdev_priv() dereference is invalid, and we get this when we emit a NETDEV_PRECHANGEUPPER event with a VLAN as its new upper: $ ip link add dummy1 type dummy $ ip link add link dummy1 name dummy1.100 type vlan id 100 [43.309174] ========== ========== [43.316456] BUG: KASAN: slab-out-of-bounds in dsa_user_prechangeupper+0x30/0xe8 [43.323835] Read of size 8 at addr ffff3f86481d2990 by task ip/374 [43.330058] [43.342436] Call trace: [43.366542] dsa_user_prechangeupper+0x30/0xe8 [43.371024] dsa_user_netdevice_event+0xb38/0xee8 [43.375768] notifier_call_chain+0xa4/0x210 [43.379985] raw_notifier_call_chain+0x24/0x38 [43.384464] __netdev_upper_dev_link+0x3ec/0x5d8 [43.389120] netdev_upper_dev_link+0x70/0xa8 [43.393424] register_vlan_dev+0x1bc/0x310 [43.397554] vlan_newlink+0x210/0x248 [43.401247] rtnl_newlink+0x9fc/0xe30 [43.404942] rtnetlink_rcv_msg+0x378/0x580 Avoid the kernel oops by dereferencing after the type check, as customary.
CVE-2024-40999	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net: ena: Add validation for completion descriptors consistency Validate that `first` flag is set only for the first descriptor in multi-buffer packets. In case of an invalid descriptor, a reset will occur. A new reset reason for RX data corruption has been added.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-49926	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: rcu-tasks: Fix access non-existent percpu rtpcp variable in rcu_tasks_need_gpcb() For kernels built with CONFIG_FORCE_NR_CPUS=y, the nr_cpu_ids is defined as NR_CPUS instead of the number of possible cpus, this will cause the following system panic: smpboot: Allowing 4 CPUs, 0 hotplug CPUs ... setup_percpu: NR_CPUS:512 nr_cpumask_bits:512 nr_cpu_ids:512 nr_node_ids:1 ... BUG: unable to handle page fault for address: ffffffff9911c8c8 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 0 PID: 15 Comm: rcu_tasks_trace Tainted: G W 6.6.21 #1 5dc7acf91a5e8e9ac9dcfc35bee0245691283ea6 RIP: 0010:rcu_tasks_need_gpcb+0x25d/0x2c0 RSP: 0018:ffffa371c00a3e60 EFLAGS: 00010082 CR2: ffffffff9911c8c8 CR3: 000000040fa20005 CR4: 00000000001706f0 Call Trace: ? __die+0x23/0x80 ? page_fault_oops+0xa4/0x180 ? exc_page_fault+0x152/0x180 ? asm_exc_page_fault+0x26/0x40 ? rcu_tasks_need_gpcb+0x25d/0x2c0 ? __pfx_rcu_tasks_kthread+0x40/0x40 rcu_tasks_one_gp+0x69/0x180 rcu_tasks_kthread+0x94/0xc0 kthread+0xe8/0x140 ? __pfx_kthread+0x40/0x40 ret_from_fork+0x34/0x80 ? __pfx_kthread+0x40/0x40 ret_from_fork_asm+0x1b/0x80 Considering that there may be holes in the CPU numbers, use the maximum possible cpu number, instead of nr_cpu_ids, for configuring enqueue and dequeue limits. [neeraj.upadhyay: Fix htmldocs build error reported by Stephen Rothwell]
CVE-2024-50137	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: reset: starfive: jh71x0: Fix accessing the empty member on JH7110 SoC data->asserted will be NULL on JH7110 SoC since commit 82327b127d41 ("reset: starfive: Add StarFive JH7110 reset driver") was added. Add the judgment condition to avoid errors when calling reset_control_status on JH7110 SoC.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-53128	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: sched/task_stack: fix object_is_on_stack() for KASAN tagged pointers When CONFIG_KASAN_SW_TAGS and CONFIG_KASAN_STACK are enabled, the object_is_on_stack() function may produce incorrect results due to the presence of tags in the obj pointer, while the stack pointer does not have tags. This discrepancy can lead to incorrect stack object detection and subsequently trigger warnings if CONFIG_DEBUG_OBJECTS is also enabled. Example of the warning: ODEBUG: object 3eff800082ea7bb0 is NOT on stack ffff800082ea0000, but annotated. ----------[cut here]---------- WARNING: CPU: 0 PID: 1 at lib/debugobjects.c:557 __debug_object_init+0x330/0x364 Modules linked in: CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.0-rc5 #4 Hardware name: linux,dummy-virt (DT) pstate: 600000c5 (nZCv daIF -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __debug_object_init+0x330/0x364 lr : __debug_object_init+0x330/0x364 sp : ffff800082ea7b40 x29: ffff800082ea7b40 x28: 98ff0000c0164518 x27: 98ff0000c0164534 x26: ffff800082d93ec8 x25: 0000000000000001 x24: 1cff0000c00172a0 x23: 0000000000000000 x22: ffff800082d93ed0 x21: ffff800081a24418 x20: 3eff800082ea7bb0 x19: efff800000000000 x18: 0000000000000000 x17: 00000000000000ff x16: 0000000000000047 x15: 206b63617473206e x14: 0000000000000018 x13: ffff800082ea7780 x12: 0ffff800082ea78e x11: 0ffff800082ea790 x10: 0ffff800082ea79d x9 : 34d77febe173e800 x8 : 34d77febe173e800 x7 : 0000000000000001 x6 : 0000000000000001 x5 : feff800082ea74b8 x4 : ffff800082870a90 x3 : ffff80008018d3c4 x2 : 0000000000000001 x1 : ffff800082858810 x0 : 0000000000000050 Call trace: __debug_object_init+0x330/0x364 debug_object_init_on_stack+0x30/0x3c schedule_hrtimeout_range_clock+0xac/0x26c schedule_hrtimeout+0x1c/0x30 wait_task_inactive+0x1d4/0x25c kthread_bind_mask+0x28/0x98 init_rescuer+0x1e8/0x280 workqueue_init+0x1a0/0x3cc kernel_init_freeable+0x118/0x200 kernel_init+0x28/0x1f0 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- ODEBUG: object 3eff800082ea7bb0 is NOT on stack ffff800082ea0000, but annotated. ----------[cut here]----------
CVE-2024-50028	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Reference count the zone in thermal_zone_get_by_id() There are places in the thermal netlink code where nothing prevents the thermal zone object from going away while being accessed after it has been returned by thermal_zone_get_by_id(). To address this, make thermal_zone_get_by_id() get a reference on the thermal zone device object to be returned with the help of get_device(), under thermal_list_lock, and adjust all of its callers to this change with the help of the cleanup.h infrastructure.
CVE-2024-53219	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: virtiofs: use pages instead of pointer for kernel direct IO When trying to insert a 10MB kernel module kept in a virtio-fs with cache disabled, the following warning was reported: ----------[cut here]---------- WARNING: CPU: 1 PID: 404 at mm/page_alloc.c:4551 ..... Modules linked in: CPU: 1 PID: 404 Comm: insmod Not tainted 6.9.0-rc5+ #123 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) ..... RIP: 0010:__alloc_pages+0x2bf/0x380 ..... Call Trace: ? __warn+0x8e/0x150 ? __alloc_pages+0x2bf/0x380 __kmalloc_large_node+0x86/0x160 __kmalloc+0x33c/0x480 virtio_fs_enqueue_req+0x240/0x6d0 virtio_fs_wake_pending_and_unlock+0x7f/0x190

CVE	Severity	CVSS	CVSS Vector	Component	Description
					queue_request_and_unlock+0x55/0x60 fuse_simple_request+0x152/0x2b0 fuse_direct_io+0x5d2/0x8c0 fuse_file_read_iter+0x121/0x160 __kernel_read+0x151/0x2d0 kernel_read+0x45/0x50 kernel_read_file+0x1a9/0x2a0 init_module_from_file+0x6a/0xe0 idempotent_init_module+0x175/0x230 __x64_sys_finit_module+0x5d/0xb0 x64_sys_call+0x1c3/0x9e0 do_syscall_64+0x3d/0xc0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 ..... ---[end trace 0000000000000000]--- The warning is triggered as follows: 1) syscall finit_module() handles the module insertion and it invokes kernel_read_file() to read the content of the module first. 2) kernel_read_file() allocates a 10MB buffer by using vmalloc() and passes it to kernel_read(). kernel_read() constructs a kvec iter by using iov_iter_kvec() and passes it to fuse_file_read_iter(). 3) virtio-fs disables the cache, so fuse_file_read_iter() invokes fuse_direct_io(). As for now, the maximal read size for kvec iter is only limited by fc->max_read. For virtio-fs, max_read is UINT_MAX, so fuse_direct_io() doesn't split the 10MB buffer. It saves the address and the size of the 10MB-sized buffer in out_args[0] of a fuse request and passes the fuse request to virtio_fs_wake_pending_and_unlock(). 4) virtio_fs_wake_pending_and_unlock() uses virtio_fs_enqueue_req() to queue the request. Because virtiofs need DMA-able address, so virtio_fs_enqueue_req() uses kmalloc() to allocate a bounce buffer for all fuse args, copies these args into the bounce buffer and passed the physical address of the bounce buffer to virtiofsd. The total length of these fuse args for the passed fuse request is about 10MB, so copy_args_to_argbuf() invokes kmalloc() with a 10MB size parameter and it triggers the warning in __alloc_pages(): if (WARN_ON_ONCE_GFP(order > MAX_PAGE_ORDER, gfp)) return NULL; 5) virtio_fs_enqueue_req() will retry the memory allocation in a kworker, but it won't help, because kmalloc() will always return NULL due to the abnormal size and finit_module() will hang forever. A feasible solution is to limit the value of max_read for virtio-fs, so the length passed to kmalloc() will be limited. However it will affect the maximal read size for normal read. And for virtio-fs write initiated from kernel, it has the similar problem but now there is no way to limit fc->max_write in kernel. So instead of limiting both the values of max_read and max_write in kernel, introducing use_pages_for_kvec_io in fuse_conn and setting it as true in virtiofs. When use_pages_for_kvec_io is enabled, fuse will use pages instead of pointer to pass the KVEC_IO data. After switching to pages for KVEC_IO data, these pages will be used for DMA through virtio-fs. If these pages are backed by vmalloc(), {flush|invalidate}_kernel_vmap_range() are necessary to flush or invalidate the cache before the DMA operation. So add two new fields in fuse_args_pages to record the base address of vmalloc area and the condition indicating whether invalidation is needed. Perform the flush in fuse_get_user_pages() for write operations and the invalidation in fuse_release_user_pages() for read operations. It may seem necessary to introduce another fie ---truncated---

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-50090	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe/oa: Fix overflow in oa batch buffer By default xe_bb_create_job() appends a MI_BATCH_BUFFER_END to batch buffer, this is not a problem if batch buffer is only used once but oa reuses the batch buffer for the same metric and at each call it appends a MI_BATCH_BUFFER_END, printing the warning below and then overflowing. [381.072016] ----------[cut here]---------- [381.072019] xe 0000:00:02.0: [drm] Assertion `bb->len * 4 + bb_prefetch(q->gt) <= size` failed! platform: LUNARLAKE subplatform: 1 graphics: Xe2_LPG / Xe2_HPG 20.04 step B0 media: Xe2_LPM / Xe2_HPM 20.00 step B0 tile: 0 VRAM 0 B GT: 0 type 1 So here checking if batch buffer already have MI_BATCH_BUFFER_END if not append it. v2: - simply fix, suggestion from Ashutosh (cherry picked from commit 9ba0e0f30ca42a98af3689460063edfb6315718a)
CVE-2024-26785	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: iommufd: Fix protection fault in iommufd_test_syz_conv_iova Syzkaller reported the following bug: general protection fault, probably for non-canonical address 0xdfffffc0000000038: 0000 [#1] SMP KASAN KASAN: null-ptr-deref in range [0x0000000000001c0-0x0000000000001c7] Call Trace: lock_acquire lock_acquire+0x1ce/0x4f0 down_read+0x93/0x4a0 iommufd_test_syz_conv_iova+0x56/0x1f0 iommufd_test_access_rw.isra.0+0x2ec/0x390 iommufd_test+0x1058/0x1e30 iommufd_fops_ioctl+0x381/0x510 vfs_ioctl __do_sys_ioctl __se_sys_ioctl __x64_sys_ioctl+0x170/0x1e0 do_syscall_x64 do_syscall_64+0x71/0x140 This is because the new iommufd_access_change_ioas() sets access->ioas to NULL during its process, so the lock might be gone in a concurrent racing context. Fix this by doing the same access->ioas sanity as iommufd_access_rw() and iommufd_access_pin_pages() functions do.
CVE-2024-49994	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: block: fix integer overflow in BLKSECDISCARD I independently rediscovered commit 22d24a544b0d49bbcbd61c8c0eaf77d3c9297155 block: fix overflow in blk_ioctl_discard() but for secure erase. Same problem: uint64_t r[2] = {512, 18446744073709551104ULL}; ioctl(fd, BLKSECDISCARD, r); will enter near infinite loop inside blkdev_issue_secure_erase(): a.out: attempt to access beyond end of device loop0: rw=5, sector=3399043073, nr_sectors = 1024 limit=2048 bio_check_eod: 3286214 callbacks suppressed
CVE-2024-42066	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix potential integer overflow in page size calculation Explicitly cast tbo->page_alignment to u64 before bit-shifting to prevent overflow when assigning to min_page_size.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-53187	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: io_uring: check for overflows in io_pin_pages WARNING: CPU: 0 PID: 5834 at io_uring/memmap.c:144 io_pin_pages+0x149/0x180 io_uring/memmap.c:144 CPU: 0 UID: 0 PID: 5834 Comm: syz-executor825 Not tainted 6.12.0-next-20241118-syzkaller #0 Call Trace: __io_uaddr_map+0xfb/0x2d0 io_uring/memmap.c:183 io_rings_map io_uring/io_uring.c:2611 [inline] io_allocate_scq_urings+0x1c0/0x650 io_uring/io_uring.c:3470 io_uring_create+0x5b5/0xc00 io_uring/io_uring.c:3692 io_uring_setup io_uring/io_uring.c:3781 [inline] ... io_pin_pages()'s uaddr parameter came directly from the user and can be garbage. Don't just add size to it as it can overflow.
CVE-2024-25740	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	A memory leak flaw was found in the UBI driver in drivers/mtd/ubi/attach.c in the Linux kernel through 6.7.4 for UBI_IOCATT, because kobj->name is not released.
CVE-2024-53084	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: Break an object reference loop When remaining resources are being cleaned up on driver close, outstanding VM mappings may result in resources being leaked, due to an object reference loop, as shown below, with each object (or set of objects) referencing the object below it: PVR GEM Object GPU scheduler "finished" fence GPU scheduler “scheduled” fence PVR driver “done” fence PVR Context PVR VM Context PVR VM Mappings PVR GEM Object The reference that the PVR VM Context has on the VM mappings is a soft one, in the sense that the freeing of outstanding VM mappings is done as part of VM context destruction; no reference counts are involved, as is the case for all the other references in the loop. To break the reference loop during cleanup, free the outstanding VM mappings before destroying the PVR Context associated with the VM context.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-27011	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix memleak in map from abort path The delete set command does not rely on the transaction object for element removal, therefore, a combination of delete element + delete set from the abort path could result in restoring twice the refcount of the mapping. Check for inactive element in the next generation for the delete element command in the abort path, skip restoring state if next generation bit has been already cleared. This is similar to the activate logic using the set walk iterator. [6170.286929] ------------[cut here]------------ [6170.286939] WARNING: CPU: 6 PID: 790302 at net/netfilter/nf_tables_api.c:2086 nf_tables_chain_destroy+0x1f7/0x220 [nf_tables] [6170.287071] Modules linked in: [...] [6170.287633] CPU: 6 PID: 790302 Comm: kworker/6:2 Not tainted 6.9.0-rc3+ #365 [6170.287768] RIP: 0010:nf_tables_chain_destroy+0x1f7/0x220 [nf_tables] [6170.287886] Code: df 48 8d 7d 58 e8 69 2e 3b df 48 8b 7d 58 e8 80 1b 37 df 48 8d 7d 68 e8 57 2e 3b df 48 8b 7d 68 e8 6e 1b 37 df 48 89 ef eb c4 <0f> 0b 48 83 c4 08 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc 0f [6170.287895] RSP: 0018:ffff888134b8fd08 EFLAGS: 00010202 [6170.287904] RAX: 0000000000000001 RBX: ffff888125bffb28 RCX: dffffc0000000000 [6170.287912] RDX: 0000000000000003 RSI: ffffffffa20298ab RDI: ffff88811ebe4750 [6170.287919] RBP: ffff88811ebe4700 R08: ffff88838e812650 R09: fffffbfff0623a55 [6170.287926] R10: ffffffff8311d2af R11: 0000000000000001 R12: ffff888125bffb10 [6170.287933] R13: ffff888125bffb10 R14: dead000000000122 R15: dead000000000100 [6170.287940] FS: 0000000000000000(0000) GS:ffff888390b00000(0000) knlGS:0000000000000000 [6170.287948] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [6170.287955] CR2: 00007fd31fc00710 CR3: 0000000133f60004 CR4: 00000000001706f0 [6170.287962] Call Trace: [6170.287967] [6170.287973] ? __warn+0x9f/0x1a0 [6170.287986] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables] [6170.288092] ? report_bug+0x1b1/0x1e0 [6170.287986] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables] [6170.288092] ? report_bug+0x1b1/0x1e0 [6170.288104] ? handle_bug+0x3c/0x70 [6170.288112] ? exc_invalid_op+0x17/0x40 [6170.288120] ? asm_exc_invalid_op+0x1a/0x20 [6170.288132] ? nf_tables_chain_destroy+0x2b/0x220 [nf_tables] [6170.288243] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables] [6170.288366] ? nf_tables_chain_destroy+0x2b/0x220 [nf_tables] [6170.288483] nf_tables_trans_destroy_work+0x588/0x590 [nf_tables]

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-27012	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: restore set elements when delete set fails From abort path, nft_mapelem_activate() needs to restore refcounters to the original state. Currently, it uses the set->ops->walk() to iterate over these set elements. The existing set iterator skips inactive elements in the next generation, this does not work from the abort path to restore the original state since it has to skip active elements instead (not inactive ones). This patch moves the check for inactive elements to the set iterator callback, then it reverses the logic for the .activate case which needs to skip active elements. Toggle next generation bit for elements when delete set command is invoked and call nft_clear() from .activate (abort) path to restore the next generation bit. The splat below shows an object in mappings memleak: [43929.457523] ------------ [cut here]------------ [43929.457532] WARNING: CPU: 0 PID: 1139 at include/net/netfilter/nf_tables.h:1237 nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables] [...] [43929.458014] RIP: 0010:nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables] [43929.458076] Code: 83 f8 01 77 ab 49 8d 7c 24 08 e8 37 5e d0 de 49 8b 6c 24 08 48 8d 7d 50 e8 e9 5c d0 de 8b 45 50 8d 50 ff 89 55 50 85 c0 75 86 <0f> 0b eb 82 0f 0b eb b3 0f 1f 40 00 90 90 90 90 90 90 90 90 90 90 [43929.458081] RSP: 0018:ffff888140f9f4b0 EFLAGS: 00010246 [43929.458086] RAX: 0000000000000000 RBX: ffff8881434f5288 RCX: dffffc0000000000 [43929.458090] RDX: 00000000ffffffff RSI: ffffffffa26d28a7 RDI: ffff88810ecc9550 [43929.458093] RBP: ffff88810ecc9500 R08: 0000000000000001 R09: ffffed10281f3e8f [43929.458096] R10: 0000000000000003 R11: ffff0000ffff0000 R12: ffff8881434f52a0 [43929.458100] R13: ffff888140f9f5f4 R14: ffff888151c7a800 R15: 0000000000000002 [43929.458103] FS: 00007f0c687c4740(0000) GS:ffff888390800000(0000) knlGS:0000000000000000 [43929.458107] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [43929.458111] CR2: 00007f58dbe5b008 CR3: 0000000123602005 CR4: 00000000001706f0 [43929.458114] Call Trace: [43929.458118] [43929.458121] ? __warn+0x9f/0x1a0 [43929.458127] ? nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables] [43929.458188] ? report_bug+0x1b1/0x1e0 [43929.458196] ? handle_bug+0x3c/0x70 [43929.458200] ? exc_invalid_op+0x17/0x40 [43929.458211] ? nft_setelem_data_deactivate+0xd7/0xf0 [nf_tables] [43929.458271] ? nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables] [43929.458332] nft_mapelem_deactivate+0x24/0x30 [nf_tables] [43929.458392] nft_rhash_walk+0xdd/0x180 [nf_tables] [43929.458453] ? __pfx_nft_rhash_walk+0x10/0x10 [nf_tables] [43929.458512] ? rb_insert_color+0x2e/0x280 [43929.458520] nft_map_deactivate+0xdc/0x1e0 [nf_tables] [43929.458582] ? __pfx_nft_map_deactivate+0x10/0x10 [nf_tables] [43929.458642] ? __pfx_nft_mapelem_deactivate+0x10/0x10 [nf_tables] [43929.458701] ? __rcu_read_unlock+0x46/0x70 [43929.458709] nft_delset+0xff/0x110 [nf_tables] [43929.458769] nft_flush_table+0x16f/0x460 [nf_tables] [43929.458830] nf_tables_deltable+0x501/0x580 [nf_tables]

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-43913	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: nvme: apple: fix device reference counting Drivers must call nvme_uninit_ctrl after a successful nvme_init_ctrl. Split the allocation side out to make the error handling boundary easier to navigate. The apple driver had been doing this wrong, leaking the controller device memory on a tagset failure.
CVE-2024-41023	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: sched/deadline: Fix task_struct reference leak During the execution of the following stress test with linux-rt: stress-ng --cyclic 30 --timeout 30 --minimize --quiet kmemleak frequently reported a memory leak concerning the task_struct: unreferenced object 0xffff8881305b8000 (size 16136): comm "stress-ng", pid 614, jiffies 4294883961 (age 286.412s) object hex dump (first 32 bytes): 02 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .@.............. 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... debug hex dump (first 16 bytes): 53 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 S..... backtrace: [<00000000046b6790>] dup_task_struct+0x30/0x540 [<00000000c5ca0f0b>] copy_process+0x3d9/0x50e0 [<00000000ced59777>] kernel_clone+0xb0/0x770 [<00000000a50befdc>] __do_sys_clone+0xb6/0xf0 [<000000001dbf2008>] do_syscall_64+0x5d/0xf0 [<00000000552900ff>] entry_SYSCALL_64_after_hwframe+0x6e/0x76 The issue occurs in start_dl_timer(), which increments the task_struct reference count and sets a timer. The timer callback, dl_task_timer, is supposed to decrement the reference count upon expiration. However, if enqueue_task_dl() is called before the timer expires and cancels it, the reference count is not decremented, leading to the leak. This patch fixes the reference leak by ensuring the task_struct reference count is properly decremented when the timer is canceled.
CVE-2024-57872	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: pltfrm: Dellocate HBA during ufshcd_pltfrm_remove() This will ensure that the scsi host is cleaned up properly using scsi_host_dev_release(). Otherwise, it may lead to memory leaks.
CVE-2024-56712	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: udmabuf: fix memory leak on last export_udmabuf() error path In export_udmabuf(), if dma_buf_fd() fails because the FD table is full, a dma_buf owning the udmabuf has already been created; but the error handling in udmabuf_create() will tear down the udmabuf without doing anything about the containing dma_buf. This leaves a dma_buf in memory that contains a dangling pointer; though that doesn't seem to lead to anything bad except a memory leak. Fix it by moving the dma_buf_fd() call out of export_udmabuf() so that we can give it different error handling. Note that the shape of this code changed a lot in commit 5e72b2b41a21 ("udmabuf: convert udmabuf driver to use folios"); but the memory leak seems to have existed since the introduction of udmabuf.
CVE-2024-40979	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix kernel crash during resume Currently during resume, QMI target memory is not properly handled, resulting in kernel crash in case DMA remap is not supported: BUG: Bad page state in process kworker/u16:54 pfn:36e80 page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x36e80 page dumped because: nonzero _refcount Call Trace: bad_page free_page_is_bad_report __free_pages_ok __free_pages dma_direct_free dma_free_attrs

CVE	Severity	CVSS	CVSS Vector	Component	Description
					ath12k_qmi_free_target_mem_chunk ath12k_qmi_msg_mem_request_cb The reason is: Once ath12k module is loaded, firmware sends memory request to host. In case DMA remap not supported, ath12k refuses the first request due to failure in allocating with large segment size: ath12k_pci 0000:04:00.0: qmi firmware request memory request ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 7077888 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 8454144 ath12k_pci 0000:04:00.0: qmi dma allocation failed (7077888 B type 1), will try later with small size ath12k_pci 0000:04:00.0: qmi delays mem_request 2 ath12k_pci 0000:04:00.0: qmi firmware request memory request Later firmware comes back with more but small segments and allocation succeeds: ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 262144 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 524288 ath12k_pci 0000:04:00.0: qmi mem seg type 4 size 65536 ath12k_pci 0000:04:00.0: qmi mem seg type 1 size 524288 Now ath12k is working. If suspend is triggered, firmware will be reloaded during resume. As same as before, firmware requests two large segments at first. In ath12k_qmi_msg_mem_request_cb() segment count and size are assigned: ab->qmi.mem_seg_count == 2 ab->qmi.target_mem[0].size == 7077888 ab->qmi.target_mem[1].size == 8454144 Then allocation failed like before and ath12k_qmi_free_target_mem_chunk() is called to free all allocated segments. Note the first segment is skipped because its v.addr is cleared due to allocation failure: chunk->v.addr = dma_alloc_coherent() Also note that this leaks that segment because it has not been freed. While freeing the second segment, a size of 8454144 is passed to dma_free_coherent(). However remember that this segment is allocated at the first time firmware is loaded, before suspend. So its real size is 524288, much smaller than 8454144. As a result kernel found we are freeing some memory which is in use and thus cras ---truncated---
[CVE-2024-23848]()	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel through 6.7.1, there is a use-after-free in cec_queue_msg_fh, related to drivers/media/cec/core/cec-adap.c and drivers/media/cec/core/cec-api.c.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-42075	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix remap of arena. The bpf arena logic didn't account for mremap operation. Add a refcnt for multiple mmap events to prevent use-after-free in arena_vm_close.
CVE-2025-21861	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: mm/migrate_device: don't add folio to be freed to LRU in migrate_device_finalize() If migration succeeded, we called folio_migrate_flags()->mem_cgroup_migrate() to migrate the memcg from the old to the new folio. This will set memcg_data of the old folio to 0. Similarly, if migration failed, memcg_data of the dst folio is left unset. If we call folio_putback_lru() on such folios (memcg_data == 0), we will add the folio to be freed to the LRU, making memcg code unhappy. Running the hmm selftests: # ./hmm-tests ... # RUN hmm.hmm_device_private.migrate ... [102.078007][T14893] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x7ff27d200 pfn:0x13cc00 [102.079974] [T14893] anon flags: 0x17ff00000020018(uptodate|dirty|swapbacked|node=0|zone=2|lastcpupid=0x7ff) [102.082037] [T14893] raw: 017ff00000020018 dead000000000100 dead000000000122 ffff8881353896c9 [102.083687][T14893] raw: 00000007ff27d200 0000000000000000 00000001ffffffff 0000000000000000 [102.085331][T14893] page dumped because: VM_WARN_ON_ONCE_FOLIO(!memcg && !mem_cgroup_disabled()) [102.087230] [T14893] ----------[cut here]---------- [102.088279][T14893] WARNING: CPU: 0 PID: 14893 at ./include/linux/memcontrol.h:726 folio_lruvec_lock_irqsave+0x10e/0x170 [102.090478][T14893] Modules linked in: [102.091244][T14893] CPU: 0 UID: 0 PID: 14893 Comm: hmm-tests Not tainted 6.13.0-09623-g6c216bc522fd #151 [102.093089][T14893] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014 [102.094848][T14893] RIP: 0010:folio_lruvec_lock_irqsave+0x10e/0x170 [102.096104][T14893] Code: ... [102.099908] [T14893] RSP: 0018:ffffc900236c37b0 EFLAGS: 00010293 [102.101152][T14893] RAX: 0000000000000000 RBX: ffffea0004f30000 RCX: ffffffff8183f426 [102.102684][T14893] RDX: ffff8881063cb880 RSI: ffffffff81b8117f RDI: ffff8881063cb880 [102.104227][T14893] RBP: 0000000000000000 R08: 0000000000000005 R09: 0000000000000000 [102.105757][T14893] R10: 0000000000000001 R11: 0000000000000002 R12: ffffc900236c37d8 [102.107296][T14893] R13: ffff888277a2bcb0 R14: 000000000000001f R15: 0000000000000000 [102.108830][T14893] FS: 00007ff27dbdd740(0000) GS:ffff888277a00000(0000) knlGS:0000000000000000 [102.110643] [T14893] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [102.111924][T14893] CR2: 00007ff27d400000 CR3: 000000010866e000 CR4: 0000000000750ef0 [102.113478][T14893] PKRU: 55555554 [102.114172][T14893] Call Trace: [102.114805][T14893] [102.115397] [T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.116547][T14893] ? __warn.cold+0x110/0x210 [102.117461][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.118667][T14893] ? report_bug+0x1b9/0x320 [102.119571][T14893] ? handle_bug+0x54/0x90 [102.120494][T14893] ? exc_invalid_op+0x17/0x50 [102.121433][T14893] ? asm_exc_invalid_op+0x1a/0x20 [102.122435] [T14893] ? __wake_up_klogd.part.0+0x76/0xd0 [102.123506][T14893] ? dump_page+0x4f/0x60 [102.124352][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.125500][T14893]

CVE	Severity	CVSS	CVSS Vector	Component	Description
					folio_batch_move_lru+0xd4/0x200 [102.126577][T14893] ? __pfx_lru_add+0x10/0x10 [102.127505][T14893] __folio_batch_add_and_move+0x391/0x720 [102.128633][T14893] ? __pfx_lru_add+0x10/0x10 [102.129550][T14893] folio_putback_lru+0x16/0x80 [102.130564][T14893] migrate_device_finalize+0x9b/0x530 [102.131640][T14893] dmirror_migrate_to_device.constprop.0+0x7c5/0xad0 [102.133047][T14893] dmirror_fops_unlocked_ioctl+0x89b/0xc80 Likely, nothing else goes wrong: putting the last folio reference will remove the folio from the LRU again. So besides memcg complaining, adding the folio to be freed to the LRU is just an unnecessary step. The new flow resembles what we have in migrate_folio_move(): add the dst to the lru, rem ---truncated---
CVE-2024-50027	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Free tzp copy along with the thermal zone The object pointed to by tz->tzp may still be accessed after being freed in thermal_zone_device_unregister(), so move the freeing of it to the point after the removal completion has been completed at which it cannot be accessed any more.
CVE-2022-38096	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	A NULL pointer dereference vulnerability was found in vmwgfx driver in drivers/gpu/vmxgfx/vmxgfx_execbuf.c in GPU component of Linux kernel with device file '/dev/dri/renderD128 (or Dxxx)'. This flaw allows a local attacker with a user account on the system to gain privilege, causing a denial of service(DoS).
CVE-2024-58012	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-dai: Ensure DAI widget is valid during params Each cpu DAI should associate with a widget. However, the topology might not create the right number of DAI widgets for aggregated amps. And it will cause NULL pointer deference. Check that the DAI widget associated with the CPU DAI is valid to prevent NULL pointer deference due to missing DAI widgets in topologies with aggregated amps.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-56702	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark raw_tp arguments with PTR_MAYBE_NULL Arguments to a raw tracepoint are tagged as trusted, which carries the semantics that the pointer will be non-NULL. However, in certain cases, a raw tracepoint argument may end up being NULL. More context about this issue is available in [0]. Thus, there is a discrepancy between the reality, that raw_tp arguments can actually be NULL, and the verifier's knowledge, that they are never NULL, causing explicit NULL checks to be deleted, and accesses to such pointers potentially crashing the kernel. To fix this, mark raw_tp arguments as PTR_MAYBE_NULL, and then special case the dereference and pointer arithmetic to permit it, and allow passing them into helpers/kfuncs; these exceptions are made for raw_tp programs only. Ensure that we don't do this when ref_obj_id > 0, as in that case this is an acquired object and doesn't need such adjustment. The reason we do mask_raw_tp_trusted_reg logic is because other will recheck in places whether the register is a trusted_reg, and then consider our register as untrusted when detecting the presence of the PTR_MAYBE_NULL flag. To allow safe dereference, we enable PROBE_MEM marking when we see loads into trusted pointers with PTR_MAYBE_NULL. While trusted raw_tp arguments can also be passed into helpers or kfuncs where such broken assumption may cause issues, a future patch set will tackle their case separately, as PTR_TO_BTF_ID (without PTR_TRUSTED) can already be passed into helpers and causes similar problems. Thus, they are left alone for now. It is possible that these checks also permit passing non-raw_tp args that are trusted PTR_TO_BTF_ID with null marking. In such a case, allowing dereference when pointer is NULL expands allowed behavior, so won't regress existing programs, and the case of passing these into helpers is the same as above and will be dealt with later. Also update the failure case in tp_btf_nullable selftest to capture the new behavior, as the verifier will no longer cause an error when directly dereference a raw tracepoint argument marked as __nullable. [0]: https://lore.kernel.org/bpf/ZrCZS6nisraEqehw@jlelli-thinkpadt14gen4.remote.csb
CVE-2024-42151	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: mark bpf_dummy_struct_ops.test_1 parameter as nullable Test case dummy_st_ops/dummy_init_ret_value passes NULL as the first parameter of the test_1() function. Mark this parameter as nullable to make verifier aware of such possibility. Otherwise, NULL check in the test_1() code: SEC("struct_ops/test_1") int BPF_PROG(test_1, struct bpf_dummy_ops_state *state) { if (!state) return ...; ... access state ... } Might be removed by verifier, thus triggering NULL pointer dereference under certain conditions.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2023-52920	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: support non-r10 register spill/fill to/from stack in precision tracking Use instruction (jump) history to record instructions that performed register spill/fill to/from stack, regardless if this was done through read-only r10 register, or any other register after copying r10 into it *and* potentially adjusting offset. To make this work reliably, we push extra per-instruction flags into instruction history, encoding stack slot index (spi) and stack frame number in extra 10 bit flags we take away from prev_idx in instruction history. We don't touch idx field for maximum performance, as it's checked most frequently during backtracking. This change removes basically the last remaining practical limitation of precision backtracking logic in BPF verifier. It fixes known deficiencies, but also opens up new opportunities to reduce number of verified states, explored in the subsequent patches. There are only three differences in selftests' BPF object files according to veristat, all in the positive direction (less states). File Program Insns (A) Insns (B) Insns (DIFF) States (A) States (B) States (DIFF) ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- test_cls_redirect_dynptr.bpf.linked3.o cls_redirect 2987 2864 -123 (-4.12%) 240 231 -9 (-3.75%) xdp_synproxy_kern.bpf.linked3.o syncookie_tc 82848 82661 -187 (-0.23%) 5107 5073 -34 (-0.67%) xdp_synproxy_kern.bpf.linked3.o syncookie_xdp 85116 84964 -152 (-0.18%) 5162 5130 -32 (-0.62%) Note, I avoided renaming jmp_history to more generic insn_hist to minimize number of lines changed and potential merge conflicts between bpf and bpf-next trees. Notice also cur_hist_entry pointer reset to NULL at the beginning of instruction verification loop. This pointer avoids the problem of relying on last jump history entry's insn_idx to determine whether we already have entry for current instruction or not. It can happen that we added jump history entry because current instruction is_jmp_point(), but also we need to add instruction flags for stack access. In this case, we don't want to entries, so we need to reuse last added entry, if it is present. Relying on insn_idx comparison has the same ambiguity problem as the one that was fixed recently in [0], so we avoid that. [0] https://patchwork.kernel.org/project/netdevbpf/patch/20231110002638.4168352-3-andrii@kernel.org/
CVE-2024-50009	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: add check for cpufreq_cpu_get's return value cpufreq_cpu_get may return NULL. To avoid NULL-dereference check it and return in case of error. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-41085	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: cxl/mem: Fix no cxl_nvd during pmem region auto-assembling When CXL subsystem is auto-assembling a pmem region during cxl endpoint port probing, always hit below calltrace. BUG: kernel NULL pointer dereference, address: 0000000000000078 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page RIP: 0010:cxl_pmem_region_probe+0x22e/0x360 [cxl_pmem] Call Trace: ? __die+0x24/0x70 ? page_fault_oops+0x82/0x160 ? do_user_addr_fault+0x65/0x6b0 ? exc_page_fault+0x7d/0x170 ? asm_exc_page_fault+0x26/0x30 ? cxl_pmem_region_probe+0x22e/0x360 [cxl_pmem] ? cxl_pmem_region_probe+0x1ac/0x360 [cxl_pmem] cxl_bus_probe+0x1b/0x60 [cxl_core] really_probe+0x173/0x410 ? __pfx___device_attach_driver+0x10/0x10 __driver_probe_device+0x80/0x170 driver_probe_device+0x1e/0x90 __device_attach_driver+0x90/0x120 bus_for_each_drv+0x84/0xe0 __device_attach+0xbc/0x1f0 bus_probe_device+0x90/0xa0 device_add+0x51c/0x710 devm_cxl_add_pmem_region+0x1b5/0x380 [cxl_core] cxl_bus_probe+0x1b/0x60 [cxl_core] The cxl_nvd of the memdev needs to be available during the pmem region probe. Currently the cxl_nvd is registered after the endpoint port probe. The endpoint probe, in the case of autoassembly of regions, can cause a pmem region probe requiring the not yet available cxl_nvd. Adjust the sequence so this dependency is met. This requires adding a port parameter to cxl_find_nvdimm_bridge() that can be used to query the ancestor root port. The endpoint port is not yet available, but will share a common ancestor with its parent, so start the query from there instead.
CVE-2024-26948	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add a dc_state NULL check in dc_state_release [How] Check wheather state is NULL before releasing it.
CVE-2024-42081	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe/xe_devcoredump: Check NULL before assignments Assign 'xe_devcoredump_snapshot *' and 'xe_device *' only if 'coredump' is not NULL. v2 - Fix commit messages. v3 - Define variables before code. (Ashutosh/Jose) v4 - Drop return check for coredump_to_xe. (Jose/Rodrigo) v5 - Modify misleading commit message. (Matt)
CVE-2024-42065	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Add a NULL check in xe_ttm_stolen_mgr_init Add an explicit check to ensure that the mgr is not NULL.
CVE-2024-46705	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe: reset mmio mappings with devm Set our various mmio mappings to NULL. This should make it easier to catch something rogue trying to mess with mmio after device removal. For example, we might unmap everything and then start hitting some mmio address which has already been unmamped by us and then remapped by something else, causing all kinds of carnage.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2025-22070	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: fs/9p: fix NULL pointer dereference on mkdir When a 9p tree was mounted with option 'posixacl', parent directory had a default ACL set for its subdirectories, e.g.: setfacl -m default:group:simpsons:rwx parentdir then creating a subdirectory crashed 9p client, as v9fs_fid_add() call in function v9fs_vfs_mkdir_dotl() sets the passed 'fid' pointer to NULL (since dafbe689736) even though the subsequent v9fs_set_create_acl() call expects a valid non-NULL 'fid' pointer: [37.273191] BUG: kernel NULL pointer dereference, address: 0000000000000000 ... [37.322338] Call Trace: [37.323043] [37.323621] ? __die (arch/x86/kernel/dumpstack.c:421 arch/x86/kernel/dumpstack.c:434) [37.324448] ? page_fault_oops (arch/x86/mm/fault.c:714) [37.325532] ? search_module_extables (kernel/module/main.c:3733) [37.326742] ? p9_client_walk (net/9p/client.c:1165) 9pnet [37.328006] ? search_bpf_extables (kernel/bpf/core.c:804) [37.329142] ? exc_page_fault (./arch/x86/include/asm/paravirt.h:686 arch/x86/mm/fault.c:1488 arch/x86/mm/fault.c:1538) [37.330196] ? asm_exc_page_fault (./arch/x86/include/asm/idtentry.h:574) [37.331330] ? p9_client_walk (net/9p/client.c:1165) 9pnet [37.332562] ? v9fs_fid_xattr_get (fs/9p/xattr.c:30) 9p [37.333824] v9fs_fid_xattr_set (fs/9p/fid.h:23 fs/9p/xattr.c:121) 9p [37.335077] v9fs_set_acl (fs/9p/acl.c:276) 9p [37.336112] v9fs_set_create_acl (fs/9p/acl.c:307) 9p [37.337326] v9fs_vfs_mkdir_dotl (fs/9p/vfs_inode_dotl.c:411) 9p [37.338590] vfs_mkdir (fs/namei.c:4313) [37.339535] do_mkdirat (fs/namei.c:4336) [37.340465] __x64_sys_mkdir (fs/namei.c:4354) [37.341455] do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) [37.342447] entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Fix this by simply swapping the sequence of these two calls in v9fs_vfs_mkdir_dotl(), i.e. calling v9fs_set_create_acl() before v9fs_fid_add().

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-42083	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ionic: fix kernel panic due to multi-buffer handling Currently, the ionic_run_xdp() doesn't handle multi-buffer packets properly for XDP_TX and XDP_REDIRECT. When a jumbo frame is received, the ionic_run_xdp() first makes xdp frame with all necessary pages in the rx descriptor. And if the action is either XDP_TX or XDP_REDIRECT, it should unmap dma-mapping and reset page pointer to NULL for all pages, not only the first page. But it doesn't for SG pages. So, SG pages unexpectedly will be reused. It eventually causes kernel panic. Oops: general protection fault, probably for non-canonical address 0x504f4e4dbebc64ff: 0000 [#1] PREEMPT SMP NOPTI CPU: 3 PID: 0 Comm: swapper/3 Not tainted 6.10.0-rc3+ #25 RIP: 0010:xdp_return_frame+0x42/0x90 Code: 01 75 12 5b 4c 89 e6 5d 31 c9 41 5c 31 d2 41 5d e9 73 fd ff ff 44 8b 6b 20 0f b7 43 0a 49 81 ed 68 01 00 00 49 29 c5 49 01 fd <41> 80 7d0 RSP: 0018:ffff99d00122ce08 EFLAGS: 00010202 RAX: 0000000000005453 RBX: ffff8d325f904000 RCX: 0000000000000001 RDX: 00000000670e1000 RSI: 000000011f90d000 RDI: 504f4e4d4c4b4a49 RBP: ffff99d003907740 R08: 0000000000000000 R09: 0000000000000000 R10: 000000011f90d000 R11: 0000000000000000 R12: ffff8d325f904010 R13: 504f4e4dbebc64fd R14: ffff8d3242b070c8 R15: ffff99d0039077c0 FS: 0000000000000000(0000) GS:ffff8d399f780000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f41f6c85e38 CR3: 000000037ac30000 CR4: 00000000007506f0 PKRU: 55555554 Call Trace: ? die_addr+0x33/0x90 ? exc_general_protection+0x251/0x2f0 ? asm_exc_general_protection+0x22/0x30 ? xdp_return_frame+0x42/0x90 ionic_tx_clean+0x211/0x280 [ionic 15881354510e6a9c655c59c54812b319ed2cd015] ionic_tx_cq_service+0xd3/0x210 [ionic 15881354510e6a9c655c59c54812b319ed2cd015] ionic_txrx_napi+0x41/0x1b0 [ionic 15881354510e6a9c655c59c54812b319ed2cd015] __napi_poll.constprop.0+0x29/0x1b0 net_rx_action+0x2c4/0x350 handle_softirqs+0xf4/0x320 irq_exit_rcu+0x78/0xa0 common_interrupt+0x77/0x90
CVE-2025-22037	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference in alloc_preauth_hash() The Client send malformed smb2 negotiate request. ksmbd return error response. Subsequently, the client can send smb2 session setup even thought conn->preauth_info is not allocated. This patch add KSMBD_SESS_NEED_SETUP status of connection to ignore session setup request if smb2 negotiate phase is not complete.
CVE-2024-43826	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: nfs: pass explicit offset/count to trace events nfs_folio_length is unsafe to use without having the folio locked and a check for a NULL ->f_mapping that protects against truncations and can lead to kernel crashes. E.g. when running xfstests generic/065 with all nfs trace points enabled. Follow the model of the XFS trace points and pass in an explicit offset and length. This has the additional benefit that these values can be more accurate as some of the users touch partial folio ranges.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-36478	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: null_blk: fix null-ptr-dereference while configuring 'power' and 'submit_queues' Writing 'power' and 'submit_queues' concurrently will trigger kernel panic: Test script: modprobe null_blk nr_devices=0 mkdir -p /sys/kernel/config/nullb/nullb0 while true; do echo 1 > submit_queues; echo 4 > submit_queues; done & while true; do echo 1 > power; echo 0 > power; done Test result: BUG: kernel NULL pointer dereference, address: 0000000000000148 Oops: 0000 [#1] PREEMPT SMP RIP: 0010:__lock_acquire+0x41d/0x28f0 Call Trace: lock_acquire+0x121/0x450 down_write+0x5f/0x1d0 simple_recursive_removal+0x12f/0x5c0 blk_mq_debugfs_unregister_hctxs+0x7c/0x100 blk_mq_update_nr_hw_queues+0x4a3/0x720 nullb_update_nr_hw_queues+0x71/0xf0 [null_blk] nullb_device_submit_queues_store+0x79/0xf0 [null_blk] configfs_write_iter+0x119/0x1e0 vfs_write+0x326/0x730 ksys_write+0x74/0x150 This is because del_gendisk() can concurrent with blk_mq_update_nr_hw_queues(): nullb_device_power_store nullb_apply_submit_queues null_del_dev del_gendisk nullb_update_nr_hw_queues if (!dev->nullb) // still set while gendisk is deleted return 0 blk_mq_update_nr_hw_queues dev->nullb = NULL Fix this problem by resuing the global mutex to protect nullb_device_power_store() and nullb_update_nr_hw_queues() from configfs.
CVE-2024-53205	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: phy: realtek: usb: fix NULL deref in rtk_usb2phy_probe In rtk_usb2phy_probe() devm_kzalloc() may return NULL but this returned value is not checked.
CVE-2024-53204	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: phy: realtek: usb: fix NULL deref in rtk_usb3phy_probe In rtk_usb3phy_probe() devm_kzalloc() may return NULL but this returned value is not checked.
CVE-2025-21723	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix possible crash when setting up bsg fails If bsg_setup_queue() fails, the bsg_queue is assigned a non-NULL value. Consequently, in mpi3mr_bsg_exit(), the condition "if(!mrioc->bsg_queue)" will not be satisfied, preventing execution from entering bsg_remove_queue(), which could lead to the following crash: BUG: kernel NULL pointer dereference, address: 000000000000041c Call Trace: mpi3mr_bsg_exit+0x1f/0x50 [mpi3mr] mpi3mr_remove+0x6f/0x340 [mpi3mr] pci_device_remove+0x3f/0xb0 device_release_driver_internal+0x19d/0x220 unbind_store+0xa4/0xb0 kernfs_fop_write_iter+0x11f/0x200 vfs_write+0x1fc/0x3e0 ksys_write+0x67/0xe0 do_syscall_64+0x38/0x80 entry_SYSCALL_64_after_hwframe+0x78/0xe2

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-56620	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: qcom: Only free platform MSIs when ESI is enabled Otherwise, it will result in a NULL pointer dereference as below: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 Call trace: mutex_lock+0xc/0x54 platform_device_msi_free_irqs_all+0x14/0x20 ufs_qcom_remove+0x34/0x48 [ufs_qcom] platform_remove+0x28/0x44 device_remove+0x4c/0x80 device_release_driver_internal+0xd8/0x178 driver_detach+0x50/0x9c bus_remove_driver+0x6c/0xbc driver_unregister+0x30/0x60 platform_driver_unregister+0x14/0x20 ufs_qcom_pltform_exit+0x18/0xb94 [ufs_qcom] __arm64_sys_delete_module+0x180/0x260 invoke_syscall+0x44/0x100 el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x34/0xdc el0t_64_sync_handler+0xc0/0xc4 el0t_64_sync+0x190/0x194
CVE-2025-22062	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: sctp: add mutual exclusion in proc_sctp_do_udp_port() We must serialize calls to sctp_udp_sock_stop() and sctp_udp_sock_start() or risk a crash as syzbot reported: Oops: general protection fault, probably for non-canonical address 0xdfffffc000000000d: 0000 [#1] SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000068-0x000000000000006f] CPU: 1 UID: 0 PID: 6551 Comm: syz.1.44 Not tainted 6.14.0-syzkaller-g7f2ff7b62617 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 RIP: 0010:kernel_sock_shutdown+0x47/0x70 net/socket.c:3653 Call Trace: udp_tunnel_sock_release+0x68/0x80 net/ipv4/udp_tunnel_core.c:181 sctp_udp_sock_stop+0x71/0x160 net/sctp/protocol.c:930 proc_sctp_do_udp_port+0x264/0x450 net/sctp/sysctl.c:553 proc_sys_call_handler+0x3d0/0x5b0 fs/proc/proc_sysctl.c:601 iter_file_splice_write+0x91c/0x1150 fs/splice.c:738 do_splice_from fs/splice.c:935 [inline] direct_splice_actor+0x18f/0x6c0 fs/splice.c:1158 splice_direct_to_actor+0x342/0xa30 fs/splice.c:1102 do_splice_direct_actor fs/splice.c:1201 [inline] do_splice_direct+0x174/0x240 fs/splice.c:1227 do_sendfile+0xafd/0xe50 fs/read_write.c:1368 __do_sys_sendfile64 fs/read_write.c:1429 [inline] __se_sys_sendfile64 fs/read_write.c:1415 [inline] __x64_sys_sendfile64+0x1d8/0x220 fs/read_write.c:1415 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline]
CVE-2025-37860	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: sfc: fix NULL dereferences in ef100_process_design_param() Since cited commit, ef100_probe_main() and hence also ef100_check_design_params() run before efx->net_dev is created; consequently, we cannot netif_set_tso_max_size() or _segs() at this point. Move those netif calls to ef100_probe_netdev(), and also replace netif_err within the design params code with pci_err.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-56544	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: udmabuf: change folios array from kmalloc to kvmalloc When PAGE_SIZE 4096, MAX_PAGE_ORDER 10, 64bit machine, page_alloc only support 4MB. If above this, trigger this warn and return NULL. udmabuf can change size limit, if change it to 3072(3GB), and then alloc 3GB udmabuf, will fail create. [4080.876581] ------------[cut here]------------ [4080.876843] WARNING: CPU: 3 PID: 2015 at mm/page_alloc.c:4556 __alloc_pages+0x2c8/0x350 [4080.878839] RIP: 0010:__alloc_pages+0x2c8/0x350 [4080.879470] Call Trace: [4080.879473] [4080.879473] ? __alloc_pages+0x2c8/0x350 [4080.879475] ? __warn.cold+0x8e/0xe8 [4080.880647] ? __alloc_pages+0x2c8/0x350 [4080.880909] ? report_bug+0xff/0x140 [4080.881175] ? handle_bug+0x3c/0x80 [4080.881556] ? exc_invalid_op+0x17/0x70 [4080.881559] ? asm_exc_invalid_op+0x1a/0x20 [4080.882077] ? udmabuf_create+0x131/0x400 Because MAX_PAGE_ORDER, kmalloc can max alloc 4096 * (1 << 10), 4MB memory, each array entry is pointer(8byte), so can save 524288 pages(2GB). Further more, costly order(order 3) may not be guaranteed that it can be applied for, due to fragmentation. This patch change udmabuf array use kvmalloc_array, this can fallback alloc into vmalloc, which can guarantee allocation for any size and does not affect the performance of kmalloc allocations.
CVE-2024-46698	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: video/aperture: optionally match the device in sysfb_disable() In aperture_remove_conflicting_pci_devices(), we currently only call sysfb_disable() on vga class devices. This leads to the following problem when the pimary device is not VGA compatible: 1. A PCI device with a non-VGA class is the boot display 2. That device is probed first and it is not a VGA device so sysfb_disable() is not called, but the device resources are freed by aperture_detach_platform_device() 3. Non-primary GPU has a VGA class and it ends up calling sysfb_disable() 4. NULL pointer dereference via sysfb_disable() since the resources have already been freed by aperture_detach_platform_device() when it was called by the other device. Fix this by passing a device pointer to sysfb_disable() and checking the device to determine if we should execute it or not. v2: Fix build when CONFIG_SCREEN_INFO is not set v3: Move device check into the mutex Drop primary variable in aperture_remove_conflicting_pci_devices() Drop __init on pci sysfb_pci_dev_is_enabled()

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-42252	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: closures: Change BUG_ON() to WARN_ON() If a BUG_ON() can be hit in the wild, it shouldn't be a BUG_ON() For reference, this has popped up once in the CI, and we'll need more info to debug it: 03240 ------------[cut here]------------ 03240 kernel BUG at lib/closure.c:21! 03240 kernel BUG at lib/closure.c:21! 03240 Internal error: Oops - BUG: 00000000f2000800 [#1] SMP 03240 Modules linked in: 03240 CPU: 15 PID: 40534 Comm: kworker/u80:1 Not tainted 6.10.0-rc4-ktest-ga56da69799bd #25570 03240 Hardware name: linux,dummy-virt (DT) 03240 Workqueue: btree_update btree_interior_update_work 03240 pstate: 00001005 (nzcv daif -PAN -UAO -TCO -DIT +SSBS BTYPE=--) 03240 pc : closure_put+0x224/0x2a0 03240 lr : closure_put+0x24/0x2a0 03240 sp : ffff0000d12071c0 03240 x29: ffff0000d12071c0 x28: dfff800000000000 x27: ffff0000d1207360 03240 x26: 0000000000000040 x25: 0000000000000040 x24: 0000000000000040 03240 x23: ffff0000c1f20180 x22: 0000000000000000 x21: ffff0000c1f20168 03240 x20: 0000000040000000 x19: ffff0000c1f20140 x18: 0000000000000001 03240 x17: 0000000000003aa0 x16: 0000000000003ad0 x15: 1fffe0001c326974 03240 x14: 0000000000000a1e x13: 0000000000000000 x12: 1fffe000183e402d 03240 x11: ffff6000183e402d x10: dfff800000000000 x9 : ffff6000183e402e 03240 x8 : 0000000000000001 x7 : 00009fffe7c1bfd3 x6 : ffff0000c1f2016b 03240 x5 : ffff0000c1f20168 x4 : ffff6000183e402e x3 : ffff800081391954 03240 x2 : 0000000000000001 x1 : 0000000000000000 x0 : 00000000a8000000 03240 Call trace: 03240 closure_put+0x224/0x2a0 03240 bch2_check_for_deadlock+0x910/0x1028 03240 bch2_six_check_for_deadlock+0x1c/0x30 03240 six_lock_slowpath.isra.0+0x29c/0xed0 03240 six_lock_ip_waiter+0xa8/0xf8 03240 __bch2_btree_node_lock_write+0x14c/0x298 03240 bch2_trans_lock_write+0x6d4/0xb10 03240 __bch2_trans_commit+0x135c/0x5520 03240 btree_interior_update_work+0x1248/0x1c10 03240 process_scheduled_works+0x53c/0xd90 03240 worker_thread+0x370/0x8c8 03240 kthread+0x258/0x2e8 03240 ret_from_fork+0x10/0x20 03240 Code: aa1303e0 d63f0020 a94363f7 17ffff8c (d4210000) 03240 ---[end trace 0000000000000000]--- 03240 Kernel panic - not syncing: Oops - BUG: Fatal exception 03240 SMP: stopping secondary CPUs 03241 SMP: failed to stop secondary CPUs 13,15 03241 Kernel Offset: disabled 03241 CPU features: 0x00,00000003,80000008,4240500b 03241 Memory Limit: none 03241 ---[end Kernel panic - not syncing: Oops - BUG: Fatal exception]--- 03246 ========= FAILED TIMEOUT copygc_torture_no_checksum in 7200s
CVE-2024-44956	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe/preempt_fence: enlarge the fence critical section It is really easy to introduce subtle deadlocks in preempt_fence_work_func() since we operate on single global ordered-wq for signalling our preempt fences behind the scenes, so even though we signal a particular fence, everything in the callback should be in the fence critical section, since blocking in the callback will prevent other published fences from signalling. If we enlarge the fence critical section to cover the entire callback, then lockdep should be able to understand this better, and complain if we grab a sensitive lock like vm->lock, which is also held when waiting on preempt fences.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-47736	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: erofs: handle overlapped pclusters out of crafted images properly syzbot reported a task hang issue due to a deadlock case where it is waiting for the folio lock of a cached folio that will be used for cache I/Os. After looking into the crafted fuzzed image, I found it's formed with several overlapped big pclusters as below: Ext: logical offset | length : physical offset | length 0: 0.. 16384 | 16384 : 151552.. 167936 | 16384 1: 16384.. 32768 | 16384 : 155648.. 172032 | 16384 2: 32768.. 49152 | 16384 : 537223168.. 537239552 | 16384 ... Here, extent 0/1 are physically overlapped although it's entirely _impossible_ for normal filesystem images generated by mkfs. First, managed folios containing compressed data will be marked as up-to-date and then unlocked immediately (unlike in-place folios) when compressed I/Os are complete. If physical blocks are not submitted in the incremental order, there should be separate BIOs to avoid dependency issues. However, the current code mis-arranges z_erofs_fill_bio_vec() and BIO submission which causes unexpected BIO waits. Second, managed folios will be connected to their own pclusters for efficient inter-queries. However, this is somewhat hard to implement easily if overlapped big pclusters exist. Again, these only appear in fuzzed images so let's simply fall back to temporary short-lived pages for correctness. Additionally, it justifies that referenced managed folios cannot be truncated for now and reverts part of commit 2080ca1ed3e4 ("erofs: tidy up `struct z_erofs_bvec`") for simplicity although it shouldn't be any difference.
CVE-2024-26686	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: fs/proc: do_task_stat: use sig->stats_lock to gather the threads/children stats lock_task_sighand() can trigger a hard lockup. If NR_CPUS threads call do_task_stat() at the same time and the process has NR_THREADS, it will spin with irqs disabled O(NR_CPUS * NR_THREADS) time. Change do_task_stat() to use sig->stats_lock to gather the statistics outside of ->siglock protected section, in the likely case this code will run lockless.
CVE-2024-41080	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: io_uring: fix possible deadlock in io_register_iowq_max_workers() The io_register_iowq_max_workers() function calls io_put_sq_data(), which acquires the sqd->lock without releasing the uring_lock. Similar to the commit 009ad9f0c6ee ("io_uring: drop ctx->uring_lock before acquiring sqd->lock"), this can lead to a potential deadlock situation. To resolve this issue, the uring_lock is released before calling io_put_sq_data(), and then it is re-acquired after the function call. This change ensures that the locks are acquired in the correct order, preventing the possibility of a deadlock.
CVE-2025-37802	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix WARNING "do not call blocking ops when !TASK_RUNNING" wait_event_timeout() will set the state of the current task to TASK_UNINTERRUPTIBLE, before doing the condition check. This means that ksmbd_durable_scavenger_alive() will try to acquire the mutex while already in a sleeping state. The scheduler warns us by giving the following warning: do not call blocking ops when !TASK_RUNNING; state=2 set at [<0000000061515a6f>] prepare_to_wait_event+0x9f/0x6c0 WARNING: CPU: 2 PID: 4147 at kernel/sched/core.c:10099 __might_sleep+0x12f/0x160 mutex lock is not needed in ksmbd_durable_scavenger_alive().

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-35808	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: md/dm-raid: don't call md_reap_sync_thread() directly Currently md_reap_sync_thread() is called from raid_message() directly without holding 'reconfig_mutex', this is definitely unsafe because md_reap_sync_thread() can change many fields that is protected by 'reconfig_mutex'. However, hold 'reconfig_mutex' here is still problematic because this will cause deadlock, for example, commit 130443d60b1b ("md: refactor idle/frozen_sync_thread() to fix deadlock"). Fix this problem by using stop_sync_thread() to unregister sync_thread, like md/raid did.
CVE-2024-57977	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: memcg: fix soft lockup in the OOM process A soft lockup issue was found in the product with about 56,000 tasks were in the OOM cgroup, it was traversing them when the soft lockup was triggered. watchdog: BUG: soft lockup - CPU#2 stuck for 23s! [VM Thread:1503066] CPU: 2 PID: 1503066 Comm: VM Thread Kdump: loaded Tainted: G Hardware name: Huawei Cloud OpenStack Nova, BIOS RIP: 0010:console_unlock+0x343/0x540 RSP: 0000:ffffb751447db9a0 EFLAGS: 00000247 ORIG_RAX: ffffffffffffff13 RAX: 0000000000000001 RBX: 0000000000000000 RCX: 00000000ffffffff RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000247 RBP: ffffffffafc71f90 R08: 0000000000000000 R09: 0000000000000040 R10: 0000000000000080 R11: 0000000000000000 R12: ffffffffafc74bd0 R13: ffffffffaf60a220 R14: 0000000000000247 R15: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f2fe6ad91f0 CR3: 00000004b2076003 CR4: 0000000000360ee0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: vprintk_emit+0x193/0x280 printk+0x52/0x6e dump_task+0x114/0x130 mem_cgroup_scan_tasks+0x76/0x100 dump_header+0x1fe/0x210 oom_kill_process+0xd1/0x100 out_of_memory+0x125/0x570 mem_cgroup_out_of_memory+0xb5/0xd0 try_charge+0x720/0x770 mem_cgroup_try_charge+0x86/0x180 mem_cgroup_try_charge_delay+0x1c/0x40 do_anonymous_page+0xb5/0x390 handle_mm_fault+0xc4/0x1f0 This is because thousands of processes are in the OOM cgroup, it takes a long time to traverse all of them. As a result, this lead to soft lockup in the OOM process. To fix this issue, call 'cond_resched' in the 'mem_cgroup_scan_tasks' function per 1000 iterations. For global OOM, call 'touch_softlockup_watchdog' per 1000 iterations to avoid this issue.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-27010	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: net/sched: Fix mirred deadlock on device recursion When the mirred action is used on a classful egress qdisc and a packet is mirrored or redirected to self we hit a qdisc lock deadlock. See trace below. [..... other info removed for brevity....] [82.890906] [82.890906] == [82.890906] WARNING: possible recursive locking detected [82.890906] 6.8.0-05205-g77fadd89fe2d-dirty #213 Tainted: G W [82.890906] -- [82.890906] ping/418 is trying to acquire lock: [82.890906] ffff888006994110 (&sch->q.lock){+.-.}-{3:3}, at: __dev_queue_xmit+0x1778/0x3550 [82.890906] [82.890906] but task is already holding lock: [82.890906] ffff888006994110 (&sch->q.lock){+.-.}-{3:3}, at: __dev_queue_xmit+0x1778/0x3550 [82.890906] [82.890906] other info that might help us debug this: [82.890906] Possible unsafe locking scenario: [82.890906] [82.890906] CPU0 [82.890906] ---- [82.890906] lock(&sch->q.lock); [82.890906] lock(&sch->q.lock); [82.890906] [82.890906] *** DEADLOCK *** [82.890906] [..... other info removed for brevity....] Example setup (eth0->eth0) to recreate tc qdisc add dev eth0 root handle 1: htb default 30 tc filter add dev eth0 handle 1: protocol ip prio 2 matchall \ action mirred egress redirect dev eth0 Another example(eth0->eth1->eth0) to recreate tc qdisc add dev eth0 root handle 1: htb default 30 tc filter add dev eth0 handle 1: protocol ip prio 2 matchall \ action mirred egress redirect dev eth1 tc qdisc add dev eth1 root handle 1: htb default 30 tc filter add dev eth1 handle 1: protocol ip prio 2 matchall \ action mirred egress redirect dev eth0 We fix this by adding an owner field (CPU id) to struct Qdisc set after root qdisc is entered. When the softirq enters it a second time, if the qdisc owner is the same CPU, the packet is dropped to break the loop.
CVE-2024-25741	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	printer_write in drivers/usb/gadget/function/f_printer.c in the Linux kernel through 6.7.4 does not properly call usb_ep_queue, which might allow attackers to cause a denial of service or have unspecified other impact.
CVE-2024-25739	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	create_empty_lvol in drivers/mtd/ubi/vtbl.c in the Linux kernel through 6.7.4 can attempt to allocate zero bytes, and crash, because of a missing check for ubi->leb_size.
CVE-2024-26811	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate payload size in ipc response If installing malicious ksmbd-tools, ksmbd.mountd can return invalid ipc response to ksmbd kernel server. ksmbd should validate payload size of ipc response from ksmbd.mountd to avoid memory overrun or slab-out-of-bounds. This patch validate 3 ipc response that has payload.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-42071	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ionic: use dev_consume_skb_any outside of napi If we're not in a NAPI softirq context, we need to be careful about how we call napi_consume_skb(), specifically we need to call it with budget==0 to signal to it that we're not in a safe context. This was found while running some configuration stress testing of traffic and a change queue config loop running, and this curious note popped out: [4371.402645] BUG: using smp_processor_id() in preemptible [00000000] code: ethtool/20545 [4371.402897] caller is napi_skb_cache_put+0x16/0x80 [4371.403120] CPU: 25 PID: 20545 Comm: ethtool Kdump: loaded Tainted: G OE 6.10.0-rc3-netnext+ #8 [4371.403302] Hardware name: HPE ProLiant DL360 Gen10/ProLiant DL360 Gen10, BIOS U32 01/23/2021 [4371.403460] Call Trace: [4371.403613] [4371.403758] dump_stack_lvl+0x4f/0x70 [4371.403904] check_preemption_disabled+0xc1/0xe0 [4371.404051] napi_skb_cache_put+0x16/0x80 [4371.404199] ionic_tx_clean+0x18a/0x240 [ionic] [4371.404354] ionic_tx_cq_service+0xc4/0x200 [ionic] [4371.404505] ionic_tx_flush+0x15/0x70 [ionic] [4371.404653] ? ionic_lif_qcq_deinit.isra.23+0x5b/0x70 [ionic] [4371.404805] ionic_txrx_deinit+0x71/0x190 [ionic] [4371.404956] ionic_reconfigure_queues+0x5f5/0xff0 [ionic] [4371.405111] ionic_set_ringparam+0x2e8/0x3e0 [ionic] [4371.405265] ethnl_set_rings+0x1f1/0x300 [4371.405418] ethnl_default_set_doit+0xbb/0x160 [4371.405571] genl_family_rcv_msg_doit+0xff/0x130 [...] I found that ionic_tx_clean() calls napi_consume_skb() which calls napi_skb_cache_put(), but before that last call is the note /* Zero budget indicate non-NAPI context called us, like netpoll */ and DEBUG_NET_WARN_ON_ONCE(!in_softirq()); Those are pretty big hints that we're doing it wrong. We can pass a context hint down through the calls to let ionic_tx_clean() know what we're doing so it can call napi_consume_skb() correctly.
CVE-2024-36288	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix loop termination condition in gss_free_in_token_pages() The in_token->pages[] array is not NULL terminated. This results in the following KASAN splat: KASAN: maybe wild-memory-access in range [0x04a2013400000008-0x04a201340000000f]

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-47794	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent tailcall infinite loop caused by freplace There is a potential infinite loop issue that can occur when using a combination of tail calls and freplace. In an upcoming selftest, the attach target for entry_freplace of tailcall_freplace.c is subprog_tc of tc_bpf2bpf.c, while the tail call in entry_freplace leads to entry_tc. This results in an infinite loop: entry_tc -> subprog_tc -> entry_freplace --tailcall-> entry_tc. The problem arises because the tail_call_cnt in entry_freplace resets to zero each time entry_freplace is executed, causing the tail call mechanism to never terminate, eventually leading to a kernel panic. To fix this issue, the solution is twofold: 1. Prevent updating a program extended by an freplace program to a prog_array map. 2. Prevent extending a program that is already part of a prog_array map with an freplace program. This ensures that: * If a program or its subprogram has been extended by an freplace program, it can no longer be updated to a prog_array map. * If a program has been added to a prog_array map, neither it nor its subprograms can be extended by an freplace program. Moreover, an extension program should not be tailcalled. As such, return -EINVAL if the program has a type of BPF_PROG_TYPE_EXT when adding it to a prog_array map. Additionally, fix a minor code style issue by replacing eight spaces with a tab for proper formatting.
CVE-2024-46701	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: libfs: fix infinite directory reads for offset dir After we switch tmpfs dir operations from simple_dir_operations to simple_offset_dir_operations, every rename happened will fill new dentry to dest dir's maple tree(&SHMEM_I(inode)->dir_offsets->mt) with a free key starting with octx->newx_offset, and then set newx_offset equals to free key + 1. This will lead to infinite readdir combine with rename happened at the same time, which fail generic/736 in xfstests(detail show as below). 1. create 5000 files(1 2 3...) under one dir 2. call readdir(man 3 readdir) once, and get one entry 3. rename(entry, "TEMPFILE"), then rename("TEMPFILE", entry) 4. loop 2~3, until readdir return nothing or we loop too many times(tmpfs break test with the second condition) We choose the same logic what commit 9b378f6ad48cf ("btrfs: fix infinite directory reads") to fix it, record the last_index when we open dir, and do not emit the entry which index >= last_index. The file->private_data now used in offset dir can use directly to do this, and we also update the last_index when we llseek the dir file. [brauner: only update last_index after seek when offset is zero like Jan suggested]

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-58097	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix RCU stall while reaping monitor destination ring While processing the monitor destination ring, MSDUs are reaped from the link descriptor based on the corresponding buf_id. However, sometimes the driver cannot obtain a valid buffer corresponding to the buf_id received from the hardware. This causes an infinite loop in the destination processing, resulting in a kernel crash. kernel log: ath11k_pci 0000:58:00.0: data msdu_pop: invalid buf_id 309 ath11k_pci 0000:58:00.0: data dp_rx_monitor_link_desc_return failed ath11k_pci 0000:58:00.0: data msdu_pop: invalid buf_id 309 ath11k_pci 0000:58:00.0: data dp_rx_monitor_link_desc_return failed Fix this by skipping the problematic buf_id and reaping the next entry, replacing the break with the next MSDU processing. Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3.6510.30 Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKSWPL_SILICONZ-1
CVE-2024-49990	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: drm/xe/hdcp: Check GSC structure validity Sometimes xe_gsc is not initialized when checked at HDCP capability check. Add gsc structure check to avoid null pointer error.
CVE-2024-50014	MEDIUM	5.5	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: ext4: fix access to uninitialised lock in fc replay path The following kernel trace can be triggered with fstest generic/629 when executed against a filesystem with fast-commit feature enabled: INFO: trying to register non-static key. The code is fine but needs lockdep annotation, or maybe you didn't initialize this object before use? turning off the locking correctness validator. CPU: 0 PID: 866 Comm: mount Not tainted 6.10.0+ #11 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.2-3-gd478f380-prebuilt.qemu.org 04/01/2014 Call Trace: dump_stack_lvl+0x66/0x90 register_lock_class+0x759/0x7d0 __lock_acquire+0x85/0x2630 ? __find_get_block+0xb4/0x380 lock_acquire+0xd1/0x2d0 ? __ext4_journal_get_write_access+0xd5/0x160 _raw_spin_lock+0x33/0x40 ? __ext4_journal_get_write_access+0xd5/0x160 __ext4_journal_get_write_access+0xd5/0x160 ext4_reserve_inode_write+0x61/0xb0 __ext4_mark_inode_dirty+0x79/0x270 ? ext4_ext_replay_set_iblocks+0x2f8/0x450 ext4_ext_replay_set_iblocks+0x330/0x450 ext4_fc_replay+0x14c8/0x1540 ? jread+0x88/0x2e0 ? rcu_is_watching+0x11/0x40 do_one_pass+0x447/0xd00 jbd2_journal_recover+0x139/0x1b0 jbd2_journal_load+0x96/0x390 ext4_load_and_init_journal+0x253/0xd40 ext4_fill_super+0x2cc6/0x3180 ... In the replay path there's an attempt to lock sbi->s_bdev_wb_lock in function ext4_check_bdev_write_error(). Unfortunately, at this point this spinlock has not been initialized yet. Moving it's initialization to an earlier point in __ext4_fill_super() fixes this splat.

CVE	Severity	CVSS	CVSS Vector	Component	Description
CVE-2024-35843	MEDIUM	6.8	CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:H	linux-rittal 6.6.96+git	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Use device rbtree in iopf reporting path The existing I/O page fault handler currently locates the PCI device by calling pci_get_domain_bus_and_slot(). This function searches the list of all PCI devices until the desired device is found. To improve lookup efficiency, replace it with device_rbtree_find() to search the device within the probed device rbtree. The I/O page fault is initiated by the device, which does not have any synchronization mechanism with the software to ensure that the device stays in the probed device tree. Theoretically, a device could be released by the IOMMU subsystem after device_rbtree_find() and before iopf_get_dev_fault_param(), which would cause a use-after-free problem. Add a mutex to synchronize the I/O page fault reporting path and the IOMMU release device path. This lock doesn't introduce any performance overhead, as the conflict between I/O page fault reporting and device releasing is very rare.
CVE-2025-46394	LOW	3.3	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N	busybox 1.36.1	In tar in BusyBox through 1.37.0, a TAR archive can have filenames hidden from a listing through the use of terminal escape sequences.
CVE-2023-52979	UNASSIGNED	N/A	N/A	linux-rittal 6.6.96+git	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.

Recommended Actions

Rittal recommends updating affected devices to firmware version **v10.1.2** or later to address the vulnerabilities listed in this bulletin.

Contact

For questions regarding this bulletin, please contact:
Rittal PSIRT – psirt@rittal.com

Disclaimer