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

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

في ضوء دور الهيئة الوطنية للأمن السيبراني للمساعدة في حماية الفضاء السيبراني الوطني، تود الهيئة مشاركتكم النشرة الأسبوعية للثغرات المسجلة من قبل the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) للأسبوع من ٢٥ أغسطس إلى ٣١ أغسطس. علماً أنه يتم تصنيف هذه الثغرات باستخدام معيار Common Vulnerability Scoring System (CVSS) حيث يتم تصنيف الثغرات بناء على التالي:

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

As part of NCA duties to help securing the cyberspace and protecting national interests, NCA provides the weekly summary of published vulnerabilities by the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) for the week from 25th of August to 31st of August. Vulnerabilities are scored using the Common Vulnerability Scoring System (CVSS) standard as per the following severity:

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

CVE ID & Source	Vendor - Product	Description	Publish Date	CVSS Score	Severity
CVE-2024-8193	Google	Heap buffer overflow in Skia in Google Chrome prior to 128.0.6613.113 allowed a remote attacker who had compromised the renderer process to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2024-08-28	8.8	High
CVE-2024-8194	Google	Type Confusion in V8 in Google Chrome prior to 128.0.6613.113 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2024-08-28	8.8	High
CVE-2024-8198	Google	Heap buffer overflow in Skia in Google Chrome prior to 128.0.6613.113 allowed a remote attacker who had compromised the renderer process to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2024-08-28	8.8	High
CVE-2024-20446	Cisco	A vulnerability in the DHCPv6 relay agent of Cisco NX-OS Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to improper handling of specific fields in a DHCPv6 RELAY-REPLY message. An attacker could exploit this vulnerability by sending a crafted DHCPv6 packet to any IPv6 address that is configured on an affected device. A successful exploit could allow the attacker to cause the dhcp_snoop process to crash and restart multiple times, causing the affected device to reload and resulting in a DoS condition.	2024-08-28	8.6	High
CVE-2024-5546	ManageEngine	Zohocorp ManageEngine Password Manager Pro versions before 12431 and ManageEngine PAM360 versions before 7001 are affected by authenticated SQL Injection vulnerability via a global search option.	2024-08-28	8.3	High
CVE-2024-6204	ManageEngine	Zohocorp ManageEngine Exchange Reporter Plus versions before 5715 are vulnerable to SQL Injection in the reports module.	2024-08-30	8.3	High
CVE-2024-38868	ManageEngine	Zohocorp ManageEngine Endpoint Central affected by Incorrect authorization vulnerability while isolating the devices.This issue affects Endpoint Central: before 11.3.2406.08 and before 11.3.2400.15	2024-08-30	8.3	High
CVE-2024-39584	Dell	Dell Client Platform BIOS contains a Use of Default Cryptographic Key Vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Secure Boot bypass and arbitrary code execution.	2024-08-28	8.2	High
CVE-2024-35133	IBM	IBM Security Verify Access 10.0.0 through 10.0.8 OIDC Provider could allow a remote attacker to conduct phishing attacks, using an open redirect attack. By persuading a victim to visit a specially crafted Web site, a remote attacker could exploit this vulnerability to spoof the URL displayed to redirect a user to a malicious Web site that would appear to be trusted. This could allow the attacker to obtain highly sensitive information or conduct further attacks against the victim.	2024-08-29	8.2	High
CVE-2024-39747	IBM	IBM Sterling Connect:Direct Web Services 6.0, 6.1, 6.2, and 6.3 uses default credentials for potentially critical functionality.	2024-08-31	8.1	High

CVE-2024-43888	Linux	In the Linux kernel, the following vulnerability has been resolved: mm: list_lru: fix UAF for memory cgroup The mem_cgroup_from_slab_obj() is supposed to be called under rcu lock or cgroup_mutex or others which could prevent returned memcg from being freed. Fix it by adding missing rcu read lock. Found by code inspection. [songmuchun@bytedance.com: only grab rcu lock when necessary, per Vlastimil] Link: https://lkml.kernel.org/r/20240801024603.1865-1-songmuchun@bytedance.com	2024-08-26	7.8	High
CVE-2024-43900	Linux	In the Linux kernel, the following vulnerability has been resolved: media: xc2028: avoid use-after-free in load_firmware_cb() syzkaller reported use-after-free in load_firmware_cb() [1]. The reason is because the module allocated a struct tuner in tuner_probe(), and then the module initialization failed, the struct tuner was released. A worker which created during module initialization accesses this struct tuner later, it caused use-after-free. The process is as follows: task-6504 worker_thread tuner_probe <= alloc dvb_frontend [2] ... request_firmware_nowait <= create a worker ... tuner_remove <= free dvb_frontend ... request_firmware_work_func <= the firmware is ready load_firmware_cb <= but now the dvb_frontend has been freed To fix the issue, check the dvd_frontend in load_firmware_cb(), if it is null, report a warning and just return. [1]: ========== BUG: KASAN: use-after-free in load_firmware_cb+0x1310/0x17a0 Read of size 8 at addr ffff8000d7ca2308 by task kworker/2:3/6504 Call trace: load_firmware_cb+0x1310/0x17a0 request_firmware_work_func+0x128/0x220 process_one_work+0x770/0x1824 worker_thread+0x488/0xea0 kthread+0x300/0x430 ret_from_fork+0x10/0x20 Allocated by task 6504: kzalloc tuner_probe+0xb0/0x1430 i2c_device_probe+0x92c/0xaf0 really_probe+0x678/0xcd0 driver_probe_device+0x280/0x370 __device_attach_driver+0x220/0x330 bus_for_each_drv+0x134/0x1c0 __device_attach+0x1f4/0x410 device_initial_probe+0x20/0x30 bus_probe_device+0x184/0x200 device_add+0x924/0x12c0 device_register+0x24/0x30 i2c_new_device+0x4e0/0xc44 v4l2_i2c_new_subdev_board+0xbc/0x290 v4l2_i2c_new_subdev+0xc8/0x104 em28xx_v4l2_init+0x1dd0/0x3770	2024-08-26	7.8	High

		Freed by task 6504: kfree+0x238/0x4e4 tuner_remove+0x144/0x1c0 i2c_device_remove+0xc8/0x290 __device_release_driver+0x314/0x5fc device_release_driver+0x30/0x44 bus_remove_device+0x244/0x490 device_del+0x350/0x900 device_unregister+0x28/0xd0 i2c_unregister_device+0x174/0x1d0 v4l2_device_unregister+0x224/0x380 em28xx_v4l2_init+0x1d90/0x3770 The buggy address belongs to the object at ffff8000d7ca2000 which belongs to the cache kmalloc-2k of size 2048 The buggy address is located 776 bytes inside of 2048-byte region [ffff8000d7ca2000, ffff8000d7ca2800) The buggy address belongs to the page: page:ffff7fe00035f280 count:1 mapcount:0 mapping:ffff8000c001f000 index:0x0 flags: 0x7ff800000000100(slab) raw: 07ff800000000100 ffff7fe00049d880 0000000300000003 ffff8000c001f000 raw: 0000000000000000 0000000080100010 00000001ffffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff8000d7ca2200: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ffff8000d7ca2280: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb >ffff8000d7ca2300: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff8000d7ca2380: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ffff8000d7ca2400: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ========== ========== [2] Actually, it is allocated for struct tuner, and dvb_frontend is inside.			
CVE-2024-44932	Linux	In the Linux kernel, the following vulnerability has been resolved: idpf: fix UAFs when destroying the queues The second tagged commit started sometimes (very rarely, but possible) throwing WARNs from net/core/page_pool.c:page_pool_disable_direct_recycling(). Turned out idpf frees interrupt vectors with embedded NAPIs *before* freeing the queues making page_pools' NAPI pointers lead to freed memory before these pools are destroyed by libeth. It's not clear whether there are other accesses to the freed vectors when destroying the queues, but anyway, we usually free queue/interrupt vectors only when the queues are destroyed and the NAPIs are guaranteed to not be referenced anywhere. Invert the allocation and freeing logic making queue/interrupt vectors be allocated first and freed last. Vectors don't require queues to be present, so this is safe. Additionally, this change allows to remove that useless queue->q_vector pointer cleanup, as vectors are still valid when freeing the queues (+ both are freed within one function, so it's not clear why nullify the pointers at all).	2024-08-26	7.8	High
CVE-2024-44934	Linux	In the Linux kernel, the following vulnerability has been resolved: net: bridge: mcast: wait for previous gc cycles when removing port syzbot hit a use-after-free[1] which is caused because the bridge doesn't make sure that all previous garbage has been collected when removing a port. What happens is: CPU 1 CPU 2	2024-08-26	7.8	High

		start gc cycle remove port acquire gc lock first wait for lock call br_multicasg_gc() directly acquire lock now but free port the port can be freed while grp timers still running Make sure all previous gc cycles have finished by using flush_work before freeing the port. [1] BUG: KASAN: slab-use-after-free in br_multicast_port_group_expired+0x4c0/0x550 net/bridge/br_multicast.c:861 Read of size 8 at addr ffff888071d6d000 by task syz.5.1232/9699 CPU: 1 PID: 9699 Comm: syz.5.1232 Not tainted 6.10.0-rc5-syzkaller-00021-g24ca36a562d6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 06/07/2024 Call Trace: <IRQ> dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:114 print_address_description mm/kasan/report.c:377 [inline] print_report+0xc3/0x620 mm/kasan/report.c:488 kasan_report+0xd9/0x110 mm/kasan/report.c:601 br_multicast_port_group_expired+0x4c0/0x550 net/bridge/br_multicast.c:861 call_timer_fn+0x1a3/0x610 kernel/time/timer.c:1792 expire_timers kernel/time/timer.c:1843 [inline] __run_timers+0x74b/0xaf0 kernel/time/timer.c:2417 __run_timer_base kernel/time/timer.c:2428 [inline] __run_timer_base kernel/time/timer.c:2421 [inline] run_timer_base+0x111/0x190 kernel/time/timer.c:2437			
CVE-2024-41879	Adobe	Acrobat Reader versions 127.0.2651.105 and earlier are affected by an out-of-bounds write vulnerability that could result in arbitrary code execution in the context of the current user. Exploitation of this issue requires user interaction in that a victim must open a malicious file.	2024-08-26	7.8	High
CVE-2024-44942	Linux	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on F2FS_INLINE_DATA flag in inode during GC syzbot reports a f2fs bug as below: ------------[cut here]------------ kernel BUG at fs/f2fs/inline.c:258! CPU: 1 PID: 34 Comm: kworker/u8:2 Not tainted 6.9.0-rc6-syzkaller-00012-g9e4bc4bcae01 #0 RIP: 0010:f2fs_write_inline_data+0x781/0x790 fs/f2fs/inline.c:258 Call Trace: f2fs_write_single_data_page+0xb65/0x1d60 fs/f2fs/data.c:2834 f2fs_write_cache_pages fs/f2fs/data.c:3133 [inline] __f2fs_write_data_pages fs/f2fs/data.c:3288 [inline] f2fs_write_data_pages+0x1efe/0x3a90 fs/f2fs/data.c:3315 do_writepages+0x35b/0x870 mm/page-writeback.c:2612 __writeback_single_inode+0x165/0x10b0 fs/fs-writeback.c:1650 writeback_sb_inodes+0x905/0x1260 fs/fs-writeback.c:1941 wb_writeback+0x457/0xce0 fs/fs-writeback.c:2117 wb_do_writeback fs/fs-writeback.c:2264 [inline] wb_workfn+0x410/0x1090 fs/fs-writeback.c:2304 process_one_work kernel/workqueue.c:3254 [inline] process_scheduled_works+0xa12/0x17c0 kernel/workqueue.c:3335 worker_thread+0x86d/0xd70 kernel/workqueue.c:3416 kthread+0x2f2/0x390 kernel/kthread.c:388 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 The root cause is: inline_data inode can be fuzzed, so that there may be valid blkaddr in its direct node, once f2fs triggers background GC to migrate the block, it will hit f2fs_bug_on() during dirty page writeback.	2024-08-26	7.8	High

		Let's add sanity check on F2FS_INLINE_DATA flag in inode during GC, so that, it can forbid migrating inline_data inode's data block for fixing.			
CVE-2024-8234	Zyxel	** UNSUPPORTED WHEN ASSIGNED ** A command injection vulnerability in the functions formSysCmd(), formUpgradeCert(), and formDelcert() in the Zyxel NWA1100-N firmware version 1.00(AACE.1)C0 could allow an unauthenticated attacker to execute some OS commands to access system files on an affected device.	2024-08-30	7.5	High
CVE-2023-43078	Dell	Dell Dock Firmware and Dell Client Platform contain an Improper Link Resolution vulnerability during installation resulting in arbitrary folder deletion, which could lead to Privilege Escalation or Denial of Service.	2024-08-28	6.7	Medium
CVE-2024-20411	Cisco	A vulnerability in Cisco NX-OS Software could allow an authenticated, local attacker with privileges to access the Bash shell to execute arbitrary code as root on an affected device. This vulnerability is due to insufficient security restrictions when executing commands from the Bash shell. An attacker with privileges to access the Bash shell could exploit this vulnerability by executing a specific crafted command on the underlying operating system. A successful exploit could allow the attacker to execute arbitrary code with the privileges of root.	2024-08-28	6.7	Medium
CVE-2024-20413	Cisco	A vulnerability in Cisco NX-OS Software could allow an authenticated, local attacker with privileges to access the Bash shell to elevate privileges to network-admin on an affected device. This vulnerability is due to insufficient security restrictions when executing application arguments from the Bash shell. An attacker with privileges to access the Bash shell could exploit this vulnerability by executing crafted commands on the underlying operating system. A successful exploit could allow the attacker to create new users with the privileges of network-admin.	2024-08-28	6.7	Medium
CVE-2024-39579	Dell	Dell PowerScale OneFS versions 8.2.2.x through 9.8.0.0 contains an incorrect privilege assignment vulnerability. A local high privileged attacker could potentially exploit this vulnerability to gain root-level access.	2024-08-31	6.7	Medium
CVE-2024-20478	Cisco	A vulnerability in the software upgrade component of Cisco Application Policy Infrastructure Controller (APIC) and Cisco Cloud Network Controller, formerly Cisco Cloud APIC, could allow an authenticated, remote attacker with Administrator-level privileges to install a modified software image, leading to arbitrary code injection on an affected system. This vulnerability is due to insufficient signature validation of software images. An attacker could exploit this vulnerability by installing a modified software image. A successful exploit could allow the attacker to execute arbitrary code on the affected system and elevate their privileges to root. Note: Administrators should always validate the hash of any upgrade image before uploading it to Cisco APIC and Cisco Cloud Network Controller.	2024-08-28	6.5	Medium
CVE-2024-39578	Dell	Dell PowerScale OneFS versions 8.2.2.x through 9.8.0.1 contains a UNIX symbolic link (symlink) following vulnerability. A local high privileged attacker could potentially exploit this vulnerability, leading to denial of service, information tampering.	2024-08-31	6.3	Medium
CVE-2024-7608	Trellix	An authenticated user can access the restricted files from NX, EX, FX, AX, IVX and CMS using path traversal.	2024-08-27	5.9	Medium
CVE-2024-43884	Linux	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Add error handling to pair_device() hci_conn_params_add() never checks for a NULL value and could lead to a NULL pointer dereference causing a crash. Fixed by adding error handling in the function.	2024-08-26	5.5	Medium
CVE-2024-43885	Linux	In the Linux kernel, the following vulnerability has been resolved:	2024-08-26	5.5	Medium

		btrfs: fix double inode unlock for direct IO sync writes If we do a direct IO sync write, at btrfs_sync_file(), and we need to skip inode logging or we get an error starting a transaction or an error when flushing delalloc, we end up unlocking the inode when we shouldn't under the 'out_release_extents' label, and then unlock it again at btrfs_direct_write(). Fix that by checking if we have to skip inode unlocking under that label.			
CVE-2024-43886	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check in resource_log_pipe_topology_update [WHY] When switching from "Extend" to "Second Display Only" we sometimes call resource_get_otg_master_for_stream on a stream for the eDP, which is disconnected. This leads to a null pointer dereference. [HOW] Added a null check in dc_resource.c/resource_log_pipe_topology_update.	2024-08-26	5.5	Medium
CVE-2024-43889	Linux	In the Linux kernel, the following vulnerability has been resolved: padata: Fix possible divide-by-0 panic in padata_mt_helper() We are hit with a not easily reproducible divide-by-0 panic in padata.c at bootup time. [10.017908] Oops: divide error: 0000 1 PREEMPT SMP NOPTI [10.017908] CPU: 26 PID: 2627 Comm: kworker/u1666:1 Not tainted 6.10.0-15.el10.x86_64 #1 [10.017908] Hardware name: Lenovo ThinkSystem SR950 [7X12CTO1WW]/[7X12CTO1WW], BIOS [PSE140J-2.30] 07/20/2021 [10.017908] Workqueue: events_unbound padata_mt_helper [10.017908] RIP: 0010:padata_mt_helper+0x39/0xb0 : [10.017963] Call Trace: [10.017968] <TASK> [10.018004] ? padata_mt_helper+0x39/0xb0 [10.018084] process_one_work+0x174/0x330 [10.018093] worker_thread+0x266/0x3a0 [10.018111] kthread+0xcf/0x100 [10.018124] ret_from_fork+0x31/0x50 [10.018138] ret_from_fork_asm+0x1a/0x30 [10.018147] </TASK> Looking at the padata_mt_helper() function, the only way a divide-by-0 panic can happen is when ps->chunk_size is 0. The way that chunk_size is initialized in padata_do_multithreaded(), chunk_size can be 0 when the min_chunk in the passed-in padata_mt_job structure is 0. Fix this divide-by-0 panic by making sure that chunk_size will be at least 1 no matter what the input parameters are.	2024-08-26	5.5	Medium
CVE-2024-43890	Linux	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix overflow in get_free_elt() "tracing_map->next_elt" in get_free_elt() is at risk of overflowing. Once it overflows, new elements can still be inserted into the tracing_map even though the maximum number of elements (`max_elts`) has been reached. Continuing to insert elements after the overflow could result in the tracing_map containing "tracing_map->max_size" elements, leaving no empty entries.	2024-08-26	5.5	Medium

		If any attempt is made to insert an element into a full tracing_map using `__tracing_map_insert()`, it will cause an infinite loop with preemption disabled, leading to a CPU hang problem. Fix this by preventing any further increments to "tracing_map->next_elt" once it reaches "tracing_map->max_elt".			
CVE-2024-43896	Linux	In the Linux kernel, the following vulnerability has been resolved: ASoC: cs-amp-lib: Fix NULL pointer crash if efi.get_variable is NULL Call efi_rt_services_supported() to check that efi.get_variable exists before calling it.	2024-08-26	5.5	Medium

CVE-2024-43897	Linux	In the Linux kernel, the following vulnerability has been resolved:
		
		net: drop bad gso csum_start and offset in virtio_net_hdr
		
		Tighten csum_start and csum_offset checks in
		virtio_net_hdr_to_skb
		for GSO packets.
		
		The function already checks that a checksum requested with
		VIRTIO_NET_HDR_F_NEEDS_CSUM is in skb linear. But for GSO packets
		this might not hold for segs after segmentation.
		
		Syzkaller demonstrated to reach this warning in
		skb_checksum_help
		
		offset = skb_checksum_start_offset(skb);
		ret = -EINVAL;
		if (WARN_ON_ONCE(offset >= skb_headlen(skb)))
		
		By injecting a TSO packet:
		
		WARNING: CPU: 1 PID: 3539 at net/core/dev.c:3284
		skb_checksum_help+0x3d0/0x5b0
		ip_do_fragment+0x209/0x1b20 net/ipv4/ip_output.c:774
		ip_finish_output_gso net/ipv4/ip_output.c:279 [inline]
		__ip_finish_output+0x2bd/0x4b0 net/ipv4/ip_output.c:301
		iptunnel_xmit+0x50c/0x930 net/ipv4/ip_tunnel_core.c:82
		ip_tunnel_xmit+0x2296/0x2c70 net/ipv4/ip_tunnel.c:813
		__gre_xmit net/ipv4/ip_gre.c:469 [inline]
		ipgre_xmit+0x759/0xa60 net/ipv4/ip_gre.c:661
		__netdev_start_xmit include/linux/netdevice.h:4850 [inline]
		netdev_start_xmit include/linux/netdevice.h:4864 [inline]
		xmit_one net/core/dev.c:3595 [inline]
		dev_hard_start_xmit+0x261/0x8c0 net/core/dev.c:3611
		__dev_queue_xmit+0x1b97/0x3c90 net/core/dev.c:4261
		packet_snd net/packet/af_packet.c:3073 [inline]
		
		The geometry of the bad input packet at tcp_gso_segment:
		
		[52.003050][T8403] skb len=12202 headroom=244
		headlen=12093 tailroom=0
		[52.003050][T8403] mac=(168,24) mac_len=24 net=(192,52)
		trans=244
		[52.003050][T8403] shinfo(txflags=0 nr_frags=1 gso(size=1552
		type=3 segs=0))
		[52.003050][T8403] csum(0x60000c7 start=199 offset=1536
		ip_summed=3 complete_sw=0 valid=0 level=0)
		
		Mitigate with stricter input validation.
		
		csum_offset: for GSO packets, deduce the correct value from
		gso_type.
		This is already done for USO. Extend it to TSO. Let UFO be:
		udp[46]_ufo_fragment ignores these fields and always computes the
		checksum in software.
		
		csum_start: finding the real offset requires parsing to the transport
		header. Do not add a parser, use existing segmentation parsing. Thanks
		to SKB_GSO_DODGY, that also catches bad packets that are hw offloaded.
			2024-08-26	5.5	Medium

		Again test both TSO and USO. Do not test UFO for the above reason, and do not test UDP tunnel offload. GSO packet are almost always CHECKSUM_PARTIAL. USO packets may be CHECKSUM_NONE since commit 10154dbded6d6 ("udp: Allow GSO transmit from devices with no checksum offload"), but then still these fields are initialized correctly in udp4_hwcsum/udp6_hwcsum_outgoing. So no need to test for ip_summed == CHECKSUM_PARTIAL first. This revises an existing fix mentioned in the Fixes tag, which broke small packets with GSO offload, as detected by kselftests.			
CVE-2024-43898	Linux	In the Linux kernel, the following vulnerability has been resolved: ext4: sanity check for NULL pointer after ext4_force_shutdown Test case: 2 threads write short inline data to a file. In ext4_page_mkwrite the resulting inline data is converted. Handling ext4_grp_locked_error with description "block bitmap and bg descriptor inconsistent: X vs Y free clusters" calls ext4_force_shutdown. The conversion clears EXT4_STATE_MAY_INLINE_DATA but fails for ext4_destroy_inline_data_nolock and ext4_mark_iloc_dirty due to ext4_forced_shutdown. The restoration of inline data fails for the same reason not setting EXT4_STATE_MAY_INLINE_DATA. Without the flag set a regular process path in ext4_da_write_end follows trying to dereference page folio private pointer that has not been set. The fix calls early return with -EIO error shall the pointer to private be NULL. Sample crash report: Unable to handle kernel paging request at virtual address dfff800000000004 KASAN: null-ptr-deref in range [0x0000000000000020-0x0000000000000027] Mem abort info: ESR = 0x0000000096000005 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault Data abort info: ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [dfff800000000004] address between user and kernel address ranges Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP Modules linked in: CPU: 1 PID: 20274 Comm: syz-executor185 Not tainted 6.9.0-rc7-syzkaller-gfda5695d692c #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/27/2024 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __block_commit_write+0x64/0x2b0 fs/buffer.c:2167 lr : __block_commit_write+0x3c/0x2b0 fs/buffer.c:2160 sp : ffff8000a1957600 x29: ffff8000a1957610 x28: dfff800000000000 x27: ffff0000e30e34b0 x26: 0000000000000000 x25: dfff800000000000 x24: dfff800000000000 x23: fffffdffc397c9e0 x22: 0000000000000020 x21: 0000000000000020 x20: 0000000000000040 x19: fffffdffc397c9c0 x18: 1fffe000367bd196 x17: ffff80008eead000 x16: ffff80008ae89e3c x15: 00000000200000c0 x14: 1fffe0001cbe4e04 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000001 x10: 0000000000ff0100 x9 : 0000000000000000 x8 : 0000000000000004 x7 : 0000000000000000 x6 : 0000000000000000 x5 : fffffdffc397c9c0 x4 : 0000000000000020 x3 : 0000000000000020 x2 : 0000000000000040 x1 : 0000000000000020 x0 :	2024-08-26	5.5	Medium

		ffffffdffc397c9c0 Call trace: __block_commit_write+0x64/0x2b0 fs/buffer.c:2167 block_write_end+0xb4/0x104 fs/buffer.c:2253 ext4_da_do_write_end fs/ext4/inode.c:2955 [inline] ext4_da_write_end+0x2c4/0xa40 fs/ext4/inode.c:3028 generic_perform_write+0x394/0x588 mm/filemap.c:3985 ext4_buffered_write_iter+0x2c0/0x4ec fs/ext4/file.c:299 ext4_file_write_iter+0x188/0x1780 call_write_iter include/linux/fs.h:2110 [inline] new_sync_write fs/read_write.c:497 [inline] vfs_write+0x968/0xc3c fs/read_write.c:590 ksys_write+0x15c/0x26c fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __arm64_sys_write+0x7c/0x90 fs/read_write.c:652 __invoke_syscall arch/arm64/kernel/syscall.c:34 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:48 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:133 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:152 el0_svc+0x54/0x168 arch/arm64/kernel/entry-common.c:712 el0t_64_sync_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:598 Code: 97f85911 f94002da 91008356 d343fec8 (38796908) ---[end trace 0000000000000000]--- ---------- Code disassembly (best guess): 0: 97f85911 bl 0xffffffffffe16444 4: f94002da ldr x26, [x22] 8: 91008356 add x22, x26, #0x20 c: d343fec8 lsr x8, x22, #3 * 10: 38796908 ldrb w8, [x8, x25] <-- trapping instruction			
		In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null pointer deref in dcn20_resource.c Fixes a hang thats triggered when MPV is run on a DCN401 dGPU: mpv --hwdec=vaapi --vo=gpu --hwdec-codecs=all and then enabling fullscreen playback (double click on the video) The following calltrace will be seen: [181.843989] BUG: kernel NULL pointer dereference, address: 0000000000000000 [181.843997] #PF: supervisor instruction fetch in kernel mode [181.844003] #PF: error_code(0x0010) - not-present page [181.844009] PGD 0 P4D 0 [181.844020] Oops: 0010 [#1] PREEMPT SMP NOPTI [181.844028] CPU: 6 PID: 1892 Comm: gnome-shell Tainted: G W OE 6.5.0-41-generic #41~22.04.2-Ubuntu [181.844038] Hardware name: System manufacturer System Product Name/CROSSHAIR VI HERO, BIOS 6302 10/23/2018 [181.844044] RIP: 0010:0x0 [181.844079] Code: Unable to access opcode bytes at 0xffffffffffffffd6. [181.844084] RSP: 0018:ffffb593c2b8f7b0 EFLAGS: 00010246 [181.844093] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000004 [181.844099] RDX: ffffb593c2b8f804 RSI: ffffb593c2b8f7e0 RDI: ffff9e3c8e758400 [181.844105] RBP: ffffb593c2b8f7b8 R08: ffffb593c2b8f9c8 R09: ffffb593c2b8f96c [181.844110] R10: 0000000000000000 R11: 0000000000000000 R12: ffffb593c2b8f9c8 [181.844115] R13: 0000000000000001 R14: ffff9e3c88000000 R15: 0000000000000005 [181.844121] FS: 00007c6e323bb5c0(0000) GS:ffff9e3f85f80000(0000) knlGS:0000000000000000 [181.844128] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [181.844134] CR2: ffffffffffffffd6 CR3: 0000000140fbe000 CR4: 00000000003506e0 [181.844141] Call Trace: [181.844146] <TASK> [181.844153] ? show_regs+0x6d/0x80 [181.844167] ? __die+0x24/0x80 [181.844179] ? page_fault_oops+0x99/0x1b0			
CVE-2024-43899	Linux		2024-08-26	5.5	Medium

		[181.844192] ? do_user_addr_fault+0x31d/0x6b0 [181.844204] ? exc_page_fault+0x83/0x1b0 [181.844216] ? asm_exc_page_fault+0x27/0x30 [181.844237] dcn20_get_dcc_compression_cap+0x23/0x30 [amdgpu] [181.845115] amdgpu_dm_plane_validate_dcc.constprop.0+0xe5/0x180 [amdgpu] [181.845985] amdgpu_dm_plane_fill_plane_buffer_attributes+0x300/0x580 [amdgpu] [181.846848] fill_dc_plane_info_and_addr+0x258/0x350 [amdgpu] [181.847734] fill_dc_plane_attributes+0x162/0x350 [amdgpu] [181.848748] dm_update_plane_state.constprop.0+0x4e3/0x6b0 [amdgpu] [181.849791] ? dm_update_plane_state.constprop.0+0x4e3/0x6b0 [amdgpu] [181.850840] amdgpu_dm_atomic_check+0xdfe/0x1760 [amdgpu]			
CVE-2024-43901	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix NULL pointer dereference for DTN log in DCN401 When users run the command: cat /sys/kernel/debug/dri/0/amdgpu_dm_dtn_log The following NULL pointer dereference happens: [+0.000003] BUG: kernel NULL pointer dereference, address: NULL [+0.000005] #PF: supervisor instruction fetch in kernel mode [+0.000002] #PF: error_code(0x0010) - not-present page [+0.000002] PGD 0 P4D 0 [+0.000004] Oops: 0010 [#1] PREEMPT SMP NOPTI [+0.000003] RIP: 0010:0x0 [+0.000008] Code: Unable to access opcode bytes at 0xffffffffffffffd6. [...] [+0.000002] PKRU: 55555554 [+0.000002] Call Trace: [+0.000002] <TASK> [+0.000003] ? show_regs+0x65/0x70 [+0.000006] ? __die+0x24/0x70 [+0.000004] ? page_fault_oops+0x160/0x470 [+0.000006] ? do_user_addr_fault+0x2b5/0x690 [+0.000003] ? prb_read_valid+0x1c/0x30 [+0.000005] ? exc_page_fault+0x8c/0x1a0 [+0.000005] ? asm_exc_page_fault+0x27/0x30 [+0.000012] dcn10_log_color_state+0xf9/0x510 [amdgpu] [+0.000306] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000003] ? vsnprintf+0x2fb/0x600 [+0.000009] dcn10_log_hw_state+0xfd0/0xfe0 [amdgpu] [+0.000218] ? __mod_memcg_lruvec_state+0xe8/0x170 [+0.000008] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000002] ? debug_smp_processor_id+0x17/0x20 [+0.000003] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000002] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000002] ? set_ptes.isra.0+0x2b/0x90 [+0.000004] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000002] ? _raw_spin_unlock+0x19/0x40 [+0.000004] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000002] ? do_anonymous_page+0x337/0x700 [+0.000004] dtn_log_read+0x82/0x120 [amdgpu] [+0.000207] full_proxy_read+0x66/0x90 [+0.000007] vfs_read+0xb0/0x340 [+0.000005] ? __count_memcg_events+0x79/0xe0 [+0.000002] ? srso_alias_return_thunk+0x5/0xfbef5 [+0.000003] ? count_memcg_events.constprop.0+0x1e/0x40 [+0.000003] ? handle_mm_fault+0xb2/0x370 [+0.000003] ksys_read+0x6b/0xf0 [+0.000004] __x64_sys_read+0x19/0x20 [+0.000003] do_syscall_64+0x60/0x130 [+0.000004] entry_SYSCALL_64_after_hwframe+0x6e/0x76 [+0.000003] RIP: 0033:0x7fdf32f147e2 [...]	This error happens when the color log tries to read the gamut	2024-08-26	5.5	Medium

		remap information from DCN401 which is not initialized in the dcn401_dpp_funcs which leads to a null pointer dereference. This commit addresses this issue by adding a proper guard to access the gamut_remap callback in case the specific ASIC did not implement this function.			
CVE-2024-43902	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null checker before passing variables Checks null pointer before passing variables to functions. This fixes 3 NULL_RETURNS issues reported by Coverity.	2024-08-26	5.5	Medium
CVE-2024-43903	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for 'afb' before dereferencing in amdgpu_dm_plane_handle_cursor_update This commit adds a null check for the 'afb' variable in the amdgpu_dm_plane_handle_cursor_update function. Previously, 'afb' was assumed to be null, but was used later in the code without a null check. This could potentially lead to a null pointer dereference. Fixes the below: drivers/gpu/drm/amd/amdgpu/../display/amdgpu_dm/amdgpu_dm_plane.c:1298 amdgpu_dm_plane_handle_cursor_update() error: we previously assumed 'afb' could be null (see line 1252)	2024-08-26	5.5	Medium
CVE-2024-43904	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null checks for 'stream' and 'plane' before dereferencing This commit adds null checks for the 'stream' and 'plane' variables in the dcn30_apply_idle_power_optimizations function. These variables were previously assumed to be null at line 922, but they were used later in the code without checking if they were null. This could potentially lead to a null pointer dereference, which would cause a crash. The null checks ensure that 'stream' and 'plane' are not null before they are used, preventing potential crashes. Fixes the below static smatch checker: drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn30/dcn30_hwseq.c:938 dcn30_apply_idle_power_optimizations() error: we previously assumed 'stream' could be null (see line 922) drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn30/dcn30_hwseq.c:940 dcn30_apply_idle_power_optimizations() error: we previously assumed 'plane' could be null (see line 922)	2024-08-26	5.5	Medium
CVE-2024-43905	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Fix the null pointer dereference for vega10_hwmgr Check return value and conduct null pointer handling to avoid null pointer dereference.	2024-08-26	5.5	Medium
CVE-2024-43906	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/admgpu: fix dereferencing null pointer context When user space sets an invalid ta type, the pointer context will be empty. So it need to check the pointer context before using it	2024-08-26	5.5	Medium
CVE-2024-43907	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm: Fix the null pointer dereference in apply_state_adjust_rules Check the pointer value to fix potential null pointer dereference	2024-08-26	5.5	Medium
CVE-2024-43908	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix the null pointer dereference to ras_manager	2024-08-26	5.5	Medium

		Check ras_manager before using it			
CVE-2024-43909	Linux	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm: Fix the null pointer dereference for smu7 optimize the code to avoid pass a null pointer (hwmgr->backend) to function smu7_update_edc_leakage_table.	2024-08-26	5.5	Medium
CVE-2024-43910	Linux	In the Linux kernel, the following vulnerability has been resolved: bpf: add missing check_func_arg_reg_off() to prevent out-of-bounds memory accesses Currently, it's possible to pass in a modified CONST_PTR_TO_DYNPTR to a global function as an argument. The adverse effects of this is that BPF helpers can continue to make use of this modified CONST_PTR_TO_DYNPTR from within the context of the global function, which can unintentionally result in out-of-bounds memory accesses and therefore compromise overall system stability i.e. [244.157771] BUG: KASAN: slab-out-of-bounds in bpf_dynptr_data+0x137/0x140 [244.161345] Read of size 8 at addr ffff88810914be68 by task test_progs/302 [244.167151] CPU: 0 PID: 302 Comm: test_progs Tainted: G O E 6.10.0-rc3-00131-g66b586715063 #533 [244.174318] Call Trace: [244.175787] <TASK> [244.177356] dump_stack_lvl+0x66/0xa0 [244.179531] print_report+0xce/0x670 [244.182314] ? __virt_addr_valid+0x200/0x3e0 [244.184908] kasan_report+0xd7/0x110 [244.187408] ? bpf_dynptr_data+0x137/0x140 [244.189714] ? bpf_dynptr_data+0x137/0x140 [244.192020] bpf_dynptr_data+0x137/0x140 [244.194264] bpf_prog_b02a02fdd2bdc5fa_global_call_bpf_dynptr_data+0x22/0x26 [244.198044] bpf_prog_b0fe7b9d7dc3abde_callback_adjust_bpf_dynptr_reg_off+0x1f/0x23 [244.202136] bpf_user_ringbuf_drain+0x2c7/0x570 [244.204744] ? 0xffffffffc0009e58 [244.206593] ? __pfx_bpf_user_ringbuf_drain+0x10/0x10 [244.209795] bpf_prog_33ab33f6a804ba2d_user_ringbuf_callback_const_ptr_to_dynptr_reg_off+0x47/0x4b [244.215922] bpf_trampoline_6442502480+0x43/0xe3 [244.218691] __x64_sys_prlimit64+0x9/0xf0 [244.220912] do_syscall_64+0xc1/0x1d0 [244.223043] entry_SYSCALL_64_after_hwframe+0x77/0x7f [244.226458] RIP: 0033:0x7ffa3eb8f059 [244.228582] Code: 08 89 e8 5b 5d c3 66 2e 0f 1f 84 00 00 00 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 8f 1d 0d 00 f7 d8 64 89 01 48 [244.241307] RSP: 002b:00007ffa3e9c6eb8 EFLAGS: 00000206 ORIG_RAX: 000000000000012e [244.246474] RAX: ffffffffffffffda RBX: 00007ffa3e9c7cdc RCX: 00007ffa3eb8f059 [244.250478] RDX: 00007ffa3eb162b4 RSI: 0000000000000000 RDI: 00007ffa3e9c7fb0 [244.255396] RBP: 00007ffa3e9c6ed0 R08: 00007ffa3e9c76c0 R09: 0000000000000000 [244.260195] R10: 0000000000000000 R11: 0000000000000206 R12: ffffffffffffff80 [244.264201] R13: 000000000000001c R14: 00007ffc5d6b4260 R15: 00007ffa3e1c7000 [244.268303] </TASK> Add a check_func_arg_reg_off() to the path in which the BPF verifier verifies the arguments of global function arguments, specifically those which take an argument of type ARG_PTR_TO_DYNPTR | MEM_RDONLY. Also, process_dynptr_func() doesn't appear to perform any explicit and strict type matching on the supplied register type, so	2024-08-26	5.5	Medium

		let's also enforce that a register either type PTR_TO_STACK or CONST_PTR_TO_DYNPTR is by the caller.			
		In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix NULL dereference at band check in starting tx ba session In MLD connection, link_data/link_conf are dynamically allocated. They don't point to vif->bss_conf. So, there will be no chanreq assigned to vif->bss_conf and then the chan will be NULL. Tweak the code to check ht_supported/vht_supported/has_he/has_eht on sta deflink. Crash log (with rtw89 version under MLO development): [9890.526087] BUG: kernel NULL pointer dereference, address: 0000000000000000 [9890.526102] #PF: supervisor read access in kernel mode [9890.526105] #PF: error_code(0x0000) - not-present page [9890.526109] PGD 0 P4D 0 [9890.526114] Oops: 0000 [#1] PREEMPT SMP PTI [9890.526119] CPU: 2 PID: 6367 Comm: kworker/u16:2 Kdump: loaded Tainted: G OE 6.9.0 #1 [9890.526123] Hardware name: LENOVO 2356AD1/2356AD1, BIOS G7ETB3WW (2.73) 11/28/2018 [9890.526126] Workqueue: phy2 rtw89_core_ba_work [rtw89_core] [9890.526203] RIP: 0010:ieee80211_start_tx_ba_session (net/mac80211/agg-tx.c:618 (discriminator 1)) mac80211 [9890.526279] Code: f7 e8 d5 93 3e ea 48 83 c4 28 89 d8 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc 49 8b 84 24 e0 f1 ff ff 48 8b 80 90 1b 00 00 <83> 38 03 0f 84 37 fe ff ff bb ea ff ff ff eb cc 49 8b 84 24 10 f3 All code ======== 0: f7 e8 imul %eax 2: d5 (bad) 3: 93 xchg %eax,%ebx 4: 3e ea ds (bad) 6: 48 83 c4 28 add $0x28,%rsp a: 89 d8 mov %ebx,%eax c: 5b pop %rbx d: 41 5c pop %r12 f: 41 5d pop %r13 11: 41 5e pop %r14 13: 41 5f pop %r15 15: 5d pop %rbp 16: c3 retq 17: cc int3 18: cc int3 19: cc int3 1a: cc int3 1b: 49 8b 84 24 e0 f1 ff mov -0xe20(%r12),%rax 22: ff 23: 48 8b 80 90 1b 00 00 mov 0x1b90(%rax),%rax 2a:* 83 38 03 cmpl $0x3,(%rax) <-- trapping instruction 2d: 0f 84 37 fe ff ff je 0xffffffffffffe6a 33: bb ea ff ff ff mov $0xffffffea,%ebx 38: eb cc jmp 0x6 3a: 49 rex.WB 3b: 8b .byte 0x8b 3c: 84 24 10 test %ah,(%rax,%rdx,1) 3f: f3 repz Code starting with the faulting instruction ========== 0: 83 38 03 cmpl $0x3,(%rax) 3: 0f 84 37 fe ff ff je 0xffffffffffffe40 9: bb ea ff ff ff mov $0xffffffea,%ebx e: eb cc jmp 0xffffffffffffffdc 10: 49 rex.WB 11: 8b .byte 0x8b 12: 84 24 10 test %ah,(%rax,%rdx,1) 15: f3 repz [9890.526285] RSP: 0018:ffffb8db09013d68 EFLAGS: 00010246 [9890.526291] RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffff9308e0d656c8 [9890.526295] RDX: 0000000000000000 RSI: ffffffffab99460b RDI: ffffffffab9a7685			
CVE-2024-43911	Linux		2024-08-26	5.5	Medium

		[9890.526300] RBP: ffffb8db09013db8 R08: 0000000000000000 R09: 0000000000000873 [9890.526304] R10: ffff9308e0d64800 R11: 0000000000000002 R12: ffff9308e5ff6e70 [9890.526308] R13: ffff930952500e20 R14: ffff9309192a8c00 R15: 0000000000000000 [9890.526313] FS: 0000000000000000(0000) GS:ffff930b4e700000(0000) knlGS:0000000000000000 [9890.526316] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [9890.526318] CR2: 0000000000000000 CR3: 0000000391c58005 CR4: 00000000001706f0 [9890.526321] Call Trace: [9890.526324] <TASK> [9890.526327] ? show_regs (arch/x86/kernel/dumpstack.c:479) [9890.526335] ? __die (arch/x86/kernel/dumpstack.c:421 arch/x86/kernel/dumpstack.c:434) [9890.526340] ? page_fault_oops (arch/x86/mm/fault.c:713) [9890.526347] ? search_module_extables (kernel/module/main.c:3256 (discriminator ---truncated---			
CVE-2024-43912	Linux	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: disallow setting special AP channel widths Setting the AP channel width is meant for use with the normal 20/40/... MHz channel width progression, and switching around in S1G or narrow channels isn't supported. Disallow that.	2024-08-26	5.5	Medium
CVE-2024-43913	Linux	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.	2024-08-26	5.5	Medium
CVE-2024-43914	Linux	In the Linux kernel, the following vulnerability has been resolved: md/raid5: avoid BUG_ON() while continue reshape after reassembling Currently, mdadm support --revert-reshape to abort the reshape while reassembling, as the test 07revert-grow. However, following BUG_ON() can be triggerred by the test: kernel BUG at drivers/md/raid5.c:6278! invalid opcode: 0000 [#1] PREEMPT SMP PTI irq event stamp: 158985 CPU: 6 PID: 891 Comm: md0_reshape Not tainted 6.9.0-03335-g7592a0b0049a #94 RIP: 0010:reshape_request+0x3f1/0xe60 Call Trace: <TASK> raid5_sync_request+0x43d/0x550 md_do_sync+0xb7a/0x2110 md_thread+0x294/0x2b0 kthread+0x147/0x1c0 ret_from_fork+0x59/0x70 ret_from_fork_asm+0x1a/0x30 </TASK> Root cause is that --revert-reshape update the raid_disks from 5 to 4, while reshape position is still set, and after reassembling the array, reshape position will be read from super block, then during reshape the checking of 'writepos' that is caculated by old reshape position will fail. Fix this panic the easy way first, by converting the BUG_ON() to WARN_ON(), and stop the reshape if checkings fail. Noted that mdadm must fix --revert-shape as well, and probably md/raid should enhance metadata validation as well, however this means	2024-08-26	5.5	Medium

		reassemble will fail and there must be user tools to fix the wrong metadata.			
CVE-2024-44931	Linux	In the Linux kernel, the following vulnerability has been resolved: gpio: prevent potential speculation leaks in gpio_device_get_desc() Userspace may trigger a speculative read of an address outside the gpio descriptor array. Users can do that by calling gpio_ioctl() with an offset out of range. Offset is copied from user and then used as an array index to get the gpio descriptor without sanitization in gpio_device_get_desc(). This change ensures that the offset is sanitized by using array_index_nospec() to mitigate any possibility of speculative information leaks. This bug was discovered and resolved using Coverity Static Analysis Security Testing (SAST) by Synopsys, Inc.	2024-08-26	5.5	Medium
CVE-2024-44933	Linux	In the Linux kernel, the following vulnerability has been resolved: bnxt_en : Fix memory out-of-bounds in bnxt_fill_hw_rss_tbl() A recent commit has modified the code in __bnxt_reserve_rings() to set the default RSS indirection table to default only when the number of RX rings is changing. While this works for newer firmware that requires RX ring reservations, it causes the regression on older firmware not requiring RX ring resrvations (BNXT_NEW_RM() returns false). With older firmware, RX ring reservations are not required and so hw_resc->resv_rx_rings is not always set to the proper value. The comparison: if (old_rx_rings != bp->hw_resc.resv_rx_rings) in __bnxt_reserve_rings() may be false even when the RX rings are changing. This will cause __bnxt_reserve_rings() to skip setting the default RSS indirection table to default to match the current number of RX rings. This may later cause bnxt_fill_hw_rss_tbl() to use an out-of-range index. We already have bnxt_check_rss_tbl_no_rmgr() to handle exactly this scenario. We just need to move it up in bnxt_need_reserve_rings() to be called unconditionally when using older firmware. Without the fix, if the TX rings are changing, we'll skip the bnxt_check_rss_tbl_no_rmgr() call and __bnxt_reserve_rings() may also skip the bnxt_set_dflt_rss_indir_tbl() call for the reason explained in the last paragraph. Without setting the default RSS indirection table to default, it causes the regression: BUG: KASAN: slab-out-of-bounds in __bnxt_hwrm_vnic_set_rss+0xb79/0xe40 Read of size 2 at addr ffff8881c5809618 by task ethtool/31525 Call Trace: __bnxt_hwrm_vnic_set_rss+0xb79/0xe40 bnxt_hwrm_vnic_rss_cfg_p5+0xf7/0x460 __bnxt_setup_vnic_p5+0x12e/0x270 __bnxt_open_nic+0x2262/0x2f30 bnxt_open_nic+0x5d/0xf0 ethnl_set_channels+0x5d4/0xb30 ethnl_default_set_doit+0x2f1/0x620	2024-08-26	5.5	Medium
CVE-2024-44935	Linux	In the Linux kernel, the following vulnerability has been resolved: sctp: Fix null-ptr-deref in reuseport_add_sock(). syzbot reported a null-ptr-deref while accessing sk2->sk_reuseport_cb in reuseport_add_sock(). [0] The repro first creates a listener with SO_REUSEPORT. Then, it	2024-08-26	5.5	Medium

		creates another listener on the same port and concurrently closes the first listener. The second listen() calls reuseport_add_sock() with the first listener as sk2, where sk2->sk_reuseport_cb is not expected to be cleared concurrently, but the close() does clear it by reuseport_detach_sock(). The problem is SCTP does not properly synchronise reuseport_alloc(), reuseport_add_sock(), and reuseport_detach_sock(). The caller of reuseport_alloc() and reuseport_{add,detach}_sock() must provide synchronisation for sockets that are classified into the same reuseport group. Otherwise, such sockets form multiple identical reuseport groups, and all groups except one would be silently dead. 1. Two sockets call listen() concurrently 2. No socket in the same group found in sctp_ep_hashtable[] 3. Two sockets call reuseport_alloc() and form two reuseport groups 4. Only one group hit first in __sctp_rcv_lookup_endpoint() receives incoming packets Also, the reported null-ptr-deref could occur. TCP/UDP guarantees that would not happen by holding the hash bucket lock. Let's apply the locking strategy to __sctp_hash_endpoint() and __sctp_unhash_endpoint(). [0]: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000002: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] CPU: 1 UID: 0 PID: 10230 Comm: syz-executor119 Not tainted 6.10.0-syzkaller-12585-g301927d2d2eb #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 06/27/2024 RIP: 0010:reuseport_add_sock+0x27e/0x5e0 net/core/sock_reuseport.c:350 Code: 00 0f b7 5d 00 bf 01 00 00 00 89 de e8 1b a4 ff f7 83 fb 01 0f 85 a3 01 00 00 e8 6d a0 ff f7 49 8d 7e 12 48 89 f8 48 c1 e8 03 <42> 0f b6 04 28 84 c0 0f 85 4b 02 00 00 41 0f b7 5e 12 49 8d 7e 14 RSP: 0018:ffffc9000b947c98 EFLAGS: 00010202 RAX: 0000000000000002 RBX: ffff8880252ddf98 RCX: ffff888079478000 RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000012 RBP: 0000000000000001 R08: ffffffff8993e18d R09: 1fffffff1fef385 R10: dffffc0000000000 R11: fffffbfff1fef386 R12: ffff8880252ddac0 R13: dffffc0000000000 R14: 0000000000000000 R15: 0000000000000000 FS: 00007f24e45b96c0(0000) GS:ffff8880b9300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffcced5f7b8 CR3: 00000000241be000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __sctp_hash_endpoint net/sctp/input.c:762 [inline] sctp_hash_endpoint+0x52a/0x600 net/sctp/input.c:790 sctp_listen_start net/sctp/socket.c:8570 [inline] sctp_inet_listen+0x767/0xa20 net/sctp/socket.c:8625 __sys_listen_socket net/socket.c:1883 [inline]			

		__sys_listen+0x1b7/0x230 net/socket.c:1894 __do_sys_listen net/socket.c:1902 [inline] __se_sys_listen net/socket.c:1900 [inline] __x64_sys_listen+0x5a/0x70 net/socket.c:1900 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f24e46039b9 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 91 1a 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f24e45b9228 EFLAGS: 00000246 ORIG_RAX: 0000000000000032 RAX: ffffffffffffffda RBX: 00007f24e468e428 RCX: 00007f24e46039b9 RDX: 00007f24e46039b9 RSI: 0000000000000003 RDI: 0000000000000004 RBP: 00007f24e468e420 R08: 00007f24e45b96c0 R09: 00007f24e45b96c0 R10: 00007f24e45b96c0 R11: 0000000000000246 R12: 00007f24e468e42c R13: ---truncated---			
CVE-2024-44936	Linux	In the Linux kernel, the following vulnerability has been resolved: power: supply: rt5033: Bring back i2c_set_clientdata Commit 3a93da231c12 ("power: supply: rt5033: Use devm_power_supply_register() helper") reworked the driver to use devm. While at it, the i2c_set_clientdata was dropped along with the remove callback. Unfortunately other parts of the driver also rely on i2c clientdata so this causes kernel oops. Bring the call back to fix the driver.	2024-08-26	5.5	Medium
CVE-2024-44937	Linux	In the Linux kernel, the following vulnerability has been resolved: platform/x86: intel-vbtn: Protect ACPI notify handler against recursion Since commit e2ffcda16290 ("ACPI: OSL: Allow Notify () handlers to run on all CPUs") ACPI notify handlers like the intel-vbtn notify_handler() may run on multiple CPU cores racing with themselves. This race gets hit on Dell Venue 7140 tablets when undocking from the keyboard, causing the handler to try and register priv->switches_dev twice, as can be seen from the dev_info() message getting logged twice: [83.861800] intel-vbtn INT33D6:00: Registering Intel Virtual Switches input-dev after receiving a switch event [83.861858] input: Intel Virtual Switches as /devices/pci0000:00/0000:00:1f.0/PNP0C09:00/INT33D6:00/input/input17 [83.861865] intel-vbtn INT33D6:00: Registering Intel Virtual Switches input-dev after receiving a switch event After which things go seriously wrong: [83.861872] sysfs: cannot create duplicate filename '/devices/pci0000:00/0000:00:1f.0/PNP0C09:00/INT33D6:00/input/input17' ... [83.861967] kobject: kobject_add_internal failed for input17 with -EEXIST, don't try to register things with the same name in the same directory. [83.877338] BUG: kernel NULL pointer dereference, address: 0000000000000018 ... Protect intel-vbtn notify_handler() from racing with itself with a mutex to fix this.	2024-08-26	5.5	Medium
CVE-2024-44944	Linux	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: use helper function to calculate expect ID	2024-08-30	5.5	Medium

		Delete expectation path is missing a call to the nf_expect_get_id() helper function to calculate the expectation ID, otherwise LSB of the expectation object address is leaked to userspace.			
CVE-2022-48944	Linux	In the Linux kernel, the following vulnerability has been resolved: sched: Fix yet more sched_fork() races Where commit 4ef0c5c6b5ba ("kernel/sched: Fix sched_fork() access an invalid sched_task_group") fixed a fork race vs cgroup, it opened up a race vs syscalls by not placing the task on the runqueue before it gets exposed through the pidhash. Commit 13765de8148f ("sched/fair: Fix fault in reweight_entity") is trying to fix a single instance of this, instead fix the whole class of issues, effectively reverting this commit.	2024-08-30	5.5	Medium
CVE-2024-44946	Linux	In the Linux kernel, the following vulnerability has been resolved: kcm: Serialise kcm_sendmsg() for the same socket. syzkaller reported UAF in kcm_release(). [0] The scenario is 1. Thread A builds a skb with MSG_MORE and sets kcm->seq_skb. 2. Thread A resumes building skb from kcm->seq_skb but is blocked by sk_stream_wait_memory() 3. Thread B calls sendmsg() concurrently, finishes building kcm->seq_skb and puts the skb to the write queue 4. Thread A faces an error and finally frees skb that is already in the write queue 5. kcm_release() does double-free the skb in the write queue When a thread is building a MSG_MORE skb, another thread must not touch it. Let's add a per-sk mutex and serialise kcm_sendmsg(). [0]: BUG: KASAN: slab-use-after-free in __skb_unlink include/linux/skbuff.h:2366 [inline] BUG: KASAN: slab-use-after-free in __skb_dequeue include/linux/skbuff.h:2385 [inline] BUG: KASAN: slab-use-after-free in __skb_queue_purge_reason include/linux/skbuff.h:3175 [inline] BUG: KASAN: slab-use-after-free in __skb_queue_purge include/linux/skbuff.h:3181 [inline] BUG: KASAN: slab-use-after-free in kcm_release+0x170/0x4c8 net/kcm/kcmsock.c:1691 Read of size 8 at addr ffff0000ced0fc80 by task syz-executor329/6167 CPU: 1 PID: 6167 Comm: syz-executor329 Tainted: G B 6.8.0-rc5-syzkaller-g9abbc24128bc #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/25/2024 Call trace: dump_backtrace+0x1b8/0x1e4 arch/arm64/kernel/stacktrace.c:291 show_stack+0x2c/0x3c arch/arm64/kernel/stacktrace.c:298 __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd0/0x124 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:377 [inline] print_report+0x178/0x518 mm/kasan/report.c:488 kasan_report+0xd8/0x138 mm/kasan/report.c:601 __asan_report_load8_noabort+0x20/0x2c mm/kasan/report_generic.c:381 __skb_unlink include/linux/skbuff.h:2366 [inline] __skb_dequeue include/linux/skbuff.h:2385 [inline] __skb_queue_purge_reason include/linux/skbuff.h:3175 [inline]	2024-08-31	5.5	Medium

		__skb_queue_purge include/linux/skbuff.h:3181 [inline] kcm_release+0x170/0x4c8 net/kcm/kcmsock.c:1691 __sock_release net/socket.c:659 [inline] sock_close+0xa4/0x1e8 net/socket.c:1421 __fput+0x30c/0x738 fs/file_table.c:376 ____fput+0x20/0x30 fs/file_table.c:404 task_work_run+0x230/0x2e0 kernel/task_work.c:180 exit_task_work include/linux/task_work.h:38 [inline] do_exit+0x618/0x1f64 kernel/exit.c:871 do_group_exit+0x194/0x22c kernel/exit.c:1020 get_signal+0x1500/0x15ec kernel/signal.c:2893 do_signal+0x23c/0x3b44 arch/arm64/kernel/signal.c:1249 do_notify_resume+0x74/0x1f4 arch/arm64/kernel/entry-common.c:148 exit_to_user_mode_prepare arch/arm64/kernel/entry-common.c:169 [inline] exit_to_user_mode arch/arm64/kernel/entry-common.c:178 [inline] el0_svc+0xac/0x168 arch/arm64/kernel/entry-common.c:713 el0t_64_sync_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:598 Allocated by task 6166: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x40/0x78 mm/kasan/common.c:68 kasan_save_alloc_info+0x70/0x84 mm/kasan/generic.c:626 unpoison_slab_object mm/kasan/common.c:314 [inline] __kasan_slab_alloc+0x74/0x8c mm/kasan/common.c:340 kasan_slab_alloc include/linux/kasan.h:201 [inline] slab_post_alloc_hook mm/slub.c:3813 [inline] slab_alloc_node mm/slub.c:3860 [inline] kmem_cache_alloc_node+0x204/0x4c0 mm/slub.c:3903 __alloc_skb+0x19c/0x3d8 net/core/skbuff.c:641 alloc_skb include/linux/skbuff.h:1296 [inline] kcm_sendmsg+0x1d3c/0x2124 net/kcm/kcmsock.c:783 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] sock_sendmsg+0x220/0x2c0 net/socket.c:768 splice_to_socket+0x7cc/0xd58 fs/splice.c:889 do_splice_from fs/splice.c:941 [inline] direct_splice_actor+0xec/0x1d8 fs/splice.c:1164 splice_direct_to_actor+0x438/0xa0c fs/splice.c:1108 do_splice_direct_actor ---truncated---			
CVE-2024-8210	D-Link	A vulnerability was found in D-Link DNS-120, DNR-202L, DNS-315L, DNS-320, DNS-320L, DNS-320LW, DNS-321, DNR-322L, DNS-323, DNS-325, DNS-326, DNS-327L, DNR-326, DNS-340L, DNS-343, DNS-345, DNS-726-4, DNS-1100-4, DNS-1200-05 and DNS-1550-04 up to 20240814. It has been classified as critical. This affects the function sprintf of the file /cgi-bin/hd_config.cgi. The manipulation of the argument f_mount leads to command injection. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used. NOTE: This vulnerability only affects products that are no longer supported by the maintainer. NOTE: Vendor was contacted early and confirmed that the product is end-of-life. It should be retired and replaced.	2024-08-27	5.3	Medium
CVE-2024-8211	D-Link	A vulnerability was found in D-Link DNS-120, DNR-202L, DNS-315L, DNS-320, DNS-320L, DNS-320LW, DNS-321, DNR-322L, DNS-323, DNS-325, DNS-326, DNS-327L, DNR-326, DNS-340L, DNS-343, DNS-345, DNS-726-4, DNS-1100-4, DNS-1200-05 and DNS-1550-04 up to 20240814. It has been declared as critical. This vulnerability affects the function cgi_FMT_Std2R1_DiskMGR of the file /cgi-bin/hd_config.cgi. The manipulation of the argument f_newly_dev leads to command injection. The attack can be initiated remotely. The exploit has been disclosed to the public and may be used. NOTE: This vulnerability only affects products that are no longer supported by the maintainer. NOTE: Vendor was contacted early and confirmed that the product is end-of-life. It should be retired and replaced.	2024-08-27	5.3	Medium
CVE-2024-8212	D-Link	A vulnerability was found in D-Link DNS-120, DNR-202L, DNS-315L, DNS-320, DNS-320L, DNS-320LW, DNS-321, DNR-322L, DNS-323, DNS-325, DNS-326, DNS-327L, DNR-326, DNS-340L, DNS-343, DNS-345, DNS-726-4, DNS-1100-4, DNS-1200-05 and DNS-1550-04 up to 20240814. It has been rated as critical. This issue affects the function cgi_FMT_R12R5_2nd_DiskMGR of the file /cgi-bin/hd_config.cgi. The manipulation of the argument f_source_dev leads to command injection. The attack may be initiated remotely. The exploit has been disclosed to the public and may be used. NOTE: This vulnerability only affects products	2024-08-27	5.3	Medium

		that are no longer supported by the maintainer. NOTE: Vendor was contacted early and confirmed that the product is end-of-life. It should be retired and replaced.			
CVE-2024-8213	D-Link	A vulnerability classified as critical has been found in D-Link DNS-120, DNR-202L, DNS-315L, DNS-320, DNS-320L, DNS-320LW, DNS-321, DNR-322L, DNS-323, DNS-325, DNS-326, DNS-327L, DNR-326, DNS-340L, DNS-343, DNS-345, DNS-726-4, DNS-1100-4, DNS-1200-05 and DNS-1550-04 up to 20240814. Affected is the function cgi_FMT_R12R5_1st_DiskMGR of the file /cgi-bin/hd_config.cgi. The manipulation of the argument f_source_dev leads to command injection. It is possible to launch the attack remotely. The exploit has been disclosed to the public and may be used. NOTE: This vulnerability only affects products that are no longer supported by the maintainer. NOTE: Vendor was contacted early and confirmed that the product is end-of-life. It should be retired and replaced.	2024-08-27	5.3	Medium
CVE-2024-8214	D-Link	A vulnerability classified as critical was found in D-Link DNS-120, DNR-202L, DNS-315L, DNS-320, DNS-320L, DNS-320LW, DNS-321, DNR-322L, DNS-323, DNS-325, DNS-326, DNS-327L, DNR-326, DNS-340L, DNS-343, DNS-345, DNS-726-4, DNS-1100-4, DNS-1200-05 and DNS-1550-04 up to 20240814. Affected by this vulnerability is the function cgi_FMT_Std2R5_2nd_DiskMGR of the file /cgi-bin/hd_config.cgi. The manipulation of the argument f_source_dev leads to command injection. The attack can be launched remotely. The exploit has been disclosed to the public and may be used. NOTE: This vulnerability only affects products that are no longer supported by the maintainer. NOTE: Vendor was contacted early and confirmed that the product is end-of-life. It should be retired and replaced.	2024-08-27	5.3	Medium
CVE-2024-20284	Cisco	A vulnerability in the Python interpreter of Cisco NX-OS Software could allow an authenticated, low-privileged, local attacker to escape the Python sandbox and gain unauthorized access to the underlying operating system of the device. The vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by manipulating specific functions within the Python interpreter. A successful exploit could allow an attacker to escape the Python sandbox and execute arbitrary commands on the underlying operating system with the privileges of the authenticated user. Note: An attacker must be authenticated with Python execution privileges to exploit these vulnerabilities. For more information regarding Python execution privileges, see product-specific documentation, such as the section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.	2024-08-28	5.3	Medium
CVE-2024-20285	Cisco	A vulnerability in the Python interpreter of Cisco NX-OS Software could allow an authenticated, low-privileged, local attacker to escape the Python sandbox and gain unauthorized access to the underlying operating system of the device. The vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by manipulating specific functions within the Python interpreter. A successful exploit could allow an attacker to escape the Python sandbox and execute arbitrary commands on the underlying operating system with the privileges of the authenticated user. Note: An attacker must be authenticated with Python execution privileges to exploit these vulnerabilities. For more information regarding Python execution privileges, see product-specific documentation, such as the section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.	2024-08-28	5.3	Medium
CVE-2024-20286	Cisco	A vulnerability in the Python interpreter of Cisco NX-OS Software could allow an authenticated, low-privileged, local attacker to escape the Python sandbox and gain unauthorized access to the underlying operating system of the device. The vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by manipulating specific functions within the Python interpreter. A successful exploit could allow an attacker to escape the Python sandbox and execute arbitrary commands on the underlying operating system with the privileges of the authenticated user.	2024-08-28	5.3	Medium

		Note: An attacker must be authenticated with Python execution privileges to exploit these vulnerabilities. For more information regarding Python execution privileges, see product-specific documentation, such as the section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.			
CVE-2024-38303	Dell	Dell PowerEdge Platform, 14G Intel BIOS version(s) prior to 2.22.x, contains an Improper Input Validation vulnerability. A high privileged attacker with local access could potentially exploit this vulnerability, leading to Information disclosure.	2024-08-29	5.3	Medium
CVE-2024-43887	Linux	In the Linux kernel, the following vulnerability has been resolved: net/tcp: Disable TCP-AO static key after RCU grace period The lifetime of TCP-AO static_key is the same as the last tcp_ao_info. On the socket destruction tcp_ao_info ceases to be with RCU grace period, while tcp-ao static branch is currently deferred destructed. The static key definition is : DEFINE_STATIC_KEY_DEFERRED_FALSE(tcp_ao_needed, HZ); which means that if RCU grace period is delayed by more than a second and tcp_ao_needed is in the process of disablement, other CPUs may yet see tcp_ao_info which atent dead, but soon-to-be. And that breaks the assumption of static_key_fast_inc_not_disabled(). See the comment near the definition: > * The caller must make sure that the static key can't get disabled while > * in this function. It doesn't patch jump labels, only adds a user to > * an already enabled static key. Originally it was introduced in commit eb8c507296f6 ("jump_label: Prevent key->enabled int overflow"), which is needed for the atomic contexts, one of which would be the creation of a full socket from a request socket. In that atomic context, it's known by the presence of the key (md5/ao) that the static branch is already enabled. So, the ref counter for that static branch is just incremented instead of holding the proper mutex. static_key_fast_inc_not_disabled() is just a helper for such usage case. But it must not be used if the static branch could get disabled in parallel as it's not protected by jump_label_mutex and as a result, races with jump_label_update() implementation details. Happened on netdev test-bot[1], so not a theoretical issue: [] jump_label: Fatal kernel bug, unexpected op at tcp_inbound_hash+0x1a7/0x870 [ffffffffa8c4e9b7] (eb 50 0f 1f 44 != 66 90 0f 1f 00)) size:2 type:1 [] ------------[cut here]------------ [] kernel BUG at arch/x86/kernel/jump_label.c:73! [] Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI [] CPU: 3 PID: 243 Comm: kworker/3:3 Not tainted 6.10.0-virtme #1 [] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 [] Workqueue: events jump_label_update_timeout [] RIP: 0010:__jump_label_patch+0x2f6/0x350 ... [] Call Trace: [] <TASK> [] arch_jump_label_transform_queue+0x6c/0x110 [] __jump_label_update+0xef/0x350 [] __static_key_slow_dec_cpuslocked.part.0+0x3c/0x60 [] jump_label_update_timeout+0x2c/0x40 [] process_one_work+0xe3b/0x1670 [] worker_thread+0x587/0xce0 [] kthread+0x28a/0x350 [] ret_from_fork+0x31/0x70 [] ret_from_fork_asm+0x1a/0x30 [] </TASK> [] Modules linked in: veth [] ---[end trace 0000000000000000]---	2024-08-26	4.7	Medium

		[] RIP: 0010:__jump_label_patch+0x2f6/0x350 [1]: https://netdev-3.bots.linux.dev/vmksft-tcp-ao-dbg/results/696681/5-connect-deny-ipv6/stderr			
		In the Linux kernel, the following vulnerability has been resolved: tracing: Have format file honor EVENT_FILE_FL_FREED When eventfs was introduced, special care had to be done to coordinate the freeing of the file meta data with the files that are exposed to user space. The file meta data would have a ref count that is set when the file is created and would be decremented and freed after the last user that opened the file closed it. When the file meta data was to be freed, it would set a flag (EVENT_FILE_FL_FREED) to denote that the file is freed, and any new references made (like new opens or reads) would fail as it is marked freed. This allowed other meta data to be freed after this flag was set (under the event_mutex). All the files that were dynamically created in the events directory had a pointer to the file meta data and would call event_release() when the last reference to the user space file was closed. This would be the time that it is safe to free the file meta data. A shortcut was made for the "format" file. It's i_private would point to the "call" entry directly and not point to the file's meta data. This is because all format files are the same for the same "call", so it was thought there was no reason to differentiate them. The other files maintain state (like the "enable", "trigger", etc). But this meant if the file were to disappear, the "format" file would be unaware of it. This caused a race that could be trigger via the user_events test (that would create dynamic events and free them), and running a loop that would read the user_events format files: In one console run: # cd tools/testing/selftests/user_events # while true; do ./ftrace_test; done And in another console run: # cd /sys/kernel/tracing/ # while true; do cat events/user_events/__test_event/format; done 2>/dev/null With KASAN memory checking, it would trigger a use-after-free bug report (which was a real bug). This was because the format file was not checking the file's meta data flag "EVENT_FILE_FL_FREED", so it would access the event that the file meta data pointed to after the event was freed. After inspection, there are other locations that were found to not check the EVENT_FILE_FL_FREED flag when accessing the trace_event_file. Add a new helper function: event_file_file() that will make sure that the event_mutex is held, and will return NULL if the trace_event_file has the EVENT_FILE_FL_FREED flag set. Have the first reference of the struct file pointer use event_file_file() and check for NULL. Later uses can still use			
CVE-2024-43891	Linux	the event_file_data() helper function if the event_mutex is still	2024-08-26	4.7	Medium

		held and was not released since the event_file_file() call.			
		In the Linux kernel, the following vulnerability has been resolved: memcg: protect concurrent access to mem_cgroup_idr Commit 73f576c04b94 ("mm: memcontrol: fix cgroup creation failure after many small jobs") decoupled the memcg IDs from the CSS ID space to fix the cgroup creation failures. It introduced IDR to maintain the memcg ID space. The IDR depends on external synchronization mechanisms for modifications. For the mem_cgroup_idr, the idr_alloc() and idr_replace() happen within css callback and thus are protected through cgroup_mutex from concurrent modifications. However idr_remove() for mem_cgroup_idr was not protected against concurrency and can be run concurrently for different memcgs when they hit their refcnt to zero. Fix that. We have been seeing list_lru based kernel crashes at a low frequency in our fleet for a long time. These crashes were in different part of list_lru code including list_lru_add(), list_lru_del() and reparenting code. Upon further inspection, it looked like for a given object (dentry and inode), the super_block's list_lru didn't have list_lru_one for the memcg of that object. The initial suspicions were either the object is not allocated through kmem_cache_alloc_lru() or somehow memcg_list_lru_alloc() failed to allocate list_lru_one() for a memcg but returned success. No evidence were found for these cases. Looking more deeply, we started seeing situations where valid memcg's id is not present in mem_cgroup_idr and in some cases multiple valid memcgs have same id and mem_cgroup_idr is pointing to one of them. So, the most reasonable explanation is that these situations can happen due to race between multiple idr_remove() calls or race between idr_alloc()/idr_replace() and idr_remove(). These races are causing multiple memcgs to acquire the same ID and then offlining of one of them would cleanup list_lrus on the system for all of them. Later access from other memcgs to the list_lru cause crashes due to missing list_lru_one.			
CVE-2024-43892	Linux		2024-08-26	4.7	Medium
CVE-2024-35118	IBM	IBM MaaS360 for Android 6.31 through 8.60 is using hard coded credentials that can be obtained by a user with physical access to the device.	2024-08-29	4.6	Medium
CVE-2024-20289	Cisco	A vulnerability in the CLI of Cisco NX-OS Software could allow an authenticated, low-privileged, local attacker to execute arbitrary commands on the underlying operating system of an affected device. This vulnerability is due to insufficient validation of arguments for a specific CLI command. An attacker could exploit this vulnerability by including crafted input as the argument of the affected command. A successful exploit could allow the attacker to execute arbitrary commands on the underlying operating system with the privileges of the currently logged-in user.	2024-08-28	4.4	Medium
CVE-2024-20279	Cisco	A vulnerability in the restricted security domain implementation of Cisco Application Policy Infrastructure Controller (APIC) could allow an authenticated, remote attacker to modify the behavior of default system policies, such as quality of service (QoS) policies, on an affected system. This vulnerability is due to improper access control when restricted security domains are used to implement multi-tenancy. An attacker with a valid user account associated with a restricted security domain could exploit this	2024-08-28	4.3	Medium

		vulnerability. A successful exploit could allow the attacker to read, modify, or delete child policies created under default system policies, which are implicitly used by all tenants in the fabric, resulting in disruption of network traffic. Exploitation is not possible for policies under tenants that an attacker has no authorization to access.			
CVE-2024-6053	TeamViewer	Improper access control in the clipboard synchronization feature in TeamViewer Full Client prior version 15.57 and TeamViewer Meeting prior version 15.55.3 can lead to unintentional sharing of the clipboard with the current presenter of a meeting.	2024-08-28	4.3	Medium
CVE-2024-38304	Dell	Dell PowerEdge Platform, 14G Intel BIOS version(s) prior to 2.22.x, contains an Access of Memory Location After End of Buffer vulnerability. A low privileged attacker with local access could potentially exploit this vulnerability, leading to Information disclosure.	2024-08-29	3.8	Low

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

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