الهيئة الوطنية للأمن السيبراني
Nat onal Cybersecurity Authority

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

As part of NCA duties to help securing the cyberspace and protecting national interests, NCA provides the weekly summary of published vulnerabilities by the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) for the week from 21st of October to 27th of October. 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**

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

في ضوء دور الهيئة الوطنية للأمن السيبراني للمساعدة في حماية الفضاء السيبراني الوطني، تود الهيئة مشاركتكم النشرة الأسبوعية للثغرات المسجلة من قبل the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) للأسبوع من ٢١ أكتوبر إلى ٢٧ أكتوبر. 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**

CVE ID & Source	Vendor - Product	Description	Publish Date	CVSS Score	Severity
CVE-2024-47901	siemens - intermesh_7177_hybrid_2.0_subscriber	A vulnerability has been identified in InterMesh 7177 Hybrid 2.0 Subscriber (All versions < V8.2.12), InterMesh 7707 Fire Subscriber (All versions < V7.2.12 only if the IP interface is enabled (which is not the default configuration)). The web server of affected devices does not sanitize the input parameters in specific GET requests that allow for code execution on operating system level. In combination with other vulnerabilities (CVE-2024-47902, CVE-2024-47903, CVE-2024-47904) this could allow an unauthenticated remote attacker to execute arbitrary code with root privileges.	2024-10-23	10	Critical
CVE-2024-20329	cisco - Cisco Adaptive Security Appliance (ASA) Software	A vulnerability in the SSH subsystem of Cisco Adaptive Security Appliance (ASA) Software could allow an authenticated, remote attacker to execute operating system commands as root. This vulnerability is due to insufficient validation of user input. An attacker could exploit this vulnerability by submitting crafted input when executing remote CLI commands over SSH. A successful exploit could allow the attacker to execute commands on the underlying operating system with root-level privileges. An	2024-10-23	9.9	Critical

		attacker with limited user privileges could use this vulnerability to gain complete control over the system.			
CVE-2024-20424	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to execute arbitrary commands on the underlying operating system as root. This vulnerability is due to insufficient input validation of certain HTTP requests. An attacker could exploit this vulnerability by authenticating to the web-based management interface of an affected device and then sending a crafted HTTP request to the device. A successful exploit could allow the attacker to execute arbitrary commands with root permissions on the underlying operating system of the Cisco FMC device or to execute commands on managed Cisco Firepower Threat Defense (FTD) devices. To exploit this vulnerability, the attacker would need valid credentials for a user account with at least the role of Security Analyst (Read Only).	2024-10-23	9.9	Critical
CVE-2024-43177	ibm - multiple products	IBM Concert 1.0.0 and 1.0.1 vulnerable to attacks that rely on the use of cookies without the SameSite attribute.	2024-10-22	9.8	Critical
CVE-2024-47575	fortinet - multiple products	A missing authentication for critical function in FortiManager 7.6.0, FortiManager 7.4.0 through 7.4.4, FortiManager 7.2.0 through 7.2.7, FortiManager 7.0.0 through 7.0.12, FortiManager 6.4.0 through 6.4.14, FortiManager 6.2.0 through 6.2.12, Fortinet FortiManager Cloud 7.4.1 through 7.4.4, FortiManager Cloud 7.2.1 through 7.2.7, FortiManager Cloud 7.0.1 through 7.0.13, FortiManager Cloud 6.4.1 through 6.4.7 allows attacker to execute arbitrary code or commands via specially crafted requests.	2024-10-23	9.8	Critical
CVE-2024-20412	cisco - Cisco Firepower Threat Defense Software	A vulnerability in Cisco Firepower Threat Defense (FTD) Software for Cisco Firepower 1000, 2100, 3100, and 4200 Series could allow an unauthenticated, local attacker to access an affected system using static credentials. This vulnerability is due to the presence of static accounts with hard-coded passwords on an affected system. An attacker could exploit this vulnerability by logging in to the CLI of an affected device with these credentials. A successful exploit could allow the attacker to access the affected system and retrieve sensitive information, perform limited troubleshooting actions, modify some configuration options, or render the device unable to boot to the operating system, requiring a reimage of the device.	2024-10-23	9.3	Critical

CVE-2024-47685	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject_ipv6: fix nf_reject_ip6_tcphdr_put() syzbot reported that nf_reject_ip6_tcphdr_put() was possibly sending garbage on the four reserved tcp bits (th->res1) Use skb_put_zero() to clear the whole TCP header, as done in nf_reject_ip_tcphdr_put() BUG: KMSAN: uninit-value in nf_reject_ip6_tcphdr_put+0x688/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:255 nf_reject_ip6_tcphdr_put+0x688/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:255 nf_send_reset6+0xd84/0x15b0 net/ipv6/netfilter/nf_reject_ipv6.c:344 nft_reject_inet_eval+0x3c1/0x880 net/netfilter/nft_reject_inet.c:48 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x438/0x22a0 net/netfilter/nf_tables_core.c:288 nft_do_chain_inet+0x41a/0x4f0 net/netfilter/nft_chain_filter.c:161 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] NF_HOOK include/linux/netfilter.h:312 [inline] ipv6_rcv+0x29b/0x390 net/ipv6/ip6_input.c:310 __netif_receive_skb_one_core net/core/dev.c:5661 [inline] __netif_receive_skb+0x1da/0xa00 net/core/dev.c:5775 process_backlog+0x4ad/0xa50 net/core/dev.c:6108 __napi_poll+0xe7/0x980 net/core/dev.c:6772 napi_poll net/core/dev.c:6841 [inline] net_rx_action+0xa5a/0x19b0 net/core/dev.c:6963 handle_softirqs+0x1ce/0x800 kernel/softirq.c:554 __do_softirq+0x14/0x1a kernel/softirq.c:588 do_softirq+0x9a/0x100 kernel/softirq.c:455 __local_bh_enable_ip+0x9f/0xb0 kernel/softirq.c:382 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:908 [inline] __dev_queue_xmit+0x2692/0x5610 net/core/dev.c:4450 dev_queue_xmit include/linux/netdevice.h:3105 [inline] neigh_resolve_output+0x9ca/0xae0	2024-10-21	9.1	Critical

		net/core/neighbour.c:1565 neigh_output include/net/neighbour.h:542 [inline] ip6_finish_output2+0x2347/0x2ba0 net/ipv6/ip6_output.c:141 __ip6_finish_output net/ipv6/ip6_output.c:215 [inline] ip6_finish_output+0xbb8/0x14b0 net/ipv6/ip6_output.c:226 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip6_output+0x356/0x620 net/ipv6/ip6_output.c:247 dst_output include/net/dst.h:450 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] ip6_xmit+0x1ba6/0x25d0 net/ipv6/ip6_output.c:366 inet6_csk_xmit+0x442/0x530 net/ipv6/inet6_connection_sock.c:135 __tcp_transmit_skb+0x3b07/0x4880 net/ipv4/tcp_output.c:1466 tcp_transmit_skb net/ipv4/tcp_output.c:1484 [inline] tcp_connect+0x35b6/0x7130 net/ipv4/tcp_output.c:4143 tcp_v6_connect+0x1bcc/0x1e40 net/ipv6/tcp_ipv6.c:333 __inet_stream_connect+0x2ef/0x1730 net/ipv4/af_inet.c:679 inet_stream_connect+0x6a/0xd0 net/ipv4/af_inet.c:750 __sys_connect_file net/socket.c:2061 [inline] __sys_connect+0x606/0x690 net/socket.c:2078 __do_sys_connect net/socket.c:2088 [inline] __se_sys_connect net/socket.c:2085 [inline] __x64_sys_connect+0x91/0xe0 net/socket.c:2085 x64_sys_call+0x27a5/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:43 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was stored to memory at: nf_reject_ip6_tcphdr_put+0x60c/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:249 nf_send_reset6+0xd84/0x15b0 net/ipv6/netfilter/nf_reject_ipv6.c:344 nft_reject_inet_eval+0x3c1/0x880 net/netfilter/nft_reject_inet.c:48 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x438/0x22a0			

		net/netfilter/nf_tables_core.c:288 nft_do_chain_inet+0x41a/0x4f0 net/netfilter/nft_chain_filter.c:161 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] NF_HOOK include/linux/netfilter.h:312 [inline] ipv6_rcv+0x29b/0x390 net/ipv6/ip6_input.c:310 __netif_receive_skb_one_core ---truncated---			
CVE-2024-46902	trendmicro - multiple products	A vulnerability in Trend Micro Deep Discovery Inspector (DDI) versions 5.8 and above could allow an attacker to disclose sensitive information affected installations. Please note: an attacker must first obtain the ability to execute high-privileged code (admin user rights) on the target system in order to exploit this vulnerability.	2024-10-22	9.1	Critical
CVE-2024-26271	liferay - multiple products	Cross-site request forgery (CSRF) vulnerability in the My Account widget in Liferay Portal 7.4.3.75 through 7.4.3.111, and Liferay DXP 2023.Q4.0 through 2023.Q4.2, 2023.Q3.1 through 2023.Q3.5, 7.4 update 75 through update 92 and 7.3 update 32 through update 36 allows remote attackers to (1) change user passwords, (2) shut down the server, (3) execute arbitrary code in the scripting console, (4) and perform other administrative actions via the _com_liferay_my_account_web_portlet_MyAccountPortlet_backURL parameter.	2024-10-22	8.8	High
CVE-2024-26272	liferay - multiple products	Cross-site request forgery (CSRF) vulnerability in the content page editor in Liferay Portal 7.3.2 through 7.4.3.107, and Liferay DXP 2023.Q4.0 through 2023.Q4.2, 2023.Q3.1 through 2023.Q3.5, 7.4 GA through update 92 and 7.3 GA through update 35 allows remote attackers to (1) change user passwords, (2) shut down the server, (3) execute arbitrary code in the scripting console, (4) and perform other administrative actions via the p_l_back_url parameter.	2024-10-22	8.8	High
CVE-2024-26273	liferay - multiple products	Cross-site request forgery (CSRF) vulnerability in the content page editor in Liferay Portal 7.4.0 through 7.4.3.103, and Liferay DXP 2023.Q4.0 through 2023.Q4.2, 2023.Q3.1 through 2023.Q3.5, 7.4 GA through update 92 and 7.3 update 29 through update 35 allows remote attackers to (1) change user passwords, (2) shut down the server, (3) execute arbitrary code in the scripting console, (4) and perform other administrative actions via the _com_liferay_commerce_catalog_web_internal_portlet_CommerceCatalogsPortlet_redirect parameter.	2024-10-22	8.8	High

CVE-2024-38002	liferay - multiple products	The workflow component in Liferay Portal 7.3.2 through 7.4.3.111, and Liferay DXP 2023.Q4.0 through 2023.Q4.5, 2023.Q3.1 through 2023.Q3.8, 7.4 GA through update 92 and 7.3 GA through update 36 does not properly check user permissions before updating a workflow definition, which allows remote authenticated users to modify workflow definitions and execute arbitrary code (RCE) via the headless API.	2024-10-22	8.8	High
CVE-2024-45518	zimbra - multiple products	An issue was discovered in Zimbra Collaboration (ZCS) 10.1.x before 10.1.1, 10.0.x before 10.0.9, 9.0.0 before Patch 41, and 8.8.15 before Patch 46. It allows authenticated users to exploit Server-Side Request Forgery (SSRF) due to improper input sanitization and misconfigured domain whitelisting. This issue permits unauthorized HTTP requests to be sent to internal services, which can lead to Remote Code Execution (RCE) by chaining Command Injection within the internal service. When combined with existing XSS vulnerabilities, this SSRF issue can further facilitate Remote Code Execution (RCE).	2024-10-22	8.8	High
CVE-2024-10230	google - chrome	Type Confusion in V8 in Google Chrome prior to 130.0.6723.69 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2024-10-22	8.8	High
CVE-2024-10231	google - chrome	Type Confusion in V8 in Google Chrome prior to 130.0.6723.69 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)	2024-10-22	8.8	High
CVE-2024-47014	google - Android	Android before 2024-10-05 on Google Pixel devices allows privilege escalation in the ABL component, A-330537292.	2024-10-25	8.8	High
CVE-2024-20260	cisco - multiple products	A vulnerability in the VPN and management web servers of the Cisco Adaptive Security Virtual Appliance (ASAv) and Cisco Secure Firewall Threat Defense Virtual (FTDv), formerly Cisco Firepower Threat Defense Virtual, platforms could allow an unauthenticated, remote attacker to cause the virtual devices to run out of system memory, which could cause SSL VPN connection processing to slow down and eventually cease all together. This vulnerability is due to a lack of proper memory management for new incoming SSL/TLS connections on the virtual platforms. An attacker could exploit this vulnerability by sending a large number of new incoming SSL/TLS connections to the targeted virtual platform. A successful exploit could allow the attacker to deplete system memory,&nbsp;resulting in a denial of service (DoS) condition. The memory could be reclaimed slowly if the attack traffic is stopped, but a manual reload may be required to restore operations quickly.	2024-10-23	8.6	High

CVE-2024-20330	cisco - Cisco Firepower Threat Defense Software	A vulnerability in the Snort 2 and Snort 3 TCP and UDP detection engine of Cisco Firepower Threat Defense (FTD) Software for Cisco Firepower 2100 Series Appliances could allow an unauthenticated, remote attacker to cause memory corruption, which could cause the Snort detection engine to restart unexpectedly. This vulnerability is due to improper memory management when the Snort detection engine processes specific TCP or UDP packets. An attacker could exploit this vulnerability by sending crafted TCP or UDP packets through a device that is inspecting traffic using the Snort detection engine. A successful exploit could allow the attacker to restart the Snort detection engine repeatedly, which could cause a denial of service (DoS) condition. The DoS condition impacts only the traffic through the device that is examined by the Snort detection engine. The device can still be managed over the network. Note: Once a memory block is corrupted, it cannot be cleared until the Cisco Firepower 2100 Series Appliance is manually reloaded. This means that the Snort detection engine could crash repeatedly, causing traffic that is processed by the Snort detection engine to be dropped until the device is manually reloaded.	2024-10-23	8.6	High
CVE-2024-20339	cisco - Cisco Firepower Threat Defense Software	A vulnerability in the TLS processing feature of Cisco Firepower Threat Defense (FTD) Software for Cisco Firepower 2100 Series could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to an issue that occurs when TLS traffic is processed. An attacker could exploit this vulnerability by sending certain TLS traffic over IPv4 through an affected device. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition and impacting traffic to and through the affected device.	2024-10-23	8.6	High
CVE-2024-20351	cisco - Cisco Firepower Threat Defense Software	A vulnerability in the TCP/IP traffic handling function of the Snort Detection Engine of Cisco Firepower Threat Defense (FTD) Software and Cisco FirePOWER Services could allow an unauthenticated, remote attacker to cause legitimate network traffic to be dropped, resulting in a denial of service (DoS) condition. This vulnerability is due to the improper handling of TCP/IP network traffic. An attacker could exploit this vulnerability by sending a large amount of TCP/IP network traffic through the affected device. A successful exploit could allow the attacker to cause the Cisco FTD device to drop network traffic, resulting in a DoS condition. The affected device must be rebooted to resolve the DoS condition.	2024-10-23	8.6	High

CVE-2024-20402	cisco - multiple products	A vulnerability in the SSL VPN feature for Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause an affected device to reload unexpectedly, resulting in a denial of service (DoS) condition. This vulnerability is due to a logic error in memory management when the device is handling SSL VPN connections. An attacker could exploit this vulnerability by sending crafted SSL/TLS packets to the SSL VPN server of the affected device. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition.	2024-10-23	8.6	High
CVE-2024-20426	cisco - multiple products	A vulnerability in the Internet Key Exchange version 2 (IKEv2) protocol for VPN termination of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device. This vulnerability is due to insufficient input validation. An attacker could exploit this vulnerability by sending crafted IKEv2 traffic to an affected device. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition.	2024-10-23	8.6	High
CVE-2024-20494	cisco - multiple products	A vulnerability in the TLS cryptography functionality of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause the device to reload unexpectedly, resulting in a denial of service (DoS) condition. This vulnerability is due to improper data validation during the TLS 1.3 handshake. An attacker could exploit this vulnerability by sending a crafted TLS 1.3 packet to an affected system through a TLS 1.3-enabled listening socket. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition. Note: This vulnerability can also impact the integrity of a device by causing VPN HostScan communication failures or file transfer failures when Cisco ASA Software is upgraded using Cisco Adaptive Security Device Manager (ASDM).	2024-10-23	8.6	High
CVE-2024-20495	cisco - multiple products	A vulnerability in the Remote Access VPN feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause the device to reload unexpectedly, resulting in a denial of service (DoS) condition on an affected device. This vulnerability is due to improper validation of client key data after the TLS session is established. An attacker could exploit this vulnerability by sending a crafted key value to an	2024-10-23	8.6	High

		affected system over the secure TLS session. A successful exploit could allow the attacker to cause the device to reload, resulting in a DoS condition.			
CVE-2024-47904	siemens - intermesh_7177_hybrid_2.0_subscriber	A vulnerability has been identified in InterMesh 7177 Hybrid 2.0 Subscriber (All versions < V8.2.12), InterMesh 7707 Fire Subscriber (All versions < V7.2.12 only if the IP interface is enabled (which is not the default configuration)). The affected devices contain a SUID binary that could allow an authenticated local attacker to execute arbitrary commands with root privileges.	2024-10-23	8.5	High
CVE-2024-5608	manageengine - ADAudit Plus	Zohocorp ManageEngine ADAudit Plus versions below 8121 are vulnerable to SQL Injection in the technician reports feature.	2024-10-24	8.3	High
CVE-2024-10229	google - chrome	Inappropriate implementation in Extensions in Google Chrome prior to 130.0.6723.69 allowed a remote attacker to bypass site isolation via a crafted Chrome Extension. (Chromium security severity: High)	2024-10-22	8.1	High
CVE-2024-47023	google - android	there is a possible man-in-the-middle attack due to a logic error in the code. This could lead to remote escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	8.1	High
CVE-2024-47675	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix use-after-free in bpf_uprobe_multi_link_attach() If bpf_link_prime() fails, bpf_uprobe_multi_link_attach() goes to the error_free label and frees the array of bpf_uprobe's without calling bpf_uprobe_unregister(). This leaks bpf_uprobe->uprobe and worse, this frees bpf_uprobe->consumer without removing it from the uprobe->consumers list.	2024-10-21	7.8	High
CVE-2024-47676	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb.c: fix UAF of vma in hugetlb fault pathway Syzbot reports a UAF in hugetlb_fault(). This happens because vmf_anon_prepare() could drop the per-VMA lock and allow the current VMA to be freed before hugetlb_vma_unlock_read() is called. We can fix this by using a modified version of vmf_anon_prepare() that doesn't release the VMA lock on failure, and then release it ourselves after hugetlb_vma_unlock_read().	2024-10-21	7.8	High
CVE-2024-47682	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: sd: Fix off-by-one error in	2024-10-21	7.8	High

		sd_read_block_characteristics() Ff the device returns page 0xb1 with length 8 (happens with qemu v2.x, for example), sd_read_block_characteristics() may attempt an out-of-bounds memory access when accessing the zoned field at offset 8.			
CVE-2024-47691	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid use-after-free in f2fs_stop_gc_thread() syzbot reports a f2fs bug as below: __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:114 print_report+0xe8/0x550 mm/kasan/report.c:491 kasan_report+0x143/0x180 mm/kasan/report.c:601 kasan_check_range+0x282/0x290 mm/kasan/generic.c:189 instrument_atomic_read_write include/linux/instrumented.h:96 [inline] atomic_fetch_add_relaxed include/linux/atomic/atomic-instrumented.h:252 [inline] __refcount_add include/linux/refcount.h:184 [inline] __refcount_inc include/linux/refcount.h:241 [inline] refcount_inc include/linux/refcount.h:258 [inline] get_task_struct include/linux/sched/task.h:118 [inline] kthread_stop+0xca/0x630 kernel/kthread.c:704 f2fs_stop_gc_thread+0x65/0xb0 fs/f2fs/gc.c:210 f2fs_do_shutdown+0x192/0x540 fs/f2fs/file.c:2283 f2fs_ioc_shutdown fs/f2fs/file.c:2325 [inline] __f2fs_ioctl+0x443a/0xbe60 fs/f2fs/file.c:4325 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xfc/0x170 fs/ioctl.c:893 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 The root cause is below race condition, it may cause use-after-free issue in sbi->gc_th pointer. - remount - f2fs_remount - f2fs_stop_gc_thread - kfree(gc_th) - f2fs_ioc_shutdown - f2fs_do_shutdown - f2fs_stop_gc_thread - kthread_stop(gc_th->f2fs_gc_task) : sbi->gc_thread = NULL;	2024-10-21	7.8	High

		We will call f2fs_do_shutdown() in two paths: - for f2fs_ioc_shutdown() path, we should grab sb->s_umount semaphore for fixing. - for f2fs_shutdown() path, it's safe since caller has already grabbed sb->s_umount semaphore.			
[CVE-2024-47695]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs-clt: Reset cid to con_num - 1 to stay in bounds In the function init_conns(), after the create_con() and create_cm() for loop if something fails. In the cleanup for loop after the destroy tag, we access out of bound memory because cid is set to clt_path->s.con_num. This commits resets the cid to clt_path->s.con_num - 1, to stay in bounds in the cleanup loop later.	2024-10-21	7.8	High
[CVE-2024-47696]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix WARNING:at_kernel/workqueue.c:#check_flush_dependency In the commit aee2424246f9 ("RDMA/iwcm: Fix a use-after-free related to destroying CM IDs"), the function flush_workqueue is invoked to flush the work queue iwcm_wq. But at that time, the work queue iwcm_wq was created via the function alloc_ordered_workqueue without the flag WQ_MEM_RECLAIM. Because the current process is trying to flush the whole iwcm_wq, if iwcm_wq doesn't have the flag WQ_MEM_RECLAIM, verify that the current process is not reclaiming memory or running on a workqueue which doesn't have the flag WQ_MEM_RECLAIM as that can break forward-progress guarantee leading to a deadlock. The call trace is as below: [125.350876][T1430] Call Trace: [125.356281][T1430] <TASK> [125.361285][T1430] ? __warn (kernel/panic.c:693) [125.367640][T1430] ? check_flush_dependency (kernel/workqueue.c:3706 (discriminator 9)) [125.375689][T1430] ? report_bug (lib/bug.c:180 lib/bug.c:219)	2024-10-21	7.8	High

		[125.382505][T1430] ? handle_bug (arch/x86/kernel/traps.c:239) [125.388987][T1430] ? exc_invalid_op (arch/x86/kernel/traps.c:260 (discriminator 1)) [125.395831][T1430] ? asm_exc_invalid_op (arch/x86/include/asm/idtentry.h:621) [125.403125][T1430] ? check_flush_dependency (kernel/workqueue.c:3706 (discriminator 9)) [125.410984][T1430] ? check_flush_dependency (kernel/workqueue.c:3706 (discriminator 9)) [125.418764][T1430] __flush_workqueue (kernel/workqueue.c:3970) [125.426021][T1430] ? __pfx___might_resched (kernel/sched/core.c:10151) [125.433431][T1430] ? destroy_cm_id (drivers/infiniband/core/iwcm.c:375) iw_cm [125.441209][T1430] ? __pfx___flush_workqueue (kernel/workqueue.c:3910) [125.473900][T1430] ? _raw_spin_lock_irqsave (arch/x86/include/asm/atomic.h:107 include/linux/atomic/atomic-arch-fallback.h:2170 include/linux/atomic/atomic-instrumented.h:1302 include/asm-generic/qspinlock.h:111 include/linux/spinlock.h:187 include/linux/spinlock_api_smp.h:111 kernel/locking/spinlock.c:162) [125.473909][T1430] ? __pfx__raw_spin_lock_irqsave (kernel/locking/spinlock.c:161) [125.482537][T1430] _destroy_id (drivers/infiniband/core/cma.c:2044) rdma_cm [125.495072][T1430] nvme_rdma_free_queue (drivers/nvme/host/rdma.c:656 drivers/nvme/host/rdma.c:650) nvme_rdma [125.505827][T1430] nvme_rdma_reset_ctrl_work (drivers/nvme/host/rdma.c:2180) nvme_rdma [125.505831][T1430] process_one_work (kernel/workqueue.c:3231) [125.515122][T1430] worker_thread (kernel/workqueue.c:3306 kernel/workqueue.c:3393) [125.515127][T1430] ? __pfx_worker_thread (kernel/workqueue.c:3339) [125.531837][T1430] kthread (kernel/kthread.c:389) [125.539864][T1430] ? __pfx_kthread (kernel/kthread.c:342) [125.550628][T1430] ret_from_fork (arch/x86/kernel/process.c:147) [125.558840][T1430] ? __pfx_kthread (kernel/kthread.c:342) [125.558844][T1430] ret_from_fork_asm (arch/x86/entry/entry_64.S:257)			

		[125.566487][T1430] </TASK> [125.566488][T1430] ---[end trace 0000000000000000]---			
CVE-2024-47697	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2830: fix an out-of-bounds write error Ensure index in rtl2830_pid_filter does not exceed 31 to prevent out-of-bounds access. dev->filters is a 32-bit value, so set_bit and clear_bit functions should only operate on indices from 0 to 31. If index is 32, it will attempt to access a non-existent 33rd bit, leading to out-of-bounds access. Change the boundary check from index > 32 to index >= 32 to resolve this issue.	2024-10-21	7.8	High
CVE-2024-47698	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2832: fix an out-of-bounds write error Ensure index in rtl2832_pid_filter does not exceed 31 to prevent out-of-bounds access. dev->filters is a 32-bit value, so set_bit and clear_bit functions should only operate on indices from 0 to 31. If index is 32, it will attempt to access a non-existent 33rd bit, leading to out-of-bounds access. Change the boundary check from index > 32 to index >= 32 to resolve this issue. [hverkuil: added fixes tag, rtl2830_pid_filter -> rtl2832_pid_filter in logmsg]	2024-10-21	7.8	High
CVE-2024-47701	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid OOB when system.data xattr changes underneath the filesystem When looking up for an entry in an inlined directory, if e_value_offs is changed underneath the filesystem by some change in the block device, it will lead to an out-of-bounds access that KASAN detects as an UAF. EXT4-fs (loop0): mounted filesystem 00000000-0000-0000-0000-000000000000 r/w without journal. Quota mode: none. loop0: detected capacity change from 2048 to 2047	2024-10-21	7.8	High

		== BUG: KASAN: use-after-free in ext4_search_dir+0xf2/0x1c0 fs/ext4/namei.c:1500 Read of size 1 at addr ffff88803e91130f by task syz-executor269/5103 CPU: 0 UID: 0 PID: 5103 Comm: syz-executor269 Not tainted 6.11.0-rc4-syzkaller #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 ext4_search_dir+0xf2/0x1c0 fs/ext4/namei.c:1500 ext4_find_inline_entry+0x4be/0x5e0 fs/ext4/inline.c:1697 __ext4_find_entry+0x2b4/0x1b30 fs/ext4/namei.c:1573 ext4_lookup_entry fs/ext4/namei.c:1727 [inline] ext4_lookup+0x15f/0x750 fs/ext4/namei.c:1795 lookup_one_qstr_excl+0x11f/0x260 fs/namei.c:1633 filename_create+0x297/0x540 fs/namei.c:3980 do_symlinkat+0xf9/0x3a0 fs/namei.c:4587 __do_sys_symlinkat fs/namei.c:4610 [inline] __se_sys_symlinkat fs/namei.c:4607 [inline] __x64_sys_symlinkat+0x95/0xb0 fs/namei.c:4607 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: 0033:0x7f3e73ced469 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 21 18 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 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fff4d40c258 EFLAGS: 00000246 ORIG_RAX: 000000000000010a RAX: ffffffffffffffda RBX: 0032656c69662f2e RCX: 00007f3e73ced469 RDX: 0000000020000200 RSI: 00000000ffffff9c RDI: 00000000200001c0 RBP: 0000000000000000 R08: 00007fff4d40c290 R09: 00007fff4d40c290 R10: 0023706f6f6c2f76 R11: 0000000000000246 R12: 00007fff4d40c27c R13: 0000000000000003 R14: 431bde82d7b634db			

		R15: 00007fff4d40c2b0 </TASK> Calling ext4_xattr_ibody_find right after reading the inode with ext4_get_inode_loc will lead to a check of the validity of the xattrs, avoiding this problem.			
CVE-2024-47711	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: af_unix: Don't return OOB skb in manage_oob(). syzbot reported use-after-free in unix_stream_recv_urg(). [0] The scenario is 1. send(MSG_OOB) 2. recv(MSG_OOB) -> The consumed OOB remains in recv queue 3. send(MSG_OOB) 4. recv() -> manage_oob() returns the next skb of the consumed OOB -> This is also OOB, but unix_sk(sk)->oob_skb is not cleared 5. recv(MSG_OOB) -> unix_sk(sk)->oob_skb is used but already freed The recent commit 8594d9b85c07 ("af_unix: Don't call skb_get() for OOB skb.") uncovered the issue. If the OOB skb is consumed and the next skb is peeked in manage_oob(), we still need to check if the skb is OOB. Let's do so by falling back to the following checks in manage_oob() and add the test case in selftest. Note that we need to add a similar check for SIOCATMARK. [0]: BUG: KASAN: slab-use-after-free in unix_stream_read_actor+0xa6/0xb0 net/unix/af_unix.c:2959 Read of size 4 at addr ffff8880326abcc4 by task syz-executor178/5235 CPU: 0 UID: 0 PID: 5235 Comm: syz-executor178 Not tainted 6.11.0-rc5-syzkaller-00742-gfbdaffe41adc #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488	2024-10-21	7.8	High

		kasan_report+0x143/0x180 mm/kasan/report.c:601 unix_stream_read_actor+0xa6/0xb0 net/unix/af_unix.c:2959 unix_stream_recv_urg+0x1df/0x320 net/unix/af_unix.c:2640 unix_stream_read_generic+0x2456/0x2520 net/unix/af_unix.c:2778 unix_stream_recvmsg+0x22b/0x2c0 net/unix/af_unix.c:2996 sock_recvmsg_nosec net/socket.c:1046 [inline] sock_recvmsg+0x22f/0x280 net/socket.c:1068 ____sys_recvmsg+0x1db/0x470 net/socket.c:2816 ___sys_recvmsg net/socket.c:2858 [inline] __sys_recvmsg+0x2f0/0x3e0 net/socket.c:2888 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: 0033:0x7f5360d6b4e9 Code: 48 83 c4 28 c3 e8 37 17 00 00 0f 1f 80 00 00 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fff29b3a458 EFLAGS: 00000246 ORIG_RAX: 000000000000002f RAX: ffffffffffffffda RBX: 00007fff29b3a638 RCX: 00007f5360d6b4e9 RDX: 0000000000002001 RSI: 0000000020000640 RDI: 0000000000000003 RBP: 00007f5360dde610 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000001 R13: 00007fff29b3a628 R14: 0000000000000001 R15: 0000000000000001 </TASK> Allocated by task 5235: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 unpoison_slab_object mm/kasan/common.c:312 [inline] __kasan_slab_alloc+0x66/0x80 mm/kasan/common.c:338 kasan_slab_alloc include/linux/kasan.h:201 [inline] slab_post_alloc_hook mm/slub.c:3988 [inline] slab_alloc_node mm/slub.c:4037 [inline] kmem_cache_alloc_node_noprof+0x16b/0x320 mm/slub.c:4080 __alloc_skb+0x1c3/0x440 net/core/skbuff.c:667 alloc_skb include/linux/skbuff.h:1320 [inline]			

		alloc_skb_with_frags+0xc3/0x770 net/core/skbuff.c:6528 sock_alloc_send_pskb+0x91a/0xa60 net/core/sock.c:2815 sock_alloc_send_skb include/net/sock.h:1778 [inline] queue_oob+0x108/0x680 net/unix/af_unix.c:2198 unix_stream_sendmsg+0xd24/0xf80 net/unix/af_unix.c:2351 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:745 ____sys_sendmsg+0x525/0x7d0 net/socket.c:2597 ___sys_sendmsg net/socket.c:2651 [inline] __sys_sendmsg+0x2b0/0x3a0 net/socket.c:2680 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 Freed by task 5235: kasan_save_stack mm/kasan/common.c:47 ---truncated---			
CVE-2024-47718	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: always wait for both firmware loading attempts In 'rtw_wait_firmware_completion()', always wait for both (regular and wowlan) firmware loading attempts. Otherwise if 'rtw_usb_intf_init()' has failed in 'rtw_usb_probe()', 'rtw_usb_disconnect()' may issue 'ieee80211_free_hw()' when one of 'rtw_load_firmware_cb()' (usually the wowlan one) is still in progress, causing UAF detected by KASAN.	2024-10-21	7.8	High
CVE-2024-47719	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommufd: Protect against overflow of ALIGN() during iova allocation Userspace can supply an iova and uptr such that the target iova alignment becomes really big and ALIGN() overflows which corrupts the selected area range during allocation. CONFIG_IOMMUFD_TEST can detect this: WARNING: CPU: 1 PID: 5092 at drivers/iommu/iommufd/io_pagetable.c:268 iopt_alloc_area_pages drivers/iommu/iommufd/io_pagetable.c:268 [inline] WARNING: CPU: 1 PID: 5092 at drivers/iommu/iommufd/io_pagetable.c:268 iopt_map_pages+0xf95/0x1050	2024-10-21	7.8	High

		drivers/iommu/iommufd/io_pagetable.c:352 Modules linked in: CPU: 1 PID: 5092 Comm: syz-executor294 Not tainted 6.10.0-rc5-syzkaller-00294-g3ffea9a7a6f7 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 06/07/2024 RIP: 0010:iopt_alloc_area_pages drivers/iommu/iommufd/io_pagetable.c:268 [inline] RIP: 0010:iopt_map_pages+0xf95/0x1050 drivers/iommu/iommufd/io_pagetable.c:352 Code: fc e9 a4 f3 ff ff e8 1a 8b 4c fc 41 be e4 ff ff ff e9 8a f3 ff ff e8 0a 8b 4c fc 90 0f 0b 90 e9 37 f5 ff ff e8 fc 8a 4c fc 90 <0f> 0b 90 e9 68 f3 ff ff 48 c7 c1 ec 82 ad 8f 80 e1 07 80 c1 03 38 RSP: 0018:ffffc90003ebf9e0 EFLAGS: 00010293 RAX: ffffffff85499fa4 RBX: 00000000ffffffef RCX: ffff888079b49e00 RDX: 0000000000000000 RSI: 00000000ffffffef RDI: 0000000000000000 RBP: ffffc90003ebfc50 R08: ffffffff85499b30 R09: ffffffff85499942 R10: 0000000000000002 R11: ffff888079b49e00 R12: ffff8880228e0010 R13: 0000000000000000 R14: 1ffff920007d7f68 R15: ffffc90003ebfd00 FS: 000055557d760380(0000) GS:ffff8880b9500000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000005fdeb8 CR3: 000000007404a000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> iommufd_ioas_copy+0x610/0x7b0 drivers/iommu/iommufd/ioas.c:274 iommufd_fops_ioctl+0x4d9/0x5a0 drivers/iommu/iommufd/main.c:421 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xfc/0x170 fs/ioctl.c:893 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			

		Cap the automatic alignment to the huge page size, which is probably a better idea overall. Huge automatic alignments can fragment and chew up the available IOVA space without any reason.			
CVE-2024-47727	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Fix "in-kernel MMIO" check TDX only supports kernel-initiated MMIO operations. The handle_mmio() function checks if the #VE exception occurred in the kernel and rejects the operation if it did not. However, userspace can deceive the kernel into performing MMIO on its behalf. For example, if userspace can point a syscall to an MMIO address, syscall does get_user() or put_user() on it, triggering MMIO #VE. The kernel will treat the #VE as in-kernel MMIO. Ensure that the target MMIO address is within the kernel before decoding instruction.	2024-10-21	7.8	High
CVE-2024-47730	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: hisilicon/qm - inject error before stopping queue The master ooo cannot be completely closed when the accelerator core reports memory error. Therefore, the driver needs to inject the qm error to close the master ooo. Currently, the qm error is injected after stopping queue, memory may be released immediately after stopping queue, causing the device to access the released memory. Therefore, error is injected to close master ooo before stopping queue to ensure that the device does not access the released memory.	2024-10-21	7.8	High
CVE-2024-47732	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: crypto: iaa - Fix potential use after free bug The free_device_compression_mode(iaa_device, device_mode) function frees "device_mode" but it iss passed to iaa_compression_modes[i]->free() a few lines later resulting in a use after free. The good news is that, so far as I can tell, nothing implements the ->free() function and the use after free happens in dead code. But, with	2024-10-21	7.8	High

		this fix, when something does implement it, we'll be ready. :)			
CVE-2024-47742	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Block path traversal Most firmware names are hardcoded strings, or are constructed from fairly constrained format strings where the dynamic parts are just some hex numbers or such. However, there are a couple codepaths in the kernel where firmware file names contain string components that are passed through from a device or semi-privileged userspace; the ones I could find (not counting interfaces that require root privileges) are: - lpfc_sli4_request_firmware_update() seems to construct the firmware filename from "ModelName", a string that was previously parsed out of some descriptor ("Vital Product Data") in lpfc_fill_vpd() - nfp_net_fw_find() seems to construct a firmware filename from a model name coming from nfp_hwinfo_lookup(pf->hwinfo, "nffw.partno"), which I think parses some descriptor that was read from the device. (But this case likely isn't exploitable because the format string looks like "netronome/nic_%s", and there shouldn't be any *folders* starting with "netronome/nic_". The previous case was different because there, the "%s" is *at the start* of the format string.) - module_flash_fw_schedule() is reachable from the ETHTOOL_MSG_MODULE_FW_FLASH_ACT netlink command, which is marked as GENL_UNS_ADMIN_PERM (meaning CAP_NET_ADMIN inside a user namespace is enough to pass the privilege check), and takes a userspace-provided firmware name. (But I think to reach this case, you need to have CAP_NET_ADMIN over a network namespace that a special kind of ethernet	2024-10-21	7.8	High

		device is mapped into, so I think this is not a viable attack path in practice.) Fix it by rejecting any firmware names containing ".." path components. For what it's worth, I went looking and haven't found any USB device drivers that use the firmware loader dangerously.			

CVE-2024-47745	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		mm: call the security_mmap_file() LSM hook in remap_file_pages()
		
		The remap_file_pages syscall handler calls do_mmap() directly, which doesn't contain the LSM security check. And if the process has called personality(READ_IMPLIES_EXEC) before and remap_file_pages() is called for RW pages, this will actually result in remapping the pages to RWX, bypassing a W^X policy enforced by SELinux.
		
		So we should check prot by security_mmap_file LSM hook in the remap_file_pages syscall handler before do_mmap() is called. Otherwise, it potentially permits an attacker to bypass a W^X policy enforced by SELinux.
		
		The bypass is similar to CVE-2016-10044, which bypass the same thing via AIO and can be found in [1].
		
		The PoC:
		
		$ cat > test.c
		
		int main(void) {
		size_t pagesz = sysconf(_SC_PAGE_SIZE);
		int mfd = syscall(SYS_memfd_create, "test", 0);
		const char *buf = mmap(NULL, 4 * pagesz, PROT_READ | PROT_WRITE, MAP_SHARED, mfd, 0);
		unsigned int old = syscall(SYS_personality, 0xffffffff);
		syscall(SYS_personality, READ_IMPLIES_EXEC | old);
		syscall(SYS_remap_file_pages, buf, pagesz, 0, 2, 0);
		syscall(SYS_personality, old);
			2024-10-21	7.8	High

		// show the RWX page exists even if W^X policy is enforced
		int fd = open("/proc/self/maps", O_RDONLY);
		unsigned char buf2[1024];
		while (1) {
		int ret = read(fd, buf2, 1024);
		if (ret <= 0) break;
		write(1, buf2, ret);
		}
		close(fd);
		}
		
		$ gcc test.c -o test
		$ ./test | grep rwx
		7f1836c34000-7f1836c35000 rwxs 00002000 00:01 2050 /memfd:test (deleted)
		
		[PM: subject line tweaks]			

CVE-2024-47748	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vhost_vdpa: assign irq bypass producer token correctly We used to call irq_bypass_unregister_producer() in vhost_vdpa_setup_vq_irq() which is problematic as we don't know if the token pointer is still valid or not. Actually, we use the eventfd_ctx as the token so the life cycle of the token should be bound to the VHOST_SET_VRING_CALL instead of vhost_vdpa_setup_vq_irq() which could be called by set_status(). Fixing this by setting up irq bypass producer's token when handling VHOST_SET_VRING_CALL and un-registering the producer before calling vhost_vring_ioctl() to prevent a possible use after free as eventfd could have been released in vhost_vring_ioctl(). And such registering and unregistering will only be done if DRIVER_OK is set.	2024-10-21	7.8	High
CVE-2024-47750	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix Use-After-Free of rsv_qp on HIP08	2024-10-21	7.8	High

		Currently rsv_qp is freed before ib_unregister_device() is called on HIP08. During the time interval, users can still dereg MR and rsv_qp will be used in this process, leading to a UAF. Move the release of rsv_qp after calling ib_unregister_device() to fix it.			
CVE-2024-47751	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PCI: kirin: Fix buffer overflow in kirin_pcie_parse_port() Within kirin_pcie_parse_port(), the pcie->num_slots is compared to pcie->gpio_id_reset size (MAX_PCI_SLOTS) which is correct and would lead to an overflow. Thus, fix condition to pcie->num_slots + 1 >= MAX_PCI_SLOTS and move pcie->num_slots increment below the if-statement to avoid out-of-bounds array access. Found by Linux Verification Center (linuxtesting.org) with SVACE. [kwilczynski: commit log]	2024-10-21	7.8	High
CVE-2024-49852	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: elx: libefc: Fix potential use after free in efc_nport_vport_del() The kref_put() function will call nport->release if the refcount drops to zero. The nport->release release function is _efc_nport_free() which frees "nport". But then we dereference "nport" on the next line which is a use after free. Re-order these lines to avoid the use after free.	2024-10-21	7.8	High
CVE-2024-49853	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix double free in OPTEE transport Channels can be shared between protocols, avoid	2024-10-21	7.8	High

		freeing the same channel descriptors twice when unloading the stack.			
CVE-2024-49854	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix uaf for accessing waker_bfqq after splitting After commit 42c306ed7233 ("block, bfq: don't break merge chain in bfq_split_bfqq()"), if the current procress is the last holder of bfqq, the bfqq can be freed after bfq_split_bfqq(). Hence recored the bfqq and then access bfqq->waker_bfqq may trigger UAF. What's more, the waker_bfqq may in the merge chain of bfqq, hence just recored waker_bfqq is still not safe. Fix the problem by adding a helper bfq_waker_bfqq() to check if bfqq->waker_bfqq is in the merge chain, and current procress is the only holder.	2024-10-21	7.8	High
CVE-2024-49865	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vm: move xa_alloc to prevent UAF Evil user can guess the next id of the vm before the ioctl completes and then call vm destroy ioctl to trigger UAF since create ioctl is still referencing the same vm. Move the xa_alloc all the way to the end to prevent this. v2: - Rebase (cherry picked from commit dcfd3971327f3ee92765154baebbaece833d3ca9)	2024-10-21	7.8	High
CVE-2024-49869	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: btrfs: send: fix buffer overflow detection when copying path to cache entry Starting with commit c0247d289e73 ("btrfs: send: annotate struct	2024-10-21	7.8	High

		name_cache_entry with __counted_by()") we annotated the variable length array "name" from the name_cache_entry structure with __counted_by() to improve overflow detection. However that alone was not correct, because the length of that array does not match the "name_len" field - it matches that plus 1 to include the NUL string terminator, so that makes a fortified kernel think there's an overflow and report a splat like this: strcpy: detected buffer overflow: 20 byte write of buffer size 19 WARNING: CPU: 3 PID: 3310 at __fortify_report+0x45/0x50 CPU: 3 UID: 0 PID: 3310 Comm: btrfs Not tainted 6.11.0-prnet #1 Hardware name: CompuLab Ltd. sbc-ihsw/Intense-PC2 (IPC2), BIOS IPC2_3.330.7 X64 03/15/2018 RIP: 0010:__fortify_report+0x45/0x50 Code: 48 8b 34 (...) RSP: 0018:ffff97ebc0d6f650 EFLAGS: 00010246 RAX: 7749924ef60fa600 RBX: ffff8bf5446a521a RCX: 0000000000000027 RDX: 00000000ffffdfff RSI: ffff97ebc0d6f548 RDI: ffff8bf84e7a1cc8 RBP: ffff8bf548574080 R08: ffffffffa8c40e10 R09: 0000000000005ffd R10: 0000000000000004 R11: ffffffffa8c70e10 R12: ffff8bf551eef400 R13: 0000000000000000 R14: 0000000000000013 R15: 00000000000003a8 FS: 00007fae144de8c0(0000) GS:ffff8bf84e780000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fae14691690 CR3: 00000001027a2003 CR4: 00000000001706f0 Call Trace: <TASK> ? __warn+0x12a/0x1d0 ? __fortify_report+0x45/0x50 ? report_bug+0x154/0x1c0 ? handle_bug+0x42/0x70 ? exc_invalid_op+0x1a/0x50 ? asm_exc_invalid_op+0x1a/0x20 ? __fortify_report+0x45/0x50 __fortify_panic+0x9/0x10			

		__get_cur_name_and_parent+0x3bc/0x3c0 get_cur_path+0x207/0x3b0 send_extent_data+0x709/0x10d0 ? find_parent_nodes+0x22df/0x25d0 ? mas_nomem+0x13/0x90 ? mtree_insert_range+0xa5/0x110 ? btrfs_lru_cache_store+0x5f/0x1e0 ? iterate_extent_inodes+0x52d/0x5a0 process_extent+0xa96/0x11a0 ? __pfx_lookup_backref_cache+0x10/0x10 ? __pfx_store_backref_cache+0x10/0x10 ? __pfx_iterate_backrefs+0x10/0x10 ? __pfx_check_extent_item+0x10/0x10 changed_cb+0x6fa/0x930 ? tree_advance+0x362/0x390 ? memcmp_extent_buffer+0xd7/0x160 send_subvol+0xf0a/0x1520 btrfs_ioctl_send+0x106b/0x11d0 ? __pfx___clone_root_cmp_sort+0x10/0x10 _btrfs_ioctl_send+0x1ac/0x240 btrfs_ioctl+0x75b/0x850 __se_sys_ioctl+0xca/0x150 do_syscall_64+0x85/0x160 ? __count_memcg_events+0x69/0x100 ? handle_mm_fault+0x1327/0x15c0 ? __se_sys_rt_sigprocmask+0xf1/0x180 ? syscall_exit_to_user_mode+0x75/0xa0 ? do_syscall_64+0x91/0x160 ? do_user_addr_fault+0x21d/0x630 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7fae145eeb4f Code: 00 48 89 (...) RSP: 002b:00007ffdf1cb09b0 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffffffffffda RBX: 0000000000000004 RCX: 00007fae145eeb4f RDX: 00007ffdf1cb0ad0 RSI: 0000000040489426 RDI: 0000000000000004 RBP: 00000000000078fe R08: 00007fae144006c0 R09: 00007ffdf1cb0927 R10: 0000000000000008 R11: 0000000000000246 R12: 00007ffdf1cb1ce8 R13: 0000000000000003 R14: 000055c499fab2e0 R15: 0000000000000004 </TASK> Fix this by not storing the NUL string terminator since we don't actually need it for name cache entries, this way "name_len" corresponds to the actual size of the "name" array. This requires marking				

		the "name" array field with __nonstring and using memcpy() instead of strcpy() as recommended by the guidelines at: https://github.com/KSPP/linux/issues/90			
CVE-2024-49876	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe: fix UAF around queue destruction We currently do stuff like queuing the final destruction step on a random system wq, which will outlive the driver instance. With bad timing we can teardown the driver with one or more work workqueue still being alive leading to various UAF splats. Add a fini step to ensure user queues are properly torn down. At this point GuC should already be nuked so queue itself should no longer be referenced from hw pov. v2 (Matt B) - Looks much safer to use a waitqueue and then just wait for the xa_array to become empty before triggering the drain. (cherry picked from commit 861108666cc0e999cffeab6aff17b662e68774e3)	2024-10-21	7.8	High
CVE-2024-49880	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: fix off by one issue in alloc_flex_gd() Wesley reported an issue: == EXT4-fs (dm-5): resizing filesystem from 7168 to 786432 blocks ------------[cut here]------------ kernel BUG at fs/ext4/resize.c:324! CPU: 9 UID: 0 PID: 3576 Comm: resize2fs Not tainted 6.11.0+ #27 RIP: 0010:ext4_resize_fs+0x1212/0x12d0 Call Trace: __ext4_ioctl+0x4e0/0x1800 ext4_ioctl+0x12/0x20 __x64_sys_ioctl+0x99/0xd0	2024-10-21	7.8	High

<table>
<tr>
<td></td>
<td></td>
<td> x64_sys_call+0x1206/0x20d0
 do_syscall_64+0x72/0x110
 entry_SYSCALL_64_after_hwframe+0x76/0x7e
===

While reviewing the patch, Honza found that when adjusting resize_bg in
alloc_flex_gd(), it was possible for flex_gd->resize_bg to be bigger than
flexbg_size.

The reproduction of the problem requires the following:

 o_group = flexbg_size * 2 * n;
 o_size = (o_group + 1) * group_size;
 n_group: [o_group + flexbg_size, o_group + flexbg_size * 2)
 o_size = (n_group + 1) * group_size;

Take n=0,flexbg_size=16 as an example:

             last:15
|o---------------|--------------n-|
o_group:0   resize to   n_group:30

The corresponding reproducer is:

img=test.img
rm -f $img
truncate -s 600M $img
mkfs.ext4 -F $img -b 1024 -G 16 8M
dev=`losetup -f --show $img`
mkdir -p /tmp/test
mount $dev /tmp/test
resize2fs $dev 248M

Delete the problematic plus 1 to fix the issue, and add a WARN_ON_ONCE()
to prevent the issue from happening again.

[Note: another reproucer which this commit fixes is:

  img=test.img
  rm -f $img
  truncate -s 25MiB $img
  mkfs.ext4 -b 4096 -E nodiscard,lazy_itable_init=0,lazy_journal_init=0 $img
  truncate -s 3GiB $img
  dev=`losetup -f --show $img`
  mkdir -p /tmp/test</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

		mount $dev /tmp/test resize2fs $dev 3G umount $dev losetup -d $dev -- TYT]			

CVE-2024-49882	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ext4: fix double brelse() the buffer of the extents path
		
		In ext4_ext_try_to_merge_up(), set path[1].p_bh to NULL after it has been
		released, otherwise it may be released twice. An example of what triggers
		this is as follows:
		
		  split2 map split1
		|--------|-------|--------|
		
		ext4_ext_map_blocks
		 ext4_ext_handle_unwritten_extents
		  ext4_split_convert_extents
		   // path->p_depth == 0
		   ext4_split_extent
		     // 1. do split1
		     ext4_split_extent_at
		       |ext4_ext_insert_extent
		       | ext4_ext_create_new_leaf
		       | ext4_ext_grow_indepth
		       | le16_add_cpu(&neh->eh_depth, 1)
		       | ext4_find_extent
		       | // return -ENOMEM
		       |// get error and try zeroout
		       |path = ext4_find_extent
		       | path->p_depth = 1
		       |ext4_ext_try_to_merge
		       | ext4_ext_try_to_merge_up
		       | path->p_depth = 0
		       | brelse(path[1].p_bh) ---> not set to NULL here
		       |// zeroout success
		     // 2. update path
		     ext4_find_extent
		     // 3. do split2
		     ext4_split_extent_at
		       ext4_ext_insert_extent
		         ext4_ext_create_new_leaf
		           ext4_ext_grow_indepth
		             le16_add_cpu(&neh->eh_depth, 1)
		           ext4_find_extent
		             path[0].p_bh = NULL;
			2024-10-21	7.8	High

		path->p_depth = 1
		read_extent_tree_block ---> return err
		// path[1].p_bh is still the old value
		ext4_free_ext_path
		ext4_ext_drop_refs
		// path->p_depth == 1
		brelse(path[1].p_bh) ---> brelse a buffer twice
		
		Finally got the following WARRNING when removing the buffer from lru:
		
		==
		VFS: brelse: Trying to free free buffer
		WARNING: CPU: 2 PID: 72 at fs/buffer.c:1241 __brelse+0x58/0x90
		CPU: 2 PID: 72 Comm: kworker/u19:1 Not tainted 6.9.0-dirty #716
		RIP: 0010:__brelse+0x58/0x90
		Call Trace:
		<TASK>
		__find_get_block+0x6e7/0x810
		bdev_getblk+0x2b/0x480
		__ext4_get_inode_loc+0x48a/0x1240
		ext4_get_inode_loc+0xb2/0x150
		ext4_reserve_inode_write+0xb7/0x230
		__ext4_mark_inode_dirty+0x144/0x6a0
		ext4_ext_insert_extent+0x9c8/0x3230
		ext4_ext_map_blocks+0xf45/0x2dc0
		ext4_map_blocks+0x724/0x1700
		ext4_do_writepages+0x12d6/0x2a70
		[...]
		==			

CVE-2024-49883	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: aovid use-after-free in ext4_ext_insert_extent() As Ojaswin mentioned in Link, in ext4_ext_insert_extent(), if the path is reallocated in ext4_ext_create_new_leaf(), we'll use the stale path and cause UAF. Below is a sample trace with dummy values: ext4_ext_insert_extent path = *ppath = 2000 ext4_ext_create_new_leaf(ppath) ext4_find_extent(ppath) path = *ppath = 2000 if (depth > path[0].p_maxdepth)	2024-10-21	7.8	High

		kfree(path = 2000);
		*ppath = path = NULL;
		path = kcalloc() = 3000
		*ppath = 3000;
		return path;
		/* here path is still 2000, UAF! */
		eh = path[depth].p_hdr
		
		==
		BUG: KASAN: slab-use-after-free in ext4_ext_insert_extent+0x26d4/0x3330
		Read of size 8 at addr ffff8881027bf7d0 by task kworker/u36:1/179
		CPU: 3 UID: 0 PID: 179 Comm: kworker/u6:1 Not tainted 6.11.0-rc2-dirty #866
		Call Trace:
		<TASK>
		ext4_ext_insert_extent+0x26d4/0x3330
		ext4_ext_map_blocks+0xe22/0x2d40
		ext4_map_blocks+0x71e/0x1700
		ext4_do_writepages+0x1290/0x2800
		[...]
		
		Allocated by task 179:
		ext4_find_extent+0x81c/0x1f70
		ext4_ext_map_blocks+0x146/0x2d40
		ext4_map_blocks+0x71e/0x1700
		ext4_do_writepages+0x1290/0x2800
		ext4_writepages+0x26d/0x4e0
		do_writepages+0x175/0x700
		[...]
		
		Freed by task 179:
		kfree+0xcb/0x240
		ext4_find_extent+0x7c0/0x1f70
		ext4_ext_insert_extent+0xa26/0x3330
		ext4_ext_map_blocks+0xe22/0x2d40
		ext4_map_blocks+0x71e/0x1700
		ext4_do_writepages+0x1290/0x2800
		ext4_writepages+0x26d/0x4e0
		do_writepages+0x175/0x700
		[...]
		==
		
		So use *ppath to update the path to avoid the above problem.			

CVE-2024-49884	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: fix slab-use-after-free in ext4_split_extent_at()	2024-10-21	7.8	High

		We hit the following use-after-free: == BUG: KASAN: slab-use-after-free in ext4_split_extent_at+0xba8/0xcc0 Read of size 2 at addr ffff88810548ed08 by task kworker/u20:0/40 CPU: 0 PID: 40 Comm: kworker/u20:0 Not tainted 6.9.0-dirty #724 Call Trace: <TASK> kasan_report+0x93/0xc0 ext4_split_extent_at+0xba8/0xcc0 ext4_split_extent.isra.0+0x18f/0x500 ext4_split_convert_extents+0x275/0x750 ext4_ext_handle_unwritten_extents+0x73e/0x1580 ext4_ext_map_blocks+0xe20/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] Allocated by task 40: __kmalloc_noprof+0x1ac/0x480 ext4_find_extent+0xf3b/0x1e70 ext4_ext_map_blocks+0x188/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] Freed by task 40: kfree+0xf1/0x2b0 ext4_find_extent+0xa71/0x1e70 ext4_ext_insert_extent+0xa22/0x3260 ext4_split_extent_at+0x3ef/0xcc0 ext4_split_extent.isra.0+0x18f/0x500 ext4_split_convert_extents+0x275/0x750 ext4_ext_handle_unwritten_extents+0x73e/0x1580 ext4_ext_map_blocks+0xe20/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] == The flow of issue triggering is as follows: ext4_split_extent_at path = *ppath ext4_ext_insert_extent(ppath) ext4_ext_create_new_leaf(ppath) ext4_find_extent(orig_path) path = *orig_path			

		read_extent_tree_block
		// return -ENOMEM or -EIO
		ext4_free_ext_path(path)
		kfree(path)
		*orig_path = NULL
		a. If err is -ENOMEM:
		ext4_ext_dirty(path + path->p_depth)
		// path use-after-free !!!
		b. If err is -EIO and we have EXT_DEBUG defined:
		ext4_ext_show_leaf(path)
		eh = path[depth].p_hdr
		// path also use-after-free !!!
		
		So when trying to zeroout or fix the extent length, call ext4_find_extent()
		to update the path.
		
		In addition we use *ppath directly as an ext4_ext_show_leaf() input to
		avoid possible use-after-free when EXT_DEBUG is defined, and to avoid
		unnecessary path updates.			

CVE-2024-49889	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ext4: avoid use-after-free in ext4_ext_show_leaf()
		
		In ext4_find_extent(), path may be freed by error or be reallocated, so
		using a previously saved *ppath may have been freed and thus may trigger
		use-after-free, as follows:
		
		ext4_split_extent
		path = *ppath;
		ext4_split_extent_at(ppath)
		path = ext4_find_extent(ppath)
		ext4_split_extent_at(ppath)
		// ext4_find_extent fails to free path
		// but zeroout succeeds
		ext4_ext_show_leaf(inode, path)
		eh = path[depth].p_hdr
		// path use-after-free !!!
		
		Similar to ext4_split_extent_at(), we use *ppath directly as an input to
		ext4_ext_show_leaf(). Fix a spelling error by the way.
		
		Same problem in ext4_ext_handle_unwritten_extents(). Since 'path' is only
			2024-10-21	7.8	High

		used in ext4_ext_show_leaf(), remove 'path' and use *ppath directly. This issue is triggered only when EXT_DEBUG is defined and therefore does not affect functionality.			
CVE-2024-49894	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in degamma hardware format translation Fixes index out of bounds issue in `cm_helper_translate_curve_to_degamma_hw_format` function. The issue could occur when the index 'i' exceeds the number of transfer function points (TRANSFER_FUNC_POINTS). The fix adds a check to ensure 'i' is within bounds before accessing the transfer function points. If 'i' is out of bounds the function returns false to indicate an error. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/dcn10/dcn10_cm_common.c:594 cm_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.red' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn10/dcn10_cm_common.c:595 cm_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.green' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn10/dcn10_cm_common.c:596 cm_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.blue' 1025 <= s32max	2024-10-21	7.8	High
CVE-2024-49895	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in DCN30 degamma hardware format translation This commit addresses a potential index out of bounds issue in the `cm3_helper_translate_curve_to_degamma_hw_format` function in the DCN30	2024-10-21	7.8	High

		color management module. The issue could occur when the index 'i' exceeds the number of transfer function points (TRANSFER_FUNC_POINTS). The fix adds a check to ensure 'i' is within bounds before accessing the transfer function points. If 'i' is out of bounds, the function returns false to indicate an error. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:338 cm3_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.red' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:339 cm3_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.green' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:340 cm3_helper_translate_curve_to_degamma_hw_format() error: buffer overflow 'output_tf->tf_pts.blue' 1025 <= s32max			
CVE-2024-49924	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fbdev: pxafb: Fix possible use after free in pxafb_task() In the pxafb_probe function, it calls the pxafb_init_fbinfo function, after which &fbi->task is associated with pxafb_task. Moreover, within this pxafb_init_fbinfo function, the pxafb_blank function within the &pxafb_ops struct is capable of scheduling work. If we remove the module which will call pxafb_remove to make cleanup, it will call unregister_framebuffer function which can call do_unregister_framebuffer to free fbi->fb through put_fb_info(fb_info), while the work mentioned above will be used. The sequence of operations that may lead to a UAF bug is as follows:	2024-10-21	7.8	High

		CPU0 CPU1 | pxafb_task pxafb_remove | unregister_framebuffer(info) | do_unregister_framebuffer(fb_info) | put_fb_info(fb_info) | // free fbi->fb | set_ctrlr_state(fbi, state) | __pxafb_lcd_power(fbi, 0) | fbi->lcd_power(on, &fbi->fb.var) | //use fbi->fb Fix it by ensuring that the work is canceled before proceeding with the cleanup in pxafb_remove. Note that only root user can remove the driver at runtime.			
CVE-2024-49930	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix array out-of-bound access in SoC stats Currently, the ath11k_soc_dp_stats::hal_reo_error array is defined with a maximum size of DP_REO_DST_RING_MAX. However, the ath11k_dp_process_rx() function access ath11k_soc_dp_stats::hal_reo_error using the REO destination SRNG ring ID, which is incorrect. SRNG ring ID differ from normal ring ID, and this usage leads to out-of-bounds array access. To fix this issue, modify ath11k_dp_process_rx() to use the normal ring ID directly instead of the SRNG ring ID to avoid out-of-bounds array access. Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKSWPL_SILICONZ-1	2024-10-21	7.8	High
CVE-2024-49931	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix array out-of-bound access in SoC stats Currently, the ath12k_soc_dp_stats::hal_reo_error array is defined with a maximum size of DP_REO_DST_RING_MAX. However,	2024-10-21	7.8	High

		the ath12k_dp_rx_process() function access ath12k_soc_dp_stats::hal_reo_error using the REO destination SRNG ring ID, which is incorrect. SRNG ring ID differ from normal ring ID, and this usage leads to out-of-bounds array access. To fix this issue, modify ath12k_dp_rx_process() to use the normal ring ID directly instead of the SRNG ring ID to avoid out-of-bounds array access. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.0.1-00029-QCAHKSWPL_SILICONZ-1			
CVE-2024-49936	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/xen-netback: prevent UAF in xenvif_flush_hash() During the list_for_each_entry_rcu iteration call of xenvif_flush_hash, kfree_rcu does not exist inside the rcu read critical section, so if kfree_rcu is called when the rcu grace period ends during the iteration, UAF occurs when accessing head->next after the entry becomes free. Therefore, to solve this, you need to change it to list_for_each_entry_safe.	2024-10-21	7.8	High
CVE-2024-49950	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix uaf in l2cap_connect [Syzbot reported] BUG: KASAN: slab-use-after-free in l2cap_connect.constprop.0+0x10d8/0x1270 net/bluetooth/l2cap_core.c:3949 Read of size 8 at addr ffff8880241e9800 by task kworker/u9:0/54 CPU: 0 UID: 0 PID: 54 Comm: kworker/u9:0 Not tainted 6.11.0-rc6-syzkaller-00268-g788220eee30d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Workqueue: hci2 hci_rx_work Call Trace: <TASK>	2024-10-21	7.8	High

		__dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:119 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 l2cap_connect.constprop.0+0x10d8/0x1270 net/bluetooth/l2cap_core.c:3949 l2cap_connect_req net/bluetooth/l2cap_core.c:4080 [inline] l2cap_bredr_sig_cmd net/bluetooth/l2cap_core.c:4772 [inline] l2cap_sig_channel net/bluetooth/l2cap_core.c:5543 [inline] l2cap_recv_frame+0xf0b/0x8eb0 net/bluetooth/l2cap_core.c:6825 l2cap_recv_acldata+0x9b4/0xb70 net/bluetooth/l2cap_core.c:7514 hci_acldata_packet net/bluetooth/hci_core.c:3791 [inline] hci_rx_work+0xaab/0x1610 net/bluetooth/hci_core.c:4028 process_one_work+0x9c5/0x1b40 kernel/workqueue.c:3231 process_scheduled_works kernel/workqueue.c:3312 [inline] worker_thread+0x6c8/0xed0 kernel/workqueue.c:3389 kthread+0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 ... Freed by task 5245: kasan_save_stack+0x33/0x60 mm/kasan/common.c:47 kasan_save_track+0x14/0x30 mm/kasan/common.c:68 kasan_save_free_info+0x3b/0x60 mm/kasan/generic.c:579 poison_slab_object+0xf7/0x160 mm/kasan/common.c:240 __kasan_slab_free+0x32/0x50 mm/kasan/common.c:256 kasan_slab_free include/linux/kasan.h:184 [inline] slab_free_hook mm/slub.c:2256 [inline] slab_free mm/slub.c:4477 [inline] kfree+0x12a/0x3b0 mm/slub.c:4598 l2cap_conn_free net/bluetooth/l2cap_core.c:1810			

		[inline] kref_put include/linux/kref.h:65 [inline] l2cap_conn_put net/bluetooth/l2cap_core.c:1822 [inline] l2cap_conn_del+0x59d/0x730 net/bluetooth/l2cap_core.c:1802 l2cap_connect_cfm+0x9e6/0xf80 net/bluetooth/l2cap_core.c:7241 hci_connect_cfm include/net/bluetooth/hci_core.h:1960 [inline] hci_conn_failed+0x1c3/0x370 net/bluetooth/hci_conn.c:1265 hci_abort_conn_sync+0x75a/0xb50 net/bluetooth/hci_sync.c:5583 abort_conn_sync+0x197/0x360 net/bluetooth/hci_conn.c:2917 hci_cmd_sync_work+0x1a4/0x410 net/bluetooth/hci_sync.c:328 process_one_work+0x9c5/0x1b40 kernel/workqueue.c:3231 process_scheduled_works kernel/workqueue.c:3312 [inline] worker_thread+0x6c8/0xed0 kernel/workqueue.c:3389 kthread+0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244			
CVE-2024-49960	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: fix timer use-after-free on failed mount Syzbot has found an ODEBUG bug in ext4_fill_super The del_timer_sync function cancels the s_err_report timer, which reminds about filesystem errors daily. We should guarantee the timer is no longer active before kfree(sbi). When filesystem mounting fails, the flow goes to failed_mount3, where an error occurs when ext4_stop_mmpd is called, causing a read I/O failure. This triggers the ext4_handle_error function that ultimately re-arms the timer, leaving the s_err_report timer active before	2024-10-21	7.8	High

		kfree(sbi) is called. Fix the issue by canceling the s_err_report timer after calling ext4_stop_mmpd.			
CVE-2024-49967	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: no need to continue when the number of entries is 1	2024-10-21	7.8	High
CVE-2024-49969	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in DCN30 color transformation This commit addresses a potential index out of bounds issue in the `cm3_helper_translate_curve_to_hw_format` function in the DCN30 color management module. The issue could occur when the index 'i' exceeds the number of transfer function points (TRANSFER_FUNC_POINTS). The fix adds a check to ensure 'i' is within bounds before accessing the transfer function points. If 'i' is out of bounds, the function returns false to indicate an error. drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:180 cm3_helper_translate_curve_to_hw_format() error: buffer overflow 'output_tf->tf_pts.red' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:181 cm3_helper_translate_curve_to_hw_format() error: buffer overflow 'output_tf->tf_pts.green' 1025 <= s32max drivers/gpu/drm/amd/amdgpu/../display/dc/dcn30/dcn30_cm_common.c:182 cm3_helper_translate_curve_to_hw_format() error: buffer overflow 'output_tf->tf_pts.blue' 1025 <= s32max	2024-10-21	7.8	High
CVE-2024-49982	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: aoe: fix the potential use-after-free problem in more places	2024-10-21	7.8	High

		For fixing CVE-2023-6270, f98364e92662 ("aoe: fix the potential use-after-free problem in aoecmd_cfg_pkts") makes tx() calling dev_put() instead of doing in aoecmd_cfg_pkts(). It avoids that the tx() runs into use-after-free. Then Nicolai Stange found more places in aoe have potential use-after-free problem with tx(). e.g. revalidate(), aoecmd_ata_rw(), resend(), probe() and aoecmd_cfg_rsp(). Those functions also use aoenet_xmit() to push packet to tx queue. So they should also use dev_hold() to increase the refcnt of skb->dev. On the other hand, moving dev_put() to tx() causes that the refcnt of skb->dev be reduced to a negative value, because corresponding dev_hold() are not called in revalidate(), aoecmd_ata_rw(), resend(), probe(), and aoecmd_cfg_rsp(). This patch fixed this issue.			
[CVE-2024-49983](https://nvd.nist.gov/vuln/detail/CVE-2024-49983)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: drop ppath from ext4_ext_replay_update_ex() to avoid double-free When calling ext4_force_split_extent_at() in ext4_ext_replay_update_ex(), the 'ppath' is updated but it is the 'path' that is freed, thus potentially triggering a double-free in the following process: ext4_ext_replay_update_ex ppath = path ext4_force_split_extent_at(&ppath) ext4_split_extent_at ext4_ext_insert_extent ext4_ext_create_new_leaf ext4_ext_grow_indepth ext4_find_extent if (depth > path[0].p_maxdepth) kfree(path) ---> path First freed *orig_path = path = NULL ---> null ppath kfree(path) ---> path double-free !!!	2024-10-21	7.8	High

		So drop the unnecessary ppath and use path directly to avoid this problem. And use ext4_find_extent() directly to update path, avoiding unnecessary memory allocation and freeing. Also, propagate the error returned by ext4_find_extent() instead of using strange error codes.			
[CVE-2024-49984](https://nvd.nist.gov/vuln/detail/CVE-2024-49984)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Prevent out of bounds access in performance query extensions Check that the number of perfmons userspace is passing in the copy and reset extensions is not greater than the internal kernel storage where the ids will be copied into.	2024-10-21	7.8	High
[CVE-2024-49986](https://nvd.nist.gov/vuln/detail/CVE-2024-49986)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: platform/x86: x86-android-tablets: Fix use after free on platform_device_register() errors x86_android_tablet_remove() frees the pdevs[] array, so it should not be used after calling x86_android_tablet_remove(). When platform_device_register() fails, store the pdevs[x] PTR_ERR() value into the local ret variable before calling x86_android_tablet_remove() to avoid using pdevs[] after it has been freed.	2024-10-21	7.8	High
[CVE-2024-49989](https://nvd.nist.gov/vuln/detail/CVE-2024-49989)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix double free issue during amdgpu module unload Flexible endpoints use DIGs from available inflexible endpoints, so only the encoders of inflexible links need to be freed. Otherwise, a double free issue may occur when unloading the amdgpu module. [279.190523] RIP: 0010:__slab_free+0x152/0x2f0 [279.190577] Call Trace: [279.190580] <TASK>	2024-10-21	7.8	High

		[279.190582] ? show_regs+0x69/0x80 [279.190590] ? die+0x3b/0x90 [279.190595] ? do_trap+0xc8/0xe0 [279.190601] ? do_error_trap+0x73/0xa0 [279.190605] ? __slab_free+0x152/0x2f0 [279.190609] ? exc_invalid_op+0x56/0x70 [279.190616] ? __slab_free+0x152/0x2f0 [279.190642] ? asm_exc_invalid_op+0x1f/0x30 [279.190648] ? dcn10_link_encoder_destroy+0x19/0x30 [amdgpu] [279.191096] ? __slab_free+0x152/0x2f0 [279.191102] ? dcn10_link_encoder_destroy+0x19/0x30 [amdgpu] [279.191469] kfree+0x260/0x2b0 [279.191474] dcn10_link_encoder_destroy+0x19/0x30 [amdgpu] [279.191821] link_destroy+0xd7/0x130 [amdgpu] [279.192248] dc_destruct+0x90/0x270 [amdgpu] [279.192666] dc_destroy+0x19/0x40 [amdgpu] [279.193020] amdgpu_dm_fini+0x16e/0x200 [amdgpu] [279.193432] dm_hw_fini+0x26/0x40 [amdgpu] [279.193795] amdgpu_device_fini_hw+0x24c/0x400 [amdgpu] [279.194108] amdgpu_driver_unload_kms+0x4f/0x70 [amdgpu] [279.194436] amdgpu_pci_remove+0x40/0x80 [amdgpu] [279.194632] pci_device_remove+0x3a/0xa0 [279.194638] device_remove+0x40/0x70 [279.194642] device_release_driver_internal+0x1ad/0x210 [279.194647] driver_detach+0x4e/0xa0 [279.194650] bus_remove_driver+0x6f/0xf0 [279.194653] driver_unregister+0x33/0x60 [279.194657] pci_unregister_driver+0x44/0x90 [279.194662] amdgpu_exit+0x19/0x1f0 [amdgpu] [279.194939] __do_sys_delete_module.isra.0+0x198/0x2f0 [279.194946] __x64_sys_delete_module+0x16/0x20 [279.194950] do_syscall_64+0x58/0x120 [279.194954] entry_SYSCALL_64_after_hwframe+0x6e/0x76 [279.194980] </TASK>			
CVE-2024-49991	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: amdkfd_free_gtt_mem clear the correct pointer	2024-10-21	7.8	High

		Pass pointer reference to amdgpu_bo_unref to clear the correct pointer, otherwise amdgpu_bo_unref clear the local variable, the original pointer not set to NULL, this could cause use-after-free bug.			
CVE-2024-49992	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/stm: Avoid use-after-free issues with crtc and plane ltdc_load() calls functions drm_crtc_init_with_planes(), drm_universal_plane_init() and drm_encoder_init(). These functions should not be called with parameters allocated with devm_kzalloc() to avoid use-after-free issues [1]. Use allocations managed by the DRM framework. Found by Linux Verification Center (linuxtesting.org). [1] https://lore.kernel.org/lkml/u366i76e3qhh3ra5oxrtngjtm2u5lterkekcz6y2jkndhuxzli@diujon4h7qwb/	2024-10-21	7.8	High
CVE-2024-49995	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tipc: guard against string buffer overrun Smatch reports that copying media_name and if_name to name_parts may overwrite the destination. .../bearer.c:166 bearer_name_validate() error: strcpy() 'media_name' too large for 'name_parts->media_name' (32 vs 16) .../bearer.c:167 bearer_name_validate() error: strcpy() 'if_name' too large for 'name_parts->if_name' (1010102 vs 16) This does seem to be the case so guard against this possibility by using strscpy() and failing if truncation occurs. Introduced by commit b97bf3fd8f6a ("[TIPC] Initial merge") Compile tested only.	2024-10-21	7.8	High

CVE-2024-49996	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix buffer overflow when parsing NFS reparse points ReparseDataLength is sum of the InodeType size and DataBuffer size. So to get DataBuffer size it is needed to subtract InodeType's size from ReparseDataLength. Function cifs_strndup_from_utf16() is currentlly accessing buf->DataBuffer at position after the end of the buffer because it does not subtract InodeType size from the length. Fix this problem and correctly subtract variable len. Member InodeType is present only when reparse buffer is large enough. Check for ReparseDataLength before accessing InodeType to prevent another invalid memory access. Major and minor rdev values are present also only when reparse buffer is large enough. Check for reparse buffer size before calling reparse_mkdev().	2024-10-21	7.8	High
CVE-2024-50007	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ALSA: asihpi: Fix potential OOB array access ASIHPI driver stores some values in the static array upon a response from the driver, and its index depends on the firmware. We shouldn't trust it blindly. This patch adds a sanity check of the array index to fit in the array size.	2024-10-21	7.8	High
CVE-2022-48948	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: uvc: Prevent buffer overflow in setup handler Setup function uvc_function_setup permits control	2024-10-21	7.8	High

		transfer requests with up to 64 bytes of payload (UVC_MAX_REQUEST_SIZE), data stage handler for OUT transfer uses memcpy to copy req->actual bytes to uvc_event->data.data array of size 60. This may result in an overflow of 4 bytes.			
CVE-2022-48950	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: perf: Fix perf_pending_task() UaF Per syzbot it is possible for perf_pending_task() to run after the event is free()'d. There are two related but distinct cases: - the task_work was already queued before destroying the event; - destroying the event itself queues the task_work. The first cannot be solved using task_work_cancel() since perf_release() itself might be called from a task_work (_____fput), which means the current->task_works list is already empty and task_work_cancel() won't be able to find the perf_pending_task() entry. The simplest alternative is extending the perf_event lifetime to cover the task_work. The second is just silly, queueing a task_work while you know the event is going away makes no sense and is easily avoided by re-arranging how the event is marked STATE_DEAD and ensuring it goes through STATE_OFF on the way down.	2024-10-21	7.8	High
CVE-2022-48951	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Check bounds for second channel in snd_soc_put_volsw_sx() The bounds checks in snd_soc_put_volsw_sx() are only being applied to the	2024-10-21	7.8	High

		first channel, meaning it is possible to write out of bounds values to the second channel in stereo controls. Add appropriate checks.			
CVE-2022-48954	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: s390/qeth: fix use-after-free in hsci KASAN found that addr was dereferenced after br2dev_event_work was freed. == BUG: KASAN: use-after-free in qeth_l2_br2dev_worker+0x5ba/0x6b0 Read of size 1 at addr 00000000fdcea440 by task kworker/u760:4/540 CPU: 17 PID: 540 Comm: kworker/u760:4 Tainted: G E 6.1.0-20221128.rc7.git1.5aa3bed4ce83.300.fc36.s390x+kasan #1 Hardware name: IBM 8561 T01 703 (LPAR) Workqueue: 0.0.8000_event qeth_l2_br2dev_worker Call Trace: [<000000016944d4ce>] dump_stack_lvl+0xc6/0xf8 [<000000016942cd9c>] print_address_description.constprop.0+0x34/0x2a0 [<000000016942d118>] print_report+0x110/0x1f8 [<0000000167a7bd04>] kasan_report+0xfc/0x128 [<000000016938d79a>] qeth_l2_br2dev_worker+0x5ba/0x6b0 [<00000001673edd1e>] process_one_work+0x76e/0x1128 [<00000001673ee85c>] worker_thread+0x184/0x1098 [<000000016740718a>] kthread+0x26a/0x310 [<00000001672c606a>] __ret_from_fork+0x8a/0xe8 [<00000001694711da>] ret_from_fork+0xa/0x40 Allocated by task 108338: kasan_save_stack+0x40/0x68 kasan_set_track+0x36/0x48 __kasan_kmalloc+0xa0/0xc0 qeth_l2_switchdev_event+0x25a/0x738 atomic_notifier_call_chain+0x9c/0xf8 br_switchdev_fdb_notify+0xf4/0x110 fdb_notify+0x122/0x180 fdb_add_entry.constprop.0.isra.0+0x312/0x558 br_fdb_add+0x59e/0x858 rtnl_fdb_add+0x58a/0x928 rtnetlink_rcv_msg+0x5f8/0x8d8 netlink_rcv_skb+0x1f2/0x408	2024-10-21	7.8	High

		netlink_unicast+0x570/0x790 netlink_sendmsg+0x752/0xbe0 sock_sendmsg+0xca/0x110 ____sys_sendmsg+0x510/0x6a8 ___sys_sendmsg+0x12a/0x180 __sys_sendmsg+0xe6/0x168 __do_sys_socketcall+0x3c8/0x468 do_syscall+0x22c/0x328 __do_syscall+0x94/0xf0 system_call+0x82/0xb0 Freed by task 540: kasan_save_stack+0x40/0x68 kasan_set_track+0x36/0x48 kasan_save_free_info+0x4c/0x68 ____kasan_slab_free+0x14e/0x1a8 __kasan_slab_free+0x24/0x30 __kmem_cache_free+0x168/0x338 qeth_l2_br2dev_worker+0x154/0x6b0 process_one_work+0x76e/0x1128 worker_thread+0x184/0x1098 kthread+0x26a/0x310 __ret_from_fork+0x8a/0xe8 ret_from_fork+0xa/0x40 Last potentially related work creation: kasan_save_stack+0x40/0x68 __kasan_record_aux_stack+0xbe/0xd0 insert_work+0x56/0x2e8 __queue_work+0x4ce/0xd10 queue_work_on+0xf4/0x100 qeth_l2_switchdev_event+0x520/0x738 atomic_notifier_call_chain+0x9c/0xf8 br_switchdev_fdb_notify+0xf4/0x110 fdb_notify+0x122/0x180 fdb_add_entry.constprop.0.isra.0+0x312/0x558 br_fdb_add+0x59e/0x858 rtnl_fdb_add+0x58a/0x928 rtnetlink_rcv_msg+0x5f8/0x8d8 netlink_rcv_skb+0x1f2/0x408 netlink_unicast+0x570/0x790 netlink_sendmsg+0x752/0xbe0 sock_sendmsg+0xca/0x110 ____sys_sendmsg+0x510/0x6a8 ___sys_sendmsg+0x12a/0x180 __sys_sendmsg+0xe6/0x168 __do_sys_socketcall+0x3c8/0x468 do_syscall+0x22c/0x328 __do_syscall+0x94/0xf0 system_call+0x82/0xb0 Second to last potentially related work creation: kasan_save_stack+0x40/0x68 __kasan_record_aux_stack+0xbe/0xd0			

		kvfree_call_rcu+0xb2/0x760 kernfs_unlink_open_file+0x348/0x430 kernfs_fop_release+0xc2/0x320 __fput+0x1ae/0x768 task_work_run+0x1bc/0x298 exit_to_user_mode_prepare+0x1a0/0x1a8 __do_syscall+0x94/0xf0 system_call+0x82/0xb0 The buggy address belongs to the object at 00000000fdcea400 which belongs to the cache kmalloc-96 of size 96 The buggy address is located 64 bytes inside of 96-byte region [00000000fdcea400, 00000000fdcea460) The buggy address belongs to the physical page: page:000000005a9c26e8 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0xfdcea flags: 0x3ffff00000000200(slab|node=0|zone=1|lastcpupid=0x1ffff) raw: 3ffff00000000200 0000000000000000 0000000100000122 000000008008cc00 raw: 0000000000000000 0020004100000000 ffffffff00000001 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: 00000000fdcea300: fb fb fb fb fb fb fb fb fb fb fb fb fc fc fc fc 00000000fdcea380: fb fb fb fb fb fb f ---truncated---			
CVE-2022-48956	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: avoid use-after-free in ip6_fragment() Blamed commit claimed rcu_read_lock() was held by ip6_fragment() callers. It seems to not be always true, at least for UDP stack. syzbot reported: BUG: KASAN: use-after-free in ip6_dst_idev include/net/ip6_fib.h:245 [inline] BUG: KASAN: use-after-free in ip6_fragment+0x2724/0x2770 net/ipv6/ip6_output.c:951 Read of size 8 at addr ffff88801d403e80 by task syz-executor.3/7618 CPU: 1 PID: 7618 Comm: syz-executor.3 Not tainted	2024-10-21	7.8	High

		6.1.0-rc6-syzkaller-00012-g4312098baf37 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd1/0x138 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:284 [inline] print_report+0x15e/0x45d mm/kasan/report.c:395 kasan_report+0xbf/0x1f0 mm/kasan/report.c:495 ip6_dst_idev include/net/ip6_fib.h:245 [inline] ip6_fragment+0x2724/0x2770 net/ipv6/ip6_output.c:951 __ip6_finish_output net/ipv6/ip6_output.c:193 [inline] ip6_finish_output+0x9a3/0x1170 net/ipv6/ip6_output.c:206 NF_HOOK_COND include/linux/netfilter.h:291 [inline] ip6_output+0x1f1/0x540 net/ipv6/ip6_output.c:227 dst_output include/net/dst.h:445 [inline] ip6_local_out+0xb3/0x1a0 net/ipv6/output_core.c:161 ip6_send_skb+0xbb/0x340 net/ipv6/ip6_output.c:1966 udp_v6_send_skb+0x82a/0x18a0 net/ipv6/udp.c:1286 udp_v6_push_pending_frames+0x140/0x200 net/ipv6/udp.c:1313 udpv6_sendmsg+0x18da/0x2c80 net/ipv6/udp.c:1606 inet6_sendmsg+0x9d/0xe0 net/ipv6/af_inet6.c:665 sock_sendmsg_nosec net/socket.c:714 [inline] sock_sendmsg+0xd3/0x120 net/socket.c:734 sock_write_iter+0x295/0x3d0 net/socket.c:1108 call_write_iter include/linux/fs.h:2191 [inline] new_sync_write fs/read_write.c:491 [inline] vfs_write+0x9ed/0xdd0 fs/read_write.c:584 ksys_write+0x1ec/0x250 fs/read_write.c:637 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7fde3588c0d9 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 f1 19 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 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fde365b6168 EFLAGS: 00000246			

		ORIG_RAX: 0000000000000001 RAX: ffffffffffffffda RBX: 00007fde359ac050 RCX: 00007fde3588c0d9 RDX: 000000000000ffdc RSI: 00000000200000c0 RDI: 000000000000000a RBP: 00007fde358e7ae9 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007fde35acfb1f R14: 00007fde365b6300 R15: 0000000000022000 </TASK> Allocated by task 7618: kasan_save_stack+0x22/0x40 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 __kasan_slab_alloc+0x82/0x90 mm/kasan/common.c:325 kasan_slab_alloc include/linux/kasan.h:201 [inline] slab_post_alloc_hook mm/slab.h:737 [inline] slab_alloc_node mm/slub.c:3398 [inline] slab_alloc mm/slub.c:3406 [inline] __kmem_cache_alloc_lru mm/slub.c:3413 [inline] kmem_cache_alloc+0x2b4/0x3d0 mm/slub.c:3422 dst_alloc+0x14a/0x1f0 net/core/dst.c:92 ip6_dst_alloc+0x32/0xa0 net/ipv6/route.c:344 ip6_rt_pcpu_alloc net/ipv6/route.c:1369 [inline] rt6_make_pcpu_route net/ipv6/route.c:1417 [inline] ip6_pol_route+0x901/0x1190 net/ipv6/route.c:2254 pol_lookup_func include/net/ip6_fib.h:582 [inline] fib6_rule_lookup+0x52e/0x6f0 net/ipv6/fib6_rules.c:121 ip6_route_output_flags_noref+0x2e6/0x380 net/ipv6/route.c:2625 ip6_route_output_flags+0x76/0x320 net/ipv6/route.c:2638 ip6_route_output include/net/ip6_route.h:98 [inline] ip6_dst_lookup_tail+0x5ab/0x1620 net/ipv6/ip6_output.c:1092 ip6_dst_lookup_flow+0x90/0x1d0 net/ipv6/ip6_output.c:1222 ip6_sk_dst_lookup_flow+0x553/0x980 net/ipv6/ip6_output.c:1260 udpv6_sendmsg+0x151d/0x2c80 net/ipv6/udp.c:1554 inet6_sendmsg+0x9d/0xe0 net/ipv6/af_inet6.c:665 sock_sendmsg_nosec n ---truncated---			

CVE-2022-48960	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: hisilicon: Fix potential use-after-free in hix5hd2_rx() The skb is delivered to napi_gro_receive() which may free it, after calling this, dereferencing skb may trigger use-after-free.	2024-10-21	7.8	High
CVE-2022-48962	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: hisilicon: Fix potential use-after-free in hisi_femac_rx() The skb is delivered to napi_gro_receive() which may free it, after calling this, dereferencing skb may trigger use-after-free.	2024-10-21	7.8	High
CVE-2022-48964	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ravb: Fix potential use-after-free in ravb_rx_gbeth() The skb is delivered to napi_gro_receive() which may free it, after calling this, dereferencing skb may trigger use-after-free.	2024-10-21	7.8	High
CVE-2022-48980	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: dsa: sja1105: avoid out of bounds access in sja1105_init_l2_policing() The SJA1105 family has 45 L2 policing table entries (SJA1105_MAX_L2_POLICING_COUNT) and SJA1110 has 110 (SJA1110_MAX_L2_POLICING_COUNT). Keeping the table structure but accounting for the difference in port count (5 in SJA1105 vs 10 in SJA1110) does not fully explain the difference. Rather, the SJA1110 also has L2 ingress policers for multicast traffic. If a packet is classified as multicast, it will be processed by the policer index 99 + SRCPORT. The sja1105_init_l2_policing() function initializes all L2 policers such that they don't interfere with normal packet	2024-10-21	7.8	High

		reception by default. To have a common code between SJA1105 and SJA1110, the index of the multicast policer for the port is calculated because it's an index that is out of bounds for SJA1105 but in bounds for SJA1110, and a bounds check is performed. The code fails to do the proper thing when determining what to do with the multicast policer of port 0 on SJA1105 (ds->num_ports = 5). The "mcast" index will be equal to 45, which is also equal to table->ops->max_entry_count (SJA1105_MAX_L2_POLICING_COUNT). So it passes through the check. But at the same time, SJA1105 doesn't have multicast policers. So the code programs the SHARINDX field of an out-of-bounds element in the L2 Policing table of the static config. The comparison between index 45 and 45 entries should have determined the code to not access this policer index on SJA1105, since its memory wasn't even allocated. With enough bad luck, the out-of-bounds write could even overwrite other valid kernel data, but in this case, the issue was detected using KASAN. Kernel log: sja1105 spi5.0: Probed switch chip: SJA1105Q === BUG: KASAN: slab-out-of-bounds in sja1105_setup+0x1cbc/0x2340 Write of size 8 at addr ffffff880bd57708 by task kworker/u8:0/8 ... Workqueue: events_unbound deferred_probe_work_func Call trace: ... sja1105_setup+0x1cbc/0x2340 dsa_register_switch+0x1284/0x18d0 sja1105_probe+0x748/0x840 ... Allocated by task 8:			

		... sja1105_setup+0x1bcc/0x2340 dsa_register_switch+0x1284/0x18d0 sja1105_probe+0x748/0x840 ...			
CVE-2022-48981	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/shmem-helper: Remove errant put in error path drm_gem_shmem_mmap() doesn't own this reference, resulting in the GEM object getting prematurely freed leading to a later use-after-free.	2024-10-21	7.8	High
CVE-2022-48990	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix use-after-free during gpu recovery [Why] [754.862560] refcount_t: underflow; use-after-free. [754.862898] Call Trace: [754.862903] <TASK> [754.862913] amdgpu_job_free_cb+0xc2/0xe1 [amdgpu] [754.863543] drm_sched_main.cold+0x34/0x39 [amd_sched] [How] The fw_fence may be not init, check whether dma_fence_init is performed before job free	2024-10-21	7.8	High
CVE-2022-49006	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tracing: Free buffers when a used dynamic event is removed After 65536 dynamic events have been added and removed, the "type" field of the event then uses the first type number that is available (not currently used by other events). A type number is the identifier of the binary blobs in the tracing ring buffer (known as events) to map them to logic that can parse the binary blob. The issue is that if a dynamic event (like a kprobe event) is traced and	2024-10-21	7.8	High

		is in the ring buffer, and then that event is removed (because it is dynamic, which means it can be created and destroyed), if another dynamic event is created that has the same number that new event's logic on parsing the binary blob will be used. To show how this can be an issue, the following can crash the kernel: # cd /sys/kernel/tracing # for i in `seq 65536`; do echo 'p:kprobes/foo do_sys_openat2 $arg1:u32' > kprobe_events # done For every iteration of the above, the writing to the kprobe_events will remove the old event and create a new one (with the same format) and increase the type number to the next available on until the type number reaches over 65535 which is the max number for the 16 bit type. After it reaches that number, the logic to allocate a new number simply looks for the next available number. When an dynamic event is removed, that number is then available to be reused by the next dynamic event created. That is, once the above reaches the max number, the number assigned to the event in that loop will remain the same. Now that means deleting one dynamic event and created another will reuse the previous events type number. This is where bad things can happen. After the above loop finishes, the kprobes/foo event which reads the do_sys_openat2 function call's first parameter as an integer. # echo 1 > kprobes/foo/enable # cat /etc/passwd > /dev/null # cat trace cat-2211 [005] .... 2007.849603: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 cat-2211 [005] .... 2007.849620: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196			

		cat-2211 [005] .... 2007.849838: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 cat-2211 [005] .... 2007.849880: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 # echo 0 > kprobes/foo/enable Now if we delete the kprobe and create a new one that reads a string: # echo 'p:kprobes/foo do_sys_openat2 +0($arg2):string' > kprobe_events And now we can the trace: # cat trace sendmail-1942 [002] ..... 530.136320: foo: (do_sys_openat2+0x0/0x240) arg1= cat-2046 [004] ..... 530.930817: foo: (do_sys_openat2+0x0/0x240) arg1="��" cat-2046 [004] ..... 530.930961: foo: (do_sys_openat2+0x0/0x240) arg1="��" cat-2046 [004] ..... 530.934278: foo: (do_sys_openat2+0x0/0x240) arg1="��" cat-2046 [004] ..... 530.934563: foo: (do_sys_openat2+0x0/0x240) arg1="��������������������������������������� ---truncated---			
[CVE-2022-49014](https://nvd.nist.gov/vuln/detail/CVE-2022-49014)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: tun: Fix use-after-free in tun_detach() syzbot reported use-after-free in tun_detach() [1]. This causes call trace like below:	2024-10-21	7.8	High

		== BUG: KASAN: use-after-free in notifier_call_chain+0x1ee/0x200 kernel/notifier.c:75 Read of size 8 at addr ffff88807324e2a8 by task syz-executor.0/3673 CPU: 0 PID: 3673 Comm: syz-executor.0 Not tainted 6.1.0-rc5-syzkaller-00044-gcc675d22e422 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd1/0x138 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:284 [inline] print_report+0x15e/0x461 mm/kasan/report.c:395 kasan_report+0xbf/0x1f0 mm/kasan/report.c:495 notifier_call_chain+0x1ee/0x200 kernel/notifier.c:75 call_netdevice_notifiers_info+0x86/0x130 net/core/dev.c:1942 call_netdevice_notifiers_extack net/core/dev.c:1983 [inline] call_netdevice_notifiers net/core/dev.c:1997 [inline] netdev_wait_allrefs_any net/core/dev.c:10237 [inline] netdev_run_todo+0xbc6/0x1100 net/core/dev.c:10351 tun_detach drivers/net/tun.c:704 [inline] tun_chr_close+0xe4/0x190 drivers/net/tun.c:3467 __fput+0x27c/0xa90 fs/file_table.c:320 task_work_run+0x16f/0x270 kernel/task_work.c:179 exit_task_work include/linux/task_work.h:38 [inline] do_exit+0xb3d/0x2a30 kernel/exit.c:820 do_group_exit+0xd4/0x2a0 kernel/exit.c:950 get_signal+0x21b1/0x2440 kernel/signal.c:2858 arch_do_signal_or_restart+0x86/0x2300 arch/x86/kernel/signal.c:869 exit_to_user_mode_loop kernel/entry/common.c:168 [inline] exit_to_user_mode_prepare+0x15f/0x250 kernel/entry/common.c:203 __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline] syscall_exit_to_user_mode+0x1d/0x50 kernel/entry/common.c:296 do_syscall_64+0x46/0xb0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x63/0xcd			

		The cause of the issue is that sock_put() from __tun_detach() drops last reference count for struct net, and then notifier_call_chain() from netdev_state_change() accesses that struct net. This patch fixes the issue by calling sock_put() from tun_detach() after all necessary accesses for the struct net has done.			
CVE-2022-49015	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: hsr: Fix potential use-after-free The skb is delivered to netif_rx() which may free it, after calling this, dereferencing skb may trigger use-after-free.	2024-10-21	7.8	High
CVE-2022-49017	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tipc: re-fetch skb cb after tipc_msg_validate As the call trace shows, the original skb was freed in tipc_msg_validate(), and dereferencing the old skb cb would cause an use-after-free crash. BUG: KASAN: use-after-free in tipc_crypto_rcv_complete+0x1835/0x2240 [tipc] Call Trace: <IRQ> tipc_crypto_rcv_complete+0x1835/0x2240 [tipc] tipc_crypto_rcv+0xd32/0x1ec0 [tipc] tipc_rcv+0x744/0x1150 [tipc] ... Allocated by task 47078: kmem_cache_alloc_node+0x158/0x4d0 __alloc_skb+0x1c1/0x270 tipc_buf_acquire+0x1e/0xe0 [tipc] tipc_msg_create+0x33/0x1c0 [tipc] tipc_link_build_proto_msg+0x38a/0x2100 [tipc] tipc_link_timeout+0x8b8/0xef0 [tipc] tipc_node_timeout+0x2a1/0x960 [tipc] call_timer_fn+0x2d/0x1c0 ... Freed by task 47078: tipc_msg_validate+0x7b/0x440 [tipc] tipc_crypto_rcv_complete+0x4b5/0x2240 [tipc] tipc_crypto_rcv+0xd32/0x1ec0 [tipc]	2024-10-21	7.8	High

		tipc_rcv+0x744/0x1150 [tipc] This patch fixes it by re-fetching the skb cb from the new allocated skb after calling tipc_msg_validate().			
CVE-2022-49022	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mac8021: fix possible oob access in ieee80211_get_rate_duration Fix possible out-of-bound access in ieee80211_get_rate_duration routine as reported by the following UBSAN report: UBSAN: array-index-out-of-bounds in net/mac80211/airtime.c:455:47 index 15 is out of range for type 'u16 [12]' CPU: 2 PID: 217 Comm: kworker/u32:10 Not tainted 6.1.0-060100rc3-generic Hardware name: Acer Aspire TC-281/Aspire TC-281, BIOS R01-A2 07/18/2017 Workqueue: mt76 mt76u_tx_status_data [mt76_usb] Call Trace: <TASK> show_stack+0x4e/0x61 dump_stack_lvl+0x4a/0x6f dump_stack+0x10/0x18 ubsan_epilogue+0x9/0x43 __ubsan_handle_out_of_bounds.cold+0x42/0x47 ieee80211_get_rate_duration.constprop.0+0x22f/0x2a0 [mac80211] ? ieee80211_tx_status_ext+0x32e/0x640 [mac80211] ieee80211_calc_rx_airtime+0xda/0x120 [mac80211] ieee80211_calc_tx_airtime+0xb4/0x100 [mac80211] mt76x02_send_tx_status+0x266/0x480 [mt76x02_lib] mt76x02_tx_status_data+0x52/0x80 [mt76x02_lib] mt76u_tx_status_data+0x67/0xd0 [mt76_usb] process_one_work+0x225/0x400 worker_thread+0x50/0x3e0 ? process_one_work+0x400/0x400 kthread+0xe9/0x110 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30	2024-10-21	7.8	High
CVE-2022-49023	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix buffer overflow in elem comparison	2024-10-21	7.8	High

		For vendor elements, the code here assumes that 5 octets are present without checking. Since the element itself is already checked to fit, we only need to check the length.			
CVE-2022-49025	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix use-after-free when reverting termination table When having multiple dests with termination tables and second one or afterwards fails the driver reverts usage of term tables but doesn't reset the assignment in attr->dests[num_vport_dests].termtbl which case a use-after-free when releasing the rule. Fix by resetting the assignment of termtbl to null.	2024-10-21	7.8	High
CVE-2022-49026	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: e100: Fix possible use after free in e100_xmit_prepare In e100_xmit_prepare(), if we can't map the skb, then return -ENOMEM, so e100_xmit_frame() will return NETDEV_TX_BUSY and the upper layer will resend the skb. But the skb is already freed, which will cause UAF bug when the upper layer resends the skb. Remove the harmful free.	2024-10-21	7.8	High
CVE-2022-49029	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hwmon: (ibmpex) Fix possible UAF when ibmpex_register_bmc() fails Smatch report warning as follows: drivers/hwmon/ibmpex.c:509 ibmpex_register_bmc() warn: '&data->list' not removed from list If ibmpex_find_sensors() fails in ibmpex_register_bmc(), data will be freed, but data->list will not be removed from	2024-10-21	7.8	High

		driver_data.bmc_data, then list traversal may cause UAF. Fix by removeing it from driver_data.bmc_data before free().			
CVE-2022-49030	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: libbpf: Handle size overflow for ringbuf mmap The maximum size of ringbuf is 2GB on x86-64 host, so 2 * max_entries will overflow u32 when mapping producer page and data pages. Only casting max_entries to size_t is not enough, because for 32-bits application on 64-bits kernel the size of read-only mmap region also could overflow size_t. So fixing it by casting the size of read-only mmap region into a __u64 and checking whether or not there will be overflow during mmap.	2024-10-21	7.8	High
CVE-2024-50029	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: Fix UAF in hci_enhanced_setup_sync This checks if the ACL connection remains valid as it could be destroyed while hci_enhanced_setup_sync is pending on cmd_sync leading to the following trace: BUG: KASAN: slab-use-after-free in hci_enhanced_setup_sync+0x91b/0xa60 Read of size 1 at addr ffff888002328ffd by task kworker/u5:2/37 CPU: 0 UID: 0 PID: 37 Comm: kworker/u5:2 Not tainted 6.11.0-rc6-01300-g810be445d8d6 #7099 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> dump_stack_lvl+0x5d/0x80 ? hci_enhanced_setup_sync+0x91b/0xa60 print_report+0x152/0x4c0	2024-10-21	7.8	High

		? hci_enhanced_setup_sync+0x91b/0xa60 ? __virt_addr_valid+0x1fa/0x420 ? hci_enhanced_setup_sync+0x91b/0xa60 kasan_report+0xda/0x1b0 ? hci_enhanced_setup_sync+0x91b/0xa60 hci_enhanced_setup_sync+0x91b/0xa60 ? __pfx_hci_enhanced_setup_sync+0x10/0x10 ? __pfx___mutex_lock+0x10/0x10 hci_cmd_sync_work+0x1c2/0x330 process_one_work+0x7d9/0x1360 ? __pfx_lock_acquire+0x10/0x10 ? __pfx_process_one_work+0x10/0x10 ? assign_work+0x167/0x240 worker_thread+0x5b7/0xf60 ? __kthread_parkme+0xac/0x1c0 ? __pfx_worker_thread+0x10/0x10 ? __pfx_worker_thread+0x10/0x10 kthread+0x293/0x360 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2f/0x70 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 34: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x8f/0xa0 __hci_conn_add+0x187/0x17d0 hci_connect_sco+0x2e1/0xb90 sco_sock_connect+0x2a2/0xb80 __sys_connect+0x227/0x2a0 __x64_sys_connect+0x6d/0xb0 do_syscall_64+0x71/0x140 entry_SYSCALL_64_after_hwframe+0x76/0x7e Freed by task 37: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x101/0x160 kfree+0xd0/0x250 device_release+0x9a/0x210 kobject_put+0x151/0x280 hci_conn_del+0x448/0xbf0 hci_abort_conn_sync+0x46f/0x980 hci_cmd_sync_work+0x1c2/0x330 process_one_work+0x7d9/0x1360 worker_thread+0x5b7/0xf60 kthread+0x293/0x360			

		ret_from_fork+0x2f/0x70 ret_from_fork_asm+0x1a/0x30			
CVE-2024-50030	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/ct: prevent UAF in send_recv() Ensure we serialize with completion side to prevent UAF with fence going out of scope on the stack, since we have no clue if it will fire after the timeout before we can erase from the xa. Also we have some dependent loads and stores for which we need the correct ordering, and we lack the needed barriers. Fix this by grabbing the ct->lock after the wait, which is also held by the completion side. v2 (Badal): - Also print done after acquiring the lock and seeing timeout. (cherry picked from commit 52789ce35c55ccd30c4b67b9cc5b2af55e0122ea)	2024-10-21	7.8	High
CVE-2024-50043	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix possible badness in FREE_STATEID When multiple FREE_STATEIDs are sent for the same delegation stateid, it can lead to a possible either use-after-free or counter refcount underflow errors. In nfsd4_free_stateid() under the client lock we find a delegation stateid, however the code drops the lock before calling nfs4_put_stid(), that allows another FREE_STATE to find the stateid again. The first one will proceed to then free the stateid which leads to either use-after-free or decrementing already zeroed counter.	2024-10-21	7.8	High
CVE-2024-50047	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix UAF in async decryption	2024-10-21	7.8	High

		Doing an async decryption (large read) crashes with a slab-use-after-free way down in the crypto API. Reproducer: # mount.cifs -o ...,seal,esize=1 //srv/share /mnt # dd if=/mnt/largefile of=/dev/null ... [194.196391] == [194.196844] BUG: KASAN: slab-use-after-free in gf128mul_4k_lle+0xc1/0x110 [194.197269] Read of size 8 at addr ffff888112bd0448 by task kworker/u77:2/899 [194.197707] [194.197818] CPU: 12 UID: 0 PID: 899 Comm: kworker/u77:2 Not tainted 6.11.0-lku-00028-gfca3ca14a17a-dirty #43 [194.198400] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.2-3-gd478f380-prebuilt.qemu.org 04/01/2014 [194.199046] Workqueue: smb3decryptd smb2_decrypt_offload [cifs] [194.200032] Call Trace: [194.200191] <TASK> [194.200327] dump_stack_lvl+0x4e/0x70 [194.200558] ? gf128mul_4k_lle+0xc1/0x110 [194.200809] print_report+0x174/0x505 [194.201040] ? __pfx__raw_spin_lock_irqsave+0x10/0x10 [194.201352] ? srso_return_thunk+0x5/0x5f [194.201604] ? __virt_addr_valid+0xdf/0x1c0 [194.201868] ? gf128mul_4k_lle+0xc1/0x110 [194.202128] kasan_report+0xc8/0x150 [194.202361] ? gf128mul_4k_lle+0xc1/0x110 [194.202616] gf128mul_4k_lle+0xc1/0x110 [194.202863] ghash_update+0x184/0x210 [194.203103] shash_ahash_update+0x184/0x2a0 [194.203377] ? __pfx_shash_ahash_update+0x10/0x10 [194.203651] ? srso_return_thunk+0x5/0x5f [194.203877] ? crypto_gcm_init_common+0x1ba/0x340 [194.204142] gcm_hash_assoc_remain_continue+0x10a/0x140 [194.204434] crypt_message+0xec1/0x10a0 [cifs] [194.206489] ? __pfx_crypt_message+0x10/0x10 [cifs] [194.208507] ? srso_return_thunk+0x5/0x5f [194.209205] ? srso_return_thunk+0x5/0x5f [194.209925] ? srso_return_thunk+0x5/0x5f [194.210443] ? srso_return_thunk+0x5/0x5f			

		[194.211037] decrypt_raw_data+0x15f/0x250 [cifs] [194.212906] ? __pfx_decrypt_raw_data+0x10/0x10 [cifs] [194.214670] ? srso_return_thunk+0x5/0x5f [194.215193] smb2_decrypt_offload+0x12a/0x6c0 [cifs] This is because TFM is being used in parallel. Fix this by allocating a new AEAD TFM for async decryption, but keep the existing one for synchronous READ cases (similar to what is done in smb3_calc_signature()). Also remove the calls to aead_request_set_callback() and crypto_wait_req() since it's always going to be a synchronous operation.			
CVE-2024-50055	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: driver core: bus: Fix double free in driver API bus_register() For bus_register(), any error which happens after kset_register() will cause that @priv are freed twice, fixed by setting @priv with NULL after the first free.	2024-10-21	7.8	High
CVE-2024-9050	red hat - multiple products	A flaw was found in the libreswan client plugin for NetworkManager (NetkworkManager-libreswan), where it fails to properly sanitize the VPN configuration from the local unprivileged user. In this configuration, composed by a key-value format, the plugin fails to escape special characters, leading the application to interpret values as keys. One of the most critical parameters that could be abused by a malicious user is the `leftupdown`key. This key takes an executable command as a value and is used to specify what executes as a callback in NetworkManager-libreswan to retrieve configuration settings back to NetworkManager. As NetworkManager uses Polkit to allow an unprivileged user to control the system's network configuration, a malicious actor could achieve local privilege escalation and potential code execution as root in the targeted machine by creating a malicious configuration.	2024-10-22	7.8	High

CVE-2024-45334	trendmicro - antivirus_one	Trend Micro Antivirus One versions 3.10.4 and below (Consumer) is vulnerable to an Arbitrary Configuration Update that could allow unauthorized access to product configurations and functions.	2024-10-22	7.8	High
CVE-2024-47012	google - android	In mm_GetMobileIdIndexForNsUpdate of mm_GmmPduCodec.c, there is a possible out of bounds write due to an incorrect bounds check. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47013	google - Android	In pmucal_rae_handle_seq_int of flexpmu_cal_rae.c, there is a possible arbitrary write due to uninitialized data. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47016	google - Android	there is a possible privilege escalation due to an insecure default value. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47017	google - android	In ufshc_scsi_cmd of ufs.c, there is a possible stack variable use after free due to a use after free. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47024	google - android	In vring_size of external/headers/include/virtio/virtio_ring.h, there is a possible out of bounds write due to an integer overflow. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47027	google - android	In sm_mem_compat_get_vmm_obj of lib/sm/shared_mem.c, there is a possible arbitrary physical memory access due to improper input validation. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47033	google - android	In lwis_allocator_free of lwis_allocator.c, there is a possible memory corruption due to a use after free. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-47035	google - android	In vring_init of external/headers/include/virtio/virtio_ring.h, there is a possible out of bounds write due to a logic error in the code. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High

CVE-2024-47041	google - android	In valid_address of syscall.c, there is a possible out of bounds read due to an incorrect bounds check. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.8	High
CVE-2024-20268	cisco - multiple products	A vulnerability in the Simple Network Management Protocol (SNMP) feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an authenticated, remote attacker to cause an unexpected reload of the device. This vulnerability is due to insufficient input validation of SNMP packets. An attacker could exploit this vulnerability by sending a crafted SNMP request to an affected device using IPv4 or IPv6. A successful exploit could allow the attacker to cause the affected device to reload, resulting in a denial of service (DoS) condition. This vulnerability affects all versions of SNMP (versions 1, 2c, and 3) and requires a valid SNMP community string or valid SNMPv3 user credentials.	2024-10-23	7.7	High
CVE-2024-20408	cisco - multiple products	A vulnerability in the Dynamic Access Policies (DAP) feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an authenticated, remote attacker to cause an affected device to reload unexpectedly. To exploit this vulnerability, an attacker would need valid remote access VPN user credentials on the affected device. This vulnerability is due to improper validation of data in HTTPS POST requests. An attacker could exploit this vulnerability by sending a crafted HTTPS POST request to an affected device. A successful exploit could allow the attacker to cause the device to reload, resulting in a denial of service (DoS) condition.	2024-10-23	7.7	High
CVE-2024-49997	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: lantiq_etop: fix memory disclosure When applying padding, the buffer is not zeroed, which results in memory disclosure. The mentioned data is observed on the wire. This patch uses skb_put_padto() to pad Ethernet frames properly.	2024-10-21	7.5	High

		The mentioned function zeroes the expanded buffer. In case the packet cannot be padded it is silently dropped. Statistics are also not incremented. This driver does not support statistics in the old 32-bit format or the new 64-bit format. These will be added in the future. In its current form, the patch should be easily backported to stable versions. Ethernet MACs on Amazon-SE and Danube cannot do padding of the packets in hardware, so software padding must be applied.			
CVE-2024-44100	google - android	Android before 2024-10-05 on Google Pixel devices allows information disclosure in the modem component, A-299774545.	2024-10-25	7.5	High
CVE-2024-44101	google - android	there is a possible Null Pointer Dereference (modem crash) due to improper input validation. This could lead to remote denial of service with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.5	High
CVE-2024-47020	google - android	Android before 2024-10-05 on Google Pixel devices allows information disclosure in the ABL component, A-331966488.	2024-10-25	7.5	High
CVE-2024-47021	google - android	In sms_ExtractCbLanguage of sms_CellBroadcast.c, there is a possible out of bounds read due to a missing bounds check. This could lead to remote information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.5	High
CVE-2024-47022	google - android	Android before 2024-10-05 on Google Pixel devices allows information disclosure in the ACPM component, A-331255656.	2024-10-25	7.5	High
CVE-2024-44098	google - Android	In lwis_device_event_states_clear_locked of lwis_event.c, there is a possible privilege escalation due to a double free. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	7.4	High
CVE-2024-47031	google - Android	Android before 2024-10-05 on Google Pixel devices allows privilege escalation in the ABL component, A-329163861.	2024-10-25	7.4	High
CVE-2024-10234	redhat - multiple products	A vulnerability was found in Wildfly, where a user may perform Cross-site scripting in the Wildfly deployment system. This flaw allows an attacker or insider to execute a deployment with a malicious payload, which could trigger undesired behavior against the server.	2024-10-22	7.3	High

CVE-2024-30157	mitel - micollab	A vulnerability in the Suite Applications Services component of Mitel MiCollab through 9.7.1.110 could allow an authenticated attacker with administrative privileges to conduct a SQL Injection attack due to insufficient validation of user input. A successful exploit could allow an attacker to execute arbitrary database and management operations.	2024-10-21	7.2	High
CVE-2024-30158	mitel - micollab	A vulnerability in the web conferencing component of Mitel MiCollab through 9.7.1.110 could allow an authenticated attacker with administrative privileges to conduct a SQL Injection attack due to insufficient validation of user input. A successful exploit could allow an attacker to execute arbitrary database and management operations.	2024-10-21	7.2	High
CVE-2024-47686	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ep93xx: clock: Fix off by one in ep93xx_div_recalc_rate() The psc->div[] array has psc->num_div elements. These values come from when we call clk_hw_register_div(). It's adc_divisors and ARRAY_SIZE(adc_divisors)) and so on. So this condition needs to be >= instead of > to prevent an out of bounds read.	2024-10-21	7.1	High
CVE-2024-47721	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: remove unused C2H event ID RTW89_MAC_C2H_FUNC_READ_WOW_CAM to prevent out-of-bounds reading The handler of firmware C2H event RTW89_MAC_C2H_FUNC_READ_WOW_CAM isn't implemented, but driver expects number of handlers is NUM_OF_RTW89_MAC_C2H_FUNC_WOW causing out-of-bounds access. Fix it by removing ID. Addresses-Coverity-ID: 1598775 ("Out-of-bounds read")	2024-10-21	7.1	High
CVE-2024-47723	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: jfs: fix out-of-bounds in dbNextAG() and diAlloc() In dbNextAG() , there is no check for the case where bmp->db_numag is	2024-10-21	7.1	High

		greater or same than MAXAG due to a polluted image, which causes an out-of-bounds. Therefore, a bounds check should be added in dbMount(). And in dbNextAG(), a check for the case where agpref is greater than bmp->db_numag should be added, so an out-of-bounds exception should be prevented. Additionally, a check for the case where agno is greater or same than MAXAG should be added in diAlloc() to prevent out-of-bounds.			
CVE-2024-47757	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential oob read in nilfs_btree_check_delete() The function nilfs_btree_check_delete(), which checks whether degeneration to direct mapping occurs before deleting a b-tree entry, causes memory access outside the block buffer when retrieving the maximum key if the root node has no entries. This does not usually happen because b-tree mappings with 0 child nodes are never created by mkfs.nilfs2 or nilfs2 itself. However, it can happen if the b-tree root node read from a device is configured that way, so fix this potential issue by adding a check for that case.	2024-10-21	7.1	High
CVE-2024-49860	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ACPI: sysfs: validate return type of _STR method Only buffer objects are valid return values of _STR. If something else is returned description_show() will access invalid memory.	2024-10-21	7.1	High
CVE-2024-49861	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix helper writes to read-only maps	2024-10-21	7.1	High

		Lonial found an issue that despite user- and BPF-side frozen BPF map (like in case of .rodata), it was still possible to write into it from a BPF program side through specific helpers having ARG_PTR_TO_{LONG,INT} as arguments. In check_func_arg() when the argument is as mentioned, the meta->raw_mode is never set. Later, check_helper_mem_access(), under the case of PTR_TO_MAP_VALUE as register base type, it assumes BPF_READ for the subsequent call to check_map_access_type() and given the BPF map is read-only it succeeds. The helpers really need to be annotated as ARG_PTR_TO_{LONG,INT} | MEM_UNINIT when results are written into them as opposed to read out of them. The latter indicates that it's okay to pass a pointer to uninitialized memory as the memory is written to anyway. However, ARG_PTR_TO_{LONG,INT} is a special case of ARG_PTR_TO_FIXED_SIZE_MEM just with additional alignment requirement. So it is better to just get rid of the ARG_PTR_TO_{LONG,INT} special cases altogether and reuse the fixed size memory types. For this, add MEM_ALIGNED to additionally ensure alignment given these helpers write directly into the args via *<ptr> = val. The .arg*_size has been initialized reflecting the actual sizeof(*<ptr>). MEM_ALIGNED can only be used in combination with MEM_FIXED_SIZE annotated argument types, since in !MEM_FIXED_SIZE cases the verifier does not know the buffer size a priori and therefore cannot blindly write *<ptr> = val.			
CVE-2024-49862	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: powercap: intel_rapl: Fix off by one in get_rpi() The rp->priv->rpi array is either rpi_msr or rpi_tpmi	2024-10-21	7.1	High

		which have NR_RAPL_PRIMITIVES number of elements. Thus the > needs to be >= to prevent an off by one access.			
CVE-2024-49900	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uninit-value access of new_ea in ea_buffer syzbot reports that lzo1x_1_do_compress is using uninit-value: === BUG: KMSAN: uninit-value in lzo1x_1_do_compress+0x19f9/0x2510 lib/lzo/lzo1x_compress.c:178 ... Uninit was stored to memory at: ea_put fs/jfs/xattr.c:639 [inline] ... Local variable ea_buf created at: __jfs_setxattr+0x5d/0x1ae0 fs/jfs/xattr.c:662 __jfs_xattr_set+0xe6/0x1f0 fs/jfs/xattr.c:934 === The reason is ea_buf->new_ea is not initialized properly. Fix this by using memset to empty its content at the beginning in ea_get().	2024-10-21	7.1	High
CVE-2024-49928	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid reading out of bounds when loading TX power FW elements Because the loop-expression will do one more time before getting false from cond-expression, the original code copied one more entry size beyond valid region. Fix it by moving the entry copy to loop-body.	2024-10-21	7.1	High

CVE-2022-48966	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mvneta: Prevent out of bounds read in mvneta_config_rss() The pp->indir[0] value comes from the user. It is passed to: if (cpu_online(pp->rxq_def)) inside the mvneta_percpu_elect() function. It needs bounds checkeding to ensure that it is not beyond the end of the cpu bitmap.	2024-10-21	7.1	High
CVE-2022-48967	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: NFC: nci: Bounds check struct nfc_target arrays While running under CONFIG_FORTIFY_SOURCE=y, syzkaller reported: memcpy: detected field-spanning write (size 129) of single field "target->sensf_res" at net/nfc/nci/ntf.c:260 (size 18) This appears to be a legitimate lack of bounds checking in nci_add_new_protocol(). Add the missing checks.	2024-10-21	7.1	High
CVE-2022-48999	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv4: Handle attempt to delete multipath route when fib_info contains an nh reference Gwangun Jung reported a slab-out-of-bounds access in fib_nh_match: fib_nh_match+0xf98/0x1130 linux-6.0-rc7/net/ipv4/fib_semantics.c:961 fib_table_delete+0x5f3/0xa40 linux-6.0-rc7/net/ipv4/fib_trie.c:1753 inet_rtm_delroute+0x2b3/0x380 linux-6.0-rc7/net/ipv4/fib_frontend.c:874 Separate nexthop objects are mutually exclusive with the legacy multipath spec. Fix fib_nh_match to return if the config for the to be deleted route contains a multipath spec while	2024-10-21	7.1	High

		the fib_info is using a nexthop object.			
CVE-2022-49031	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: health: afe4403: Fix oob read in afe4403_read_raw KASAN report out-of-bounds read as follows: BUG: KASAN: global-out-of-bounds in afe4403_read_raw+0x42e/0x4c0 Read of size 4 at addr ffffffffc02ac638 by task cat/279 Call Trace: afe4403_read_raw iio_read_channel_info dev_attr_show The buggy address belongs to the variable: afe4403_channel_leds+0x18/0xffffffffffffe9e0 This issue can be reproduced by singe command: $ cat /sys/bus/spi/devices/spi0.0/iio\:device0/in_intensity6_raw The array size of afe4403_channel_leds is less than channels, so access with chan->address cause OOB read in afe4403_read_raw. Fix it by moving access before use it.	2024-10-21	7.1	High
CVE-2022-49032	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iio: health: afe4404: Fix oob read in afe4404_[read|write]_raw KASAN report out-of-bounds read as follows: BUG: KASAN: global-out-of-bounds in afe4404_read_raw+0x2ce/0x380 Read of size 4 at addr ffffffffc00e4658 by task cat/278 Call Trace: afe4404_read_raw iio_read_channel_info dev_attr_show The buggy address belongs to the variable:	2024-10-21	7.1	High

		afe4404_channel_leds+0x18/0xffffffffffffe9c0 This issue can be reproduce by singe command: $ cat /sys/bus/i2c/devices/0-0058/iio\:device0/in_intensity6_raw The array size of afe4404_channel_leds and afe4404_channel_offdacs are less than channels, so access with chan->address cause OOB read in afe4404_[read|write]_raw. Fix it by moving access before use them.			
CVE-2024-50033	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: slip: make slhc_remember() more robust against malicious packets syzbot found that slhc_remember() was missing checks against malicious packets [1]. slhc_remember() only checked the size of the packet was at least 20, which is not good enough. We need to make sure the packet includes the IPv4 and TCP header that are supposed to be carried. Add iph and th pointers to make the code more readable. [1] BUG: KMSAN: uninit-value in slhc_remember+0x2e8/0x7b0 drivers/net/slip/slhc.c:666 slhc_remember+0x2e8/0x7b0 drivers/net/slip/slhc.c:666 ppp_receive_nonmp_frame+0xe45/0x35e0 drivers/net/ppp/ppp_generic.c:2455 ppp_receive_frame drivers/net/ppp/ppp_generic.c:2372 [inline] ppp_do_recv+0x65f/0x40d0 drivers/net/ppp/ppp_generic.c:2212 ppp_input+0x7dc/0xe60 drivers/net/ppp/ppp_generic.c:2327 pppoe_rcv_core+0x1d3/0x720 drivers/net/ppp/pppoe.c:379	2024-10-21	7.1	High

		sk_backlog_rcv+0x13b/0x420 include/net/sock.h:1113 __release_sock+0x1da/0x330 net/core/sock.c:3072 release_sock+0x6b/0x250 net/core/sock.c:3626 pppoe_sendmsg+0x2b8/0xb90 drivers/net/ppp/pppoe.c:903 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 ____sys_sendmsg+0x903/0xb60 net/socket.c:2602 ___sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4091 [inline] slab_alloc_node mm/slub.c:4134 [inline] kmem_cache_alloc_node_noprof+0x6bf/0xb80 mm/slub.c:4186 kmalloc_reserve+0x13d/0x4a0 net/core/skbuff.c:587 __alloc_skb+0x363/0x7b0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1322 [inline] sock_wmalloc+0xfe/0x1a0 net/core/sock.c:2732 pppoe_sendmsg+0x3a7/0xb90 drivers/net/ppp/pppoe.c:867 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 ____sys_sendmsg+0x903/0xb60 net/socket.c:2602 ___sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f			

		CPU: 0 UID: 0 PID: 5460 Comm: syz.2.33 Not tainted 6.12.0-rc2-syzkaller-00006-g87d6aab2389e #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024			
CVE-2024-50035	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ppp: fix ppp_async_encode() illegal access syzbot reported an issue in ppp_async_encode() [1] In this case, pppoe_sendmsg() is called with a zero size. Then ppp_async_encode() is called with an empty skb. BUG: KMSAN: uninit-value in ppp_async_encode drivers/net/ppp/ppp_async.c:545 [inline] BUG: KMSAN: uninit-value in ppp_async_push+0xb4f/0x2660 drivers/net/ppp/ppp_async.c:675 ppp_async_encode drivers/net/ppp/ppp_async.c:545 [inline] ppp_async_push+0xb4f/0x2660 drivers/net/ppp/ppp_async.c:675 ppp_async_send+0x130/0x1b0 drivers/net/ppp/ppp_async.c:634 ppp_channel_bridge_input drivers/net/ppp/ppp_generic.c:2280 [inline] ppp_input+0x1f1/0xe60 drivers/net/ppp/ppp_generic.c:2304 pppoe_rcv_core+0x1d3/0x720 drivers/net/ppp/pppoe.c:379 sk_backlog_rcv+0x13b/0x420 include/net/sock.h:1113 __release_sock+0x1da/0x330 net/core/sock.c:3072 release_sock+0x6b/0x250 net/core/sock.c:3626 pppoe_sendmsg+0x2b8/0xb90 drivers/net/ppp/pppoe.c:903 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 ____sys_sendmsg+0x903/0xb60 net/socket.c:2602 ___sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0	2024-10-21	7.1	High

		arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4092 [inline] slab_alloc_node mm/slub.c:4135 [inline] kmem_cache_alloc_node_noprof+0x6bf/0xb80 mm/slub.c:4187 kmalloc_reserve+0x13d/0x4a0 net/core/skbuff.c:587 __alloc_skb+0x363/0x7b0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1322 [inline] sock_wmalloc+0xfe/0x1a0 net/core/sock.c:2732 pppoe_sendmsg+0x3a7/0xb90 drivers/net/ppp/pppoe.c:867 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 ____sys_sendmsg+0x903/0xb60 net/socket.c:2602 ___sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f CPU: 1 UID: 0 PID: 5411 Comm: syz.1.14 Not tainted 6.12.0-rc1-syzkaller-00165-g360c1f1f24c6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024			
CVE-2024-47741	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix race setting file private on concurrent lseek using same fd When doing concurrent lseek(2) system calls against the same file descriptor, using multiple threads belonging to the same process, we have	2024-10-21	7	High

		a short time window where a race happens and can result in a memory leak. The race happens like this: 1) A program opens a file descriptor for a file and then spawns two threads (with the pthreads library for example), lets call them task A and task B; 2) Task A calls lseek with SEEK_DATA or SEEK_HOLE and ends up at file.c:find_desired_extent() while holding a read lock on the inode; 3) At the start of find_desired_extent(), it extracts the file's private_data pointer into a local variable named 'private', which has a value of NULL; 4) Task B also calls lseek with SEEK_DATA or SEEK_HOLE, locks the inode in shared mode and enters file.c:find_desired_extent(), where it also extracts file->private_data into its local variable 'private', which has a NULL value; 5) Because it saw a NULL file private, task A allocates a private structure and assigns to the file structure; 6) Task B also saw a NULL file private so it also allocates its own file private and then assigns it to the same file structure, since both tasks are using the same file descriptor. At this point we leak the private structure allocated by task A. Besides the memory leak, there's also the detail that both tasks end up using the same cached state record in the private structure (struct btrfs_file_private::llseek_cached_state), which can result in a use-after-free problem since one task can free it while the other is			

		still using it (only one task took a reference count on it). Also, sharing the cached state is not a good idea since it could result in incorrect results in the future - right now it should not be a problem because it end ups being used only in extent-io-tree.c:count_range_bits() where we do range validation before using the cached state. Fix this by protecting the private assignment and check of a file while holding the inode's spinlock and keep track of the task that allocated the private, so that it's used only by that task in order to prevent user-after-free issues with the cached state record as well as potentially using it incorrectly in the future.			
CVE-2024-47747	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: seeq: Fix use after free vulnerability in ether3 Driver Due to Race Condition In the ether3_probe function, a timer is initialized with a callback function ether3_ledoff, bound to &prev(dev)->timer. Once the timer is started, there is a risk of a race condition if the module or device is removed, triggering the ether3_remove function to perform cleanup. The sequence of operations that may lead to a UAF bug is as follows: CPU0 CPU1 | ether3_ledoff ether3_remove | free_netdev(dev); | put_devic | kfree(dev); | | ether3_outw(priv(dev)->regs.config2 |= CFG2_CTRLO, REG_CONFIG2); | // use dev Fix it by ensuring that the timer is canceled before proceeding with the cleanup in ether3_remove.	2024-10-21	7	High

<table>
<tr><td>CVE-2024-49855</td><td>linux - multiple products</td><td>In the Linux kernel, the following vulnerability has been resolved:

nbd: fix race between timeout and normal completion

If request timetout is handled by nbd_requeue_cmd(), normal completion has to be stopped for avoiding to complete this requeued request, other use-after-free can be triggered.

Fix the race by clearing NBD_CMD_INFLIGHT in nbd_requeue_cmd(), meantime make sure that cmd->lock is grabbed for clearing the flag and the requeue.</td><td>2024-10-21</td><td>7</td><td>High</td></tr>
<tr><td>CVE-2024-49874</td><td>linux - multiple products</td><td>In the Linux kernel, the following vulnerability has been resolved:

i3c: master: svc: Fix use after free vulnerability in svc_i3c_master Driver Due to Race Condition

In the svc_i3c_master_probe function, &master->hj_work is bound with svc_i3c_master_hj_work, &master->ibi_work is bound with svc_i3c_master_ibi_work. And svc_i3c_master_ibi_work can start the hj_work, svc_i3c_master_irq_handler can start the ibi_work.

If we remove the module which will call svc_i3c_master_remove to make cleanup, it will free master->base through i3c_master_unregister while the work mentioned above will be used. The sequence of operations that may lead to a UAF bug is as follows:

CPU0   CPU1

                              | svc_i3c_master_hj_work
svc_i3c_master_remove |
i3c_master_unregister(&master->base)|
device_unregister(&master->dev) |
device_release |
//free master->base |
                              | i3c_master_do_daa(&master->base)
                              | //use master->base</td><td>2024-10-21</td><td>7</td><td>High</td></tr>
</table>

		Fix it by ensuring that the work is canceled before proceeding with the cleanup in svc_i3c_master_remove.			
CVE-2024-49903	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uaf in dbFreeBits [syzbot reported] === BUG: KASAN: slab-use-after-free in __mutex_lock_common kernel/locking/mutex.c:587 [inline] BUG: KASAN: slab-use-after-free in __mutex_lock+0xfe/0xd70 kernel/locking/mutex.c:752 Read of size 8 at addr ffff8880229254b0 by task syz-executor357/5216 CPU: 0 UID: 0 PID: 5216 Comm: syz-executor357 Not tainted 6.11.0-rc3-syzkaller-00156-gd7a5aa4b3c00 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 06/27/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 __mutex_lock_common kernel/locking/mutex.c:587 [inline] __mutex_lock+0xfe/0xd70 kernel/locking/mutex.c:752 dbFreeBits+0x7ea/0xd90 fs/jfs/jfs_dmap.c:2390 dbFreeDmap fs/jfs/jfs_dmap.c:2089 [inline] dbFree+0x35b/0x680 fs/jfs/jfs_dmap.c:409 dbDiscardAG+0x8a9/0xa20 fs/jfs/jfs_dmap.c:1650 jfs_ioc_trim+0x433/0x670 fs/jfs/jfs_discard.c:100 jfs_ioctl+0x2d0/0x3e0 fs/jfs/ioctl.c:131 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xfc/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 Freed by task 5218:	2024-10-21	7	High

		kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:579 poison_slab_object+0xe0/0x150 mm/kasan/common.c:240 __kasan_slab_free+0x37/0x60 mm/kasan/common.c:256 kasan_slab_free include/linux/kasan.h:184 [inline] slab_free_hook mm/slub.c:2252 [inline] slab_free mm/slub.c:4473 [inline] kfree+0x149/0x360 mm/slub.c:4594 dbUnmount+0x11d/0x190 fs/jfs/jfs_dmap.c:278 jfs_mount_rw+0x4ac/0x6a0 fs/jfs/jfs_mount.c:247 jfs_remount+0x3d1/0x6b0 fs/jfs/super.c:454 reconfigure_super+0x445/0x880 fs/super.c:1083 vfs_cmd_reconfigure fs/fsopen.c:263 [inline] vfs_fsconfig_locked fs/fsopen.c:292 [inline] __do_sys_fsconfig fs/fsopen.c:473 [inline] __se_sys_fsconfig+0xb6e/0xf80 fs/fsopen.c:345 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 [Analysis] There are two paths (dbUnmount and jfs_ioc_trim) that generate race condition when accessing bmap, which leads to the occurrence of uaf. Use the lock s_umount to synchronize them, in order to avoid uaf caused by race condition.			
CVE-2024-49981	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: venus: fix use after free bug in venus_remove due to race condition in venus_probe, core->work is bound with venus_sys_error_handler, which is used to handle error. The code use core->sys_err_done to make sync work. The core->work is started in venus_event_notify. If we call venus_remove, there might be an unfished work. The possible sequence is as follows:	2024-10-21	7	High

		CPU0 CPU1 |venus_sys_error_handler venus_remove | hfi_destroy | venus_hfi_destroy | kfree(hdev); | |hfi_reinit |venus_hfi_queues_reinit |//use hdev Fix it by canceling the work in venus_remove.			
CVE-2022-48988	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: memcg: fix possible use-after-free in memcg_write_event_control() memcg_write_event_control() accesses the dentry->d_name of the specified control fd to route the write call. As a cgroup interface file can't be renamed, it's safe to access d_name as long as the specified file is a regular cgroup file. Also, as these cgroup interface files can't be removed before the directory, it's safe to access the parent too. Prior to 347c4a874710 ("memcg: remove cgroup_event->cft"), there was a call to __file_cft() which verified that the specified file is a regular cgroupfs file before further accesses. The cftype pointer returned from __file_cft() was no longer necessary and the commit inadvertently dropped the file type check with it allowing any file to slip through. With the invarients broken, the d_name and parent accesses can now race against renames and removals of arbitrary files and cause use-after-free's. Fix the bug by resurrecting the file type check in __file_cft(). Now that cgroupfs is implemented through kernfs, checking the file operations needs to go through a layer of indirection. Instead, let's check the superblock and dentry type.	2024-10-21	7	High

CVE-2022-49001	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: riscv: fix race when vmap stack overflow Currently, when detecting vmap stack overflow, riscv firstly switches to the so called shadow stack, then use this shadow stack to call the get_overflow_stack() to get the overflow stack. However, there's a race here if two or more harts use the same shadow stack at the same time. To solve this race, we introduce spin_shadow_stack atomic var, which will be swap between its own address and 0 in atomic way, when the var is set, it means the shadow_stack is being used; when the var is cleared, it means the shadow_stack isn't being used. [Palmer: Add AQ to the swap, and also some comments.]	2024-10-21	7	High
CVE-2024-50036	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: do not delay dst_entries_add() in dst_release() dst_entries_add() uses per-cpu data that might be freed at netns dismantle from ip6_route_net_exit() calling dst_entries_destroy() Before ip6_route_net_exit() can be called, we release all the dsts associated with this netns, via calls to dst_release(), which waits an rcu grace period before calling dst_destroy() dst_entries_add() use in dst_destroy() is racy, because dst_entries_destroy() could have been called already. Decrementing the number of dsts must happen sooner. Notes:	2024-10-21	7	High

		1) in CONFIG_XFRM case, dst_destroy() can call dst_release_immediate(child), this might also cause UAF if the child does not have DST_NOCOUNT set. IPSEC maintainers might take a look and see how to address this. 2) There is also discussion about removing this count of dst, which might happen in future kernels.			
CVE-2024-50059	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntb: ntb_hw_switchtec: Fix use after free vulnerability in switchtec_ntb_remove due to race condition In the switchtec_ntb_add function, it can call switchtec_ntb_init_sndev function, then &sndev->check_link_status_work is bound with check_link_status_work. switchtec_ntb_link_notification may be called to start the work. If we remove the module which will call switchtec_ntb_remove to make cleanup, it will free sndev through kfree(sndev), while the work mentioned above will be used. The sequence of operations that may lead to a UAF bug is as follows: CPU0 CPU1 | check_link_status_work switchtec_ntb_remove | kfree(sndev); | | if (sndev->link_force_down) | // use sndev Fix it by ensuring that the work is canceled before proceeding with the cleanup in switchtec_ntb_remove.	2024-10-21	7	High
CVE-2024-50061	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: i3c: master: cdns: Fix use after free vulnerability in cdns_i3c_master Driver Due to Race Condition	2024-10-21	7	High

		In the cdns_i3c_master_probe function, &master->hj_work is bound with cdns_i3c_master_hj. And cdns_i3c_master_interrupt can call cnds_i3c_master_demux_ibis function to start the work. If we remove the module which will call cdns_i3c_master_remove to make cleanup, it will free master->base through i3c_master_unregister while the work mentioned above will be used. The sequence of operations that may lead to a UAF bug is as follows: CPU0 CPU1 | cdns_i3c_master_hj cdns_i3c_master_remove | i3c_master_unregister(&master->base) | device_unregister(&master->dev) | device_release | //free master->base | | i3c_master_do_daa(&master->base) | //use master->base Fix it by ensuring that the work is canceled before proceeding with the cleanup in cdns_i3c_master_remove.			
CVE-2024-47902	siemens - intermesh_7177_hybrid_2.0_subscriber	A vulnerability has been identified in InterMesh 7177 Hybrid 2.0 Subscriber (All versions < V8.2.12), InterMesh 7707 Fire Subscriber (All versions < V7.2.12 only if the IP interface is enabled (which is not the default configuration)). The web server of affected devices does not authenticate GET requests that execute specific commands (such as `ping`) on operating system level.	2024-10-23	6.9	Medium
CVE-2024-47903	siemens - intermesh_7177_hybrid_2.0_subscriber	A vulnerability has been identified in InterMesh 7177 Hybrid 2.0 Subscriber (All versions < V8.2.12), InterMesh 7707 Fire Subscriber (All versions < V7.2.12 only if the IP interface is enabled (which is not the default configuration)). The web server of affected devices allows to write arbitrary files to the web server's DocumentRoot directory.	2024-10-23	6.9	Medium
CVE-2024-20485	cisco - multiple products	A vulnerability in the VPN web server of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an authenticated, local attacker to execute arbitrary code with root-level privileges. Administrator-level privileges are required to exploit this vulnerability.	2024-10-23	6.7	Medium

		This vulnerability is due to improper validation of a specific file when it is read from system flash memory. An attacker could exploit this vulnerability by restoring a crafted backup file to an affected device. A successful exploit could allow the attacker to execute arbitrary code on the affected device after the next reload of the device, which could alter system behavior. Because the injected code could persist across device reboots, Cisco has raised the Security Impact Rating (SIR) of this advisory from Medium to High.			
CVE-2024-44141	apple - macOS	The issue was addressed with improved checks. This issue is fixed in macOS Sonoma 14.6. A person with physical access to an unlocked Mac may be able to gain root code execution.	2024-10-24	6.6	Medium
CVE-2024-47692	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nfsd: return -EINVAL when namelen is 0 When we have a corrupted main.sqlite in /var/lib/nfs/nfsdcld/, it may result in namelen being 0, which will cause memdup_user() to return ZERO_SIZE_PTR. When we access the name.data that has been assigned the value of ZERO_SIZE_PTR in nfs4_client_to_reclaim(), null pointer dereference is triggered. [T1205] == [T1205] BUG: KASAN: null-ptr-deref in nfs4_client_to_reclaim+0xe9/0x260 [T1205] Read of size 1 at addr 0000000000000010 by task nfsdcld/1205 [T1205] [T1205] CPU: 11 PID: 1205 Comm: nfsdcld Not tainted 5.10.0-00003-g2c1423731b8d #406 [T1205] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS ?-20190727_073836-buildvm-ppc64le-16.ppc.fedoraproject.org-3.fc31 04/01/2014 [T1205] Call Trace: [T1205] dump_stack+0x9a/0xd0 [T1205] ? nfs4_client_to_reclaim+0xe9/0x260 [T1205] __kasan_report.cold+0x34/0x84 [T1205] ? nfs4_client_to_reclaim+0xe9/0x260 [T1205] kasan_report+0x3a/0x50	2024-10-21	6.5	Medium

		[T1205] nfs4_client_to_reclaim+0xe9/0x260 [T1205] ? nfsd4_release_lockowner+0x410/0x410 [T1205] cld_pipe_downcall+0x5ca/0x760 [T1205] ? nfsd4_cld_tracking_exit+0x1d0/0x1d0 [T1205] ? down_write_killable_nested+0x170/0x170 [T1205] ? avc_policy_seqno+0x28/0x40 [T1205] ? selinux_file_permission+0x1b4/0x1e0 [T1205] rpc_pipe_write+0x84/0xb0 [T1205] vfs_write+0x143/0x520 [T1205] ksys_write+0xc9/0x170 [T1205] ? __ia32_sys_read+0x50/0x50 [T1205] ? ktime_get_coarse_real_ts64+0xfe/0x110 [T1205] ? ktime_get_coarse_real_ts64+0xa2/0x110 [T1205] do_syscall_64+0x33/0x40 [T1205] entry_SYSCALL_64_after_hwframe+0x67/0xd1 [T1205] RIP: 0033:0x7fdbdb761bc7 [T1205] Code: 0f 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b7 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 514 [T1205] RSP: 002b:00007fff8c4b7248 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 [T1205] RAX: ffffffffffffffda RBX: 000000000000042b RCX: 00007fdbdb761bc7 [T1205] RDX: 000000000000042b RSI: 00007fff8c4b75f0 RDI: 0000000000000008 [T1205] RBP: 00007fdbdb761bb0 R08: 0000000000000000 R09: 0000000000000001 [T1205] R10: 0000000000000000 R11: 0000000000000246 R12: 000000000000042b [T1205] R13: 0000000000000008 R14: 00007fff8c4b75f0 R15: 0000000000000000 [T1205] == Fix it by checking namelen.			
CVE-2024-47693	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: IB/core: Fix ib_cache_setup_one error flow cleanup When ib_cache_update return an error, we exit ib_cache_setup_one instantly with no proper cleanup, even though before this we had already successfully done gid_table_setup_one, that results in the kernel WARN below. Do proper cleanup using gid_table_cleanup_one	2024-10-21	6.5	Medium

		before returning the err in order to fix the issue. WARNING: CPU: 4 PID: 922 at drivers/infiniband/core/cache.c:806 gid_table_release_one+0x181/0x1a0 Modules linked in: CPU: 4 UID: 0 PID: 922 Comm: c_repro Not tainted 6.11.0-rc1+ #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02- prebuilt.qemu.org 04/01/2014 RIP: 0010:gid_table_release_one+0x181/0x1a0 Code: 44 8b 38 75 0c e8 2f cb 34 ff 4d 8b b5 28 05 00 00 e8 23 cb 34 ff 44 89 f9 89 da 4c 89 f6 48 c7 c7 d0 58 14 83 e8 4f de 21 ff <0f> 0b 4c 8b 75 30 e9 54 ff ff ff 48 8 3 c4 10 5b 5d 41 5c 41 5d 41 RSP: 0018:ffffc90002b835b0 EFLAGS: 00010286 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff811c8527 RDX: 0000000000000000 RSI: ffffffff811c8534 RDI: 0000000000000001 RBP: ffff8881011b3d00 R08: ffff88810b3abe00 R09: 205d303839303631 R10: 666572207972746e R11: 72746e6520444947 R12: 0000000000000001 R13: ffff888106390000 R14: ffff8881011f2110 R15: 0000000000000001 FS: 00007fecc3b70800(0000) GS:ffff88813bd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000340 CR3: 000000010435a001 CR4: 00000000003706b0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? show_regs+0x94/0xa0 ? __warn+0x9e/0x1c0 ? gid_table_release_one+0x181/0x1a0 ? report_bug+0x1f9/0x340 ? gid_table_release_one+0x181/0x1a0 ? handle_bug+0xa2/0x110 ? exc_invalid_op+0x31/0xa0 ? asm_exc_invalid_op+0x16/0x20 ? __warn_printk+0xc7/0x180 ? __warn_printk+0xd4/0x180 ? gid_table_release_one+0x181/0x1a0			

		ib_device_release+0x71/0xe0 ? __pfx_ib_device_release+0x10/0x10 device_release+0x44/0xd0 kobject_put+0x135/0x3d0 put_device+0x20/0x30 rxe_net_add+0x7d/0xa0 rxe_newlink+0xd7/0x190 nldev_newlink+0x1b0/0x2a0 ? __pfx_nldev_newlink+0x10/0x10 rdma_nl_rcv_msg+0x1ad/0x2e0 rdma_nl_rcv_skb.constprop.0+0x176/0x210 netlink_unicast+0x2de/0x400 netlink_sendmsg+0x306/0x660 __sock_sendmsg+0x110/0x120 ____sys_sendmsg+0x30e/0x390 ___sys_sendmsg+0x9b/0xf0 ? kstrtouint+0x6e/0xa0 ? kstrtouint_from_user+0x7c/0xb0 ? get_pid_task+0xb0/0xd0 ? proc_fail_nth_write+0x5b/0x140 ? __fget_light+0x9a/0x200 ? preempt_count_add+0x47/0xa0 __sys_sendmsg+0x61/0xd0 do_syscall_64+0x50/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e			
CVE-2024-47726	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to wait dio completion It should wait all existing dio write IOs before block removal, otherwise, previous direct write IO may overwrite data in the block which may be reused by other inode.	2024-10-21	6.5	Medium
CVE-2024-50311	redhat - openshift_container_platform	A denial of service (DoS) vulnerability was found in OpenShift. This flaw allows attackers to exploit the GraphQL batching functionality. The vulnerability arises when multiple queries can be sent within a single request, enabling an attacker to submit a request containing thousands of aliases in one query. This issue causes excessive resource consumption, leading to application unavailability for legitimate users.	2024-10-22	6.5	Medium
CVE-2024-46903	trendmicro - multiple products	A vulnerability in Trend Micro Deep Discovery Inspector (DDI) versions 5.8 and above could allow an attacker to disclose sensitive information affected installations.	2024-10-22	6.5	Medium

		Please note: an attacker must first obtain the ability to execute low-privileged code on the target system in order to exploit this vulnerability.			
CVE-2024-20340	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to perform an SQL injection attack against an affected device. To exploit this vulnerability, an attacker must have a valid account on the device with the role of Security Approver, Intrusion Admin, Access Admin, or Network Admin. This vulnerability is due to insufficient validation of user-supplied input. An attacker could exploit this vulnerability by sending a crafted HTTP request to the web-based management interface of an affected device. A successful exploit could allow the attacker to read the contents of databases on the affected device and also obtain limited read access to the underlying operating system.	2024-10-23	6.5	Medium
CVE-2024-20374	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker with Administrator-level privileges to execute arbitrary commands on the underlying operating system. This vulnerability is due to insufficient input validation of certain HTTP request parameters that are sent to the web-based management interface. An attacker could exploit this vulnerability by authenticating to the Cisco FMC web-based management interface and sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to execute commands as the root user on the affected device. To exploit this vulnerability, an attacker would need Administrator-level credentials.	2024-10-23	6.5	Medium
CVE-2024-20379	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to read arbitrary files from the underlying operating system.	2024-10-23	6.5	Medium

		This vulnerability exists because the web-based management interface does not properly validate user-supplied input. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to read arbitrary files on the underlying operating system of the affected device. The attacker would need valid user credentials to exploit this vulnerability.			
CVE-2024-20471	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software could allow an authenticated, remote attacker to conduct SQL injection attacks on an affected system. This vulnerability exists because the web-based management interface does not validate user input adequately. An attacker could exploit this vulnerability by authenticating to the application as an Administrator and sending crafted SQL queries to an affected system. A successful exploit could allow the attacker to obtain unauthorized data from the database and make changes to the system. To exploit this vulnerability, an attacker would need Administrator-level privileges.	2024-10-23	6.5	Medium
CVE-2024-20472	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software could allow an authenticated, remote attacker to conduct SQL injection attacks on an affected system. This vulnerability exists because the web-based management interface does not validate user input adequately. An attacker could exploit this vulnerability by authenticating to the application as an Administrator and sending crafted SQL queries to an affected system. A successful exploit could allow the attacker to obtain unauthorized data from the database and make changes to the system. To exploit this vulnerability, an attacker would need Administrator-level privileges.	2024-10-23	6.5	Medium
CVE-2024-20473	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software could allow an authenticated, remote attacker to conduct SQL injection attacks on an affected system.	2024-10-23	6.5	Medium

		This vulnerability exists because the web-based management interface does not validate user input adequately. An attacker could exploit this vulnerability by authenticating to the application as an Administrator and sending crafted SQL queries to an affected system. A successful exploit could allow the attacker to obtain unauthorized data from the database and make changes to the system. To exploit this vulnerability, an attacker would need Administrator-level privileges.			
CVE-2024-20474	cisco - multiple products	A vulnerability in Internet Key Exchange version 2 (IKEv2) processing of Cisco Secure Client Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) of Cisco Secure Client. This vulnerability is due to an integer underflow condition. An attacker could exploit this vulnerability by sending a crafted IKEv2 packet to an affected system. A successful exploit could allow the attacker to cause Cisco Secure Client Software to crash, resulting in a DoS condition on the client software. Note: Cisco Secure Client Software releases 4.10 and earlier were known as Cisco AnyConnect Secure Mobility Client.	2024-10-23	6.5	Medium
CVE-2024-20482	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to elevate privileges on an affected device. To exploit this vulnerability, an attacker must have a valid account on the device that is configured with a custom read-only role. This vulnerability is due to insufficient validation of role permissions in part of the web-based management interface. An attacker could exploit this vulnerability by performing a write operation on the affected part of the web-based management interface. A successful exploit could allow the attacker to modify certain parts of the configuration.	2024-10-23	6.5	Medium
CVE-2024-47481	dell - multiple products	Dell Data Lakehouse, version(s) 1.0.0.0, 1.1.0., contain(s) an Improper Access Control vulnerability. An unauthenticated attacker with adjacent network access could potentially exploit this vulnerability, leading to Denial of service.	2024-10-25	6.5	Medium

CVE-2024-8980	liferay - multiple products	The Script Console in Liferay Portal 7.0.0 through 7.4.3.101, and Liferay DXP 2023.Q3.1 through 2023.Q3.4, 7.4 GA through update 92, 7.3 GA through update 35, 7.2 GA through fix pack 20, 7.1 GA through fix pack 28, 7.0 GA through fix pack 102 and 6.2 GA through fix pack 173 does not sufficiently protect against Cross-Site Request Forgery (CSRF) attacks, which allows remote attackers to execute arbitrary Groovy script via a crafted URL or a XSS vulnerability.	2024-10-22	6.1	Medium
CVE-2024-20273	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	6.1	Medium
CVE-2024-20275	cisco - Cisco Firepower Management Center	A vulnerability in the cluster backup feature of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to execute arbitrary commands on the underlying operating system. This vulnerability is due to insufficient validation of user data that is supplied through the web-based management interface. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to execute arbitrary operating system commands on the affected device. To exploit this vulnerability, an attacker would need valid credentials for a user account with at least the role of Network Administrator. In addition, the attacker would need to persuade a legitimate user to initiate a cluster backup on the affected device.	2024-10-23	6.1	Medium
CVE-2024-20341	cisco - multiple products	A vulnerability in the VPN web client services feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a browser that is accessing an affected device. This vulnerability is due to improper validation of user-	2024-10-23	6.1	Medium

		supplied input to application endpoints. An attacker could exploit this vulnerability by persuading a user to follow a link designed to submit malicious input to the affected application. A successful exploit could allow the attacker to execute arbitrary HTML or script code in the browser in the context of the web services page.			
[CVE-2024-20372](https://nvd.nist.gov/vuln/detail/CVE-2024-20372)	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	6.1	Medium
[CVE-2024-20382](https://nvd.nist.gov/vuln/detail/CVE-2024-20382)	cisco - multiple products	A vulnerability in the VPN web client services feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a browser that is accessing an affected device. This vulnerability is due to improper validation of user-supplied input to application endpoints. An attacker could exploit this vulnerability by persuading a user to follow a link designed to submit malicious input to the affected application. A successful exploit could allow the attacker to execute arbitrary HTML or script code in the browser in the context of the web services page.	2024-10-23	6.1	Medium
[CVE-2024-20415](https://nvd.nist.gov/vuln/detail/CVE-2024-20415)	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	6.1	Medium
[CVE-2024-20370](https://nvd.nist.gov/vuln/detail/CVE-2024-20370)	cisco - multiple products	A vulnerability in the Cisco FXOS CLI feature on specific hardware platforms for Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow	2024-10-23	6	Medium

		an authenticated, local attacker to elevate their administrative privileges to root. The attacker would need valid administrative credentials on the device to exploit this vulnerability. This vulnerability exists because certain system configurations and executable files have insecure storage and permissions. An attacker could exploit this vulnerability by authenticating on the device and then performing a series of steps that includes downloading malicious system files and accessing the Cisco FXOS CLI to configure the attack. A successful exploit could allow the attacker to obtain root access on the device.			
CVE-2024-20331	cisco - multiple products	A vulnerability in the session authentication functionality of the Remote Access SSL VPN feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to prevent users from authenticating. This vulnerability is due to insufficient entropy in the authentication process. An attacker could exploit this vulnerability by determining the handle of an authenticating user and using it to terminate their authentication session. A successful exploit could allow the attacker to force a user to restart the authentication process, preventing a legitimate user from establishing remote access VPN sessions.	2024-10-23	5.9	Medium
CVE-2024-10295	red hat - Red Hat 3scale API Management Platform 2	A flaw was found in Gateway. Sending a non-base64 'basic' auth with special characters can cause APICast to incorrectly authenticate a request. A malformed basic authentication header containing special characters bypasses authentication and allows unauthorized access to the backend. This issue can occur due to a failure in the base64 decoding process, which causes APICast to skip the rest of the authentication checks and proceed with routing the request upstream.	2024-10-24	5.9	Medium
CVE-2024-38314	ibm - Maximo Application Suite - Monitor Component	IBM Maximo Application Suite - Monitor Component 8.10, 8.11, and 9.0 could disclose information in the form of the hard-coded cryptographic key to an attacker that has compromised environment.	2024-10-24	5.9	Medium
CVE-2024-20297	cisco - multiple products	A vulnerability in the AnyConnect firewall for Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass a	2024-10-23	5.8	Medium

		configured access control list (ACL) and allow traffic that should have been denied to flow through an affected device. This vulnerability is due to a logic error in populating group ACLs when an AnyConnect client establishes a new session toward an affected device. An attacker could exploit this vulnerability by establishing an AnyConnect connection to the affected device. A successful exploit could allow the attacker to bypass configured ACL rules.			
CVE-2024-20299	cisco - multiple products	A vulnerability in the AnyConnect firewall for Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass a configured access control list (ACL) and allow traffic that should have been denied to flow through an affected device. This vulnerability is due to a logic error in populating group ACLs when an AnyConnect client establishes a new session toward an affected device. An attacker could exploit this vulnerability by establishing an AnyConnect connection to the affected device. A successful exploit could allow the attacker to bypass configured ACL rules.	2024-10-23	5.8	Medium
CVE-2024-20342	cisco - Cisco Firepower Threat Defense Software	Multiple Cisco products are affected by a vulnerability in the rate filtering feature of the Snort detection engine that could allow an unauthenticated, remote attacker to bypass a configured rate limiting filter. This vulnerability is due to an incorrect connection count comparison. An attacker could exploit this vulnerability by sending traffic through an affected device at a rate that exceeds a configured rate filter. A successful exploit could allow the attacker to successfully bypass the rate filter. This could allow unintended traffic to enter the network protected by the affected device.	2024-10-23	5.8	Medium
CVE-2024-20384	cisco - multiple products	A vulnerability in the Network Service Group (NSG) feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass a configured access control list (ACL) and allow traffic that should be denied to flow through an affected device. This vulnerability is due to a logic error that occurs when NSG ACLs are populated on an affected device. An attacker could exploit this vulnerability by establishing a connection to the affected device. A	2024-10-23	5.8	Medium

		successful exploit could allow the attacker to bypass configured ACL rules.			
CVE-2024-20407	cisco - Cisco Firepower Threat Defense Software	A vulnerability in the interaction between the TCP Intercept feature and the Snort 3 detection engine on Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass configured policies on an affected system. Devices that are configured with Snort 2 are not affected by this vulnerability. This vulnerability is due to a logic error when handling embryonic (half-open) TCP connections. An attacker could exploit this vulnerability by sending a crafted traffic pattern through an affected device. A successful exploit could allow unintended traffic to enter the network protected by the affected device.	2024-10-23	5.8	Medium
CVE-2024-20431	cisco - Cisco Firepower Threat Defense Software	A vulnerability in the geolocation access control feature of Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to bypass an access control policy. This vulnerability is due to improper assignment of geolocation data. An attacker could exploit this vulnerability by sending traffic through an affected device. A successful exploit could allow the attacker to bypass a geolocation-based access control policy and successfully send traffic to a protected device.	2024-10-23	5.8	Medium
CVE-2024-20481	cisco - multiple products	A vulnerability in the Remote Access VPN (RAVPN) service of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) of the RAVPN service. This vulnerability is due to resource exhaustion. An attacker could exploit this vulnerability by sending a large number of VPN authentication requests to an affected device. A successful exploit could allow the attacker to exhaust resources, resulting in a DoS of the RAVPN service on the affected device. Depending on the impact of the attack, a reload of the device may be required to restore the RAVPN service. Services that are not related to VPN are not affected. Cisco Talos discussed these attacks in the blog post Large-scale brute-force activity targeting VPNs, SSH services with commonly used login credentials.	2024-10-23	5.8	Medium

CVE-2024-47677	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: exfat: resolve memory leak from exfat_create_upcase_table() If exfat_load_upcase_table reaches end and returns -EINVAL, allocated memory doesn't get freed and while exfat_load_default_upcase_table allocates more memory, leading to a memory leak. Here's link to syzkaller crash report illustrating this issue: https://syzkaller.appspot.com/text?tag=CrashReport&x=1406c201980000	2024-10-21	5.5	Medium
CVE-2024-47678	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: icmp: change the order of rate limits ICMP messages are ratelimited : After the blamed commits, the two rate limiters are applied in this order: 1) host wide ratelimit (icmp_global_allow()) 2) Per destination ratelimit (inetpeer based) In order to avoid side-channels attacks, we need to apply the per destination check first. This patch makes the following change : 1) icmp_global_allow() checks if the host wide limit is reached. But credits are not yet consumed. This is deferred to 3) 2) The per destination limit is checked/updated. This might add a new node in inetpeer tree. 3) icmp_global_consume() consumes tokens if prior operations succeeded. This means that host wide ratelimit is still effective in keeping inetpeer tree small even under DDOS.	2024-10-21	5.5	Medium

		As a bonus, I removed icmp_global.lock as the fast path can use a lock-free operation.			

CVE-2024-47680	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		f2fs: check discard support for conventional zones
		
		As the helper function f2fs_bdev_support_discard() shows, f2fs checks if the target block devices support discard by calling bdev_max_discard_sectors() and bdev_is_zoned(). This check works well for most cases, but it does not work for conventional zones on zoned block devices. F2fs assumes that zoned block devices support discard, and calls __submit_discard_cmd(). When __submit_discard_cmd() is called for sequential write required zones, it works fine since __submit_discard_cmd() issues zone reset commands instead of discard commands. However, when __submit_discard_cmd() is called for conventional zones, __blkdev_issue_discard() is called even when the devices do not support discard.
		
		The inappropriate __blkdev_issue_discard() call was not a problem before the commit 30f1e7241422 ("block: move discard checks into the ioctl handler") because __blkdev_issue_discard() checked if the target devices support discard or not. If not, it returned EOPNOTSUPP. After the commit, __blkdev_issue_discard() no longer checks it. It always returns zero and sets NULL to the given bio pointer. This NULL pointer triggers f2fs_bug_on() in __submit_discard_cmd(). The BUG is recreated with the commands below at the umount step, where /dev/nullb0 is a zoned null_blk with 5GB total size, 128MB zone size and 10 conventional zones.
		
		$ mkfs.f2fs -f -m /dev/nullb0
		$ mount /dev/nullb0 /mnt
		$ for ((i=0;i<5;i++)); do dd if=/dev/zero of=/mnt/test
			2024-10-21	5.5	Medium

		bs=65536 count=1600 conv=fsync; done $ umount /mnt To fix the BUG, avoid the inappropriate __blkdev_issue_discard() call. When discard is requested for conventional zones, check if the device supports discard or not. If not, return EOPNOTSUPP.			
CVE-2024-47681	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: fix NULL pointer dereference in mt7996_mcu_sta_bfer_he Fix the NULL pointer dereference in mt7996_mcu_sta_bfer_he routine adding an sta interface to the mt7996 driver. Found by code review.	2024-10-21	5.5	Medium
CVE-2024-47683	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip Recompute DSC Params if no Stream on Link [why] Encounter NULL pointer dereference uner mst + dsc setup. BUG: kernel NULL pointer dereference, address: 0000000000000008 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 4 PID: 917 Comm: sway Not tainted 6.3.9-arch1-1 #1 124dc55df4f5272ccb409f39ef4872fc2b3376a2 Hardware name: LENOVO 20NKS01Y00/20NKS01Y00, BIOS R12ET61W(1.31) 07/28/2022 RIP: 0010:drm_dp_atomic_find_time_slots+0x5e/0x260 [drm_display_helper] Code: 01 00 00 48 8b 85 60 05 00 00 48 63 80 88 00 00 00 3b 43 28 0f 8d 2e 01 00 00 48 8b 53 30 48 8d 04 80 48 8d 04 c2 48 8b 40 18 <48> 8> RSP: 0018:ffff960cc2df77d8 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffff8afb87e81280 RCX: 0000000000000224 RDX: ffff8afb9ee37c00 RSI: ffff8afb8da1a578 RDI: ffff8afb87e81280 RBP: ffff8afb83d67000 R08: 0000000000000001			

		R09: ffff8afb9652f850 R10: ffff960cc2df7908 R11: 0000000000000002 R12: 0000000000000000 R13: ffff8afb8d7688a0 R14: ffff8afb8da1a578 R15: 0000000000000224 FS: 00007f4dac35ce00(0000) GS:ffff8afe30b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000008 CR3: 000000010ddc6000 CR4: 00000000003506e0 Call Trace: <TASK> ? __die+0x23/0x70 ? page_fault_oops+0x171/0x4e0 ? plist_add+0xbe/0x100 ? exc_page_fault+0x7c/0x180 ? asm_exc_page_fault+0x26/0x30 ? drm_dp_atomic_find_time_slots+0x5e/0x260 [drm_display_helper 0e67723696438d8e02b741593dd50d80b44c2026] ? drm_dp_atomic_find_time_slots+0x28/0x260 [drm_display_helper 0e67723696438d8e02b741593dd50d80b44c2026] compute_mst_dsc_configs_for_link+0x2ff/0xa40 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] ? fill_plane_buffer_attributes+0x419/0x510 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] compute_mst_dsc_configs_for_state+0x1e1/0x250 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] amdgpu_dm_atomic_check+0xecd/0x1190 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] drm_atomic_check_only+0x5c5/0xa40 drm_mode_atomic_ioctl+0x76e/0xbc0 [how] dsc recompute should be skipped if no mode change detected on the new request. If detected, keep checking whether the stream is already on current state or not.			
CVE-2024-47684	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tcp: check skb is non-NULL in tcp_rto_delta_us()	2024-10-21	5.5	Medium

		We have some machines running stock Ubuntu 20.04.6 which is their 5.4.0-174-generic kernel that are running ceph and recently hit a null ptr dereference in tcp_rearm_rto(). Initially hitting it from the TLP path, but then later we also saw it getting hit from the RACK case as well. Here are examples of the oops messages we saw in each of those cases: Jul 26 15:05:02 rx [11061395.780353] BUG: kernel NULL pointer dereference, address: 0000000000000020 Jul 26 15:05:02 rx [11061395.787572] #PF: supervisor read access in kernel mode Jul 26 15:05:02 rx [11061395.792971] #PF: error_code(0x0000) - not-present page Jul 26 15:05:02 rx [11061395.798362] PGD 0 P4D 0 Jul 26 15:05:02 rx [11061395.801164] Oops: 0000 [#1] SMP NOPTI Jul 26 15:05:02 rx [11061395.805091] CPU: 0 PID: 9180 Comm: msgr-worker-1 Tainted: G W 5.4.0-174-generic #193-Ubuntu Jul 26 15:05:02 rx [11061395.814996] Hardware name: Supermicro SMC 2x26 os-gen8 64C NVME-Y 256G/H12SSW-NTR, BIOS 2.5.V1.2U.NVMe.UEFI 05/09/2023 Jul 26 15:05:02 rx [11061395.825952] RIP: 0010:tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.830656] Code: 87 ca 04 00 00 00 5b 41 5c 41 5d 5d c3 c3 49 8b bc 24 40 06 00 00 eb 8d 48 bb cf f7 53 e3 a5 9b c4 20 4c 89 ef e8 0c fe 0e 00 <48> 8b 78 20 48 c1 ef 03 48 89 f8 41 8b bc 24 80 04 00 00 48 f7 e3 Jul 26 15:05:02 rx [11061395.849665] RSP: 0018:ffffb75d40003e08 EFLAGS: 00010246 Jul 26 15:05:02 rx [11061395.855149] RAX: 0000000000000000 RBX: 20c49ba5e353f7cf RCX: 0000000000000000 Jul 26 15:05:02 rx [11061395.862542] RDX: 0000000062177c30 RSI: 000000000000231c RDI: ffff9874ad283a60 Jul 26 15:05:02 rx [11061395.869933] RBP: ffffb75d40003e20 R08: 0000000000000000 R09: ffff987605e20aa8 Jul 26 15:05:02 rx [11061395.877318] R10: ffffb75d40003f00 R11: ffffb75d4460f740 R12: ffff9874ad283900 Jul 26 15:05:02 rx [11061395.884710] R13: ffff9874ad283a60 R14: ffff9874ad283980 R15:			

		ffff9874ad283d30 Jul 26 15:05:02 rx [11061395.892095] FS: 00007f1ef4a2e700(0000) GS:ffff987605e00000(0000) knlGS:0000000000000000 Jul 26 15:05:02 rx [11061395.900438] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 Jul 26 15:05:02 rx [11061395.906435] CR2: 0000000000000020 CR3: 0000003e450ba003 CR4: 0000000000760ef0 Jul 26 15:05:02 rx [11061395.913822] PKRU: 55555554 Jul 26 15:05:02 rx [11061395.916786] Call Trace: Jul 26 15:05:02 rx [11061395.919488] Jul 26 15:05:02 rx [11061395.921765] ? show_regs.cold+0x1a/0x1f Jul 26 15:05:02 rx [11061395.925859] ? __die+0x90/0xd9 Jul 26 15:05:02 rx [11061395.929169] ? no_context+0x196/0x380 Jul 26 15:05:02 rx [11061395.933088] ? ip6_protocol_deliver_rcu+0x4e0/0x4e0 Jul 26 15:05:02 rx [11061395.938216] ? ip6_sublist_rcv_finish+0x3d/0x50 Jul 26 15:05:02 rx [11061395.943000] ? __bad_area_nosemaphore+0x50/0x1a0 Jul 26 15:05:02 rx [11061395.947873] ? bad_area_nosemaphore+0x16/0x20 Jul 26 15:05:02 rx [11061395.952486] ? do_user_addr_fault+0x267/0x450 Jul 26 15:05:02 rx [11061395.957104] ? ipv6_list_rcv+0x112/0x140 Jul 26 15:05:02 rx [11061395.961279] ? __do_page_fault+0x58/0x90 Jul 26 15:05:02 rx [11061395.965458] ? do_page_fault+0x2c/0xe0 Jul 26 15:05:02 rx [11061395.969465] ? page_fault+0x34/0x40 Jul 26 15:05:02 rx [11061395.973217] ? tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.977313] ? tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.981408] tcp_send_loss_probe+0x10b/0x220 Jul 26 15:05:02 rx [11061395.985937] tcp_write_timer_handler+0x1b4/0x240 Jul 26 15:05:02 rx [11061395.990809] tcp_write_timer+0x9e/0xe0 Jul 26 15:05:02 rx [11061395.994814] ? tcp_write_timer_handler+0x240/0x240 Jul 26 15:05:02 rx [11061395.999866] call_timer_fn+0x32/0x130			

		Jul 26 15:05:02 rx [11061396.003782] __run_timers.part.0+0x180/0x280 Jul 26 15:05:02 rx [11061396.008309] ? recalibrate_cpu_khz+0x10/0x10 Jul 26 15:05:02 rx [11061396.012841] ? native_x2apic_icr_write+0x30/0x30 Jul 26 15:05:02 rx [11061396.017718] ? lapic_next_even ---truncated---			
CVE-2024-47687	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vdpa/mlx5: Fix invalid mr resource destroy Certain error paths from mlx5_vdpa_dev_add() can end up releasing mr resources which never got initialized in the first place. This patch adds the missing check in mlx5_vdpa_destroy_mr_resources() to block releasing non-initialized mr resources. Reference trace: mlx5_core 0000:08:00.2: mlx5_vdpa_dev_add:3274:(pid 2700) warning: No mac address provisioned? BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 140216067 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 8 PID: 2700 Comm: vdpa Kdump: loaded Not tainted 5.14.0-496.el9.x86_64 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:vhost_iotlb_del_range+0xf/0xe0 [vhost_iotlb] Code: [...] RSP: 0018:ff1c823ac23077f0 EFLAGS: 00010246 RAX: ffffffffc1a21a60 RBX: ffffffff899567a0 RCX: 0000000000000000 RDX: ffffffffffffffff RSI: 0000000000000000 RDI: 0000000000000000 RBP: ff1bda1f7c21e800 R08: 0000000000000000 R09: ff1c823ac2307670 R10: ff1c823ac2307668 R11: ffffffff8a9e7b68 R12: 0000000000000000 R13: 0000000000000000 R14: ff1bda1f43e341a0	2024-10-21	5.5	Medium

		R15: 00000000ffffffea FS: 00007f56eba7c740(0000) GS:ff1bda269f800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 0000000104d90001 CR4: 0000000000771ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: ? show_trace_log_lvl+0x1c4/0x2df ? show_trace_log_lvl+0x1c4/0x2df ? mlx5_vdpa_free+0x3d/0x150 [mlx5_vdpa] ? __die_body.cold+0x8/0xd ? page_fault_oops+0x134/0x170 ? __irq_work_queue_local+0x2b/0xc0 ? irq_work_queue+0x2c/0x50 ? exc_page_fault+0x62/0x150 ? asm_exc_page_fault+0x22/0x30 ? __pfx_mlx5_vdpa_free+0x10/0x10 [mlx5_vdpa] ? vhost_iotlb_del_range+0xf/0xe0 [vhost_iotlb] mlx5_vdpa_free+0x3d/0x150 [mlx5_vdpa] vdpa_release_dev+0x1e/0x50 [vdpa] device_release+0x31/0x90 kobject_cleanup+0x37/0x130 mlx5_vdpa_dev_add+0x2d2/0x7a0 [mlx5_vdpa] vdpa_nl_cmd_dev_add_set_doit+0x277/0x4c0 [vdpa] genl_family_rcv_msg_doit+0xd9/0x130 genl_family_rcv_msg+0x14d/0x220 ? __pfx_vdpa_nl_cmd_dev_add_set_doit+0x10/0x10 [vdpa] ? _copy_to_user+0x1a/0x30 ? move_addr_to_user+0x4b/0xe0 genl_rcv_msg+0x47/0xa0 ? __import_iovec+0x46/0x150 ? __pfx_genl_rcv_msg+0x10/0x10 netlink_rcv_skb+0x54/0x100 genl_rcv+0x24/0x40 netlink_unicast+0x245/0x370 netlink_sendmsg+0x206/0x440 __sys_sendto+0x1dc/0x1f0 ? do_read_fault+0x10c/0x1d0 ? do_pte_missing+0x10d/0x190 __x64_sys_sendto+0x20/0x30			

		do_syscall_64+0x5c/0xf0 ? __count_memcg_events+0x4f/0xb0 ? mm_account_fault+0x6c/0x100 ? handle_mm_fault+0x116/0x270 ? do_user_addr_fault+0x1d6/0x6a0 ? do_syscall_64+0x6b/0xf0 ? clear_bhb_loop+0x25/0x80 ? clear_bhb_loop+0x25/0x80 ? clear_bhb_loop+0x25/0x80 ? clear_bhb_loop+0x25/0x80 ? clear_bhb_loop+0x25/0x80 entry_SYSCALL_64_after_hwframe+0x78/0x80			
CVE-2024-47688	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: driver core: Fix a potential null-ptr-deref in module_add_driver() Inject fault while probing of-fpga-region, if kasprintf() fails in module_add_driver(), the second sysfs_remove_link() in exit path will cause null-ptr-deref as below because kernfs_name_hash() will call strlen() with NULL driver_name. Fix it by releasing resources based on the exit path sequence. KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] 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 [dfffffc000000000] address between user and kernel address ranges Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP Dumping ftrace buffer: (ftrace buffer empty) Modules linked in: of_fpga_region(+) fpga_region fpga_bridge cfg80211 rfkill 8021q garp mrp stp llc ipv6 [last unloaded: of_fpga_region] CPU: 2 UID: 0 PID: 2036 Comm: modprobe Not	2024-10-21	5.5	Medium

		tainted 6.11.0-rc2-g6a0e38264012 #295 Hardware name: linux,dummy-virt (DT) pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : strlen+0x24/0xb0 lr : kernfs_name_hash+0x1c/0xc4 sp : ffffffc081f97380 x29: ffffffc081f97380 x28: ffffffc081f97b90 x27: ffffff80c821c2a0 x26: ffffffedac0be418 x25: 0000000000000000 x24: ffffff80c09d2000 x23: 0000000000000000 x22: 0000000000000000 x21: 0000000000000000 x20: 0000000000000000 x19: 0000000000000000 x18: 0000000000001840 x17: 0000000000000000 x16: 0000000000000000 x15: 1ffffff8103f2e42 x14: 00000000f1f1f1f1 x13: 0000000000000004 x12: ffffffb01812d61d x11: 1ffffff01812d61c x10: ffffffb01812d61c x9 : dfffffc000000000 x8 : 0000004fe7ed29e4 x7 : ffffff80c096b0e7 x6 : 0000000000000001 x5 : ffffff80c096b0e0 x4 : 1ffffffdb990efa2 x3 : 0000000000000000 x2 : 0000000000000000 x1 : dfffffc000000000 x0 : 0000000000000000 Call trace: strlen+0x24/0xb0 kernfs_name_hash+0x1c/0xc4 kernfs_find_ns+0x118/0x2e8 kernfs_remove_by_name_ns+0x80/0x100 sysfs_remove_link+0x74/0xa8 module_add_driver+0x278/0x394 bus_add_driver+0x1f0/0x43c driver_register+0xf4/0x3c0 __platform_driver_register+0x60/0x88 of_fpga_region_init+0x20/0x1000 [of_fpga_region] do_one_initcall+0x110/0x788 do_init_module+0x1dc/0x5c8 load_module+0x3c38/0x4cac init_module_from_file+0xd4/0x128 idempotent_init_module+0x2cc/0x528 __arm64_sys_finit_module+0xac/0x100 invoke_syscall+0x6c/0x258 el0_svc_common.constprop.0+0x160/0x22c do_el0_svc+0x44/0x5c el0_svc+0x48/0xb8 el0t_64_sync_handler+0x13c/0x158 el0t_64_sync+0x190/0x194 Code: f2fbffe1 a90157f4 12000802 aa0003f5			

		(38e16861) ---[end trace 0000000000000000]--- Kernel panic - not syncing: Oops: Fatal exception			
CVE-2024-47690	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: get rid of online repaire on corrupted directory syzbot reports a f2fs bug as below: kernel BUG at fs/f2fs/inode.c:896! RIP: 0010:f2fs_evict_inode+0x1598/0x15c0 fs/f2fs/inode.c:896 Call Trace: evict+0x532/0x950 fs/inode.c:704 dispose_list fs/inode.c:747 [inline] evict_inodes+0x5f9/0x690 fs/inode.c:797 generic_shutdown_super+0x9d/0x2d0 fs/super.c:627 kill_block_super+0x44/0x90 fs/super.c:1696 kill_f2fs_super+0x344/0x690 fs/f2fs/super.c:4898 deactivate_locked_super+0xc4/0x130 fs/super.c:473 cleanup_mnt+0x41f/0x4b0 fs/namespace.c:1373 task_work_run+0x24f/0x310 kernel/task_work.c:228 ptrace_notify+0x2d2/0x380 kernel/signal.c:2402 ptrace_report_syscall include/linux/ptrace.h:415 [inline] ptrace_report_syscall_exit include/linux/ptrace.h:477 [inline] syscall_exit_work+0xc6/0x190 kernel/entry/common.c:173 syscall_exit_to_user_mode_prepare kernel/entry/common.c:200 [inline] __syscall_exit_to_user_mode_work kernel/entry/common.c:205 [inline] syscall_exit_to_user_mode+0x279/0x370 kernel/entry/common.c:218 do_syscall_64+0x100/0x230 arch/x86/entry/common.c:89 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0010:f2fs_evict_inode+0x1598/0x15c0 fs/f2fs/inode.c:896 Online repaire on corrupted directory in f2fs_lookup() can generate dirty data/meta while racing w/ readonly remount, it may leave dirty inode after filesystem becomes readonly, however, checkpoint() will skips flushing dirty inode in a state of readonly mode, result in	2024-10-21	5.5	Medium

		above panic. Let's get rid of online repaire in f2fs_lookup(), and leave the work to fsck.f2fs.			
CVE-2024-47694	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix UMR pd cleanup on error flow of driver init The cited commit moves the pd allocation from function mlx5r_umr_resource_cleanup() to a new function mlx5r_umr_cleanup(). So the fix in commit [1] is broken. In error flow, will hit panic [2]. Fix it by checking pd pointer to avoid panic if it is NULL; [1] RDMA/mlx5: Fix UMR cleanup on error flow of driver init [2] [347.567063] infiniband mlx5_0: Couldn't register device with driver model [347.591382] BUG: kernel NULL pointer dereference, address: 0000000000000020 [347.593438] #PF: supervisor read access in kernel mode [347.595176] #PF: error_code(0x0000) - not-present page [347.596962] PGD 0 P4D 0 [347.601361] RIP: 0010:ib_dealloc_pd_user+0x12/0xc0 [ib_core] [347.604171] RSP: 0018:ffff888106293b10 EFLAGS: 00010282 [347.604834] RAX: 0000000000000000 RBX: 000000000000000e RCX: 0000000000000000 [347.605672] RDX: ffff888106293ad0 RSI: 0000000000000000 RDI: 0000000000000000 [347.606529] RBP: 0000000000000000 R08: ffff888106293ae0 R09: ffff888106293ae0 [347.607379] R10: 0000000000000a06 R11: 0000000000000000 R12: 0000000000000000 [347.608224] R13: ffffffffa0704dc0 R14: 0000000000000001 R15: 0000000000000001 [347.609067] FS: 00007fdc720cd9c0(0000) GS:ffff88852c880000(0000) knlGS:0000000000000000 [347.610094] CS: 0010 DS: 0000 ES: 0000 CR0:	2024-10-21	5.5	Medium

		0000000080050033 [347.610727] CR2: 0000000000000020 CR3: 0000000103012003 CR4: 0000000000370eb0 [347.611421] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [347.612113] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [347.612804] Call Trace: [347.613130] <TASK> [347.613417] ? __die+0x20/0x60 [347.613793] ? page_fault_oops+0x150/0x3e0 [347.614243] ? free_msg+0x68/0x80 [mlx5_core] [347.614840] ? cmd_exec+0x48f/0x11d0 [mlx5_core] [347.615359] ? exc_page_fault+0x74/0x130 [347.615808] ? asm_exc_page_fault+0x22/0x30 [347.616273] ? ib_dealloc_pd_user+0x12/0xc0 [ib_core] [347.616801] mlx5r_umr_cleanup+0x23/0x90 [mlx5_ib] [347.617365] mlx5_ib_stage_pre_ib_reg_umr_cleanup+0x36/0x40 [mlx5_ib] [347.618025] __mlx5_ib_add+0x96/0xd0 [mlx5_ib] [347.618539] mlx5r_probe+0xe9/0x310 [mlx5_ib] [347.619032] ? kernfs_add_one+0x107/0x150 [347.619478] ? __mlx5_ib_add+0xd0/0xd0 [mlx5_ib] [347.619984] auxiliary_bus_probe+0x3e/0x90 [347.620448] really_probe+0xc5/0x3a0 [347.620857] __driver_probe_device+0x80/0x160 [347.621325] driver_probe_device+0x1e/0x90 [347.621770] __driver_attach+0xec/0x1c0 [347.622213] ? __device_attach_driver+0x100/0x100 [347.622724] bus_for_each_dev+0x71/0xc0 [347.623151] bus_add_driver+0xed/0x240 [347.623570] driver_register+0x58/0x100 [347.623998] __auxiliary_driver_register+0x6a/0xc0 [347.624499] ? driver_register+0xae/0x100 [347.624940] ? 0xffffffffa0893000 [347.625329] mlx5_ib_init+0x16a/0x1e0 [mlx5_ib] [347.625845] do_one_initcall+0x4a/0x2a0 [347.626273] ? gcov_event+0x2e2/0x3a0 [347.626706] do_init_module+0x8a/0x260 [347.627126] init_module_from_file+0x8b/0xd0 [347.627596] __x64_sys_finit_module+0x1ca/0x2f0 [347.628089] do_syscall_64+0x4c/0x100			

CVE-2024-47699	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential null-ptr-deref in nilfs_btree_insert() Patch series "nilfs2: fix potential issues with empty b-tree nodes". This series addresses three potential issues with empty b-tree nodes that can occur with corrupted filesystem images, including one recently discovered by syzbot. This patch (of 3): If a b-tree is broken on the device, and the b-tree height is greater than 2 (the level of the root node is greater than 1) even if the number of child nodes of the b-tree root is 0, a NULL pointer dereference occurs in nilfs_btree_prepare_insert(), which is called from nilfs_btree_insert(). This is because, when the number of child nodes of the b-tree root is 0, nilfs_btree_do_lookup() does not set the block buffer head in any of path[x].bp_bh, leaving it as the initial value of NULL, but if the level of the b-tree root node is greater than 1, nilfs_btree_get_nonroot_node(), which accesses the buffer memory of path[x].bp_bh, is called. Fix this issue by adding a check to nilfs_btree_root_broken(), which performs sanity checks when reading the root node from the device, to detect this inconsistency. Thanks to Lizhi Xu for trying to solve the bug and clarifying the cause early on.	2024-10-21	5.5	Medium
CVE-2024-47700	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: check stripe size compatibility on remount as	2024-10-21	5.5	Medium

		well We disable stripe size in __ext4_fill_super if it is not a multiple of the cluster ratio however this check is missed when trying to remount. This can leave us with cases where stripe < cluster_ratio after remount:set making EXT4_B2C(sbi->s_stripe) become 0 that can cause some unforeseen bugs like divide by 0. Fix that by adding the check in remount path as well.			
CVE-2024-47702	linux - multiple products	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;	2024-10-21	5.5	Medium

		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-2024-47703	linux - multiple products	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.	2024-10-21	5.5	Medium

		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-2024-47704	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check link_res->hpo_dp_link_enc before using it [WHAT & HOW] Functions dp_enable_link_phy and dp_disable_link_phy can pass link_res without initializing hpo_dp_link_enc and it is necessary to check for null before dereferencing. This fixes 2 FORWARD_NULL issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-47705	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: block: fix potential invalid pointer dereference in blk_add_partition The blk_add_partition() function initially used a single if-condition (IS_ERR(part)) to check for errors when adding a partition. This was modified to handle the specific case of -ENXIO separately, allowing the function to proceed without logging the error in this case. However, this change unintentionally left a path where md_autodetect_dev() could be called without confirming that part is a valid pointer. This commit separates the error handling logic by splitting the initial if-condition, improving code readability and handling specific error scenarios explicitly. The function now distinguishes the general error case from -ENXIO without altering the existing behavior of md_autodetect_dev() calls.	2024-10-21	5.5	Medium

<table>
<tr><td>CVE-2024-47706</td><td>linux - multiple products</td><td>In the Linux kernel, the following vulnerability has been resolved:

block, bfq: fix possible UAF for bfqq->bic with merge chain

1) initial state, three tasks:

Process 1   Process 2 Process 3
(BIC1)   (BIC2) (BIC3)
 | ? | ? | ?
 | | | | | |
 V | V | V |
 bfqq1   bfqq2   bfqq3
process ref: 1   1   1

2) bfqq1 merged to bfqq2:

Process 1   Process 2 Process 3
(BIC1)   (BIC2) (BIC3)
 | | | ?
 \--------------\| | |
               V   V |
 bfqq1--------->bfqq2   bfqq3
process ref: 0   2   1

3) bfqq2 merged to bfqq3:

Process 1   Process 2 Process 3
(BIC1)   (BIC2) (BIC3)
here -> ? | |
 \--------------\ \-------------\|
               V   V
 bfqq1--------->bfqq2---------->bfqq3
process ref: 0   1   3

In this case, IO from Process 1 will get bfqq2 from BIC1 first, and then
get bfqq3 through merge chain, and finially handle IO by bfqq3.
Howerver, current code will think bfqq2 is owned by BIC1, like initial
state, and set bfqq2->bic to BIC1.

bfq_insert_request
-> by Process 1
 bfqq = bfq_init_rq(rq)
  bfqq = bfq_get_bfqq_handle_split
   bfqq = bic_to_bfqq
   -> get bfqq2 from BIC1
 bfqq->ref++</td><td>2024-10-21</td><td>5.5</td><td>Medium</td></tr>
</table>

		rq->elv.priv[0] = bic rq->elv.priv[1] = bfqq if (bfqq_process_refs(bfqq) == 1) bfqq->bic = bic -> record BIC1 to bfqq2 __bfq_insert_request new_bfqq = bfq_setup_cooperator -> get bfqq3 from bfqq2->new_bfqq bfqq_request_freed(bfqq) new_bfqq->ref++ rq->elv.priv[1] = new_bfqq -> handle IO by bfqq3 Fix the problem by checking bfqq is from merge chain fist. And this might fix a following problem reported by our syzkaller(unreproducible): == BUG: KASAN: slab-use-after-free in bfq_do_early_stable_merge block/bfq-iosched.c:5692 [inline] BUG: KASAN: slab-use-after-free in bfq_do_or_sched_stable_merge block/bfq-iosched.c:5805 [inline] BUG: KASAN: slab-use-after-free in bfq_get_queue+0x25b0/0x2610 block/bfq-iosched.c:5889 Write of size 1 at addr ffff888123839eb8 by task kworker/0:1H/18595 CPU: 0 PID: 18595 Comm: kworker/0:1H Tainted: G L 6.6.0-07439-gba2303cacfda #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 Workqueue: kblockd blk_mq_requeue_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x91/0xf0 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:364 [inline] print_report+0x10d/0x610 mm/kasan/report.c:475 kasan_report+0x8e/0xc0 mm/kasan/report.c:588 bfq_do_early_stable_merge block/bfq-iosched.c:5692 [inline] bfq_do_or_sched_stable_merge block/bfq-iosched.c:5805 [inline] bfq_get_queue+0x25b0/0x2610 block/bfq-			

		iosched.c:5889 bfq_get_bfqq_handle_split+0x169/0x5d0 block/bfq-iosched.c:6757 bfq_init_rq block/bfq-iosched.c:6876 [inline] bfq_insert_request block/bfq-iosched.c:6254 [inline] bfq_insert_requests+0x1112/0x5cf0 block/bfq-iosched.c:6304 blk_mq_insert_request+0x290/0x8d0 block/blk-mq.c:2593 blk_mq_requeue_work+0x6bc/0xa70 block/blk-mq.c:1502 process_one_work kernel/workqueue.c:2627 [inline] process_scheduled_works+0x432/0x13f0 kernel/workqueue.c:2700 worker_thread+0x6f2/0x1160 kernel/workqueue.c:2781 kthread+0x33c/0x440 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:305 </TASK> Allocated by task 20776: kasan_save_stack+0x20/0x40 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 __kasan_slab_alloc+0x87/0x90 mm/kasan/common.c:328 kasan_slab_alloc include/linux/kasan.h:188 [inline] slab_post_alloc_hook mm/slab.h:763 [inline] slab_alloc_node mm/slub.c:3458 [inline] kmem_cache_alloc_node+0x1a4/0x6f0 mm/slub.c:3503 ioc_create_icq block/blk-ioc.c:370 [inline] ---truncated---			
CVE-2024-47707	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ipv6: avoid possible NULL deref in rt6_uncached_list_flush_dev() Blamed commit accidentally removed a check for rt->rt6i_idev being NULL, as spotted by syzbot: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range	2024-10-21	5.5	Medium

		[0x0000000000000000-0x0000000000000007] CPU: 1 UID: 0 PID: 10998 Comm: syz-executor Not tainted 6.11.0-rc6-syzkaller-00208-g625403177711 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010:rt6_uncached_list_flush_dev net/ipv6/route.c:177 [inline] RIP: 0010:rt6_disable_ip+0x33e/0x7e0 net/ipv6/route.c:4914 Code: 41 80 3c 04 00 74 0a e8 90 d0 9b f7 48 8b 7c 24 08 48 8b 07 48 89 44 24 10 4c 89 f0 48 c1 e8 03 48 b9 00 00 00 00 00 fc ff df <80> 3c 08 00 74 08 4c 89 f7 e8 64 d0 9b f7 48 8b 44 24 18 49 39 06 RSP: 0018:ffffc900047374e0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 1ffff1100fdf8f33 RCX: dffffc0000000000 RDX: 0000000000000000 RSI: 0000000000000004 RDI: ffff88807efc78c0 RBP: ffffc900047375d0 R08: 0000000000000003 R09: fffff520008e6e8c R10: dffffc0000000000 R11: fffff520008e6e8c R12: 1ffff1100fdf8f18 R13: ffff88807efc7998 R14: 0000000000000000 R15: ffff88807efc7930 FS: 0000000000000000(0000) GS:ffff8880b8900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020002a80 CR3: 0000000022f62000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> addrconf_ifdown+0x15d/0x1bd0 net/ipv6/addrconf.c:3856 addrconf_notify+0x3cb/0x1020 notifier_call_chain+0x19f/0x3e0 kernel/notifier.c:93 call_netdevice_notifiers_extack net/core/dev.c:2032 [inline] call_netdevice_notifiers net/core/dev.c:2046 [inline] unregister_netdevice_many_notify+0xd81/0x1c40 net/core/dev.c:11352 unregister_netdevice_many net/core/dev.c:11414 [inline] unregister_netdevice_queue+0x303/0x370 net/core/dev.c:11289			

		unregister_netdevice include/linux/netdevice.h:3129 [inline] __tun_detach+0x6b9/0x1600 drivers/net/tun.c:685 tun_detach drivers/net/tun.c:701 [inline] tun_chr_close+0x108/0x1b0 drivers/net/tun.c:3510 __fput+0x24a/0x8a0 fs/file_table.c:422 task_work_run+0x24f/0x310 kernel/task_work.c:228 exit_task_work include/linux/task_work.h:40 [inline] do_exit+0xa2f/0x27f0 kernel/exit.c:882 do_group_exit+0x207/0x2c0 kernel/exit.c:1031 __do_sys_exit_group kernel/exit.c:1042 [inline] __se_sys_exit_group kernel/exit.c:1040 [inline] __x64_sys_exit_group+0x3f/0x40 kernel/exit.c:1040 x64_sys_call+0x2634/0x2640 arch/x86/include/generated/asm/syscalls_64.h:232 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: 0033:0x7f1acc77def9 Code: Unable to access opcode bytes at 0x7f1acc77decf. RSP: 002b:00007ffeb26fa738 EFLAGS: 00000246 ORIG_RAX: 00000000000000e7 RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007f1acc77def9 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000043 RBP: 00007f1acc7dd508 R08: 00007ffeb26f84d7 R09: 0000000000000003 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000001 R13: 0000000000000003 R14: 00000000ffffffff R15: 00007ffeb26fa8e0 </TASK> Modules linked in: ---[end trace 0000000000000000]--- RIP: 0010:rt6_uncached_list_flush_dev net/ipv6/route.c:177 [inline] RIP: 0010:rt6_disable_ip+0x33e/0x7e0 net/ipv6/route.c:4914 Code: 41 80 3c 04 00 74 0a e8 90 d0 9b f7 48 8b 7c 24 08 48 8b 07 48 89 44 24 10 4c 89 f0 48 c1 e8 03 48 b9 00 00 00 00 00 fc ff df <80> 3c 08 00 74 08 4c 89 f7 e8 64 d0 9b f7 48 8b 44 24 18 49 39 06 RSP: 0018:ffffc900047374e0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 1ffff1100fdf8f33 RCX: dffffc0000000000 RDX: 0000000000000000 RSI: 0000000000000004			

		RDI: ffff88807efc78c0 R ---truncated---			
CVE-2024-47708	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: netkit: Assign missing bpf_net_context During the introduction of struct bpf_net_context handling for XDP-redirect, the netkit driver has been missed, which also requires it because NETKIT_REDIRECT invokes skb_do_redirect() which is accessing the per-CPU variables. Otherwise we see the following crash: BUG: kernel NULL pointer dereference, address: 0000000000000038 bpf_redirect() netkit_xmit() dev_hard_start_xmit() Set the bpf_net_context before invoking netkit_xmit() program within the netkit driver.	2024-10-21	5.5	Medium
CVE-2024-47709	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: can: bcm: Clear bo->bcm_proc_read after remove_proc_entry(). syzbot reported a warning in bcm_release(). [0] The blamed change fixed another warning that is triggered when connect() is issued again for a socket whose connect()ed device has been unregistered. However, if the socket is just close()d without the 2nd connect(), the remaining bo->bcm_proc_read triggers unnecessary remove_proc_entry() in bcm_release(). Let's clear bo->bcm_proc_read after remove_proc_entry() in bcm_notify(). [0] name '4986'	2024-10-21	5.5	Medium

		WARNING: CPU: 0 PID: 5234 at fs/proc/generic.c:711 remove_proc_entry+0x2e7/0x5d0 fs/proc/generic.c:711 Modules linked in: CPU: 0 UID: 0 PID: 5234 Comm: syz-executor606 Not tainted 6.11.0-rc5-syzkaller-00178-g5517ae241919 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010:remove_proc_entry+0x2e7/0x5d0 fs/proc/generic.c:711 Code: ff eb 05 e8 cb 1e 5e ff 48 8b 5c 24 10 48 c7 c7 e0 f7 aa 8e e8 2a 38 8e 09 90 48 c7 c7 60 3a 1b 8c 48 89 de e8 da 42 20 ff 90 <0f> 0b 90 90 48 8b 44 24 18 48 c7 44 24 40 0e 36 e0 45 49 c7 04 07 RSP: 0018:ffffc9000345fa20 EFLAGS: 00010246 RAX: 2a2d0aee2eb64600 RBX: ffff888032f1f548 RCX: ffff888029431e00 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffffc9000345fb08 R08: ffffffff8155b2f2 R09: 1ffff1101710519a R10: dffffc0000000000 R11: ffffed101710519b R12: ffff888011d38640 R13: 0000000000000004 R14: 0000000000000000 R15: dffffc0000000000 FS: 0000000000000000(0000) GS:ffff8880b8800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fcfb52722f0 CR3: 000000000e734000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> bcm_release+0x250/0x880 net/can/bcm.c:1578 __sock_release net/socket.c:659 [inline] sock_close+0xbc/0x240 net/socket.c:1421 __fput+0x24a/0x8a0 fs/file_table.c:422 task_work_run+0x24f/0x310 kernel/task_work.c:228 exit_task_work include/linux/task_work.h:40 [inline] do_exit+0xa2f/0x27f0 kernel/exit.c:882 do_group_exit+0x207/0x2c0 kernel/exit.c:1031 __do_sys_exit_group kernel/exit.c:1042 [inline] __se_sys_exit_group kernel/exit.c:1040 [inline] __x64_sys_exit_group+0x3f/0x40 kernel/exit.c:1040 x64_sys_call+0x2634/0x2640			

		arch/x86/include/generated/asm/syscalls_64.h:232 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: 0033:0x7fcfb51ee969 Code: Unable to access opcode bytes at 0x7fcfb51ee93f. RSP: 002b:00007ffce0109ca8 EFLAGS: 00000246 ORIG_RAX: 00000000000000e7 RAX: ffffffffffffffda RBX: 0000000000000001 RCX: 00007fcfb51ee969 RDX: 000000000000003c RSI: 00000000000000e7 RDI: 0000000000000001 RBP: 00007fcfb526f3b0 R08: ffffffffffffffb8 R09: 0000555500000000 R10: 0000555500000000 R11: 0000000000000246 R12: 00007fcfb526f3b0 R13: 0000000000000000 R14: 00007fcfb5271ee0 R15: 00007fcfb51bf160 </TASK>			
CVE-2024-47710	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sock_map: Add a cond_resched() in sock_hash_free() Several syzbot soft lockup reports all have in common sock_hash_free() If a map with a large number of buckets is destroyed, we need to yield the cpu when needed.	2024-10-21	5.5	Medium
CVE-2024-47712	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: wilc1000: fix potential RCU dereference issue in wilc_parse_join_bss_param In the `wilc_parse_join_bss_param` function, the TSF field of the `ies` structure is accessed after the RCU read-side critical section is unlocked. According to RCU usage rules, this is illegal. Reusing this pointer can lead to unpredictable behavior, including accessing memory that has been updated or causing use-after-free issues. This possible bug was identified using a static analysis tool developed	2024-10-21	5.5	Medium

		by myself, specifically designed to detect RCU-related issues. To address this, the TSF value is now stored in a local variable `ies_tsf` before the RCU lock is released. The `param->tsf_lo` field is then assigned using this local variable, ensuring that the TSF value is safely accessed.			

CVE-2024-47713	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		wifi: mac80211: use two-phase skb reclamation in ieee80211_do_stop()
		
		Since '__dev_queue_xmit()' should be called with interrupts enabled,
		the following backtrace:
		
		ieee80211_do_stop()
		 ...
		 spin_lock_irqsave(&local->queue_stop_reason_lock, flags)
		 ...
		 ieee80211_free_txskb()
		  ieee80211_report_used_skb()
		   ieee80211_report_ack_skb()
		    cfg80211_mgmt_tx_status_ext()
		     nl80211_frame_tx_status()
		      genlmsg_multicast_netns()
		       genlmsg_multicast_netns_filtered()
		        nlmsg_multicast_filtered()
		netlink_broadcast_filtered()
		  do_one_broadcast()
		   netlink_broadcast_deliver()
		    __netlink_sendskb()
		     netlink_deliver_tap()
		      __netlink_deliver_tap_skb()
		       dev_queue_xmit()
		        __dev_queue_xmit() ; with IRQS disabled
		 ...
		 spin_unlock_irqrestore(&local->queue_stop_reason_lock, flags)
		
		issues the warning (as reported by syzbot reproducer):
		
		WARNING: CPU: 2 PID: 5128 at kernel/softirq.c:362 __local_bh_enable_ip+0xc3/0x120
			2024-10-21	5.5	Medium

		Fix this by implementing a two-phase skb reclamation in 'ieee80211_do_stop()', where actual work is performed outside of a section with interrupts disabled.			
CVE-2024-47714	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: use hweight16 to get correct tx antenna The chainmask is u16 so using hweight8 cannot get correct tx_ant. Without this patch, the tx_ant of band 2 would be -1 and lead to the following issue: BUG: KASAN: stack-out-of-bounds in mt7996_mcu_add_sta+0x12e0/0x16e0 [mt7996e]	2024-10-21	5.5	Medium
CVE-2024-47715	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: fix oops on non-dbdc mt7986 mt7915_band_config() sets band_idx = 1 on the main phy for mt7986 with MT7975_ONE_ADIE or MT7976_ONE_ADIE. Commit 0335c034e726 ("wifi: mt76: fix race condition related to checking tx queue fill status") introduced a dereference of the phys array indirectly indexed by band_idx via wcid->phy_idx in mt76_wcid_cleanup(). This caused the following Oops on affected mt7986 devices: Unable to handle kernel read from unreadable memory at virtual address 0000000000000024 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 CM = 0, WnR = 0 user pgtable: 4k pages, 39-bit VAs, pgdp=0000000042545000 [0000000000000024] pgd=0000000000000000,	2024-10-21	5.5	Medium

		p4d=0000000000000000, pud=0000000000000000 Internal error: Oops: 0000000096000005 [#1] SMP Modules linked in: ... mt7915e mt76_connac_lib mt76 mac80211 cfg80211 ... CPU: 2 PID: 1631 Comm: hostapd Not tainted 5.15.150 #0 Hardware name: ZyXEL EX5700 (Telenor) (DT) pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : mt76_wcid_cleanup+0x84/0x22c [mt76] lr : mt76_wcid_cleanup+0x64/0x22c [mt76] sp : ffffffc00a803700 x29: ffffffc00a803700 x28: ffffff80008f7300 x27: ffffff80003f3c00 x26: ffffff80000a7880 x25: ffffffc008c26e00 x24: 0000000000000001 x23: ffffffc000a68114 x22: 0000000000000000 x21: ffffff8004172cc8 x20: ffffffc00a803748 x19: ffffff8004152020 x18: 0000000000000000 x17: 00000000000017c0 x16: ffffffc008ef5000 x15: 0000000000000be0 x14: ffffff8004172e28 x13: ffffff8004172e28 x12: 0000000000000000 x11: 0000000000000000 x10: ffffff8004172e30 x9 : ffffff8004172e28 x8 : 0000000000000000 x7 : ffffff8004156020 x6 : 0000000000000000 x5 : 0000000000000031 x4 : 0000000000000000 x3 : 0000000000000001 x2 : 0000000000000000 x1 : ffffff80008f7300 x0 : 0000000000000024 Call trace: mt76_wcid_cleanup+0x84/0x22c [mt76] __mt76_sta_remove+0x70/0xbc [mt76] mt76_sta_state+0x8c/0x1a4 [mt76] mt7915_eeprom_get_power_delta+0x11e4/0x23a0 [mt7915e] drv_sta_state+0x144/0x274 [mac80211] sta_info_move_state+0x1cc/0x2a4 [mac80211] sta_set_sinfo+0xaf8/0xc24 [mac80211] sta_info_destroy_addr_bss+0x4c/0x6c [mac80211] ieee80211_color_change_finish+0x1c08/0x1e70 [mac80211] cfg80211_check_station_change+0x1360/0x4710 [cfg80211] genl_family_rcv_msg_doit+0xb4/0x110 genl_rcv_msg+0xd0/0x1bc netlink_rcv_skb+0x58/0x120 genl_rcv+0x34/0x50			

		netlink_unicast+0x1f0/0x2ec netlink_sendmsg+0x198/0x3d0 ____sys_sendmsg+0x1b0/0x210 ___sys_sendmsg+0x80/0xf0 __sys_sendmsg+0x44/0xa0 __arm64_sys_sendmsg+0x20/0x30 invoke_syscall.constprop.0+0x4c/0xe0 do_el0_svc+0x40/0xd0 el0_svc+0x14/0x4c el0t_64_sync_handler+0x100/0x110 el0t_64_sync+0x15c/0x160 Code: d2800002 910092c0 52800023 f9800011 (885f7c01) ---[end trace 7e42dd9a39ed2281]--- Fix by using mt76_dev_phy() which will map band_idx to the correct phy for all hardware combinations.			
CVE-2024-47716	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ARM: 9410/1: vfp: Use asm volatile in fmrx/fmxr macros Floating point instructions in userspace can crash some arm kernels built with clang/LLD 17.0.6: BUG: unsupported FP instruction in kernel mode FPEXC == 0xc0000780 Internal error: Oops - undefined instruction: 0 [#1] ARM CPU: 0 PID: 196 Comm: vfp-reproducer Not tainted 6.10.0 #1 Hardware name: BCM2835 PC is at vfp_support_entry+0xc8/0x2cc LR is at do_undefinstr+0xa8/0x250 pc : [<c0101d50>] lr : [<c010a80c>] psr: a0000013 sp : dc8d1f68 ip : 60000013 fp : bedea19c r10: ec532b17 r9 : 00000010 r8 : 0044766c r7 : c0000780 r6 : ec532b17 r5 : c1c13800 r4 : dc8d1fb0 r3 : c10072c4 r2 : c0101c88 r1 : ec532b17 r0 : 0044766c Flags: NzCv IRQs on FIQs on Mode SVC_32 ISA ARM Segment none Control: 00c5387d Table: 0251c008 DAC: 00000051 Register r0 information: non-paged memory Register r1 information: vmalloc memory	2024-10-21	5.5	Medium

		Register r2 information: non-slab/vmalloc memory Register r3 information: non-slab/vmalloc memory Register r4 information: 2-page vmalloc region Register r5 information: slab kmalloc-cg-2k Register r6 information: vmalloc memory Register r7 information: non-slab/vmalloc memory Register r8 information: non-paged memory Register r9 information: zero-size pointer Register r10 information: vmalloc memory Register r11 information: non-paged memory Register r12 information: non-paged memory Process vfp-reproducer (pid: 196, stack limit = 0x61aaaf8b) Stack: (0xdc8d1f68 to 0xdc8d2000) 1f60: 0000081f b6f69300 0000000f c10073f4 c10072c4 dc8d1fb0 1f80: ec532b17 0c532b17 0044766c b6f9ccd8 00000000 c010a80c 00447670 60000010 1fa0: ffffffff c1c13800 00c5387d c0100f10 b6f68af8 00448fc0 00000000 bedea188 1fc0: bedea314 00000001 00448ebc b6f9d000 00447608 b6f9ccd8 00000000 bedea19c 1fe0: bede9198 bedea188 b6e1061c 0044766c 60000010 ffffffff 00000000 00000000 Call trace: [<c0101d50>] (vfp_support_entry) from [<c010a80c>] (do_undefinstr+0xa8/0x250) [<c010a80c>] (do_undefinstr) from [<c0100f10>] (__und_usr+0x70/0x80) Exception stack(0xdc8d1fb0 to 0xdc8d1ff8) 1fa0: b6f68af8 00448fc0 00000000 bedea188 1fc0: bedea314 00000001 00448ebc b6f9d000 00447608 b6f9ccd8 00000000 bedea19c 1fe0: bede9198 bedea188 b6e1061c 0044766c 60000010 ffffffff Code: 0a000061 e3877202 e594003c e3a09010 (eef16a10) ---[end trace 0000000000000000]--- Kernel panic - not syncing: Fatal exception in interrupt ---[end Kernel panic - not syncing: Fatal exception in interrupt]--- This is a minimal userspace reproducer on a Raspberry Pi Zero W: #include <stdio.h> #include <math.h> int main(void)			

		{
		 double v = 1.0;
		 printf("%fn", NAN + *(volatile double *)&v);
		 return 0;
		 }
		
		 Another way to consistently trigger the oops is:
		
		 calvin@raspberry-pi-zero-w ~$ python -c "import json"
		
		 The bug reproduces only when the kernel is built with DYNAMIC_DEBUG=n,
		 because the pr_debug() calls act as barriers even when not activated.
		
		 This is the output from the same kernel source built with the same
		 compiler and DYNAMIC_DEBUG=y, where the userspace reproducer works as
		 expected:
		
		 VFP: bounce: trigger ec532b17 fpexc c0000780
		 VFP: emulate: INST=0xee377b06 SCR=0x00000000
		 VFP: bounce: trigger eef1fa10 fpexc c0000780
		 VFP: emulate: INST=0xeeb40b40 SCR=0x00000000
		 VFP: raising exceptions 30000000
		
		 calvin@raspberry-pi-zero-w ~$ ./vfp-reproducer
		 nan
		
		 Crudely grepping for vmsr/vmrs instructions in the otherwise nearly
		 idential text for vfp_support_entry() makes the problem obvious:
		
		 vmlinux.llvm.good [0xc0101cb8] <+48>: vmrs r7, fpexc
		 vmlinux.llvm.good [0xc0101cd8] <+80>: vmsr fpexc, r0
		 vmlinux.llvm.good [0xc0101d20
		 ---truncated---			

CVE-2024-47717	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RISC-V: KVM: Don't zero-out PMU snapshot area before freeing data With the latest Linux-6.11-rc3, the below NULL pointer crash is observed when SBI PMU snapshot is enabled for the guest and	2024-10-21	5.5	Medium

		the guest is forcefully powered-off. Unable to handle kernel NULL pointer dereference at virtual address 0000000000000508 Oops [#1] Modules linked in: kvm CPU: 0 UID: 0 PID: 61 Comm: term-poll Not tainted 6.11.0-rc3-00018-g44d7178dd77a #3 Hardware name: riscv-virtio,qemu (DT) epc : __kvm_write_guest_page+0x94/0xa6 [kvm] ra : __kvm_write_guest_page+0x54/0xa6 [kvm] epc : ffffffff01590e98 ra : ffffffff01590e58 sp : ffff8f80001f39b0 gp : ffffffff81512a60 tp : ffffaf80024872c0 t0 : ffffaf800247e000 t1 : 00000000000007e0 t2 : 0000000000000000 s0 : ffff8f80001f39f0 s1 : 00007fff89ac4000 a0 : ffffffff015dd7e8 a1 : 0000000000000086 a2 : 0000000000000000 a3 : ffffaf8000000000 a4 : ffffaf80024882c0 a5 : 0000000000000000 a6 : ffffaf800328d780 a7 : 00000000000001cc s2 : ffffaf800197bd00 s3 : 00000000000828c4 s4 : ffffaf800248c000 s5 : ffffaf800247d000 s6 : 0000000000001000 s7 : 0000000000001000 s8 : 0000000000000000 s9 : 00007fff861fd500 s10: 0000000000000001 s11: 0000000000800000 t3 : 00000000000004d3 t4 : 00000000000004d3 t5 : ffffffff814126e0 t6 : ffffffff81412700 status: 0000000200000120 badaddr: 0000000000000508 cause: 000000000000000d [<ffffffff01590e98>] __kvm_write_guest_page+0x94/0xa6 [kvm] [<ffffffff015943a6>] kvm_vcpu_write_guest+0x56/0x90 [kvm] [<ffffffff015a175c>] kvm_pmu_clear_snapshot_area+0x42/0x7e [kvm] [<ffffffff015a1972>] kvm_riscv_vcpu_pmu_deinit.part.0+0xe0/0x14e [kvm] [<ffffffff015a2ad0>] kvm_riscv_vcpu_pmu_deinit+0x1a/0x24 [kvm] [<ffffffff0159b344>] kvm_arch_vcpu_destroy+0x28/0x4c [kvm] [<ffffffff0158e420>] kvm_destroy_vcpus+0x5a/0xda [kvm] [<ffffffff0159930c>]			

		kvm_arch_destroy_vm+0x14/0x28 [kvm] [<ffffffff01593260>] kvm_destroy_vm+0x168/0x2a0 [kvm] [<ffffffff015933d4>] kvm_put_kvm+0x3c/0x58 [kvm] [<ffffffff01593412>] kvm_vm_release+0x22/0x2e [kvm] Clearly, the kvm_vcpu_write_guest() function is crashing because it is being called from kvm_pmu_clear_snapshot_area() upon guest tear down. To address the above issue, simplify the kvm_pmu_clear_snapshot_area() to not zero-out PMU snapshot area from kvm_pmu_clear_snapshot_area() because the guest is anyway being tore down. The kvm_pmu_clear_snapshot_area() is also called when guest changes PMU snapshot area of a VCPU but even in this case the previous PMU snaphsot area must not be zeroed-out because the guest might have reclaimed the pervious PMU snapshot area for some other purpose.			
CVE-2024-47720	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for set_output_gamma in dcn30_set_output_transfer_func This commit adds a null check for the set_output_gamma function pointer in the dcn30_set_output_transfer_func function. Previously, set_output_gamma was being checked for nullity at line 386, but then it was being dereferenced without any nullity check at line 401. This could potentially lead to a null pointer dereference error if set_output_gamma is indeed null. To fix this, we now ensure that set_output_gamma is not null before dereferencing it. We do this by adding a nullity check for set_output_gamma before the call to set_output_gamma at line 401. If	2024-10-21	5.5	Medium

		set_output_gamma is null, we log an error message and do not call the
		function.
		
		This fix prevents a potential null pointer dereference error.
		
		drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn30/dcn30_hwseq.c:401 dcn30_set_output_transfer_func()
		error: we previously assumed 'mpc->funcs->set_output_gamma' could be null (see line 386)
		
		drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn30/dcn30_hwseq.c
		373 bool dcn30_set_output_transfer_func(struct dc *dc,
		374 struct pipe_ctx *pipe_ctx,
		375 const struct dc_stream_state *stream)
		376 {
		377 int mpcc_id = pipe_ctx->plane_res.hubp->inst;
		378 struct mpc *mpc = pipe_ctx->stream_res.opp->ctx->dc->res_pool->mpc;
		379 const struct pwl_params *params = NULL;
		380 bool ret = false;
		381
		382 /* program OGAM or 3DLUT only for the top pipe*/
		383 if (pipe_ctx->top_pipe == NULL) {
		384 /*program rmu shaper and 3dlut in MPC*/
		385 ret = dcn30_set_mpc_shaper_3dlut(pipe_ctx, stream);
		386 if (ret == false && mpc->funcs->set_output_gamma) {
		
		^^^^^^^^^^^^^^^^^^^^^^^^^^^^ If this is NULL
		
		387 if (stream->out_transfer_func.type == TF_TYPE_HWPWL)
		388 params = &stream->out_transfer_func.pwl;
		389 else if (pipe_ctx->stream->out_transfer_func.type ==
		390 TF_TYPE_DISTRIBUTED_POINTS &&
		391 cm3_helper_translate_curve_to_hw_format(
		392 &stream-			

		>out_transfer_func, 393 &mpc->blender_params, false)) 394 params = &mpc->blender_params; 395 /* there are no ROM LUTs in OUTGAM */ 396 if (stream->out_transfer_func.type == TF_TYPE_PREDEFINED) 397 BREAK_TO_DEBUGGER(); 398 } 399 } 400 --> 401 mpc->funcs->set_output_gamma(mpc, mpcc_id, params); ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ Then it will crash 402 return ret; 403 }			
[CVE-2023-52917]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ntb: intel: Fix the NULL vs IS_ERR() bug for debugfs_create_dir() The debugfs_create_dir() function returns error pointers. It never returns NULL. So use IS_ERR() to check it.	2024-10-21	5.5	Medium
[CVE-2024-47724]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: use work queue to process beacon tx event Commit 3a415daa3e8b ("wifi: ath11k: add P2P IE in beacon template") from Feb 28, 2024 (linux-next), leads to the following Smatch static checker warning: drivers/net/wireless/ath/ath11k/wmi.c:1742 ath11k_wmi_p2p_go_bcn_ie() warn: sleeping in atomic context The reason is that ath11k_bcn_tx_status_event() will directly call might sleep function ath11k_wmi_cmd_send() during RCU read-side critical sections. The call trace is like:	2024-10-21	5.5	Medium

		ath11k_bcn_tx_status_event() -> rcu_read_lock() -> ath11k_mac_bcn_tx_event() -> ath11k_mac_setup_bcn_tmpl() ...... -> ath11k_wmi_bcn_tmpl() -> ath11k_wmi_cmd_send() -> rcu_read_unlock() Commit 886433a98425 ("ath11k: add support for BSS color change") added the ath11k_mac_bcn_tx_event(), commit 01e782c89108 ("ath11k: fix warning of RCU usage for ath11k_mac_get_arvif_by_vdev_id()") added the RCU lock to avoid warning but also introduced this BUG. Use work queue to avoid directly calling ath11k_mac_bcn_tx_event() during RCU critical sections. No need to worry about the deletion of vif because cancel_work_sync() will drop the work if it doesn't start or block vif deletion until the running work is done. Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3.6510.30			
CVE-2024-47728	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: Zero former ARG_PTR_TO_{LONG,INT} args in case of error For all non-tracing helpers which formerly had ARG_PTR_TO_{LONG,INT} as input arguments, zero the value for the case of an error as otherwise it could leak memory. For tracing, it is not needed given CAP_PERFMON can already read all kernel memory anyway hence bpf_get_func_arg() and bpf_get_func_ret() is skipped in here. Also, the MTU helpers mtu_len pointer value is being written but also read. Technically, the MEM_UNINIT should not be there in order to always force init. Removing MEM_UNINIT needs more verifier rework though: MEM_UNINIT right now	2024-10-21	5.5	Medium

		implies two things actually: i) write into memory, ii) memory does not have to be initialized. If we lift MEM_UNINIT, it then becomes: i) read into memory, ii) memory must be initialized. This means that for bpf_*_check_mtu() we're readding the issue we're trying to fix, that is, it would then be able to write back into things like .rodata BPF maps. Follow-up work will rework the MEM_UNINIT semantics such that the intent can be better expressed. For now just clear the *mtu_len on error path which can be lifted later again.			
CVE-2024-47729	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Use reserved copy engine for user binds on faulting devices User binds map to engines with can fault, faults depend on user binds completion, thus we can deadlock. Avoid this by using reserved copy engine for user binds on faulting devices. While we are here, normalize bind queue creation with a helper. v2: - Pass in extensions to bind queue creation (CI) v3: - s/resevered/reserved (Lucas) - Fix NULL hwe check (Jonathan)	2024-10-21	5.5	Medium
CVE-2024-47731	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drivers/perf: Fix ali_drw_pmu driver interrupt status clearing The alibaba_uncore_pmu driver forgot to clear all interrupt status in the interrupt processing function. After the PMU counter overflow interrupt occurred, an interrupt storm occurred, causing the system to hang. Therefore, clear the correct interrupt status in the interrupt handling function to fix it.	2024-10-21	5.5	Medium

CVE-2024-47733	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfs: Delete subtree of 'fs/netfs' when netfs module exits In netfs_init() or fscache_proc_init(), we create dentry under 'fs/netfs', but in netfs_exit(), we only delete the proc entry of 'fs/netfs' without deleting its subtree. This triggers the following WARNING: == remove_proc_entry: removing non-empty directory 'fs/netfs', leaking at least 'requests' WARNING: CPU: 4 PID: 566 at fs/proc/generic.c:717 remove_proc_entry+0x160/0x1c0 Modules linked in: netfs(-) CPU: 4 UID: 0 PID: 566 Comm: rmmod Not tainted 6.11.0-rc3 #860 RIP: 0010:remove_proc_entry+0x160/0x1c0 Call Trace: <TASK> netfs_exit+0x12/0x620 [netfs] __do_sys_delete_module.isra.0+0x14c/0x2e0 do_syscall_64+0x4b/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e == Therefore use remove_proc_subtree() instead of remove_proc_entry() to fix the above problem.	2024-10-21	5.5	Medium
CVE-2024-47734	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bonding: Fix unnecessary warnings and logs from bond_xdp_get_xmit_slave() syzbot reported a WARNING in bond_xdp_get_xmit_slave. To reproduce this[1], one bond device (bond1) has xdpdrv, which increases bpf_master_redirect_enabled_key. Another bond device (bond0) which is unsupported by XDP but its slave (veth3) has xdpgeneric that returns XDP_TX. This triggers WARN_ON_ONCE() from the xdp_master_redirect(). To reduce unnecessary warnings and improve log management, we need to	2024-10-21	5.5	Medium

		delete the WARN_ON_ONCE() and add ratelimit to the netdev_err(). [1] Steps to reproduce: # Needs tx_xdp with return XDP_TX; ip l add veth0 type veth peer veth1 ip l add veth3 type veth peer veth4 ip l add bond0 type bond mode 6 # BOND_MODE_ALB, unsupported by XDP ip l add bond1 type bond # BOND_MODE_ROUNDROBIN by default ip l set veth0 master bond1 ip l set bond1 up # Increases bpf_master_redirect_enabled_key ip l set dev bond1 xdpdrv object tx_xdp.o section xdp_tx ip l set veth3 master bond0 ip l set bond0 up ip l set veth4 up # Triggers WARN_ON_ONCE() from the xdp_master_redirect() ip l set veth3 xdpgeneric object tx_xdp.o section xdp_tx			
CVE-2024-47735	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix spin_unlock_irqrestore() called with IRQs enabled Fix missuse of spin_lock_irq()/spin_unlock_irq() when spin_lock_irqsave()/spin_lock_irqrestore() was hold. This was discovered through the lock debugging, and the corresponding log is as follows: raw_local_irq_restore() called with IRQs enabled WARNING: CPU: 96 PID: 2074 at kernel/locking/irqflag-debug.c:10 warn_bogus_irq_restore+0x30/0x40 ... Call trace: warn_bogus_irq_restore+0x30/0x40 _raw_spin_unlock_irqrestore+0x84/0xc8 add_qp_to_list+0x11c/0x148 [hns_roce_hw_v2] hns_roce_create_qp_common.constprop.0+0x240/0x780 [hns_roce_hw_v2] hns_roce_create_qp+0x98/0x160 [hns_roce_hw_v2] create_qp+0x138/0x258 ib_create_qp_kernel+0x50/0xe8	2024-10-21	5.5	Medium

		create_mad_qp+0xa8/0x128 ib_mad_port_open+0x218/0x448 ib_mad_init_device+0x70/0x1f8 add_client_context+0xfc/0x220 enable_device_and_get+0xd0/0x140 ib_register_device.part.0+0xf4/0x1c8 ib_register_device+0x34/0x50 hns_roce_register_device+0x174/0x3d0 [hns_roce_hw_v2] hns_roce_init+0xfc/0x2c0 [hns_roce_hw_v2] __hns_roce_hw_v2_init_instance+0x7c/0x1d0 [hns_roce_hw_v2] hns_roce_hw_v2_init_instance+0x9c/0x180 [hns_roce_hw_v2]			
CVE-2024-47736	linux - multiple products	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	2024-10-21	5.5	Medium

		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-2024-47737	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nfsd: call cache_put if xdr_reserve_space returns NULL If not enough buffer space available, but idmap_lookup has triggered lookup_fn which calls cache_get and returns successfully. Then we missed to call cache_put here which pairs with cache_get. Reviwed-by: Jeff Layton <jlayton@kernel.org>	2024-10-21	5.5	Medium
CVE-2024-47739	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: padata: use integer wrap around to prevent deadlock on seq_nr overflow When submitting more than 2^32 padata objects to padata_do_serial, the current sorting implementation incorrectly sorts padata objects with overflowed seq_nr, causing them to be placed before existing objects in the reorder list. This leads to a deadlock in the	2024-10-21	5.5	Medium

		serialization process as padata_find_next cannot match padata->seq_nr and pd->processed because the padata instance with overflowed seq_nr will be selected next. To fix this, we use an unsigned integer wrap around to correctly sort padata objects in scenarios with integer overflow.			
CVE-2024-47743	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: KEYS: prevent NULL pointer dereference in find_asymmetric_key() In find_asymmetric_key(), if all NULLs are passed in the id_{0,1,2} arguments, the kernel will first emit WARN but then have an oops because id_2 gets dereferenced anyway. Add the missing id_2 check and move WARN_ON() to the final else branch to avoid duplicate NULL checks. Found by Linux Verification Center (linuxtesting.org) with Svace static analysis tool.	2024-10-21	5.5	Medium

CVE-2024-47744	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		KVM: Use dedicated mutex to protect kvm_usage_count to avoid deadlock
		
		Use a dedicated mutex to guard kvm_usage_count to fix a potential deadlock on x86 due to a chain of locks and SRCU synchronizations. Translating the below lockdep splat, CPU1 #6 will wait on CPU0 #1, CPU0 #8 will wait on CPU2 #3, and CPU2 #7 will wait on CPU1 #4 (if there's a writer, due to the fairness of r/w semaphores).
		
		CPU0 CPU1 CPU2
		1 lock(&kvm->slots_lock);
		2 lock(&vcpu->mutex);
		3 lock(&kvm->srcu);
		4 lock(cpu_hotplug_lock);
		5 lock(kvm_lock);
			2024-10-21	5.5	Medium

		6 lock(&kvm->slots_lock); 7 lock(cpu_hotplug_lock); 8 sync(&kvm->srcu); Note, there are likely more potential deadlocks in KVM x86, e.g. the same pattern of taking cpu_hotplug_lock outside of kvm_lock likely exists with __kvmclock_cpufreq_notifier(): cpuhp_cpufreq_online() | -> cpufreq_online() | -> cpufreq_gov_performance_limits() | -> __cpufreq_driver_target() | -> __target_index() | -> cpufreq_freq_transition_begin() | -> cpufreq_notify_transition() | -> ... __kvmclock_cpufreq_notifier() But, actually triggering such deadlocks is beyond rare due to the combination of dependencies and timings involved. E.g. the cpufreq notifier is only used on older CPUs without a constant TSC, mucking with the NX hugepage mitigation while VMs are running is very uncommon, and doing so while also onlining/offlining a CPU (necessary to generate contention on cpu_hotplug_lock) would be even more unusual. The most robust solution to the general cpu_hotplug_lock issue is likely to switch vm_list to be an RCU-protected list, e.g. so that x86's cpufreq notifier doesn't to take kvm_lock. For now, settle for fixing the most blatant deadlock, as switching to an RCU-protected list is a much more involved change, but add a comment in locking.rst to call out that care needs to be taken when walking holding kvm_lock			

		and walking vm_list. == ========== WARNING: possible circular locking dependency detected 6.10.0-smp--c257535a0c9d-pip #330 Tainted: G S O -- tee/35048 is trying to acquire lock: ff6a80eced71e0a8 (&kvm->slots_lock){+.+.}-{3:3}, at: set_nx_huge_pages+0x179/0x1e0 [kvm] but task is already holding lock: ffffffffc07abb08 (kvm_lock){+.+.}-{3:3}, at: set_nx_huge_pages+0x14a/0x1e0 [kvm] which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #3 (kvm_lock){+.+.}-{3:3}: __mutex_lock+0x6a/0xb40 mutex_lock_nested+0x1f/0x30 kvm_dev_ioctl+0x4fb/0xe50 [kvm] __se_sys_ioctl+0x7b/0xd0 __x64_sys_ioctl+0x21/0x30 x64_sys_call+0x15d0/0x2e60 do_syscall_64+0x83/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e -> #2 (cpu_hotplug_lock){++++}-{0:0}: cpus_read_lock+0x2e/0xb0 static_key_slow_inc+0x16/0x30 kvm_lapic_set_base+0x6a/0x1c0 [kvm] kvm_set_apic_base+0x8f/0xe0 [kvm] kvm_set_msr_common+0x9ae/0xf80 [kvm] vmx_set_msr+0xa54/0xbe0 [kvm_intel] __kvm_set_msr+0xb6/0x1a0 [kvm] kvm_arch_vcpu_ioctl+0xeca/0x10c0 [kvm] kvm_vcpu_ioctl+0x485/0x5b0 [kvm] __se_sys_ioctl+0x7b/0xd0 __x64_sys_ioctl+0x21/0x30 x64_sys_call+0x15d0/0x2e60 do_syscall_64+0x83/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e -> #1 (&kvm->srcu){.+.+}-{0:0}: __synchronize_srcu+0x44/0x1a0			

		---truncated---			
CVE-2024-47746	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fuse: use exclusive lock when FUSE_I_CACHE_IO_MODE is set This may be a typo. The comment has said shared locks are not allowed when this bit is set. If using shared lock, the wait in `fuse_file_cached_io_open` may be forever.	2024-10-21	5.5	Medium
CVE-2024-47749	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Added NULL check for lookup_atid The lookup_atid() function can return NULL if the ATID is invalid or does not exist in the identifier table, which could lead to dereferencing a null pointer without a check in the `act_establish()` and `act_open_rpl()` functions. Add a NULL check to prevent null pointer dereferencing. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2024-10-21	5.5	Medium
CVE-2024-47752	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix H264 stateless decoder smatch warning Fix a smatch static checker warning on vdec_h264_req_if.c. Which leads to a kernel crash when fb is NULL.	2024-10-21	5.5	Medium
CVE-2024-47753	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix VP8 stateless decoder smatch warning Fix a smatch static checker warning on vdec_vp8_req_if.c. Which leads to a kernel crash when fb is NULL.	2024-10-21	5.5	Medium
CVE-2024-47754	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix H264 multi stateless	2024-10-21	5.5	Medium

		decoder smatch warning Fix a smatch static checker warning on vdec_h264_req_multi_if.c. Which leads to a kernel crash when fb is NULL.			
CVE-2024-47756	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PCI: keystone: Fix if-statement expression in ks_pcie_quirk() This code accidentally uses && where || was intended. It potentially results in a NULL dereference. Thus, fix the if-statement expression to use the correct condition. [kwilczynski: commit log]	2024-10-21	5.5	Medium
CVE-2024-49850	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpf: correctly handle malformed BPF_CORE_TYPE_ID_LOCAL relos In case of malformed relocation record of kind BPF_CORE_TYPE_ID_LOCAL referencing a non-existing BTF type, function bpf_core_calc_relo_insn would cause a null pointer deference. Fix this by adding a proper check upper in call stack, as malformed relocation records could be passed from user space. Simplest reproducer is a program: r0 = 0 exit With a single relocation record: .insn_off = 0, /* patch first instruction */ .type_id = 100500, /* this type id does not exist */ .access_str_off = 6, /* offset of string "0" */ .kind = BPF_CORE_TYPE_ID_LOCAL, See the link for original reproducer or next commit for a test case.	2024-10-21	5.5	Medium

CVE-2024-49851	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tpm: Clean up TPM space after command failure tpm_dev_transmit prepares the TPM space before attempting command transmission. However if the command fails no rollback of this preparation is done. This can result in transient handles being leaked if the device is subsequently closed with no further commands performed. Fix this by flushing the space in the event of command transmission failure.	2024-10-21	5.5	Medium
CVE-2024-49856	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Fix deadlock in SGX NUMA node search When the current node doesn't have an EPC section configured by firmware and all other EPC sections are used up, CPU can get stuck inside the while loop that looks for an available EPC page from remote nodes indefinitely, leading to a soft lockup. Note how nid_of_current will never be equal to nid in that while loop because nid_of_current is not set in sgx_numa_mask. Also worth mentioning is that it's perfectly fine for the firmware not to setup an EPC section on a node. While setting up an EPC section on each node can enhance performance, it is not a requirement for functionality. Rework the loop to start and end on *a* node that has SGX memory. This avoids the deadlock looking for the current SGX-lacking node to show up in the loop when it never will.	2024-10-21	5.5	Medium
CVE-2024-49857	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: set the cipher for secured NDP	2024-10-21	5.5	Medium

		ranging The cipher pointer is not set, but is derefereced trying to set its content, which leads to a NULL pointer dereference. Fix it by pointing to the cipher parameter before dereferencing.			
CVE-2024-49858	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: efistub/tpm: Use ACPI reclaim memory for event log to avoid corruption The TPM event log table is a Linux specific construct, where the data produced by the GetEventLog() boot service is cached in memory, and passed on to the OS using an EFI configuration table. The use of EFI_LOADER_DATA here results in the region being left unreserved in the E820 memory map constructed by the EFI stub, and this is the memory description that is passed on to the incoming kernel by kexec, which is therefore unaware that the region should be reserved. Even though the utility of the TPM2 event log after a kexec is questionable, any corruption might send the parsing code off into the weeds and crash the kernel. So let's use EFI_ACPI_RECLAIM_MEMORY instead, which is always treated as reserved by the E820 conversion logic.	2024-10-21	5.5	Medium
CVE-2024-49863	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vhost/scsi: null-ptr-dereference in vhost_scsi_get_req() Since commit 3f8ca2e115e5 ("vhost/scsi: Extract common handling code from control queue handler") a null pointer dereference bug can be triggered when guest sends an SCSI AN request. In vhost_scsi_ctl_handle_vq(), `vc.target` is assigned with	2024-10-21	5.5	Medium

		`&v_req.tmf.lun[1]` within a switch-case block and is then passed to vhost_scsi_get_req() which extracts `vc->req` and `tpg`. However, for a `VIRTIO_SCSI_T_AN_*` request, tpg is not required, so `vc.target` is set to NULL in this branch. Later, in vhost_scsi_get_req(), `vc->target` is dereferenced without being checked, leading to a null pointer dereference bug. This bug can be triggered from guest. When this bug occurs, the vhost_worker process is killed while holding `vq->mutex` and the corresponding tpg will remain occupied indefinitely. Below is the KASAN report: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] CPU: 1 PID: 840 Comm: poc Not tainted 6.10.0+ #1 Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:vhost_scsi_get_req+0x165/0x3a0 Code: 00 fc ff df 48 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 2b 02 00 00 48 b8 00 00 00 00 00 fc ff df 4d 8b 65 30 4c 89 e2 48 c1 ea 03 <0f> b6 04 02 4c 89 e2 83 e2 07 38 d0 7f 08 84 c0 0f 85 be 01 00 00 RSP: 0018:ffff888017affb50 EFLAGS: 00010246 RAX: dffffc0000000000 RBX: ffff88801b000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff888017affcb8 RBP: ffff888017affb80 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000 R13: ffff888017affc88 R14: ffff888017affd1c R15: ffff888017993000 FS: 000055556e076500(0000) GS:ffff88806b100000(0000) knlGS:0000000000000000			

		CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000200027c0 CR3: 0000000010ed0004 CR4: 0000000000370ef0 Call Trace: <TASK> ? show_regs+0x86/0xa0 ? die_addr+0x4b/0xd0 ? exc_general_protection+0x163/0x260 ? asm_exc_general_protection+0x27/0x30 ? vhost_scsi_get_req+0x165/0x3a0 vhost_scsi_ctl_handle_vq+0x2a4/0xca0 ? __pfx_vhost_scsi_ctl_handle_vq+0x10/0x10 ? __switch_to+0x721/0xeb0 ? __schedule+0xda5/0x5710 ? __kasan_check_write+0x14/0x30 ? _raw_spin_lock+0x82/0xf0 vhost_scsi_ctl_handle_kick+0x52/0x90 vhost_run_work_list+0x134/0x1b0 vhost_task_fn+0x121/0x350 ... </TASK> ---[end trace 0000000000000000]--- Let's add a check in vhost_scsi_get_req. [whitespace fixes]			
CVE-2024-49867	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: wait for fixup workers before stopping cleaner kthread during umount During unmount, at close_ctree(), we have the following steps in this order: 1) Park the cleaner kthread - this doesn't destroy the kthread, it basically halts its execution (wake ups against it work but do nothing); 2) We stop the cleaner kthread - this results in freeing the respective struct task_struct; 3) We call btrfs_stop_all_workers() which waits for any jobs running in all the work queues and then free the work queues. Syzbot reported a case where a fixup worker resulted in a crash when doing a delayed iput on its inode while attempting to wake	2024-10-21	5.5	Medium

		up the cleaner at btrfs_add_delayed_iput(), because the task_struct of the cleaner kthread was already freed. This can happen during unmount because we don't wait for any fixup workers still running before we call kthread_stop() against the cleaner kthread, which stops and free all its resources. Fix this by waiting for any fixup workers at close_ctree() before we call kthread_stop() against the cleaner and run pending delayed iputs. The stack traces reported by syzbot were the following: BUG: KASAN: slab-use-after-free in __lock_acquire+0x77/0x2050 kernel/locking/lockdep.c:5065 Read of size 8 at addr ffff8880272a8a18 by task kworker/u8:3/52 CPU: 1 UID: 0 PID: 52 Comm: kworker/u8:3 Not tainted 6.12.0-rc1-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: btrfs-fixup btrfs_work_helper Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 __lock_acquire+0x77/0x2050 kernel/locking/lockdep.c:5065 lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5825 __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline] _raw_spin_lock_irqsave+0xd5/0x120 kernel/locking/spinlock.c:162 class_raw_spinlock_irqsave_constructor include/linux/spinlock.h:551 [inline] try_to_wake_up+0xb0/0x1480 kernel/sched/core.c:4154 btrfs_writepage_fixup_worker+0xc16/0xdf0			

		fs/btrfs/inode.c:2842 btrfs_work_helper+0x390/0xc50 fs/btrfs/async-thread.c:314 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa63/0x1850 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Allocated by task 2: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 unpoison_slab_object mm/kasan/common.c:319 [inline] __kasan_slab_alloc+0x66/0x80 mm/kasan/common.c:345 kasan_slab_alloc include/linux/kasan.h:247 [inline] slab_post_alloc_hook mm/slub.c:4086 [inline] slab_alloc_node mm/slub.c:4135 [inline] kmem_cache_alloc_node_noprof+0x16b/0x320 mm/slub.c:4187 alloc_task_struct_node kernel/fork.c:180 [inline] dup_task_struct+0x57/0x8c0 kernel/fork.c:1107 copy_process+0x5d1/0x3d50 kernel/fork.c:2206 kernel_clone+0x223/0x880 kernel/fork.c:2787 kernel_thread+0x1bc/0x240 kernel/fork.c:2849 create_kthread kernel/kthread.c:412 [inline] kthreadd+0x60d/0x810 kernel/kthread.c:765 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 Freed by task 61: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:579 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x59/0x70 mm/kasan/common.c:264			

		kasan_slab_free include/linux/kasan.h:230 [inline] slab_free_h ---truncated---			
CVE-2024-49868	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix a NULL pointer dereference when failed to start a new trasacntion [BUG] Syzbot reported a NULL pointer dereference with the following crash: FAULT_INJECTION: forcing a failure. start_transaction+0x830/0x1670 fs/btrfs/transaction.c:676 prepare_to_relocate+0x31f/0x4c0 fs/btrfs/relocation.c:3642 relocate_block_group+0x169/0xd20 fs/btrfs/relocation.c:3678 ... BTRFS info (device loop0): balance: ended with status: -12 Oops: general protection fault, probably for non-canonical address 0xdffffc00000000cc: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000660-0x0000000000000667] RIP: 0010:btrfs_update_reloc_root+0x362/0xa80 fs/btrfs/relocation.c:926 Call Trace: <TASK> commit_fs_roots+0x2ee/0x720 fs/btrfs/transaction.c:1496 btrfs_commit_transaction+0xfaf/0x3740 fs/btrfs/transaction.c:2430 del_balance_item fs/btrfs/volumes.c:3678 [inline] reset_balance_state+0x25e/0x3c0 fs/btrfs/volumes.c:3742 btrfs_balance+0xead/0x10c0 fs/btrfs/volumes.c:4574 btrfs_ioctl_balance+0x493/0x7c0 fs/btrfs/ioctl.c:3673 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xf9/0x170 fs/ioctl.c:893 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	2024-10-21	5.5	Medium

		[CAUSE] The allocation failure happens at the start_transaction() inside prepare_to_relocate(), and during the error handling we call unset_reloc_control(), which makes fs_info->balance_ctl to be NULL. Then we continue the error path cleanup in btrfs_balance() by calling reset_balance_state() which will call del_balance_item() to fully delete the balance item in the root tree. However during the small window between set_reloc_contrl() and unset_reloc_control(), we can have a subvolume tree update and created a reloc_root for that subvolume. Then we go into the final btrfs_commit_transaction() of del_balance_item(), and into btrfs_update_reloc_root() inside commit_fs_roots(). That function checks if fs_info->reloc_ctl is in the merge_reloc_tree stage, but since fs_info->reloc_ctl is NULL, it results a NULL pointer dereference. [FIX] Just add extra check on fs_info->reloc_ctl inside btrfs_update_reloc_root(), before checking fs_info->reloc_ctl->merge_reloc_tree. That DEAD_RELOC_TREE handling is to prevent further modification to the reloc tree during merge stage, but since there is no reloc_ctl at all, we do not need to bother that.			
CVE-2024-49870	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cachefiles: fix dentry leak in cachefiles_open_file() A dentry leak may be caused when a lookup cookie and a cull are concurrent:	2024-10-21	5.5	Medium

<table>
<tr>
<td></td>
<td></td>
<td><pre>
          P1 | P2
----
cachefiles_lookup_cookie
  cachefiles_look_up_object
    lookup_one_positive_unlocked
     // get dentry
                        cachefiles_cull
                          inode->i_flags |= S_KERNEL_FILE;
    cachefiles_open_file
      cachefiles_mark_inode_in_use
        __cachefiles_mark_inode_in_use
          can_use = false
          if (!(inode->i_flags & S_KERNEL_FILE))
            can_use = true
  return false
        return false
        // Returns an error but doesn't put dentry
</pre>
After that the following WARNING will be triggered when the backend folder is umounted:
<pre>
==
BUG: Dentry 000000008ad87947{i=7a,n=Dx_1_1.img}
still in use (1) [unmount of ext4 sda]
WARNING: CPU: 4 PID: 359261 at fs/dcache.c:1767
umount_check+0x5d/0x70
CPU: 4 PID: 359261 Comm: umount Not tainted 6.6.0-
dirty #25
RIP: 0010:umount_check+0x5d/0x70
Call Trace:
 <TASK>
 d_walk+0xda/0x2b0
 do_one_tree+0x20/0x40
 shrink_dcache_for_umount+0x2c/0x90
 generic_shutdown_super+0x20/0x160
 kill_block_super+0x1a/0x40
 ext4_kill_sb+0x22/0x40
 deactivate_locked_super+0x35/0x80
 cleanup_mnt+0x104/0x160
==
</pre>
Whether cachefiles_open_file() returns true or false, the reference count obtained by lookup_positive_unlocked() in cachefiles_look_up_object() should be released.

Therefore release that reference count in cachefiles_look_up_object() to fix the above issue and simplify the code.</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

CVE-2024-49871	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Input: adp5589-keys - fix NULL pointer dereference We register a devm action to call adp5589_clear_config() and then pass the i2c client as argument so that we can call i2c_get_clientdata() in order to get our device object. However, i2c_set_clientdata() is only being set at the end of the probe function which means that we'll get a NULL pointer dereference in case the probe function fails early.	2024-10-21	5.5	Medium
CVE-2024-49873	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: mm/filemap: fix filemap_get_folios_contig THP panic Patch series "memfd-pin huge page fixes". Fix multiple bugs that occur when using memfd_pin_folios with hugetlb pages and THP. The hugetlb bugs only bite when the page is not yet faulted in when memfd_pin_folios is called. The THP bug bites when the starting offset passed to memfd_pin_folios is not huge page aligned. See the commit messages for details. This patch (of 5): memfd_pin_folios on memory backed by THP panics if the requested start offset is not huge page aligned: BUG: kernel NULL pointer dereference, address: 0000000000000036 RIP: 0010:filemap_get_folios_contig+0xdf/0x290 RSP: 0018:ffffc9002092fbe8 EFLAGS: 00010202 RAX: 0000000000000002 RBX: 0000000000000002 RCX: 0000000000000002 The fault occurs here, because xas_load returns a folio with value 2: filemap_get_folios_contig() for (folio = xas_load(&xas); folio && xas.xa_index	2024-10-21	5.5	Medium

		<= end;
		folio = xas_next(&xas)) {
		...
		if (!folio_try_get(folio)) <-- BOOM
		
		"2" is an xarray sibling entry. We get it because memfd_pin_folios does not round the indices passed to filemap_get_folios_contig to huge page boundaries for THP, so we load from the middle of a huge page range see a sibling. (It does round for hugetlbfs, at the is_file_hugepages test).
		
		To fix, if the folio is a sibling, then return the next index as the starting point for the next call to filemap_get_folios_contig.			

CVE-2024-49875	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nfsd: map the EBADMSG to nfserr_io to avoid warning Ext4 will throw -EBADMSG through ext4_readdir when a checksum error occurs, resulting in the following WARNING. Fix it by mapping EBADMSG to nfserr_io. nfsd_buffered_readdir iterate_dir // -EBADMSG -74 ext4_readdir // .iterate_shared ext4_dx_readdir ext4_htree_fill_tree htree_dirblock_to_tree ext4_read_dirblock __ext4_read_dirblock ext4_dirblock_csum_verify warn_no_space_for_csum __warn_no_space_for_csum return ERR_PTR(-EFSBADCRC) // -EBADMSG -74 nfserrno // WARNING [161.115610] ------------[cut here]------------ [161.116465] nfsd: non-standard errno: -74 [161.117315] WARNING: CPU: 1 PID: 780 at fs/nfsd/nfsproc.c:878 nfserrno+0x9d/0xd0 [161.118596] Modules linked in: [161.119243] CPU: 1 PID: 780 Comm: nfsd Not tainted 5.10.0-00014-g79679361fd5d #138	2024-10-21	5.5	Medium

		[161.120684] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qe mu.org 04/01/2014 [161.123601] RIP: 0010:nfserrno+0x9d/0xd0 [161.124676] Code: 0f 87 da 30 dd 00 83 e3 01 b8 00 00 00 05 75 d7 44 89 ee 48 c7 c7 c0 57 24 98 89 44 24 04 c6 05 ce 2b 61 03 01 e8 99 20 d8 00 <0f> 0b 8b 44 24 04 eb b5 4c 89 e6 48 c7 c7 a0 6d a4 99 e8 cc 15 33 [161.127797] RSP: 0018:ffffc90000e2f9c0 EFLAGS: 00010286 [161.128794] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 [161.130089] RDX: 1ffff1103ee16f6d RSI: 0000000000000008 RDI: fffff520001c5f2a [161.131379] RBP: 0000000000000022 R08: 0000000000000001 R09: ffff8881f70c1827 [161.132664] R10: ffffed103ee18304 R11: 0000000000000001 R12: 0000000000000021 [161.133949] R13: 00000000ffffffb6 R14: ffff8881317c0000 R15: ffffc90000e2fbd8 [161.135244] FS: 0000000000000000(0000) GS:ffff8881f7080000(0000) knlGS:0000000000000000 [161.136695] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [161.137761] CR2: 00007fcaad70b348 CR3: 0000000144256006 CR4: 0000000000770ee0 [161.139041] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [161.140291] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [161.141519] PKRU: 55555554 [161.142076] Call Trace: [161.142575] ? __warn+0x9b/0x140 [161.143229] ? nfserrno+0x9d/0xd0 [161.143872] ? report_bug+0x125/0x150 [161.144595] ? handle_bug+0x41/0x90 [161.145284] ? exc_invalid_op+0x14/0x70 [161.146009] ? asm_exc_invalid_op+0x12/0x20 [161.146816] ? nfserrno+0x9d/0xd0 [161.147487] nfsd_buffered_readdir+0x28b/0x2b0 [161.148333] ? nfsd4_encode_dirent_fattr+0x380/0x380 [161.149258] ? nfsd_buffered_filldir+0xf0/0xf0 [161.150093] ? wait_for_concurrent_writes+0x170/0x170 [161.151004] ? generic_file_llseek_size+0x48/0x160 [161.151895] nfsd_readdir+0x132/0x190 [161.152606] ?			

		nfsd4_encode_dirent_fattr+0x380/0x380 [161.153516] ? nfsd_unlink+0x380/0x380 [161.154256] ? override_creds+0x45/0x60 [161.155006] nfsd4_encode_readdir+0x21a/0x3d0 [161.155850] ? nfsd4_encode_readlink+0x210/0x210 [161.156731] ? write_bytes_to_xdr_buf+0x97/0xe0 [161.157598] ? __write_bytes_to_xdr_buf+0xd0/0xd0 [161.158494] ? lock_downgrade+0x90/0x90 [161.159232] ? nfs4svc_decode_voidarg+0x10/0x10 [161.160092] nfsd4_encode_operation+0x15a/0x440 [161.160959] nfsd4_proc_compound+0x718/0xe90 [161.161818] nfsd_dispatch+0x18e/0x2c0 [161.162586] svc_process_common+0x786/0xc50 [161.163403] ? nfsd_svc+0x380/0x380 [161.164137] ? svc_printk+0x160/0x160 [161.164846] ? svc_xprt_do_enqueue.part.0+0x365/0x380 [161.165808] ? nfsd_svc+0x380/0x380 [161.166523] ? rcu_is_watching+0x23/0x40 [161.167309] svc_process+0x1a5/0x200 [161.168019] nfsd+0x1f5/0x380 [161.168663] ? nfsd_shutdown_threads+0x260/0x260 [161.169554] kthread+0x1c4/0x210 [161.170224] ? kthread_insert_work_sanity_check+0x80/0x80 [161.171246] ret_from_fork+0x1f/0x30			
CVE-2024-49877	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix possible null-ptr-deref in ocfs2_set_buffer_uptodate When doing cleanup, if flags without OCFS2_BH_READAHEAD, it may trigger NULL pointer dereference in the following ocfs2_set_buffer_uptodate() if bh is NULL.	2024-10-21	5.5	Medium
CVE-2024-49879	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm: omapdrm: Add missing check for alloc_ordered_workqueue As it may return NULL pointer and cause NULL pointer dereference. Add check for the return value of alloc_ordered_workqueue.	2024-10-21	5.5	Medium

CVE-2024-49881	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ext4: update orig_path in ext4_find_extent()
		
		In ext4_find_extent(), if the path is not big enough, we free it and set
		*orig_path to NULL. But after reallocating and successfully initializing
		the path, we don't update *orig_path, in which case the caller gets a
		valid path but a NULL ppath, and this may cause a NULL pointer dereference
		or a path memory leak. For example:
		
		ext4_split_extent
		  path = *ppath = 2000
		  ext4_find_extent
		    if (depth > path[0].p_maxdepth)
		      kfree(path = 2000);
		      *orig_path = path = NULL;
		      path = kcalloc() = 3000
		  ext4_split_extent_at(*ppath = NULL)
		    path = *ppath;
		    ex = path[depth].p_ext;
		    // NULL pointer dereference!
		
		==
		BUG: kernel NULL pointer dereference, address: 0000000000000010
		CPU: 6 UID: 0 PID: 576 Comm: fsstress Not tainted 6.11.0-rc2-dirty #847
		RIP: 0010:ext4_split_extent_at+0x6d/0x560
		Call Trace:
		 <TASK>
		 ext4_split_extent.isra.0+0xcb/0x1b0
		 ext4_ext_convert_to_initialized+0x168/0x6c0
		 ext4_ext_handle_unwritten_extents+0x325/0x4d0
		 ext4_ext_map_blocks+0x520/0xdb0
		 ext4_map_blocks+0x2b0/0x690
		 ext4_iomap_begin+0x20e/0x2c0
		[...]
		==
		
		Therefore, *orig_path is updated when the extent lookup succeeds, so that
		the caller can safely use path or *ppath.
			2024-10-21	5.5	Medium

CVE-2024-49890	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: ensure the fw_info is not null before	2024-10-21	5.5	Medium

		using it This resolves the dereference null return value warning reported by Coverity.			
CVE-2024-49891	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Validate hdwq pointers before dereferencing in reset/errata paths When the HBA is undergoing a reset or is handling an errata event, NULL ptr dereference crashes may occur in routines such as lpfc_sli_flush_io_rings(), lpfc_dev_loss_tmo_callbk(), or lpfc_abort_handler(). Add NULL ptr checks before dereferencing hdwq pointers that may have been freed due to operations colliding with a reset or errata event handler.	2024-10-21	5.5	Medium
CVE-2024-49892	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize get_bytes_per_element's default to 1 Variables, used as denominators and maybe not assigned to other values, should not be 0. bytes_per_element_y & bytes_per_element_c are initialized by get_bytes_per_element() which should never return 0. This fixes 10 DIVIDE_BY_ZERO issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49893	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check stream_status before it is used [WHAT & HOW] dc_state_get_stream_status can return null, and therefore null must be checked before stream_status is used. This fixes 1 NULL_RETURNS issue reported by Coverity.	2024-10-21	5.5	Medium

CVE-2024-49896	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check stream before comparing them [WHAT & HOW] amdgpu_dm can pass a null stream to dc_is_stream_unchanged. It is necessary to check for null before dereferencing them. This fixes 1 FORWARD_NULL issue reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49897	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check phantom_stream before it is used dcn32_enable_phantom_stream can return null, so returned value must be checked before used. This fixes 1 NULL_RETURNS issue reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49898	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null-initialized variables [WHAT & HOW] drr_timing and subvp_pipe are initialized to null and they are not always assigned new values. It is necessary to check for null before dereferencing. This fixes 2 FORWARD_NULL issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49899	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize denominators' default to 1 [WHAT & HOW] Variables used as denominators and maybe not assigned to other values, should not be 0. Change their default to 1 so they are never 0.	2024-10-21	5.5	Medium

		This fixes 10 DIVIDE_BY_ZERO issues reported by Coverity.			
CVE-2024-49901	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/msm/adreno: Assign msm_gpu->pdev earlier to avoid nullptrs There are some cases, such as the one uncovered by Commit 46d4efcccc68 ("drm/msm/a6xx: Avoid a nullptr dereference when speedbin setting fails") where msm_gpu_cleanup() : platform_set_drvdata(gpu->pdev, NULL); is called on gpu->pdev == NULL, as the GPU device has not been fully initialized yet. Turns out that there's more than just the aforementioned path that causes this to happen (e.g. the case when there's speedbin data in the catalog, but opp-supported-hw is missing in DT). Assigning msm_gpu->pdev earlier seems like the least painful solution to this, therefore do so. Patchwork: https://patchwork.freedesktop.org/patch/602742/	2024-10-21	5.5	Medium
CVE-2024-49904	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add list empty check to avoid null pointer issue Add list empty check to avoid null pointer issues in some corner cases. - list_for_each_entry_safe()	2024-10-21	5.5	Medium
CVE-2024-49905	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for 'afb' in amdgpu_dm_plane_handle_cursor_update (v2) This commit adds a null check for the 'afb' variable in the	2024-10-21	5.5	Medium

		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. Changes since v1: - Moved the null check for 'afb' to the line where 'afb' is used. (Alex) 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)			
CVE-2024-49906	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointer before try to access it [why & how] Change the order of the pipe_ctx->plane_state check to ensure that plane_state is not null before accessing it.	2024-10-21	5.5	Medium
CVE-2024-49907	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before using dc->clk_mgr [WHY & HOW] dc->clk_mgr is null checked previously in the same function, indicating it might be null. Passing "dc" to "dc->hwss.apply_idle_power_optimizations", which dereferences null "dc->clk_mgr". (The function pointer resolves to "dcn35_apply_idle_power_optimizations".) This fixes 1 FORWARD_NULL issue reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49908	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for 'afb' in amdgpu_dm_update_cursor (v2)	2024-10-21	5.5	Medium

		This commit adds a null check for the 'afb' variable in the amdgpu_dm_update_cursor function. Previously, 'afb' was assumed to be null at line 8388, but was used later in the code without a null check. This could potentially lead to a null pointer dereference. Changes since v1: - Moved the null check for 'afb' to the line where 'afb' is used. (Alex) Fixes the below: drivers/gpu/drm/amd/amdgpu/../display/amdgpu_dm/amdgpu_dm.c:8433 amdgpu_dm_update_cursor() error: we previously assumed 'afb' could be null (see line 8388)			
[CVE-2024-49909]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function pointer in dcn32_set_output_transfer_func This commit adds a null check for the set_output_gamma function pointer in the dcn32_set_output_transfer_func function. Previously, set_output_gamma was being checked for null, but then it was being dereferenced without any null check. This could lead to a null pointer dereference if set_output_gamma is null. To fix this, we now ensure that set_output_gamma is not null before dereferencing it. We do this by adding a null check for set_output_gamma before the call to set_output_gamma.	2024-10-21	5.5	Medium
[CVE-2024-49910]()	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function pointer in dcn401_set_output_transfer_func This commit adds a null check for the set_output_gamma function pointer in the dcn401_set_output_transfer_func function. Previously, set_output_gamma was being checked for null, but	2024-10-21	5.5	Medium

		then it was being dereferenced without any null check. This could lead to a null pointer dereference if set_output_gamma is null. To fix this, we now ensure that set_output_gamma is not null before dereferencing it. We do this by adding a null check for set_output_gamma before the call to set_output_gamma.			
CVE-2024-49911	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function pointer in dcn20_set_output_transfer_func This commit adds a null check for the set_output_gamma function pointer in the dcn20_set_output_transfer_func function. Previously, set_output_gamma was being checked for null at line 1030, but then it was being dereferenced without any null check at line 1048. This could potentially lead to a null pointer dereference error if set_output_gamma is null. To fix this, we now ensure that set_output_gamma is not null before dereferencing it. We do this by adding a null check for set_output_gamma before the call to set_output_gamma at line 1048.	2024-10-21	5.5	Medium
CVE-2024-49912	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Handle null 'stream_status' in 'planes_changed_for_existing_stream' This commit adds a null check for 'stream_status' in the function 'planes_changed_for_existing_stream'. Previously, the code assumed 'stream_status' could be null, but did not handle the case where it was actually null. This could lead to a null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/core/dc_resource.c:3784	2024-10-21	5.5	Medium

		planes_changed_for_existing_stream() error: we previously assumed 'stream_status' could be null (see line 3774)			
CVE-2024-49913	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for top_pipe_to_program in commit_planes_for_stream This commit addresses a null pointer dereference issue in the `commit_planes_for_stream` function at line 4140. The issue could occur when `top_pipe_to_program` is null. The fix adds a check to ensure `top_pipe_to_program` is not null before accessing its stream_res. This prevents a null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/core/dc.c:4140 commit_planes_for_stream() error: we previously assumed 'top_pipe_to_program' could be null (see line 3906)	2024-10-21	5.5	Medium
CVE-2024-49914	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for pipe_ctx->plane_state in dcn20_program_pipe This commit addresses a null pointer dereference issue in the `dcn20_program_pipe` function. The issue could occur when `pipe_ctx->plane_state` is null. The fix adds a check to ensure `pipe_ctx->plane_state` is not null before accessing. This prevents a null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn20/dcn20_hwseq.c:1925 dcn20_program_pipe() error: we previously assumed 'pipe_ctx->plane_state' could be null (see line 1877)	2024-10-21	5.5	Medium
CVE-2024-49915	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr in	2024-10-21	5.5	Medium

		dcn32_init_hw This commit addresses a potential null pointer dereference issue in the `dcn32_init_hw` function. The issue could occur when `dc->clk_mgr` is null. The fix adds a check to ensure `dc->clk_mgr` is not null before accessing its functions. This prevents a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn32/dcn32_hwseq.c:961 dcn32_init_hw() error: we previously assumed 'dc->clk_mgr' could be null (see line 782)			
CVE-2024-49916	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr and clk_mgr->funcs in dcn401_init_hw This commit addresses a potential null pointer dereference issue in the `dcn401_init_hw` function. The issue could occur when `dc->clk_mgr` or `dc->clk_mgr->funcs` is null. The fix adds a check to ensure `dc->clk_mgr` and `dc->clk_mgr->funcs` is not null before accessing its functions. This prevents a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn401/dcn401_hwseq.c:416 dcn401_init_hw() error: we previously assumed 'dc->clk_mgr' could be null (see line 225)	2024-10-21	5.5	Medium
CVE-2024-49917	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr and clk_mgr->funcs in dcn30_init_hw This commit addresses a potential null pointer dereference issue in the `dcn30_init_hw` function. The issue could occur when	2024-10-21	5.5	Medium

		`dc->clk_mgr` or `dc->clk_mgr->funcs` is null. The fix adds a check to ensure `dc->clk_mgr` and `dc->clk_mgr->funcs` is not null before accessing its functions. This prevents a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn30/dcn30_hwseq.c:789 dcn30_init_hw() error: we previously assumed 'dc->clk_mgr' could be null (see line 628)			
[CVE-2024-49918](CVE-2024-49918)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for head_pipe in dcn32_acquire_idle_pipe_for_head_pipe_in_layer This commit addresses a potential null pointer dereference issue in the `dcn32_acquire_idle_pipe_for_head_pipe_in_layer` function. The issue could occur when `head_pipe` is null. The fix adds a check to ensure `head_pipe` is not null before asserting it. If `head_pipe` is null, the function returns NULL to prevent a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/resource/dcn32/dcn32_resource.c:2690 dcn32_acquire_idle_pipe_for_head_pipe_in_layer() error: we previously assumed 'head_pipe' could be null (see line 2681)	2024-10-21	5.5	Medium
[CVE-2024-49919](CVE-2024-49919)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for head_pipe in dcn201_acquire_free_pipe_for_layer This commit addresses a potential null pointer dereference issue in the `dcn201_acquire_free_pipe_for_layer` function. The issue could occur when `head_pipe` is null. The fix adds a check to ensure `head_pipe` is not null	2024-10-21	5.5	Medium

		before asserting it. If `head_pipe` is null, the function returns NULL to prevent a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/resource/dcn201/dcn201_resource.c:1016 dcn201_acquire_free_pipe_for_layer() error: we previously assumed 'head_pipe' could be null (see line 1010)			
CVE-2024-49920	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before multiple uses [WHAT & HOW] Poniters, such as stream_enc and dc->bw_vbios, are null checked previously in the same function, so Coverity warns "implies that stream_enc and dc->bw_vbios might be null". They are used multiple times in the subsequent code and need to be checked. This fixes 10 FORWARD_NULL issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49921	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before used [WHAT & HOW] Poniters, such as dc->clk_mgr, are null checked previously in the same function, so Coverity warns "implies that "dc->clk_mgr" might be null". As a result, these pointers need to be checked when used again. This fixes 10 FORWARD_NULL issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49922	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before using them [WHAT & HOW] These pointers are null checked previously in the	2024-10-21	5.5	Medium

		same function, indicating they might be null as reported by Coverity. As a result, they need to be checked when used again. This fixes 3 FORWARD_NULL issue reported by Coverity.			
CVE-2024-49923	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Pass non-null to dcn20_validate_apply_pipe_split_flags [WHAT & HOW] "dcn20_validate_apply_pipe_split_flags" dereferences merge, and thus it cannot be a null pointer. Let's pass a valid pointer to avoid null dereference. This fixes 2 FORWARD_NULL issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49929	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: avoid NULL pointer dereference iwl_mvm_tx_skb_sta() and iwl_mvm_tx_mpdu() verify that the mvmvsta pointer is not NULL. It retrieves this pointer using iwl_mvm_sta_from_mac80211, which is dereferencing the ieee80211_sta pointer. If sta is NULL, iwl_mvm_sta_from_mac80211 will dereference a NULL pointer. Fix this by checking the sta pointer before retrieving the mvmsta from it. If sta is not NULL, then mvmsta isn't either.	2024-10-21	5.5	Medium
CVE-2024-49941	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpiolib: Fix potential NULL pointer dereference in gpiod_get_label() In `gpiod_get_label()`, it is possible that `srcu_dereference_check()` may return a NULL pointer, leading to a scenario where `label->str` is accessed without verifying if `label` itself is NULL.	2024-10-21	5.5	Medium

		This patch adds a proper NULL check for `label` before accessing `label->str`. The check for `label->str != NULL` is removed because `label->str` can never be NULL if `label` is not NULL. This fixes the issue where the label name was being printed as `(efault)` when dumping the sysfs GPIO file when `label == NULL`.			
CVE-2024-49942	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Prevent null pointer access in xe_migrate_copy xe_migrate_copy designed to copy content of TTM resources. When source resource is null, it will trigger a NULL pointer dereference in xe_migrate_copy. To avoid this situation, update lacks source flag to true for this case, the flag will trigger xe_migrate_clear rather than xe_migrate_copy. Issue trace: <7> [317.089847] xe 0000:00:02.0: [drm:xe_migrate_copy [xe]] Pass 14, sizes: 4194304 & 4194304 <7> [317.089945] xe 0000:00:02.0: [drm:xe_migrate_copy [xe]] Pass 15, sizes: 4194304 & 4194304 <1> [317.128055] BUG: kernel NULL pointer dereference, address: 0000000000000010 <1> [317.128064] #PF: supervisor read access in kernel mode <1> [317.128066] #PF: error_code(0x0000) - not-present page <6> [317.128069] PGD 0 P4D 0 <4> [317.128071] Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI <4> [317.128074] CPU: 1 UID: 0 PID: 1440 Comm: kunit_try_catch Tainted: G U N 6.11.0-rc7-xe #1 <4> [317.128078] Tainted: [U]=USER, [N]=TEST <4> [317.128080] Hardware name: Intel Corporation Lunar Lake Client Platform/LNL-M LP5 RVP1, BIOS LNLMFWI1.R00.3221.D80.2407291239 07/29/2024	2024-10-21	5.5	Medium

		<4> [317.128082] RIP: 0010:xe_migrate_copy+0x66/0x13e0 [xe] <4> [317.128158] Code: 00 00 48 89 8d e0 fe ff ff 48 8b 40 10 4c 89 85 c8 fe ff ff 44 88 8d bd fe ff ff 65 48 8b 3c 25 28 00 00 00 48 89 7d d0 31 ff <8b> 79 10 48 89 85 a0 fe ff ff 48 8b 00 48 89 b5 d8 fe ff ff 83 ff <4> [317.128162] RSP: 0018:ffffc9000167f9f0 EFLAGS: 00010246 <4> [317.128164] RAX: ffff8881120d8028 RBX: ffff88814d070428 RCX: 0000000000000000 <4> [317.128166] RDX: ffff88813cb99c00 RSI: 0000000004000000 RDI: 0000000000000000 <4> [317.128168] RBP: ffffc9000167fbb8 R08: ffff88814e7b1f08 R09: 0000000000000001 <4> [317.128170] R10: 0000000000000001 R11: 0000000000000001 R12: ffff88814e7b1f08 <4> [317.128172] R13: ffff88814e7b1f08 R14: ffff88813cb99c00 R15: 0000000000000001 <4> [317.128174] FS: 0000000000000000(0000) GS:ffff88846f280000(0000) knlGS:0000000000000000 <4> [317.128176] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 <4> [317.128178] CR2: 0000000000000010 CR3: 000000011f676004 CR4: 0000000000770ef0 <4> [317.128180] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 <4> [317.128182] DR3: 0000000000000000 DR6: 00000000ffff07f0 DR7: 0000000000000400 <4> [317.128184] PKRU: 55555554 <4> [317.128185] Call Trace: <4> [317.128187] <TASK> <4> [317.128189] ? show_regs+0x67/0x70 <4> [317.128194] ? __die_body+0x20/0x70 <4> [317.128196] ? __die+0x2b/0x40 <4> [317.128198] ? page_fault_oops+0x15f/0x4e0 <4> [317.128203] ? do_user_addr_fault+0x3fb/0x970 <4> [317.128205] ? lock_acquire+0xc7/0x2e0 <4> [317.128209] ? exc_page_fault+0x87/0x2b0 <4> [317.128212] ? asm_exc_page_fault+0x27/0x30			

		<4> [317.128216] ? xe_migrate_copy+0x66/0x13e0 [xe] <4> [317.128263] ? __lock_acquire+0xb9d/0x26f0 <4> [317.128265] ? __lock_acquire+0xb9d/0x26f0 <4> [317.128267] ? sg_free_append_table+0x20/0x80 <4> [317.128271] ? lock_acquire+0xc7/0x2e0 <4> [317.128273] ? mark_held_locks+0x4d/0x80 <4> [317.128275] ? trace_hardirqs_on+0x1e/0xd0 <4> [317.128278] ? _raw_spin_unlock_irqrestore+0x31/0x60 <4> [317.128281] ? __pm_runtime_resume+0x60/0xa0 <4> [317.128284] xe_bo_move+0x682/0xc50 [xe] <4> [317.128315] ? lock_is_held_type+0xaa/0x120 <4> [317.128318] ttm_bo_handle_move_mem+0xe5/0x1a0 [ttm] <4> [317.128324] ttm_bo_validate+0xd1/0x1a0 [ttm] <4> [317.128328] shrink_test_run_device+0x721/0xc10 [xe] <4> [317.128360] ? find_held_lock+0x31/0x90 <4> [317.128363] ? lock_release+0xd1/0x2a0 <4> [317.128365] ? __pfx_kunit_generic_run_threadfn_adapter+0x10/0x10 [kunit] <4> [317.128370] xe_bo_shrink_kunit+0x11/0x20 [xe] <4> [317.128397] kunit_try_run_case+0x6e/0x150 [kunit] <4> [317.128400] ? trace_hardirqs_on+0x1e/0xd0 <4> [317.128402] ? _raw_spin_unlock_irqrestore+0x31/0x60 <4> [317.128404] kunit_generic_run_threadfn_adapter+0x1e/0x40 [ku ---truncated---			
[CVE-2024-49943](CVE-2024-49943)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/xe/guc_submit: add missing locking in wedged_fini Any non-wedged queue can have a zero refcount here and can be running concurrently with an async queue destroy, therefore dereferencing the queue ptr to check wedge status after the lookup can trigger UAF if queue is not wedged. Fix this by keeping the submission_state lock held	2024-10-21	5.5	Medium

		around the check to postpone the free and make the check safe, before dropping again around the put() to avoid the deadlock. (cherry picked from commit d28af0b6b9580b9f90c265a7da0315b0ad20bbfd)			
CVE-2024-49945	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/ncsi: Disable the ncsi work before freeing the associated structure The work function can run after the ncsi device is freed, resulting in use-after-free bugs or kernel panic.	2024-10-21	5.5	Medium
CVE-2024-49956	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: gfs2: fix double destroy_workqueue error When gfs2_fill_super() fails, destroy_workqueue() is called within gfs2_gl_hash_clear(), and the subsequent code path calls destroy_workqueue() on the same work queue again. This issue can be fixed by setting the work queue pointer to NULL after the first destroy_workqueue() call and checking for a NULL pointer before attempting to destroy the work queue again.	2024-10-21	5.5	Medium
CVE-2024-49957	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix null-ptr-deref when journal load failed. During the mounting process, if journal_reset() fails because of too short journal, then lead to jbd2_journal_load() fails with NULL j_sb_buffer. Subsequently, ocfs2_journal_shutdown() calls jbd2_journal_flush()->jbd2_cleanup_journal_tail()-> __jbd2_update_log_tail()->jbd2_journal_update_sb_log_tail() ->lock_buffer(journal->j_sb_buffer), resulting in a null-pointer dereference error. To resolve this issue, we should check the JBD2_LOADED flag to ensure the	2024-10-21	5.5	Medium

		journal was properly loaded. Additionally, use journal instead of osb->journal directly to simplify the code.			
CVE-2024-49962	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ACPICA: check null return of ACPI_ALLOCATE_ZEROED() in acpi_db_convert_to_package() ACPICA commit 4d4547cf13cca820ff7e0f859ba83e1a610b9fd0 ACPI_ALLOCATE_ZEROED() may fail, elements might be NULL and will cause NULL pointer dereference later. [rjw: Subject and changelog edits]	2024-10-21	5.5	Medium
CVE-2024-49970	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Implement bounds check for stream encoder creation in DCN401 'stream_enc_regs' array is an array of dcn10_stream_enc_registers structures. The array is initialized with four elements, corresponding to the four calls to stream_enc_regs() in the array initializer. This means that valid indices for this array are 0, 1, 2, and 3. The error message 'stream_enc_regs' 4 <= 5 below, is indicating that there is an attempt to access this array with an index of 5, which is out of bounds. This could lead to undefined behavior Here, eng_id is used as an index to access the stream_enc_regs array. If eng_id is 5, this would result in an out-of-bounds access on the stream_enc_regs array. Thus fixing Buffer overflow error in dcn401_stream_encoder_create Found by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/resource/dcn401/dcn401_resource.c:1209	2024-10-21	5.5	Medium

		dcn401_stream_encoder_create() error: buffer overflow 'stream_enc_regs' 4 <= 5			
CVE-2024-49971	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Increase array size of dummy_boolean [WHY] dml2_core_shared_mode_support and dml_core_mode_support access the third element of dummy_boolean, i.e. hw_debug5 = &s->dummy_boolean[2], when dummy_boolean has size of 2. Any assignment to hw_debug5 causes an OVERRUN. [HOW] Increase dummy_boolean's array size to 3. This fixes 2 OVERRUN issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-49972	linux - linux_kernel	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Deallocate DML memory if allocation fails [Why] When DC state create DML memory allocation fails, memory is not deallocated subsequently, resulting in uninitialized structure that is not NULL. [How] Deallocate memory if DML memory allocation fails.	2024-10-21	5.5	Medium
CVE-2024-49973	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: r8169: add tally counter fields added with RTL8125 RTL8125 added fields to the tally counter, what may result in the chip dma'ing these new fields to unallocated memory. Therefore make sure that the allocated memory area is big enough to hold all of the tally counter values, even if we use only parts of it.	2024-10-21	5.5	Medium
CVE-2024-49974	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2024-10-21	5.5	Medium

		NFSD: Limit the number of concurrent async COPY operations Nothing appears to limit the number of concurrent async COPY operations that clients can start. In addition, AFAICT each async COPY can copy an unlimited number of 4MB chunks, so can run for a long time. Thus IMO async COPY can become a DoS vector. Add a restriction mechanism that bounds the number of concurrent background COPY operations. Start simple and try to be fair -- this patch implements a per-namespace limit. An async COPY request that occurs while this limit is exceeded gets NFS4ERR_DELAY. The requesting client can choose to send the request again after a delay or fall back to a traditional read/write style copy. If there is need to make the mechanism more sophisticated, we can visit that in future patches.			
CVE-2024-49975	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: uprobes: fix kernel info leak via "[uprobes]" vma xol_add_vma() maps the uninitialized page allocated by __create_xol_area() into userspace. On some architectures (x86) this memory is readable even without VM_READ, VM_EXEC results in the same pgprot_t as VM_EXEC|VM_READ, although this doesn't really matter, debugger can read this memory anyway.	2024-10-21	5.5	Medium
CVE-2024-49976	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Drop interface_lock in stop_kthread() stop_kthread() is the offline callback for "trace/osnoise:online", since commit 5bfbcd1ee57b ("tracing/timerlat: Add	2024-10-21	5.5	Medium

<table>
<tr><td></td><td></td><td>interface_lock around clearing
of kthread in stop_kthread()"), the following ABBA deadlock scenario is
introduced:

T1 | T2 [BP] | T3 [AP]
osnoise_hotplug_workfn() | work_for_cpu_fn()
| cpuhp_thread_fun()
| _cpu_down() |
osnoise_cpu_die()
mutex_lock(&interface_lock) | |
stop_kthread()
| cpus_write_lock() |
mutex_lock(&interface_lock)
cpus_read_lock() | cpuhp_kick_ap() |

As the interface_lock here in just for protecting the "kthread" field of
the osn_var, use xchg() instead to fix this issue. Also use
for_each_online_cpu() back in
stop_per_cpu_kthreads() as it can take
cpu_read_lock() again.</td><td></td><td></td><td></td></tr>
<tr><td>CVE-2024-49977</td><td>linux - multiple products</td><td>In the Linux kernel, the following vulnerability has been resolved:

net: stmmac: Fix zero-division error when disabling tc cbs

The commit b8c43360f6e4 ("net: stmmac: No need to calculate speed divider
when offload is disabled") allows the "port_transmit_rate_kbps" to be
set to a value of 0, which is then passed to the "div_s64" function when
tc-cbs is disabled. This leads to a zero-division error.

When tc-cbs is disabled, the idleslope, sendslope, and credit values the
credit values are not required to be configured. Therefore, adding a return
statement after setting the txQ mode to DCB when tc-cbs is disabled would
prevent a zero-division error.</td><td>2024-10-21</td><td>5.5</td><td>Medium</td></tr>
<tr><td>CVE-2024-49978</td><td>linux - multiple products</td><td>In the Linux kernel, the following vulnerability has been resolved:

gso: fix udp gso fraglist segmentation after pull from frag_list

Detect gso fraglist skbs with corrupted geometry (see</td><td>2024-10-21</td><td>5.5</td><td>Medium</td></tr>
</table>

		below) and pass these to skb_segment instead of skb_segment_list, as the first can segment them correctly. Valid SKB_GSO_FRAGLIST skbs - consist of two or more segments - the head_skb holds the protocol headers plus first gso_size - one or more frag_list skbs hold exactly one segment - all but the last must be gso_size Optional datapath hooks such as NAT and BPF (bpf_skb_pull_data) can modify these skbs, breaking these invariants. In extreme cases they pull all data into skb linear. For UDP, this causes a NULL ptr deref in __udpv4_gso_segment_list_csum at udp_hdr(seg->next)->dest. Detect invalid geometry due to pull, by checking head_skb size. Don't just drop, as this may blackhole a destination. Convert to be able to pass to regular skb_segment.			
[CVE-2024-49979](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: gso: fix tcp fraglist segmentation after pull from frag_list Detect tcp gso fraglist skbs with corrupted geometry (see below) and pass these to skb_segment instead of skb_segment_list, as the first can segment them correctly. Valid SKB_GSO_FRAGLIST skbs - consist of two or more segments - the head_skb holds the protocol headers plus first gso_size - one or more frag_list skbs hold exactly one segment - all but the last must be gso_size Optional datapath hooks such as NAT and BPF (bpf_skb_pull_data) can modify these skbs, breaking these invariants. In extreme cases they pull all data into skb linear. For	2024-10-21	5.5	Medium

		TCP, this causes a NULL ptr deref in __tcpv4_gso_segment_list_csum at tcp_hdr(seg->next). Detect invalid geometry due to pull, by checking head_skb size. Don't just drop, as this may blackhole a destination. Convert to be able to pass to regular skb_segment. Approach and description based on a patch by Willem de Bruijn.			
CVE-2024-49980	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: vrf: revert "vrf: Remove unnecessary RCU-bh critical section" This reverts commit 504fc6f4f7f681d2a03aa5f68aad549d90eab853. dev_queue_xmit_nit is expected to be called with BH disabled. __dev_queue_xmit has the following: /* Disable soft irqs for various locks below. Also * stops preemption for RCU. */ rcu_read_lock_bh(); VRF must follow this invariant. The referenced commit removed this protection. Which triggered a lockdep warning: ================================ WARNING: inconsistent lock state 6.11.0 #1 Tainted: G W -------------------------------- inconsistent {IN-SOFTIRQ-W} -> {SOFTIRQ-ON-W} usage. btserver/134819 [HC0[0]:SC0[0]:HE1:SE1] takes: ffff8882da30c118 (rlock-AF_PACKET){+.?.}-{2:2}, at: tpacket_rcv+0x863/0x3b30 {IN-SOFTIRQ-W} state was registered at: lock_acquire+0x19a/0x4f0 _raw_spin_lock+0x27/0x40 packet_rcv+0xa33/0x1320 __netif_receive_skb_core.constprop.0+0xcb0/0x3a90 __netif_receive_skb_list_core+0x2c9/0x890	2024-10-21	5.5	Medium

		netif_receive_skb_list_internal+0x610/0xcc0 [...] other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(rlock-AF_PACKET); <Interrupt> lock(rlock-AF_PACKET); *** DEADLOCK *** Call Trace: <TASK> dump_stack_lvl+0x73/0xa0 mark_lock+0x102e/0x16b0 __lock_acquire+0x9ae/0x6170 lock_acquire+0x19a/0x4f0 _raw_spin_lock+0x27/0x40 tpacket_rcv+0x863/0x3b30 dev_queue_xmit_nit+0x709/0xa40 vrf_finish_direct+0x26e/0x340 [vrf] vrf_l3_out+0x5f4/0xe80 [vrf] __ip_local_out+0x51e/0x7a0 [...]			
CVE-2024-49985	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: i2c: stm32f7: Do not prepare/unprepare clock during runtime suspend/resume In case there is any sort of clock controller attached to this I2C bus controller, for example Versaclock or even an AIC32x4 I2C codec, then an I2C transfer triggered from the clock controller clk_ops .prepare callback may trigger a deadlock on drivers/clk/clk.c prepare_lock mutex. This is because the clock controller first grabs the prepare_lock mutex and then performs the prepare operation, including its I2C access. The I2C access resumes this I2C bus controller via .runtime_resume callback, which calls clk_prepare_enable(), which attempts to grab the prepare_lock mutex again and deadlocks.	2024-10-21	5.5	Medium

		Since the clock are already prepared since probe() and unprepared in remove(), use simple clk_enable()/clk_disable() calls to enable and disable the clock on runtime suspend and resume, to avoid hitting the prepare_lock mutex.			
CVE-2024-49987	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: bpftool: Fix undefined behavior in qsort(NULL, 0, ...) When netfilter has no entry to display, qsort is called with qsort(NULL, 0, ...). This results in undefined behavior, as UBSan reports: net.c:827:2: runtime error: null pointer passed as argument 1, which is declared to never be null Although the C standard does not explicitly state whether calling qsort with a NULL pointer when the size is 0 constitutes undefined behavior, Section 7.1.4 of the C standard (Use of library functions) mentions: "Each of the following statements applies unless explicitly stated otherwise in the detailed descriptions that follow: If an argument to a function has an invalid value (such as a value outside the domain of the function, or a pointer outside the address space of the program, or a null pointer, or a pointer to non-modifiable storage when the corresponding parameter is not const-qualified) or a type (after promotion) not expected by a function with variable number of arguments, the behavior is undefined." To avoid this, add an early return when nf_link_info is NULL to prevent calling qsort with a NULL pointer.	2024-10-21	5.5	Medium
CVE-2024-49988	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2024-10-21	5.5	Medium

		ksmbd: add refcnt to ksmbd_conn struct When sending an oplock break request, opinfo->conn is used, But freed ->conn can be used on multichannel. This patch add a reference count to the ksmbd_conn struct so that it can be freed when it is no longer used.			
[CVE-2024-49990](https://nvd.nist.gov/vuln/detail/CVE-2024-49990)	linux - multiple products	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.	2024-10-21	5.5	Medium
[CVE-2024-49993](https://nvd.nist.gov/vuln/detail/CVE-2024-49993)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix potential lockup if qi_submit_sync called with 0 count If qi_submit_sync() is invoked with 0 invalidation descriptors (for instance, for DMA draining purposes), we can run into a bug where a submitting thread fails to detect the completion of invalidation_wait. Subsequently, this led to a soft lockup. Currently, there is no impact by this bug on the existing users because no callers are submitting invalidations with 0 descriptors. This fix will enable future users (such as DMA drain) calling qi_submit_sync() with 0 count. Suppose thread T1 invokes qi_submit_sync() with non-zero descriptors, while concurrently, thread T2 calls qi_submit_sync() with zero descriptors. Both threads then enter a while loop, waiting for their respective descriptors to complete. T1 detects its completion (i.e., T1's invalidation_wait status changes to QI_DONE by HW) and proceeds to call reclaim_free_desc() to reclaim all descriptors, potentially including adjacent ones of other threads that are also marked as QI_DONE.	2024-10-21	5.5	Medium

		During this time, while T2 is waiting to acquire the qi->q_lock, the IOMMU hardware may complete the invalidation for T2, setting its status to QI_DONE. However, if T1's execution of reclaim_free_desc() frees T2's invalidation_wait descriptor and changes its status to QI_FREE, T2 will not observe the QI_DONE status for its invalidation_wait and will indefinitely remain stuck. This soft lockup does not occur when only non-zero descriptors are submitted.In such cases, invalidation descriptors are interspersed among wait descriptors with the status QI_IN_USE, acting as barriers. These barriers prevent the reclaim code from mistakenly freeing descriptors belonging to other submitters. Considered the following example timeline: T1 T2 == ID1 WD1 while(WD1!=QI_DONE) unlock lock WD1=QI_DONE* WD2 while(WD2!=QI_DONE) unlock lock WD1==QI_DONE? ID1=QI_DONE WD2=DONE* reclaim() ID1=FREE WD1=FREE WD2=FREE unlock soft lockup! T2 never sees QI_DONE in WD2 Where: ID = invalidation descriptor WD = wait descriptor * Written by hardware The root of the problem is that the descriptor status QI_DONE flag is used			

		for two conflicting purposes: 1. signal a descriptor is ready for reclaim (to be freed) 2. signal by the hardware that a wait descriptor is complete The solution (in this patch) is state separation by using QI_FREE flag for #1. Once a thread's invalidation descriptors are complete, their status would be set to QI_FREE. The reclaim_free_desc() function would then only free descriptors marked as QI_FREE instead of those marked as QI_DONE. This change ensures that T2 (from the previous example) will correctly observe the completion of its invalidation_wait (marked as QI_DONE).			
[CVE-2024-49994](CVE-2024-49994)	linux - multiple products	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	2024-10-21	5.5	Medium
[CVE-2024-49999](CVE-2024-49999)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: afs: Fix the setting of the server responding flag In afs_wait_for_operation(), we set transcribe the call	2024-10-21	5.5	Medium

		responded flag to the server record that we used after doing the fileserver iteration loop - but it's possible to exit the loop having had a response from the server that we've discarded (e.g. it returned an abort or we started receiving data, but the call didn't complete). This means that op->server might be NULL, but we don't check that before attempting to set the server flag.			
CVE-2024-50000	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix NULL deref in mlx5e_tir_builder_alloc() In mlx5e_tir_builder_alloc() kvzalloc() may return NULL which is dereferenced on the next line in a reference to the modify field. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2024-10-21	5.5	Medium
CVE-2024-50001	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix error path in multi-packet WQE transmit Remove the erroneous unmap in case no DMA mapping was established The multi-packet WQE transmit code attempts to obtain a DMA mapping for the skb. This could fail, e.g. under memory pressure, when the IOMMU driver just can't allocate more memory for page tables. While the code tries to handle this in the path below the err_unmap label it erroneously unmaps one entry from the sq's FIFO list of active mappings. Since the current map attempt failed this unmap is removing some random DMA mapping that might still be required. If the PCI function now presents that IOVA, the IOMMU may assumes a rogue DMA access and e.g. on s390 puts the PCI function in error state.	2024-10-21	5.5	Medium

		The erroneous behavior was seen in a stress-test environment that created memory pressure.			

CVE-2024-50002	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		static_call: Handle module init failure correctly in static_call_del_module()
		
		Module insertion invokes static_call_add_module() to initialize the static calls in a module. static_call_add_module() invokes __static_call_init(), which allocates a struct static_call_mod to either encapsulate the built-in static call sites of the associated key into it so further modules can be added or to append the module to the module chain.
		
		If that allocation fails the function returns with an error code and the module core invokes static_call_del_module() to clean up eventually added static_call_mod entries.
		
		This works correctly, when all keys used by the module were converted over to a module chain before the failure. If not then static_call_del_module() causes a #GP as it blindly assumes that key::mods points to a valid struct static_call_mod.
		
		The problem is that key::mods is not a individual struct member of struct static_call_key, it's part of a union to save space:
		
		union {
		/* bit 0: 0 = mods, 1 = sites */
		unsigned long type;
		struct static_call_mod *mods;
		struct static_call_site *sites;
		};
		
		key::sites is a pointer to the list of built-in usage sites of the static call. The type of the pointer is differentiated by bit 0. A mods pointer has the bit clear, the sites pointer has the bit set.
			2024-10-21	5.5	Medium

		As static_call_del_module() blidly assumes that the pointer is a valid static_call_mod type, it fails to check for this failure case and dereferences the pointer to the list of built-in call sites, which is obviously bogus. Cure it by checking whether the key has a sites or a mods pointer. If it's a sites pointer then the key is not to be touched. As the sites are walked in the same order as in __static_call_init() the site walk can be terminated because all subsequent sites have not been touched by the init code due to the error exit. If it was converted before the allocation fail, then the inner loop which searches for a module match will find nothing. A fail in the second allocation in __static_call_init() is harmless and does not require special treatment. The first allocation succeeded and converted the key to a module chain. That first entry has mod::mod == NULL and mod::next == NULL, so the inner loop of static_call_del_module() will neither find a module match nor a module chain. The next site in the walk was either already converted, but can't match the module, or it will exit the outer loop because it has a static_call_site pointer and not a static_call_mod pointer.			
CVE-2024-50003	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix system hang while resume with TBT monitor [Why] Connected with a Thunderbolt monitor and do the suspend and the system may hang while resume. The TBT monitor HPD will be triggered during the resume procedure	2024-10-21	5.5	Medium

		and call the drm_client_modeset_probe() while struct drm_connector connector->dev->master is NULL. It will mess up the pipe topology after resume. [How] Skip the TBT monitor HPD during the resume procedure because we currently will probe the connectors after resume by default. (cherry picked from commit 453f86a26945207a16b8f66aaed5962dc2b95b85)			
CVE-2024-50009	linux - linux_kernel	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.	2024-10-21	5.5	Medium
CVE-2024-50011	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: soc-acpi-intel-rpl-match: add missing empty item There is no links_num in struct snd_soc_acpi_mach {}, and we test !link->num_adr as a condition to end the loop in hda_sdw_machine_select(). So an empty item in struct snd_soc_acpi_link_adr array is required.	2024-10-21	5.5	Medium
CVE-2024-50012	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: cpufreq: Avoid a bad reference count on CPU node In the parse_perf_domain function, if the call to of_parse_phandle_with_args returns an error, then the reference to the CPU device node that was acquired at the start of the function would not be properly decremented. Address this by declaring the variable with the	2024-10-21	5.5	Medium

		__free(device_node) cleanup attribute.			
CVE-2024-50013	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: exfat: fix memory leak in exfat_load_bitmap() If the first directory entry in the root directory is not a bitmap directory entry, 'bh' will not be released and reassigned, which will cause a memory leak.	2024-10-21	5.5	Medium
CVE-2024-50014	linux - multiple products	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: <TASK> 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	2024-10-21	5.5	Medium

		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-2024-50015	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		ext4: dax: fix overflowing extents beyond inode size when partially writing
		
		The dax_iomap_rw() does two things in each iteration: map written blocks and copy user data to blocks. If the process is killed by user(See signal handling in dax_iomap_iter()), the copied data will be returned and added on inode size, which means that the length of written extents may exceed the inode size, then fsck will fail. An example is given as:
		
		dd if=/dev/urandom of=file bs=4M count=1
		 dax_iomap_rw
		  iomap_iter // round 1
		   ext4_iomap_begin
		    ext4_iomap_alloc // allocate 0~2M extents(written flag)
		  dax_iomap_iter // copy 2M data
		  iomap_iter // round 2
		   iomap_iter_advance
		    iter->pos += iter->processed // iter->pos = 2M
		   ext4_iomap_begin
		    ext4_iomap_alloc // allocate 2~4M extents(written flag)
		  dax_iomap_iter
		   fatal_signal_pending
		  done = iter->pos - iocb->ki_pos // done = 2M
		 ext4_handle_inode_extension
		  ext4_update_inode_size // inode size = 2M
		
		fsck reports: Inode 13, i_size is 2097152, should be 4194304. Fix?
			2024-10-21	5.5	Medium

		Fix the problem by truncating extents if the written length is smaller than expected.			
CVE-2024-50016	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Avoid overflow assignment in link_dp_cts sampling_rate is an uint8_t but is assigned an unsigned int, and thus it can overflow. As a result, sampling_rate is changed to uint32_t. Similarly, LINK_QUAL_PATTERN_SET has a size of 2 bits, and it should only be assigned to a value less or equal than 4. This fixes 2 INTEGER_OVERFLOW issues reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-50017	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: x86/mm/ident_map: Use gbpages only where full GB page should be mapped. When ident_pud_init() uses only GB pages to create identity maps, large ranges of addresses not actually requested can be included in the resulting table; a 4K request will map a full GB. This can include a lot of extra address space past that requested, including areas marked reserved by the BIOS. That allows processor speculation into reserved regions, that on UV systems can cause system halts. Only use GB pages when map creation requests include the full GB page of space. Fall back to using smaller 2M pages when only portions of a GB page are included in the request. No attempt is made to coalesce mapping requests. If a request requires a map entry at the 2M (pmd) level, subsequent mapping requests within the same 1G region will also be at the pmd level, even if adjacent or	2024-10-21	5.5	Medium

		overlapping such requests could have been combined to map a full GB page. Existing usage starts with larger regions and then adds smaller regions, so this should not have any great consequence.			

[CVE-2024-50018](https://nvd.nist.gov/vuln/detail/CVE-2024-50018)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net: napi: Prevent overflow of napi_defer_hard_irqs
		
		In commit 6f8b12d661d0 ("net: napi: add hard irqs deferral feature")
		napi_defer_irqs was added to net_device and napi_defer_irqs_count was
		added to napi_struct, both as type int.
		
		This value never goes below zero, so there is not reason for it to be a
		signed int. Change the type for both from int to u32, and add an
		overflow check to sysfs to limit the value to S32_MAX.
		
		The limit of S32_MAX was chosen because the practical limit before this
		patch was S32_MAX (anything larger was an overflow) and thus there are
		no behavioral changes introduced. If the extra bit is needed in the
		future, the limit can be raised.
		
		Before this patch:
		
		$ sudo bash -c 'echo 2147483649 > /sys/class/net/eth4/napi_defer_hard_irqs'
		$ cat /sys/class/net/eth4/napi_defer_hard_irqs
		-2147483647
		
		After this patch:
		
		$ sudo bash -c 'echo 2147483649 > /sys/class/net/eth4/napi_defer_hard_irqs'
		bash: line 0: echo: write error: Numerical result out of range
		
		Similarly, /sys/class/net/XXXXX/tx_queue_len is defined as unsigned:
		
		include/linux/netdevice.h: unsigned int tx_queue_len;
			2024-10-21	5.5	Medium

		And has an overflow check:
		
		dev_change_tx_queue_len(..., unsigned long new_len):
		
		if (new_len != (unsigned int)new_len)
		return -ERANGE;			

CVE-2022-48946	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: udf: Fix preallocation discarding at indirect extent boundary When preallocation extent is the first one in the extent block, the code would corrupt extent tree header instead. Fix the problem and use udf_delete_aext() for deleting extent to avoid some code duplication.	2024-10-21	5.5	Medium
CVE-2022-48947	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix u8 overflow By keep sending L2CAP_CONF_REQ packets, chan->num_conf_rsp increases multiple times and eventually it will wrap around the maximum number (i.e., 255). This patch prevents this by adding a boundary check with L2CAP_MAX_CONF_RSP Btmon log: Bluetooth monitor ver 5.64 = Note: Linux version 6.1.0-rc2 (x86_64) 0.264594 = Note: Bluetooth subsystem version 2.22 0.264636 @ MGMT Open: btmon (privileged) version 1.22 {0x0001} 0.272191 = New Index: 00:00:00:00:00:00 (Primary,Virtual,hci0) [hci0] 13.877604 @ RAW Open: 9496 (privileged) version 2.22 {0x0002} 13.890741 = Open Index: 00:00:00:00:00:00 [hci0] 13.900426 (...) > ACL Data RX: Handle 200 flags 0x00 dlen 1033 #32 [hci0] 14.273106 invalid packet size (12 != 1033)	2024-10-21	5.5	Medium

		08 00 01 00 02 01 04 00 01 10 ff ff .... > ACL Data RX: Handle 200 flags 0x00 dlen 1547 #33 [hci0] 14.273561 invalid packet size (14 != 1547) 0a 00 01 00 04 01 06 00 40 00 00 00 00 00 ....@..... > ACL Data RX: Handle 200 flags 0x00 dlen 2061 #34 [hci0] 14.274390 invalid packet size (16 != 2061) 0c 00 01 00 04 01 08 00 40 00 00 00 00 00 00 04 ....@....... > ACL Data RX: Handle 200 flags 0x00 dlen 2061 #35 [hci0] 14.274932 invalid packet size (16 != 2061) 0c 00 01 00 04 01 08 00 40 00 00 00 07 00 03 00 ....@....... = bluetoothd: Bluetooth daemon 5.43 14.401828 > ACL Data RX: Handle 200 flags 0x00 dlen 1033 #36 [hci0] 14.275753 invalid packet size (12 != 1033) 08 00 01 00 04 01 04 00 40 00 00 00 ....@...			
CVE-2022-48949	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: igb: Initialize mailbox message for VF reset When a MAC address is not assigned to the VF, that portion of the message sent to the VF is not set. The memory, however, is allocated from the stack meaning that information may be leaked to the VM. Initialize the message buffer to 0 so that no information is passed to the VM in this case.	2024-10-21	5.5	Medium
CVE-2022-48952	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: PCI: mt7621: Add sentinel to quirks table Current driver is missing a sentinel in the struct soc_device_attribute array, which causes an oops when assessed by the soc_device_match(mt7621_pcie_quirks_match) call. This was only exposed once the CONFIG_SOC_MT7621 mt7621 soc_dev_attr was fixed to register the SOC as a device, in:	2024-10-21	5.5	Medium

		commit 7c18b64bba3b ("mips: ralink: mt7621: do not use kzalloc too early") Fix it by adding the required sentinel.			
CVE-2022-48953	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rtc: cmos: Fix event handler registration ordering issue Because acpi_install_fixed_event_handler() enables the event automatically on success, it is incorrect to call it before the handler routine passed to it is ready to handle events. Unfortunately, the rtc-cmos driver does exactly the incorrect thing by calling cmos_wake_setup(), which passes rtc_handler() to acpi_install_fixed_event_handler(), before cmos_do_probe(), because rtc_handler() uses dev_get_drvdata() to get to the cmos object pointer and the driver data pointer is only populated in cmos_do_probe(). This leads to a NULL pointer dereference in rtc_handler() on boot if the RTC fixed event happens to be active at the init time. To address this issue, change the initialization ordering of the driver so that cmos_wake_setup() is always called after a successful cmos_do_probe() call. While at it, change cmos_pnp_probe() to call cmos_do_probe() after the initial if () statement used for computing the IRQ argument to be passed to cmos_do_probe() which is cleaner than calling it in each branch of that if () (local variable "irq" can be of type int, because it is passed to that function as an argument of type int). Note that commit 6492fed7d8c9 ("rtc: rtc-cmos: Do	2024-10-21	5.5	Medium

		not check ACPI_FADT_LOW_POWER_S0") caused this issue to affect a larger number of systems, because previously it only affected systems with ACPI_FADT_LOW_POWER_S0 set, but it is present regardless of that commit.			
CVE-2022-48955	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: thunderbolt: fix memory leak in tbnet_open() When tb_ring_alloc_rx() failed in tbnet_open(), ida that allocated in tb_xdomain_alloc_out_hopid() is not released. Add tb_xdomain_release_out_hopid() to the error path to release ida.	2024-10-21	5.5	Medium
CVE-2022-48957	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: dpaa2-switch: Fix memory leak in dpaa2_switch_acl_entry_add() and dpaa2_switch_acl_entry_remove() The cmd_buff needs to be freed when error happened in dpaa2_switch_acl_entry_add() and dpaa2_switch_acl_entry_remove().	2024-10-21	5.5	Medium
CVE-2022-48958	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ethernet: aeroflex: fix potential skb leak in greth_init_rings() The greth_init_rings() function won't free the newly allocated skb when dma_mapping_error() returns error, so add dev_kfree_skb() to fix it. Compile tested only.	2024-10-21	5.5	Medium
CVE-2022-48959	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: dsa: sja1105: fix memory leak in sja1105_setup_devlink_regions() When dsa_devlink_region_create failed in sja1105_setup_devlink_regions(), priv->regions is not released.	2024-10-21	5.5	Medium

CVE-2022-48961	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mdio: fix unbalanced fwnode reference count in mdio_device_release() There is warning report about of_node refcount leak while probing mdio device: OF: ERROR: memory leak, expected refcount 1 instead of 2, of_node_get()/of_node_put() unbalanced - destroy cset entry: attach overlay node /spi/soc@0/mdio@710700c0/ethernet@4 In of_mdiobus_register_device(), we increase fwnode refcount by fwnode_handle_get() before associating the of_node with mdio device, but it has never been decreased in normal path. Since that, in mdio_device_release(), it needs to call fwnode_handle_put() in addition instead of calling kfree() directly. After above, just calling mdio_device_free() in the error handle path of of_mdiobus_register_device() is enough to keep the refcount balanced.	2024-10-21	5.5	Medium
CVE-2022-48963	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: wwan: iosm: fix memory leak in ipc_mux_init() When failed to alloc ipc_mux->ul_adb.pp_qlt in ipc_mux_init(), ipc_mux is not released.	2024-10-21	5.5	Medium
CVE-2022-48965	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio/rockchip: fix refcount leak in rockchip_gpiolib_register() The node returned by of_get_parent() with refcount incremented, of_node_put() needs be called when finish using it. So add it in the end of of_pinctrl_get().	2024-10-21	5.5	Medium

CVE-2022-48968	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: Fix potential memory leak in otx2_init_tc() In otx2_init_tc(), if rhashtable_init() failed, it does not free tc->tc_entries_bitmap which is allocated in otx2_tc_alloc_ent_bitmap().	2024-10-21	5.5	Medium
CVE-2022-48969	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: xen-netfront: Fix NULL sring after live migration A NAPI is setup for each network sring to poll data to kernel The sring with source host is destroyed before live migration and new sring with target host is setup after live migration. The NAPI for the old sring is not deleted until setup new sring with target host after migration. With busy_poll/busy_read enabled, the NAPI can be polled before got deleted when resume VM. BUG: unable to handle kernel NULL pointer dereference at 0000000000000008 IP: xennet_poll+0xae/0xd20 PGD 0 P4D 0 Oops: 0000 [#1] SMP PTI Call Trace: finish_task_switch+0x71/0x230 timerqueue_del+0x1d/0x40 hrtimer_try_to_cancel+0xb5/0x110 xennet_alloc_rx_buffers+0x2a0/0x2a0 napi_busy_loop+0xdb/0x270 sock_poll+0x87/0x90 do_sys_poll+0x26f/0x580 tracing_map_insert+0x1d4/0x2f0 event_hist_trigger+0x14a/0x260 finish_task_switch+0x71/0x230 __schedule+0x256/0x890 recalc_sigpending+0x1b/0x50 xen_sched_clock+0x15/0x20 __rb_reserve_next+0x12d/0x140 ring_buffer_lock_reserve+0x123/0x3d0	2024-10-21	5.5	Medium

		event_triggers_call+0x87/0xb0 trace_event_buffer_commit+0x1c4/0x210 xen_clocksource_get_cycles+0x15/0x20 ktime_get_ts64+0x51/0xf0 SyS_ppoll+0x160/0x1a0 SyS_ppoll+0x160/0x1a0 do_syscall_64+0x73/0x130 entry_SYSCALL_64_after_hwframe+0x41/0xa6 ... RIP: xennet_poll+0xae/0xd20 RSP: ffffb4f041933900 CR2: 0000000000000008 ---[end trace f8601785b354351c]--- xen frontend should remove the NAPIs for the old srings before live migration as the bond srings are destroyed There is a tiny window between the srings are set to NULL and the NAPIs are disabled, It is safe as the NAPI threads are still frozen at that time			
[CVE-2022-48970](https://nvd.nist.gov/vuln/detail/CVE-2022-48970)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: af_unix: Get user_ns from in_skb in unix_diag_get_exact(). Wei Chen reported a NULL deref in sk_user_ns() [0][1], and Paolo diagnosed the root cause: in unix_diag_get_exact(), the newly allocated skb does not have sk. [2] We must get the user_ns from the NETLINK_CB(in_skb).sk and pass it to sk_diag_fill(). [0]: BUG: kernel NULL pointer dereference, address: 0000000000000270 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 12bbce067 P4D 12bbce067 PUD 12bc40067 PMD 0 Oops: 0000 [#1] PREEMPT SMP CPU: 0 PID: 27942 Comm: syz-executor.0 Not tainted 6.1.0-rc5-next-20221118 #2 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-48-gd9c812dda519-prebuilt.qemu.org	2024-10-21	5.5	Medium

		04/01/2014 RIP: 0010:sk_user_ns include/net/sock.h:920 [inline] RIP: 0010:sk_diag_dump_uid net/unix/diag.c:119 [inline] RIP: 0010:sk_diag_fill+0x77d/0x890 net/unix/diag.c:170 Code: 89 ef e8 66 d4 2d fd c7 44 24 40 00 00 00 00 49 8d 7c 24 18 e8 54 d7 2d fd 49 8b 5c 24 18 48 8d bb 70 02 00 00 e8 43 d7 2d fd <48> 8b 9b 70 02 00 00 48 8d 7b 10 e8 33 d7 2d fd 48 8b 5b 10 48 8d RSP: 0018:ffffc90000d67968 EFLAGS: 00010246 RAX: ffff88812badaa48 RBX: 0000000000000000 RCX: ffffffff840d481d RDX: 0000000000000465 RSI: 0000000000000000 RDI: 0000000000000270 RBP: ffffc90000d679a8 R08: 0000000000000277 R09: 0000000000000000 R10: 0001ffffffffffff R11: 0001c90000d679a8 R12: ffff88812ac03800 R13: ffff88812c87c400 R14: ffff88812ae42210 R15: ffff888103026940 FS: 00007f08b4e6f700(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000270 CR3: 000000012c58b000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> unix_diag_get_exact net/unix/diag.c:285 [inline] unix_diag_handler_dump+0x3f9/0x500 net/unix/diag.c:317 __sock_diag_cmd net/core/sock_diag.c:235 [inline] sock_diag_rcv_msg+0x237/0x250 net/core/sock_diag.c:266 netlink_rcv_skb+0x13e/0x250 net/netlink/af_netlink.c:2564 sock_diag_rcv+0x24/0x40 net/core/sock_diag.c:277 netlink_unicast_kernel net/netlink/af_netlink.c:1330 [inline] netlink_unicast+0x5e9/0x6b0 net/netlink/af_netlink.c:1356 netlink_sendmsg+0x739/0x860 net/netlink/af_netlink.c:1932 sock_sendmsg_nosec net/socket.c:714 [inline]			

		sock_sendmsg net/socket.c:734 [inline] ____sys_sendmsg+0x38f/0x500 net/socket.c:2476 ___sys_sendmsg net/socket.c:2530 [inline] __sys_sendmsg+0x197/0x230 net/socket.c:2559 __do_sys_sendmsg net/socket.c:2568 [inline] __se_sys_sendmsg net/socket.c:2566 [inline] __x64_sys_sendmsg+0x42/0x50 net/socket.c:2566 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x2b/0x70 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x4697f9 Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f08b4e6ec48 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffffffffda RBX: 000000000077bf80 RCX: 00000000004697f9 RDX: 0000000000000000 RSI: 00000000200001c0 RDI: 0000000000000003 RBP: 00000000004d29e9 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 000000000077bf80 R13: 0000000000000000 R14: 000000000077bf80 R15: 00007ffdb36bc6c0 </TASK> Modules linked in: CR2: 0000000000000270 [1]: https://lore.kernel.org/netdev/CAO4mrfdvyjFpokhNsiwZiP-wpdSD0AStcJwfKcKQdAALQ9_2Qw@mail.gmail.com/ [2]: https://lore.kernel.org/netdev/e04315e7c90d9a75613f3993c2baf2d344eef7eb.camel@redhat.com/			
CVE-2022-48971	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix not cleanup led when bt_init fails bt_init() calls bt_leds_init() to register led, but if it fails later, bt_leds_cleanup() is not called to unregister it. This can cause panic if the argument "bluetooth-power" in text is freed	2024-10-21	5.5	Medium

		and then another led_trigger_register() tries to access it: BUG: unable to handle page fault for address: ffffffffc06d3bc0 RIP: 0010:strcmp+0xc/0x30 Call Trace: <TASK> led_trigger_register+0x10d/0x4f0 led_trigger_register_simple+0x7d/0x100 bt_init+0x39/0xf7 [bluetooth] do_one_initcall+0xd0/0x4e0			
CVE-2022-48972	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mac802154: fix missing INIT_LIST_HEAD in ieee802154_if_add() Kernel fault injection test reports null-ptr-deref as follows: BUG: kernel NULL pointer dereference, address: 0000000000000008 RIP: 0010:cfg802154_netdev_notifier_call+0x120/0x310 include/linux/list.h:114 Call Trace: <TASK> raw_notifier_call_chain+0x6d/0xa0 kernel/notifier.c:87 call_netdevice_notifiers_info+0x6e/0xc0 net/core/dev.c:1944 unregister_netdevice_many_notify+0x60d/0xcb0 net/core/dev.c:1982 unregister_netdevice_queue+0x154/0x1a0 net/core/dev.c:10879 register_netdevice+0x9a8/0xb90 net/core/dev.c:10083 ieee802154_if_add+0x6ed/0x7e0 net/mac802154/iface.c:659 ieee802154_register_hw+0x29c/0x330 net/mac802154/main.c:229 mcr20a_probe+0xaaa/0xcb1 drivers/net/ieee802154/mcr20a.c:1316 ieee802154_if_add() allocates wpan_dev as netdev's private data, but not init the list in struct wpan_dev. cfg802154_netdev_notifier_call() manage the list when device register/unregister, and may lead to null-ptr-deref.	2024-10-21	5.5	Medium

		Use INIT_LIST_HEAD() on it to initialize it correctly.			
CVE-2022-48973	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpio: amd8111: Fix PCI device reference count leak for_each_pci_dev() is implemented by pci_get_device(). The comment of pci_get_device() says that it will increase the reference count for the returned pci_dev and also decrease the reference count for the input pci_dev @from if it is not NULL. If we break for_each_pci_dev() loop with pdev not NULL, we need to call pci_dev_put() to decrease the reference count. Add the missing pci_dev_put() after the 'out' label. Since pci_dev_put() can handle NULL input parameter, there is no problem for the 'Device not found' branch. For the normal path, add pci_dev_put() in amd_gpio_exit().	2024-10-21	5.5	Medium
CVE-2022-48974	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: fix using __this_cpu_add in preemptible Currently in nf_conntrack_hash_check_insert(), when it fails in nf_ct_ext_valid_pre/post(), NF_CT_STAT_INC() will be called in the preemptible context, a call trace can be triggered: BUG: using __this_cpu_add() in preemptible [00000000] code: conntrack/1636 caller is nf_conntrack_hash_check_insert+0x45/0x430 [nf_conntrack] Call Trace: <TASK> dump_stack_lvl+0x33/0x46 check_preemption_disabled+0xc3/0xf0 nf_conntrack_hash_check_insert+0x45/0x430 [nf_conntrack] ctnetlink_create_conntrack+0x3cd/0x4e0 [nf_conntrack_netlink] ctnetlink_new_conntrack+0x1c0/0x450	2024-10-21	5.5	Medium

		[nf_conntrack_netlink] nfnetlink_rcv_msg+0x277/0x2f0 [nfnetlink] netlink_rcv_skb+0x50/0x100 nfnetlink_rcv+0x65/0x144 [nfnetlink] netlink_unicast+0x1ae/0x290 netlink_sendmsg+0x257/0x4f0 sock_sendmsg+0x5f/0x70 This patch is to fix it by changing to use NF_CT_STAT_INC_ATOMIC() for nf_ct_ext_valid_pre/post() check in nf_conntrack_hash_check_insert(), as well as nf_ct_ext_valid_post() in __nf_conntrack_confirm(). Note that nf_ct_ext_valid_pre() check in __nf_conntrack_confirm() is safe to use NF_CT_STAT_INC(), as it's under local_bh_disable().			
CVE-2022-48975	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: gpiolib: fix memory leak in gpiochip_setup_dev() Here is a backtrace report about memory leak detected in gpiochip_setup_dev(): unreferenced object 0xffff88810b406400 (size 512): comm "python3", pid 1682, jiffies 4295346908 (age 24.090s) backtrace: kmalloc_trace device_add device_private_init at drivers/base/core.c:3361 (inlined by) device_add at drivers/base/core.c:3411 cdev_device_add gpiolib_cdev_register gpiochip_setup_dev gpiochip_add_data_with_key gcdev_register() & gcdev_unregister() would call device_add() & device_del() (no matter CONFIG_GPIO_CDEV is enabled or not) to register/unregister device. However, if device_add() succeeds, some resource (like struct device_private allocated by device_private_init())	2024-10-21	5.5	Medium

		is not released by device_del(). Therefore, after device_add() succeeds by gcdev_register(), it needs to call put_device() to release resource in the error handle path. Here we move forward the register of release function, and let it release every piece of resource by put_device() instead of kfree(). While at it, fix another subtle issue, i.e. when gc->ngpio is equal to 0, we still call kcalloc() and, in case of further error, kfree() on the ZERO_PTR pointer, which is not NULL. It's not a bug per se, but rather waste of the resources and potentially wrong expectation about contents of the gdev->descs variable.			
CVE-2022-48976	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable_offload: fix using __this_cpu_add in preemptible flow_offload_queue_work() can be called in workqueue without bh disabled, like the call trace showed in my act_ct testing, calling NF_FLOW_TABLE_STAT_INC() there would cause a call trace: BUG: using __this_cpu_add() in preemptible [00000000] code: kworker/u4:0/138560 caller is flow_offload_queue_work+0xec/0x1b0 [nf_flow_table] Workqueue: act_ct_workqueue tcf_ct_flow_table_cleanup_work [act_ct] Call Trace: <TASK> dump_stack_lvl+0x33/0x46 check_preemption_disabled+0xc3/0xf0 flow_offload_queue_work+0xec/0x1b0 [nf_flow_table] nf_flow_table_iterate+0x138/0x170 [nf_flow_table] nf_flow_table_free+0x140/0x1a0 [nf_flow_table] tcf_ct_flow_table_cleanup_work+0x2f/0x2b0	2024-10-21	5.5	Medium

		[act_ct] process_one_work+0x6a3/0x1030 worker_thread+0x8a/0xdf0 This patch fixes it by using NF_FLOW_TABLE_STAT_INC_ATOMIC() instead in flow_offload_queue_work(). Note that for FLOW_CLS_REPLACE branch in flow_offload_queue_work(), it may not be called in preemptible path, but it's good to use NF_FLOW_TABLE_STAT_INC_ATOMIC() for all cases in flow_offload_queue_work().			
CVE-2022-48977	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: can: af_can: fix NULL pointer dereference in can_rcv_filter Analogue to commit 8aa59e355949 ("can: af_can: fix NULL pointer dereference in can_rx_register()") we need to check for a missing initialization of ml_priv in the receive path of CAN frames. Since commit 4e096a18867a ("net: introduce CAN specific pointer in the struct net_device") the check for dev->type to be ARPHRD_CAN is not sufficient anymore since bonding or tun netdevices claim to be CAN devices but do not initialize ml_priv accordingly.	2024-10-21	5.5	Medium
CVE-2022-48978	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: HID: core: fix shift-out-of-bounds in hid_report_raw_event Syzbot reported shift-out-of-bounds in hid_report_raw_event. microsoft 0003:045E:07DA.0001: hid_field_extract() called with n (128) > 32! (swapper/0) == UBSAN: shift-out-of-bounds in drivers/hid/hid-core.c:1323:20 shift exponent 127 is too large for 32-bit type 'int'	2024-10-21	5.5	Medium

		CPU: 0 PID: 0 Comm: swapper/0 Not tainted 6.1.0-rc4-syzkaller-00159-g4bbf3422df78 #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x1e3/0x2cb lib/dump_stack.c:106 ubsan_epilogue lib/ubsan.c:151 [inline] __ubsan_handle_shift_out_of_bounds+0x3a6/0x420 lib/ubsan.c:322 snto32 drivers/hid/hid-core.c:1323 [inline] hid_input_fetch_field drivers/hid/hid-core.c:1572 [inline] hid_process_report drivers/hid/hid-core.c:1665 [inline] hid_report_raw_event+0xd56/0x18b0 drivers/hid/hid-core.c:1998 hid_input_report+0x408/0x4f0 drivers/hid/hid-core.c:2066 hid_irq_in+0x459/0x690 drivers/hid/usbhid/hid-core.c:284 __usb_hcd_giveback_urb+0x369/0x530 drivers/usb/core/hcd.c:1671 dummy_timer+0x86b/0x3110 drivers/usb/gadget/udc/dummy_hcd.c:1988 call_timer_fn+0xf5/0x210 kernel/time/timer.c:1474 expire_timers kernel/time/timer.c:1519 [inline] __run_timers+0x76a/0x980 kernel/time/timer.c:1790 run_timer_softirq+0x63/0xf0 kernel/time/timer.c:1803 __do_softirq+0x277/0x75b kernel/softirq.c:571 __irq_exit_rcu+0xec/0x170 kernel/softirq.c:650 irq_exit_rcu+0x5/0x20 kernel/softirq.c:662 sysvec_apic_timer_interrupt+0x91/0xb0 arch/x86/kernel/apic/apic.c:1107 === If the size of the integer (unsigned n) is bigger than 32 in snto32(), shift exponent will be too large for 32-bit type 'int', resulting in a shift-out-of-bounds bug. Fix this by adding a check on the size of the integer (unsigned n) in snto32(). To add support for n greater than 32 bits, set n to 32, if n is greater than 32.			

CVE-2022-48979	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix array index out of bound error in DCN32 DML [Why&How] LinkCapacitySupport array is indexed with the number of voltage states and not the number of max DPPs. Fix the error by changing the array declaration to use the correct (larger) array size of total number of voltage states.	2024-10-21	5.5	Medium
CVE-2022-48982	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix crash when replugging CSR fake controllers It seems fake CSR 5.0 clones can cause the suspend notifier to be registered twice causing the following kernel panic: [71.986122] Call Trace: [71.986124] <TASK> [71.986125] blocking_notifier_chain_register+0x33/0x60 [71.986130] hci_register_dev+0x316/0x3d0 [bluetooth 99b5497ea3d09708fa1366c1dc03288bf3cca8da] [71.986154] btusb_probe+0x979/0xd85 [btusb e1e0605a4f4c01984a4b9c8ac58c3666ae287477] [71.986159] ? __pm_runtime_set_status+0x1a9/0x300 [71.986162] ? ktime_get_mono_fast_ns+0x3e/0x90 [71.986167] usb_probe_interface+0xe3/0x2b0 [71.986171] really_probe+0xdb/0x380 [71.986174] ? pm_runtime_barrier+0x54/0x90 [71.986177] __driver_probe_device+0x78/0x170 [71.986180] driver_probe_device+0x1f/0x90 [71.986183] __device_attach_driver+0x89/0x110 [71.986186] ? driver_allows_async_probing+0x70/0x70 [71.986189] bus_for_each_drv+0x8c/0xe0 [71.986192] __device_attach+0xb2/0x1e0 [71.986195] bus_probe_device+0x92/0xb0 [71.986198] device_add+0x422/0x9a0 [71.986201] ? sysfs_merge_group+0xd4/0x110 [71.986205] usb_set_configuration+0x57a/0x820	2024-10-21	5.5	Medium

		[71.986208] usb_generic_driver_probe+0x4f/0x70 [71.986211] usb_probe_device+0x3a/0x110 [71.986213] really_probe+0xdb/0x380 [71.986216] ? pm_runtime_barrier+0x54/0x90 [71.986219] __driver_probe_device+0x78/0x170 [71.986221] driver_probe_device+0x1f/0x90 [71.986224] __device_attach_driver+0x89/0x110 [71.986227] ? driver_allows_async_probing+0x70/0x70 [71.986230] bus_for_each_drv+0x8c/0xe0 [71.986232] __device_attach+0xb2/0x1e0 [71.986235] bus_probe_device+0x92/0xb0 [71.986237] device_add+0x422/0x9a0 [71.986239] ? _dev_info+0x7d/0x98 [71.986242] ? blake2s_update+0x4c/0xc0 [71.986246] usb_new_device.cold+0x148/0x36d [71.986250] hub_event+0xa8a/0x1910 [71.986255] process_one_work+0x1c4/0x380 [71.986259] worker_thread+0x51/0x390 [71.986262] ? rescuer_thread+0x3b0/0x3b0 [71.986264] kthread+0xdb/0x110 [71.986266] ? kthread_complete_and_exit+0x20/0x20 [71.986268] ret_from_fork+0x1f/0x30 [71.986273] </TASK> [71.986274] ---[end trace 0000000000000000]--- [71.986284] btusb: probe of 2-1.6:1.0 failed with error -17			
CVE-2022-48983	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: io_uring: Fix a null-ptr-deref in io_tctx_exit_cb() Syzkaller reports a NULL deref bug as follows: BUG: KASAN: null-ptr-deref in io_tctx_exit_cb+0x53/0xd3 Read of size 4 at addr 0000000000000138 by task file1/1955 CPU: 1 PID: 1955 Comm: file1 Not tainted 6.1.0-rc7-00103-gef4d3ea40565 #75 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.0-2.el7 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0xcd/0x134 ? io_tctx_exit_cb+0x53/0xd3 kasan_report+0xbb/0x1f0 ? io_tctx_exit_cb+0x53/0xd3 kasan_check_range+0x140/0x190	2024-10-21	5.5	Medium

		io_tctx_exit_cb+0x53/0xd3 task_work_run+0x164/0x250 ? task_work_cancel+0x30/0x30 get_signal+0x1c3/0x2440 ? lock_downgrade+0x6e0/0x6e0 ? lock_downgrade+0x6e0/0x6e0 ? exit_signals+0x8b0/0x8b0 ? do_raw_read_unlock+0x3b/0x70 ? do_raw_spin_unlock+0x50/0x230 arch_do_signal_or_restart+0x82/0x2470 ? kmem_cache_free+0x260/0x4b0 ? putname+0xfe/0x140 ? get_sigframe_size+0x10/0x10 ? do_execveat_common.isra.0+0x226/0x710 ? lockdep_hardirqs_on+0x79/0x100 ? putname+0xfe/0x140 ? do_execveat_common.isra.0+0x238/0x710 exit_to_user_mode_prepare+0x15f/0x250 syscall_exit_to_user_mode+0x19/0x50 do_syscall_64+0x42/0xb0 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0023:0x0 Code: Unable to access opcode bytes at 0xffffffffffffffd6. RSP: 002b:00000000fffb7790 EFLAGS: 00000200 ORIG_RAX: 000000000000000b RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 </TASK> Kernel panic - not syncing: panic_on_warn set ... This happens because the adding of task_work from io_ring_exit_work() isn't synchronized with canceling all work items from eg exec. The execution of the two are ordered in that they are both run by the task itself, but if io_tctx_exit_cb() is queued while we're canceling all work items off exec AND gets executed when the task exits to userspace rather than in the main loop in io_uring_cancel_generic(), then we can			

		find current->io_uring == NULL and hit the above crash. It's safe to add this NULL check here, because the execution of the two paths are done by the task itself. [axboe: add code comment and also put an explanation in the commit msg]			
CVE-2022-48984	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: can: slcan: fix freed work crash The LTP test pty03 is causing a crash in slcan: BUG: kernel NULL pointer dereference, address: 0000000000000008 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 348 Comm: kworker/0:3 Not tainted 6.0.8-1-default #1 openSUSE Tumbleweed 9d20364b934f5aab0a9bdf84e8f45cfdfae39dab Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.15.0-0-g2dd4b9b-rebuilt.opensuse.org 04/01/2014 Workqueue: 0x0 (events) RIP: 0010:process_one_work (/home/rich/kernel/linux/kernel/workqueue.c:706 /home/rich/kernel/linux/kernel/workqueue.c:2185) Code: 49 89 ff 41 56 41 55 41 54 55 53 48 89 f3 48 83 ec 10 48 8b 06 48 8b 6f 48 49 89 c4 45 30 e4 a8 04 b8 00 00 00 00 4c 0f 44 e0 <49> 8b 44 24 08 44 8b a8 00 01 00 00 41 83 e5 20 f6 45 10 04 75 0e RSP: 0018:ffffaf7b40f47e98 EFLAGS: 00010046 RAX: 0000000000000000 RBX: ffff9d644e1b8b48 RCX: ffff9d649e439968 RDX: 00000000ffff8455 RSI: ffff9d644e1b8b48 RDI: ffff9d64764aa6c0 RBP: ffff9d649e4335c0 R08: 0000000000000c00 R09: ffff9d64764aa734 R10: 0000000000000007 R11: 0000000000000001 R12: 0000000000000000 R13: ffff9d649e4335e8 R14: ffff9d64490da780 R15: ffff9d64764aa6c0 FS: 0000000000000000(0000) GS:ffff9d649e400000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033	2024-10-21	5.5	Medium

		CR2: 0000000000000008 CR3: 0000000036424000 CR4: 00000000000006f0 Call Trace: <TASK> worker_thread (/home/rich/kernel/linux/kernel/workqueue.c:2436) kthread (/home/rich/kernel/linux/kernel/kthread.c:376) ret_from_fork (/home/rich/kernel/linux/arch/x86/entry/entry_64.S:312) Apparently, the slcan's tx_work is freed while being scheduled. While slcan_netdev_close() (netdev side) calls flush_work(&sl->tx_work), slcan_close() (tty side) does not. So when the netdev is never set UP, but the tty is stuffed with bytes and forced to wakeup write, the work is scheduled, but never flushed. So add an additional flush_work() to slcan_close() to be sure the work is flushed under all circumstances. The Fixes commit below moved flush_work() from slcan_close() to slcan_netdev_close(). What was the rationale behind it? Maybe we can drop the one in slcan_netdev_close()? I see the same pattern in can327. So it perhaps needs the very same fix.			
CVE-2022-48986	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mm/gup: fix gup_pud_range() for dax For dax pud, pud_huge() returns true on x86. So the function works as long as hugetlb is configured. However, dax doesn't depend on hugetlb. Commit 414fd080d125 ("mm/gup: fix gup_pmd_range() for dax") fixed devmap-backed huge PMDs, but missed devmap-backed huge PUDs. Fix this as well. This fixes the below kernel panic:	2024-10-21	5.5	Medium

		general protection fault, probably for non-canonical address 0x69e7c000cc478: 0000 [#1] SMP < snip > Call Trace: <TASK> get_user_pages_fast+0x1f/0x40 iov_iter_get_pages+0xc6/0x3b0 ? mempool_alloc+0x5d/0x170 bio_iov_iter_get_pages+0x82/0x4e0 ? bvec_alloc+0x91/0xc0 ? bio_alloc_bioset+0x19a/0x2a0 blkdev_direct_IO+0x282/0x480 ? __io_complete_rw_common+0xc0/0xc0 ? filemap_range_has_page+0x82/0xc0 generic_file_direct_write+0x9d/0x1a0 ? inode_update_time+0x24/0x30 __generic_file_write_iter+0xbd/0x1e0 blkdev_write_iter+0xb4/0x150 ? io_import_iovec+0x8d/0x340 io_write+0xf9/0x300 io_issue_sqe+0x3c3/0x1d30 ? sysvec_reschedule_ipi+0x6c/0x80 __io_queue_sqe+0x33/0x240 ? fget+0x76/0xa0 io_submit_sqes+0xe6a/0x18d0 ? __fget_light+0xd1/0x100 __x64_sys_io_uring_enter+0x199/0x880 ? __context_tracking_enter+0x1f/0x70 ? irqentry_exit_to_user_mode+0x24/0x30 ? irqentry_exit+0x1d/0x30 ? __context_tracking_exit+0xe/0x70 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x61/0xcb RIP: 0033:0x7fc97c11a7be < snip > </TASK> ---[end trace 48b2e0e67debcaeb]--- RIP: 0010:internal_get_user_pages_fast+0x340/0x990 < snip > Kernel panic - not syncing: Fatal exception Kernel Offset: disabled			
CVE-2022-48987	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-dv-timings.c: fix too strict blanking sanity checks Sanity checks were added to verify the v4l2_bt_timings blanking fields in order to avoid integer overflows when userspace	2024-10-21	5.5	Medium

		passes weird values. But that assumed that userspace would correctly fill in the front porch, backporch and sync values, but sometimes all you know is the total blanking, which is then assigned to just one of these fields. And that can fail with these checks. So instead set a maximum for the total horizontal and vertical blanking and check that each field remains below that. That is still sufficient to avoid integer overflows, but it also allows for more flexibility in how userspace fills in these fields.			
CVE-2022-48992	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: soc-pcm: Add NULL check in BE reparenting Add NULL check in dpcm_be_reparent API, to handle kernel NULL pointer dereference error. The issue occurred in fuzzing test.	2024-10-21	5.5	Medium
CVE-2022-48995	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Input: raydium_ts_i2c - fix memory leak in raydium_i2c_send() There is a kmemleak when test the raydium_i2c_ts with bpf mock device: unreferenced object 0xffff88812d3675a0 (size 8): comm "python3", pid 349, jiffies 4294741067 (age 95.695s) hex dump (first 8 bytes): 11 0e 10 c0 01 00 04 00 .... backtrace: [<0000000068427125>] __kmalloc+0x46/0x1b0 [<0000000090180f91>] raydium_i2c_send+0xd4/0x2bf [raydium_i2c_ts] [<000000006e631aee>] raydium_i2c_initialize.cold+0xbc/0x3e4 [raydium_i2c_ts] [<00000000dc6fcf38>] raydium_i2c_probe+0x3cd/0x6bc [raydium_i2c_ts]	2024-10-21	5.5	Medium

		[<00000000a310de16>] i2c_device_probe+0x651/0x680 [<00000000f5a96bf3>] really_probe+0x17c/0x3f0 [<00000000096ba499>] __driver_probe_device+0xe3/0x170 [<00000000c5acb4d9>] driver_probe_device+0x49/0x120 [<00000000264fe082>] __device_attach_driver+0xf7/0x150 [<00000000f919423c>] bus_for_each_drv+0x114/0x180 [<00000000e067feca>] __device_attach+0x1e5/0x2d0 [<0000000054301fc2>] bus_probe_device+0x126/0x140 [<00000000aad93b22>] device_add+0x810/0x1130 [<00000000c086a53f>] i2c_new_client_device+0x352/0x4e0 [<000000003c2c248c>] of_i2c_register_device+0xf1/0x110 [<00000000ffec4177>] of_i2c_notify+0x100/0x160 unreferenced object 0xffff88812d3675c8 (size 8): comm "python3", pid 349, jiffies 4294741070 (age 95.692s) hex dump (first 8 bytes): 22 00 36 2d 81 88 ff ff ".6-.... backtrace: [<0000000068427125>] __kmalloc+0x46/0x1b0 [<0000000090180f91>] raydium_i2c_send+0xd4/0x2bf [raydium_i2c_ts] [<000000001d5c9620>] raydium_i2c_initialize.cold+0x223/0x3e4 [raydium_i2c_ts] [<00000000dc6fcf38>] raydium_i2c_probe+0x3cd/0x6bc [raydium_i2c_ts] [<00000000a310de16>] i2c_device_probe+0x651/0x680 [<00000000f5a96bf3>] really_probe+0x17c/0x3f0 [<00000000096ba499>] __driver_probe_device+0xe3/0x170 [<00000000c5acb4d9>] driver_probe_device+0x49/0x120 [<00000000264fe082>] __device_attach_driver+0xf7/0x150 [<00000000f919423c>] bus_for_each_drv+0x114/0x180 [<00000000e067feca>] __device_attach+0x1e5/0x2d0 [<0000000054301fc2>]			

		bus_probe_device+0x126/0x140 [<00000000aad93b22>] device_add+0x810/0x1130 [<00000000c086a53f>] i2c_new_client_device+0x352/0x4e0 [<000000003c2c248c>] of_i2c_register_device+0xf1/0x110 [<00000000ffec4177>] of_i2c_notify+0x100/0x160 After BANK_SWITCH command from i2c BUS, no matter success or error happened, the tx_buf should be freed.			
CVE-2022-49000	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix PCI device refcount leak in has_external_pci() for_each_pci_dev() is implemented by pci_get_device(). The comment of pci_get_device() says that it will increase the reference count for the returned pci_dev and also decrease the reference count for the input pci_dev @from if it is not NULL. If we break for_each_pci_dev() loop with pdev not NULL, we need to call pci_dev_put() to decrease the reference count. Add the missing pci_dev_put() before 'return true' to avoid reference count leak.	2024-10-21	5.5	Medium
CVE-2022-49002	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix PCI device refcount leak in dmar_dev_scope_init() for_each_pci_dev() is implemented by pci_get_device(). The comment of pci_get_device() says that it will increase the reference count for the returned pci_dev and also decrease the reference count for the input pci_dev @from if it is not NULL. If we break for_each_pci_dev() loop with pdev not NULL, we need to call pci_dev_put() to decrease the reference count. Add the missing	2024-10-21	5.5	Medium

		pci_dev_put() for the error path to avoid reference count leak.			
CVE-2022-49004	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: riscv: Sync efi page table's kernel mappings before switching The EFI page table is initially created as a copy of the kernel page table. With VMAP_STACK enabled, kernel stacks are allocated in the vmalloc area: if the stack is allocated in a new PGD (one that was not present at the moment of the efi page table creation or not synced in a previous vmalloc fault), the kernel will take a trap when switching to the efi page table when the vmalloc kernel stack is accessed, resulting in a kernel panic. Fix that by updating the efi kernel mappings before switching to the efi page table.	2024-10-21	5.5	Medium
CVE-2022-49005	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Fix bounds check for _sx controls For _sx controls the semantics of the max field is not the usual one, max is the number of steps rather than the maximum value. This means that our check in snd_soc_put_volsw_sx() needs to just check against the maximum value.	2024-10-21	5.5	Medium
CVE-2022-49007	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix NULL pointer dereference in nilfs_palloc_commit_free_entry() Syzbot reported a null-ptr-deref bug: NILFS (loop0): segctord starting. Construction interval = 5 seconds, CP frequency < 30 seconds general protection fault, probably for non-canonical address 0xdffffc0000000002: 0000 [#1] PREEMPT SMP KASAN	2024-10-21	5.5	Medium

		KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] CPU: 1 PID: 3603 Comm: segctord Not tainted 6.1.0-rc2-syzkaller-00105-gb229b6ca5abb #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 10/11/2022 RIP: 0010:nilfs_palloc_commit_free_entry+0xe5/0x6b0 fs/nilfs2/alloc.c:608 Code: 00 00 00 00 fc ff df 80 3c 02 00 0f 85 cd 05 00 00 48 b8 00 00 00 00 00 fc ff df 4c 8b 73 08 49 8d 7e 10 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 26 05 00 00 49 8b 46 10 be a6 00 00 00 48 c7 c7 RSP: 0018:ffffc90003dff830 EFLAGS: 00010212 RAX: dffffc0000000000 RBX: ffff88802594e218 RCX: 000000000000000d RDX: 0000000000000002 RSI: 0000000000002000 RDI: 0000000000000010 RBP: ffff888071880222 R08: 0000000000000005 R09: 000000000000003f R10: 000000000000000d R11: 0000000000000000 R12: ffff888071880158 R13: ffff88802594e220 R14: 0000000000000000 R15: 0000000000000004 FS: 0000000000000000(0000) GS:ffff8880b9b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fb1c08316a8 CR3: 0000000018560000 CR4: 0000000000350ee0 Call Trace: <TASK> nilfs_dat_commit_free fs/nilfs2/dat.c:114 [inline] nilfs_dat_commit_end+0x464/0x5f0 fs/nilfs2/dat.c:193 nilfs_dat_commit_update+0x26/0x40 fs/nilfs2/dat.c:236 nilfs_btree_commit_update_v+0x87/0x4a0 fs/nilfs2/btree.c:1940 nilfs_btree_commit_propagate_v fs/nilfs2/btree.c:2016 [inline] nilfs_btree_propagate_v fs/nilfs2/btree.c:2046 [inline] nilfs_btree_propagate+0xa00/0xd60 fs/nilfs2/btree.c:2088 nilfs_bmap_propagate+0x73/0x170 fs/nilfs2/bmap.c:337 nilfs_collect_file_data+0x45/0xd0			

		fs/nilfs2/segment.c:568 nilfs_segctor_apply_buffers+0x14a/0x470 fs/nilfs2/segment.c:1018 nilfs_segctor_scan_file+0x3f4/0x6f0 fs/nilfs2/segment.c:1067 nilfs_segctor_collect_blocks fs/nilfs2/segment.c:1197 [inline] nilfs_segctor_collect fs/nilfs2/segment.c:1503 [inline] nilfs_segctor_do_construct+0x12fc/0x6af0 fs/nilfs2/segment.c:2045 nilfs_segctor_construct+0x8e3/0xb30 fs/nilfs2/segment.c:2379 nilfs_segctor_thread_construct fs/nilfs2/segment.c:2487 [inline] nilfs_segctor_thread+0x3c3/0xf30 fs/nilfs2/segment.c:2570 kthread+0x2e4/0x3a0 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:306 </TASK> ... If DAT metadata file is corrupted on disk, there is a case where req->pr_desc_bh is NULL and blocknr is 0 at nilfs_dat_commit_end() during a b-tree operation that cascadingly updates ancestor nodes of the b-tree, because nilfs_dat_commit_alloc() for a lower level block can initialize the blocknr on the same DAT entry between nilfs_dat_prepare_end() and nilfs_dat_commit_end(). If this happens, nilfs_dat_commit_end() calls nilfs_dat_commit_free() without valid buffer heads in req->pr_desc_bh and req->pr_bitmap_bh, and causes the NULL pointer dereference above in nilfs_palloc_commit_free_entry() function, which leads to a crash. Fix this by adding a NULL check on req->pr_desc_bh and req->pr_bitmap_bh before nilfs_palloc_commit_free_entry() in nilfs_dat_commit_free(). This also calls nilfs_error() in that case to notify that there is a fatal			

		flaw in the filesystem metadata and prevent further operations.			
CVE-2022-49008	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: can: can327: can327_feed_frame_to_netdev(): fix potential skb leak when netdev is down In can327_feed_frame_to_netdev(), it did not free the skb when netdev is down, and all callers of can327_feed_frame_to_netdev() did not free allocated skb too. That would trigger skb leak. Fix it by adding kfree_skb() in can327_feed_frame_to_netdev() when netdev is down. Not tested, just compiled.	2024-10-21	5.5	Medium
CVE-2022-49009	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hwmon: (asus-ec-sensors) Add checks for devm_kcalloc As the devm_kcalloc may return NULL, the return value needs to be checked to avoid NULL poineter dereference.	2024-10-21	5.5	Medium
CVE-2022-49010	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hwmon: (coretemp) Check for null before removing sysfs attrs If coretemp_add_core() gets an error then pdata->core_data[indx] is already NULL and has been kfreed. Don't pass that to sysfs_remove_group() as that will crash in sysfs_remove_group(). [Shortened for readability] [91854.020159] sysfs: cannot create duplicate filename '/devices/platform/coretemp.0/hwmon/hwmon2/temp20_label' <cpu offline> [91855.126115] BUG: kernel NULL pointer dereference, address: 0000000000000188 [91855.165103] #PF: supervisor read access in kernel mode [91855.194506] #PF: error_code(0x0000) - not-present page	2024-10-21	5.5	Medium

		[91855.224445] PGD 0 P4D 0 [91855.238508] Oops: 0000 [#1] PREEMPT SMP PTI ... [91855.342716] RIP: 0010:sysfs_remove_group+0xc/0x80 ... [91855.796571] Call Trace: [91855.810524] coretemp_cpu_offline+0x12b/0x1dd [coretemp] [91855.841738] ? coretemp_cpu_online+0x180/0x180 [coretemp] [91855.871107] cpuhp_invoke_callback+0x105/0x4b0 [91855.893432] cpuhp_thread_fun+0x8e/0x150 ... Fix this by checking for NULL first.			
CVE-2022-49011	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: hwmon: (coretemp) fix pci device refcount leak in nv1a_ram_new() As comment of pci_get_domain_bus_and_slot() says, it returns a pci device with refcount increment, when finish using it, the caller must decrement the reference count by calling pci_dev_put(). So call it after using to avoid refcount leak.	2024-10-21	5.5	Medium
CVE-2022-49012	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: afs: Fix server->active leak in afs_put_server The atomic_read was accidentally replaced with atomic_inc_return, which prevents the server from getting cleaned up and causes rmmod to hang with a warning: Can't purge s=00000001	2024-10-21	5.5	Medium
CVE-2022-49013	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: sctp: fix memory leak in sctp_stream_outq_migrate() When sctp_stream_outq_migrate() is called to release stream out resources, the memory pointed to by prio_head in stream out is	2024-10-21	5.5	Medium

		not released. The memory leak information is as follows: unreferenced object 0xffff88801fe79f80 (size 64): comm "sctp_repo", pid 7957, jiffies 4294951704 (age 36.480s) hex dump (first 32 bytes): 80 9f e7 1f 80 88 ff ff 80 9f e7 1f 80 88 ff ff .... 90 9f e7 1f 80 88 ff ff 90 9f e7 1f 80 88 ff ff .... backtrace: [<ffffffff81b215c6>] kmalloc_trace+0x26/0x60 [<ffffffff88ae517c>] sctp_sched_prio_set+0x4cc/0x770 [<ffffffff88ad64f2>] sctp_stream_init_ext+0xd2/0x1b0 [<ffffffff88aa2604>] sctp_sendmsg_to_asoc+0x1614/0x1a30 [<ffffffff88ab7ff1>] sctp_sendmsg+0xda1/0x1ef0 [<ffffffff87f765ed>] inet_sendmsg+0x9d/0xe0 [<ffffffff8754b5b3>] sock_sendmsg+0xd3/0x120 [<ffffffff8755446a>] __sys_sendto+0x23a/0x340 [<ffffffff87554651>] __x64_sys_sendto+0xe1/0x1b0 [<ffffffff89978b49>] do_syscall_64+0x39/0xb0 [<ffffffff89a0008b>] entry_SYSCALL_64_after_hwframe+0x63/0xcd			
[CVE-2022-49016]()	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: mdiobus: fix unbalanced node reference count I got the following report while doing device(mscc-miim) load test with CONFIG_OF_UNITTEST and CONFIG_OF_DYNAMIC enabled: OF: ERROR: memory leak, expected refcount 1 instead of 2, of_node_get()/of_node_put() unbalanced - destroy cset entry: attach overlay node /spi/soc@0/mdio@7107009c/ethernet-phy@0 If the 'fwnode' is not an acpi node, the refcount is get in fwnode_mdiobus_phy_device_register(), but it has never been put when the device is freed in the normal path. So call	2024-10-21	5.5	Medium

		fwnode_handle_put() in phy_device_release() to avoid leak. If it's an acpi node, it has never been get, but it's put in the error path, so call fwnode_handle_get() before phy_device_register() to keep get/put operation balanced.			
CVE-2022-49018	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix sleep in atomic at close time Matt reported a splat at msk close time: BUG: sleeping function called from invalid context at net/mptcp/protocol.c:2877 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 155, name: packetdrill preempt_count: 201, expected: 0 RCU nest depth: 0, expected: 0 4 locks held by packetdrill/155: #0: ffff888001536990 (&sb->s_type->i_mutex_key#6){+.+.}-{3:3}, at: __sock_release (net/socket.c:650) #1: ffff88800b498130 (sk_lock-AF_INET){+.+.}-{0:0}, at: mptcp_close (net/mptcp/protocol.c:2973) #2: ffff88800b49a130 (sk_lock-AF_INET/1){+.+.}-{0:0}, at: __mptcp_close_ssk (net/mptcp/protocol.c:2363) #3: ffff88800b49a0b0 (slock-AF_INET){+...}-{2:2}, at: __lock_sock_fast (include/net/sock.h:1820) Preemption disabled at: 0x0 CPU: 1 PID: 155 Comm: packetdrill Not tainted 6.1.0-rc5 #365 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:107 (discriminator 4)) __might_resched.cold (kernel/sched/core.c:9891) __mptcp_destroy_sock (include/linux/kernel.h:110) __mptcp_close (net/mptcp/protocol.c:2959) mptcp_subflow_queue_clean (include/net/sock.h:1777) __mptcp_close_ssk (net/mptcp/protocol.c:2363) mptcp_destroy_common (net/mptcp/protocol.c:3170) mptcp_destroy (include/net/sock.h:1495)	2024-10-21	5.5	Medium

		__mptcp_destroy_sock (net/mptcp/protocol.c:2886) __mptcp_close (net/mptcp/protocol.c:2959) mptcp_close (net/mptcp/protocol.c:2974) inet_release (net/ipv4/af_inet.c:432) __sock_release (net/socket.c:651) sock_close (net/socket.c:1367) __fput (fs/file_table.c:320) task_work_run (kernel/task_work.c:181 (discriminator 1)) exit_to_user_mode_prepare (include/linux/resume_user_mode.h:49) syscall_exit_to_user_mode (kernel/entry/common.c:130) do_syscall_64 (arch/x86/entry/common.c:87) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120) We can't call mptcp_close under the 'fast' socket lock variant, replace it with a sock_lock_nested() as the relevant code is already under the listening msk socket lock protection.			
CVE-2022-49019	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: nixge: fix NULL dereference In function nixge_hw_dma_bd_release() dereference of NULL pointer priv->rx_bd_v is possible for the case of its allocation failure in nixge_hw_dma_bd_init(). Move for() loop with priv->rx_bd_v dereference under the check for its validity. Found by Linux Verification Center (linuxtesting.org) with SVACE.	2024-10-21	5.5	Medium
CVE-2022-49020	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/9p: Fix a potential socket leak in p9_socket_open Both p9_fd_create_tcp() and p9_fd_create_unix() will call p9_socket_open(). If the creation of p9_trans_fd fails, p9_fd_create_tcp() and p9_fd_create_unix() will return an error directly instead of releasing the cscoket, which	2024-10-21	5.5	Medium

		will result in a socket leak. This patch adds sock_release() to fix the leak issue.			

CVE-2022-49021	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		net: phy: fix null-ptr-deref while probe() failed
		
		I got a null-ptr-deref report as following when doing fault injection test:
		
		BUG: kernel NULL pointer dereference, address: 0000000000000058
		Oops: 0000 [#1] PREEMPT SMP KASAN PTI
		CPU: 1 PID: 253 Comm: 507-spi-dm9051 Tainted: G B N 6.1.0-rc3+
		Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014
		RIP: 0010:klist_put+0x2d/0xd0
		Call Trace:
		 <TASK>
		 klist_remove+0xf1/0x1c0
		 device_release_driver_internal+0x23e/0x2d0
		 bus_remove_device+0x1bd/0x240
		 device_del+0x357/0x770
		 phy_device_remove+0x11/0x30
		 mdiobus_unregister+0xa5/0x140
		 release_nodes+0x6a/0xa0
		 devres_release_all+0xf8/0x150
		 device_unbind_cleanup+0x19/0xd0
		
		//probe path:
		phy_device_register()
		  device_add()
		
		phy_connect
		  phy_attach_direct() //set device driver
		    probe() //it's failed, driver is not bound
		    device_bind_driver() // probe failed, it's not called
		
		//remove path:
		phy_device_remove()
		  device_del()
		    device_release_driver_internal()
		      __device_release_driver() //dev->drv is not NULL
		        klist_remove() <- knode_driver is not added yet, cause null-ptr-deref
		
		In phy_attach_direct(), after setting the 'dev->driver', probe() fails,
			2024-10-21	5.5	Medium

		device_bind_driver() is not called, so the knode_driver->n_klist is not set, then it causes null-ptr-deref in __device_release_driver() while deleting device. Fix this by setting dev->driver to NULL in the error path in phy_attach_direct().			
CVE-2022-49024	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: can: m_can: pci: add missing m_can_class_free_dev() in probe/remove methods In m_can_pci_remove() and error handling path of m_can_pci_probe(), m_can_class_free_dev() should be called to free resource allocated by m_can_class_allocate_dev(), otherwise there will be memleak.	2024-10-21	5.5	Medium
CVE-2022-49027	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix error handling in iavf_init_module() The iavf_init_module() won't destroy workqueue when pci_register_driver() failed. Call destroy_workqueue() when pci_register_driver() failed to prevent the resource leak. Similar to the handling of u132_hcd_init in commit f276e002793c ("usb: u132-hcd: fix resource leak")	2024-10-21	5.5	Medium
CVE-2022-49028	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ixgbevf: Fix resource leak in ixgbevf_init_module() ixgbevf_init_module() won't destroy the workqueue created by create_singlethread_workqueue() when pci_register_driver() failed. Add destroy_workqueue() in fail path to prevent the resource leak. Similar to the handling of u132_hcd_init in commit f276e002793c ("usb: u132-hcd: fix resource leak")	2024-10-21	5.5	Medium
CVE-2022-49033	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:	2024-10-21	5.5	Medium

		btrfs: qgroup: fix sleep from invalid context bug in btrfs_qgroup_inherit() Syzkaller reported BUG as follows: BUG: sleeping function called from invalid context at include/linux/sched/mm.h:274 Call Trace: <TASK> dump_stack_lvl+0xcd/0x134 __might_resched.cold+0x222/0x26b kmem_cache_alloc+0x2e7/0x3c0 update_qgroup_limit_item+0xe1/0x390 btrfs_qgroup_inherit+0x147b/0x1ee0 create_subvol+0x4eb/0x1710 btrfs_mksubvol+0xfe5/0x13f0 __btrfs_ioctl_snap_create+0x2b0/0x430 btrfs_ioctl_snap_create_v2+0x25a/0x520 btrfs_ioctl+0x2a1c/0x5ce0 __x64_sys_ioctl+0x193/0x200 do_syscall_64+0x35/0x80 Fix this by calling qgroup_dirty() on @dstqgroup, and update limit item in btrfs_run_qgroups() later outside of the spinlock context.			
CVE-2024-50019	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: kthread: unpark only parked kthread Calling into kthread unparking unconditionally is mostly harmless when the kthread is already unparked. The wake up is then simply ignored because the target is not in TASK_PARKED state. However if the kthread is per CPU, the wake up is preceded by a call to kthread_bind() which expects the task to be inactive and in TASK_PARKED state, which obviously isn't the case if it is unparked. As a result, calling kthread_stop() on an unparked per-cpu kthread triggers such a warning: WARNING: CPU: 0 PID: 11 at kernel/kthread.c:525 __kthread_bind_mask kernel/kthread.c:525 <TASK>	2024-10-21	5.5	Medium

		kthread_stop+0x17a/0x630 kernel/kthread.c:707 destroy_workqueue+0x136/0xc40 kernel/workqueue.c:5810 wg_destruct+0x1e2/0x2e0 drivers/net/wireguard/device.c:257 netdev_run_todo+0xe1a/0x1000 net/core/dev.c:10693 default_device_exit_batch+0xa14/0xa90 net/core/dev.c:11769 ops_exit_list net/core/net_namespace.c:178 [inline] cleanup_net+0x89d/0xcc0 net/core/net_namespace.c:640 process_one_work kernel/workqueue.c:3231 [inline] process_scheduled_works+0xa2c/0x1830 kernel/workqueue.c:3312 worker_thread+0x86d/0xd70 kernel/workqueue.c:3393 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Fix this with skipping unecessary unparking while stopping a kthread.			
CVE-2024-50020	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper handling of refcount in ice_sriov_set_msix_vec_count() This patch addresses an issue with improper reference count handling in the ice_sriov_set_msix_vec_count() function. First, the function calls ice_get_vf_by_id(), which increments the reference count of the vf pointer. If the subsequent call to ice_get_vf_vsi() fails, the function currently returns an error without decrementing the reference count of the vf pointer, leading to a reference count leak. The correct behavior, as implemented in this patch, is to decrement the reference count using ice_put_vf(vf) before returning an error when vsi is NULL. Second, the function calls ice_sriov_get_irqs(), which	2024-10-21	5.5	Medium

		sets vf->first_vector_idx. If this call returns a negative value, indicating an error, the function returns an error without decrementing the reference count of the vf pointer, resulting in another reference count leak. The patch addresses this by adding a call to ice_put_vf(vf) before returning an error when vf->first_vector_idx < 0. This bug was identified by an experimental static analysis tool developed by our team. The tool specializes in analyzing reference count operations and identifying potential mismanagement of reference counts. In this case, the tool flagged the missing decrement operation as a potential issue, leading to this patch.			
CVE-2024-50021	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper handling of refcount in ice_dpll_init_rclk_pins() This patch addresses a reference count handling issue in the ice_dpll_init_rclk_pins() function. The function calls ice_dpll_get_pins(), which increments the reference count of the relevant resources. However, if the condition WARN_ON((!vsi || !vsi->netdev)) is met, the function currently returns an error without properly releasing the resources acquired by ice_dpll_get_pins(), leading to a reference count leak. To resolve this, the check has been moved to the top of the function. This ensures that the function verifies the state before any resources are acquired, avoiding the need for additional resource management in the error path. This bug was identified by an experimental static analysis tool developed by our team. The tool specializes in analyzing reference count operations	2024-10-21	5.5	Medium

		and detecting potential issues where resources are not properly managed. In this case, the tool flagged the missing release operation as a potential problem, which led to the development of this patch.			
CVE-2024-50022	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: device-dax: correct pgoff align in dax_set_mapping() pgoff should be aligned using ALIGN_DOWN() instead of ALIGN(). Otherwise, vmf->address not aligned to fault_size will be aligned to the next alignment, that can result in memory failure getting the wrong address. It's a subtle situation that only can be observed in page_mapped_in_vma() after the page is page fault handled by dev_dax_huge_fault. Generally, there is little chance to perform page_mapped_in_vma in dev-dax's page unless in specific error injection to the dax device to trigger an MCE - memory-failure. In that case, page_mapped_in_vma() will be triggered to determine which task is accessing the failure address and kill that task in the end. We used self-developed dax device (which is 2M aligned mapping) , to perform error injection to random address. It turned out that error injected to non-2M-aligned address was causing endless MCE until panic. Because page_mapped_in_vma() kept resulting wrong address and the task accessing the failure address was never killed properly: [3783.719419] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3784.049006] mce: Uncorrected hardware memory error in user-access at 200c9742380	2024-10-21	5.5	Medium

		[3784.049190] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3784.448042] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3784.448186] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3784.792026] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3784.792179] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3785.162502] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3785.162633] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3785.461116] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3785.461247] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3785.764730] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3785.764859] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3786.042128] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3786.042259] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3786.464293] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3786.464423] Memory failure: 0x200c9742: recovery action for dax page: Recovered [3786.818090] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3786.818217] Memory failure: 0x200c9742: recovery action for dax page: Recovered			

		[3787.085297] mce: Uncorrected hardware memory error in user-access at 200c9742380 [3787.085424] Memory failure: 0x200c9742: recovery action for dax page: Recovered It took us several weeks to pinpoint this problem, but we eventually used bpftrace to trace the page fault and mce address and successfully identified the issue. Joao added: ; Likely we never reproduce in production because we always pin : device-dax regions in the region align they provide (Qemu does : similarly with prealloc in hugetlb/file backed memory). I think this : bug requires that we touch *unpinned* device-dax regions unaligned to : the device-dax selected alignment (page size i.e. 4K/2M/1G)			
CVE-2024-50023	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: phy: Remove LED entry from LEDs list on unregister Commit c938ab4da0eb ("net: phy: Manual remove LEDs to ensure correct ordering") correctly fixed a problem with using devm_ but missed removing the LED entry from the LEDs list. This cause kernel panic on specific scenario where the port for the PHY is torn down and up and the kmod for the PHY is removed. On setting the port down the first time, the assosiacted LEDs are correctly unregistered. The associated kmod for the PHY is now removed. The kmod is now added again and the port is now put up, the associated LED are registered again. On putting the port down again for the second time	2024-10-21	5.5	Medium

		after these step, the LED list now have 4 elements. With the first 2 already unregistered previously and the 2 new one registered again. This cause a kernel panic as the first 2 element should have been removed. Fix this by correctly removing the element when LED is unregistered.			
CVE-2024-50024	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net: Fix an unsafe loop on the list The kernel may crash when deleting a genetlink family if there are still listeners for that family: Oops: Kernel access of bad area, sig: 11 [#1] ... NIP [c000000000c080bc] netlink_update_socket_mc+0x3c/0xc0 LR [c000000000c0f764] __netlink_clear_multicast_users+0x74/0xc0 Call Trace: __netlink_clear_multicast_users+0x74/0xc0 genl_unregister_family+0xd4/0x2d0 Change the unsafe loop on the list to a safe one, because inside the loop there is an element removal from this list.	2024-10-21	5.5	Medium
CVE-2024-50025	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: fnic: Move flush_work initialization out of if block After commit 379a58caa199 ("scsi: fnic: Move fnic_fnic_flush_tx() to a work queue"), it can happen that a work item is sent to an uninitialized work queue. This may has the effect that the item being queued is never actually queued, and any further actions depending on it will not proceed. The following warning is observed while the fnic driver is loaded:	2024-10-21	5.5	Medium

		kernel: WARNING: CPU: 11 PID: 0 at ../kernel/workqueue.c:1524 __queue_work+0x373/0x410 kernel: <IRQ> kernel: queue_work_on+0x3a/0x50 kernel: fnic_wq_copy_cmpl_handler+0x54a/0x730 [fnic 62fbff0c42e7fb825c60a55cde2fb91facb2ed24] kernel: fnic_isr_msix_wq_copy+0x2d/0x60 [fnic 62fbff0c42e7fb825c60a55cde2fb91facb2ed24] kernel: __handle_irq_event_percpu+0x36/0x1a0 kernel: handle_irq_event_percpu+0x30/0x70 kernel: handle_irq_event+0x34/0x60 kernel: handle_edge_irq+0x7e/0x1a0 kernel: __common_interrupt+0x3b/0xb0 kernel: common_interrupt+0x58/0xa0 kernel: </IRQ> It has been observed that this may break the rediscovery of Fibre Channel devices after a temporary fabric failure. This patch fixes it by moving the work queue initialization out of an if block in fnic_probe().			
[CVE-2024-50026](#)	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: scsi: wd33c93: Don't use stale scsi_pointer value A regression was introduced with commit dbb2da557a6a ("scsi: wd33c93: Move the SCSI pointer to private command data") which results in an oops in wd33c93_intr(). That commit added the scsi_pointer variable and initialized it from hostdata->connected. However, during selection, hostdata->connected is not yet valid. Fix this by getting the current scsi_pointer from hostdata->selecting.	2024-10-21	5.5	Medium
[CVE-2024-50027](#)	linux - multiple products	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	2024-10-21	5.5	Medium

		to the point after the removal completion has been completed at which it cannot be accessed any more.			
CVE-2024-50028	linux - multiple products	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.	2024-10-21	5.5	Medium
CVE-2024-50031	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Stop the active perfmon before being destroyed When running `kmscube` with one or more performance monitors enabled via `GALLIUM_HUD`, the following kernel panic can occur: [55.008324] Unable to handle kernel paging request at virtual address 00000000052004a4 [55.008368] Mem abort info: [55.008377] ESR = 0x0000000096000005 [55.008387] EC = 0x25: DABT (current EL), IL = 32 bits [55.008402] SET = 0, FnV = 0 [55.008412] EA = 0, S1PTW = 0 [55.008421] FSC = 0x05: level 1 translation fault [55.008434] Data abort info: [55.008442] ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000 [55.008455] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [55.008467] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [55.008481] user pgtable: 4k pages, 39-bit VAs, pgdp=00000001046c6000	2024-10-21	5.5	Medium

		[55.008497] [00000000052004a4] pgd=0000000000000000, p4d=0000000000000000, pud=0000000000000000 [55.008525] Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP [55.008542] Modules linked in: rfcomm [...] vc4 v3d snd_soc_hdmi_codec drm_display_helper gpu_sched drm_shmem_helper cec drm_dma_helper drm_kms_helper i2c_brcmstb drm drm_panel_orientation_quirks snd_soc_core snd_compress snd_pcm_dmaengine snd_pcm snd_timer snd backlight [55.008799] CPU: 2 PID: 166 Comm: v3d_bin Tainted: G C 6.6.47+rpt-rpi-v8 #1 Debian 1:6.6.47-1+rpt1 [55.008824] Hardware name: Raspberry Pi 4 Model B Rev 1.5 (DT) [55.008838] pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [55.008855] pc : __mutex_lock.constprop.0+0x90/0x608 [55.008879] lr : __mutex_lock.constprop.0+0x58/0x608 [55.008895] sp : ffffffc080673cf0 [55.008904] x29: ffffffc080673cf0 x28: 0000000000000000 x27: ffffff8106188a28 [55.008926] x26: ffffff8101e78040 x25: ffffff8101baa6c0 x24: ffffffd9d989f148 [55.008947] x23: ffffffda1c2a4008 x22: 0000000000000002 x21: ffffffc080673d38 [55.008968] x20: ffffff8101238000 x19: ffffff8104f83188 x18: 0000000000000000 [55.008988] x17: 0000000000000000 x16: ffffffda1bd04d18 x15: 00000055bb08bc90 [55.009715] x14: 0000000000000000 x13: 0000000000000000 x12: ffffffda1bd4cbb0 [55.010433] x11: 00000000fa83b2da x10: 0000000000001a40 x9 : ffffffda1bd04d04 [55.011162] x8 : ffffff8102097b80 x7 : 0000000000000000 x6 : 00000000030a5857 [55.011880] x5 : 00ffffffffffffff x4 : 0300000005200470 x3 : 0300000005200470 [55.012598] x2 : ffffff8101238000 x1 : 0000000000000021 x0 : 0300000005200470 [55.013292] Call trace: [55.013959] __mutex_lock.constprop.0+0x90/0x608 [55.014646] __mutex_lock_slowpath+0x1c/0x30 [55.015317] mutex_lock+0x50/0x68 [55.015961] v3d_perfmon_stop+0x40/0xe0 [v3d] [55.016627] v3d_bin_job_run+0x10c/0x2d8 [v3d]			

		[55.017282] drm_sched_main+0x178/0x3f8 [gpu_sched] [55.017921] kthread+0x11c/0x128 [55.018554] ret_from_fork+0x10/0x20 [55.019168] Code: f9400260 f1001c1f 54001ea9 927df000 (b9403401) [55.019776] ---[end trace 0000000000000000]--- [55.020411] note: v3d_bin[166] exited with preempt_count 1 This issue arises because, upon closing the file descriptor (which happens when we interrupt `kmscube`), the active performance monitor is not stopped. Although all perfmons are destroyed in `v3d_perfmon_close_file()`, the active performance monitor's pointer (`v3d->active_perfmon`) is still retained. If `kmscube` is run again, the driver will attempt to stop the active performance monitor using the stale pointer in `v3d->active_perfmon`. However, this pointer is no longer valid because the previous process has already terminated, and all performance monitors associated with it have been destroyed and freed. To fix this, when the active performance monitor belongs to a given process, explicitly stop it before destroying and freeing it.			
CVE-2024-50032	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: rcu/nocb: Fix rcuog wake-up from offline softirq After a CPU has set itself offline and before it eventually calls rcutree_report_cpu_dead(), there are still opportunities for callbacks to be enqueued, for example from a softirq. When that happens on NOCB, the rcuog wake-up is deferred through an IPI to an online CPU in order not to call into the scheduler and risk arming the RT-bandwidth after hrtimers have been migrated out and disabled.	2024-10-21	5.5	Medium

		But performing a synchronized IPI from a softirq is buggy as reported in the following scenario: WARNING: CPU: 1 PID: 26 at kernel/smp.c:633 smp_call_function_single Modules linked in: rcutorture torture CPU: 1 UID: 0 PID: 26 Comm: migration/1 Not tainted 6.11.0-rc1-00012-g9139f93209d1 #1 Stopper: multi_cpu_stop+0x0/0x320 <- __stop_cpus+0xd0/0x120 RIP: 0010:smp_call_function_single <IRQ> swake_up_one_online __call_rcu_nocb_wake __call_rcu_common ? rcu_torture_one_read call_timer_fn __run_timers run_timer_softirq handle_softirqs irq_exit_rcu ? tick_handle_periodic sysvec_apic_timer_interrupt </IRQ> Fix this with forcing deferred rcuog wake up through the NOCB timer when the CPU is offline. The actual wake up will happen from rcutree_report_cpu_dead().			
CVE-2024-50034	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix lacks of icsk_syn_mss with IPPROTO_SMC Eric report a panic on IPPROTO_SMC, and give the facts that when INET_PROTOSW_ICSK was set, icsk->icsk_sync_mss must be set too. Bug: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000086000005 EC = 0x21: IABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault	2024-10-21	5.5	Medium

		user pgtable: 4k pages, 48-bit VAs, pgdp=00000001195d1000 [0000000000000000] pgd=0800000109c46003, p4d=0800000109c46003, pud=0000000000000000 Internal error: Oops: 0000000086000005 [#1] PREEMPT SMP Modules linked in: CPU: 1 UID: 0 PID: 8037 Comm: syz.3.265 Not tainted 6.11.0-rc7-syzkaller-g5f5673607153 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT - SSBS BTYPE=--) pc : 0x0 lr : cipso_v4_sock_setattr+0x2a8/0x3c0 net/ipv4/cipso_ipv4.c:1910 sp : ffff80009b887a90 x29: ffff80009b887aa0 x28: ffff80008db94050 x27: 0000000000000000 x26: 1fffe0001aa6f5b3 x25: dfff800000000000 x24: ffff0000db75da00 x23: 0000000000000000 x22: ffff0000d8b78518 x21: 0000000000000000 x20: ffff0000d537ad80 x19: ffff0000d8b78000 x18: 1fffe000366d79ee x17: ffff8000800614a8 x16: ffff800080569b84 x15: 0000000000000001 x14: 000000008b336894 x13: 00000000cd96feaa x12: 0000000000000003 x11: 0000000000040000 x10: 00000000000020a3 x9 : 1fffe0001b16f0f1 x8 : 0000000000000000 x7 : 0000000000000000 x6 : 000000000000003f x5 : 0000000000000040 x4 : 0000000000000001 x3 : 0000000000000000 x2 : 0000000000000002 x1 : 0000000000000000 x0 : ffff0000d8b78000 Call trace: 0x0 netlbl_sock_setattr+0x2e4/0x338 net/netlabel/netlabel_kapi.c:1000 smack_netlbl_add+0xa4/0x154 security/smack/smack_lsm.c:2593 smack_socket_post_create+0xa8/0x14c security/smack/smack_lsm.c:2973 security_socket_post_create+0x94/0xd4 security/security.c:4425 __sock_create+0x4c8/0x884 net/socket.c:1587 sock_create net/socket.c:1622 [inline]			

		__sys_socket_create net/socket.c:1659 [inline] __sys_socket+0x134/0x340 net/socket.c:1706 __do_sys_socket net/socket.c:1720 [inline] __se_sys_socket net/socket.c:1718 [inline] __arm64_sys_socket+0x7c/0x94 net/socket.c:1718 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 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: ???????? ???????? ???????? ???????? (????????) ---[end trace 0000000000000000]--- This patch add a toy implementation that performs a simple return to prevent such panic. This is because MSS can be set in sock_create_kern or smc_setsockopt, similar to how it's done in AF_SMC. However, for AF_SMC, there is currently no way to synchronize MSS within __sys_connect_file. This toy implementation lays the groundwork for us to support such feature for IPPROTO_SMC in the future.			
CVE-2024-50037	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/fbdev-dma: Only cleanup deferred I/O if necessary Commit 5a498d4d06d6 ("drm/fbdev-dma: Only install deferred I/O if necessary") initializes deferred I/O only if it is used. drm_fbdev_dma_fb_destroy() however calls fb_deferred_io_cleanup() unconditionally with struct fb_info.fbdefio == NULL. KASAN with the out-of-tree Apple silicon display driver posts following warning from __flush_work() of a random struct work_struct	2024-10-21	5.5	Medium

		instead of the expected NULL pointer derefs. [22.053799] ------------[cut here]------------ [22.054832] WARNING: CPU: 2 PID: 1 at kernel/workqueue.c:4177 __flush_work+0x4d8/0x580 [22.056597] Modules linked in: uhid bnep uinput nls_ascii ip6_tables ip_tables i2c_dev loop fuse dm_multipath nfnetlink zram hid_magicmouse btrfs xor xor_neon brcmfmac_wcc raid6_pq hci_bcm4377 bluetooth brcmfmac hid_apple brcmutil nvmem_spmi_mfd simple_mfd_spmi dockchannel_hid cfg80211 joydev regmap_spmi nvme_apple ecdh_generic ecc macsmc_hid rfkill dwc3 appledrm snd_soc_macaudio macsmc_power nvme_core apple_isp phy_apple_atc apple_sart apple_rtkit_helper apple_dockchannel tps6598x macsmc_hwmon snd_soc_cs42l84 videobuf2_v4l2 spmi_apple_controller nvmem_apple_efuses videobuf2_dma_sg apple_z2 videobuf2_memops spi_nor panel_summit videobuf2_common asahi videodev pwm_apple apple_dcp snd_soc_apple_mca apple_admac spi_apple clk_apple_nco i2c_pasemi_platform snd_pcm_dmaengine mc i2c_pasemi_core mux_core ofpart adpdrm drm_dma_helper apple_dart apple_soc_cpufreq leds_pwm phram [22.073768] CPU: 2 UID: 0 PID: 1 Comm: systemd-shutdow Not tainted 6.11.2-asahi+ #asahi-dev [22.075612] Hardware name: Apple MacBook Pro (13-inch, M2, 2022) (DT) [22.077032] pstate: 01400005 (nzcv daif +PAN -UAO -TCO +DIT -SSBS BTYPE=--) [22.078567] pc : __flush_work+0x4d8/0x580 [22.079471] lr : __flush_work+0x54/0x580 [22.080345] sp : ffffc000836ef820 [22.081089] x29: ffffc000836ef880 x28: 0000000000000000 x27: ffff80002ddb7128 [22.082678] x26: dfffc00000000000 x25: 1ffff000096f0c57 x24: ffffc00082d3e358 [22.084263] x23: ffff80004b7862b8 x22: dfffc00000000000 x21: ffff80005aa1d470 [22.085855] x20: ffff80004b786000 x19: ffff80004b7862a0 x18: 0000000000000000 [22.087439] x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000005 [22.089030] x14: 1ffff800106ddf0a x13: 0000000000000000 x12: 0000000000000000 [22.090618] x11: ffffb800106ddf0f x10: dfffc00000000000 x9 : 1ffff800106ddf0e				

		[22.092206] x8 : 0000000000000000 x7 : aaaaaaaaaaaaaaaa x6 : 0000000000000001 [22.093790] x5 : ffffc000836ef728 x4 : 0000000000000000 x3 : 0000000000000020 [22.095368] x2 : 0000000000000008 x1 : 00000000000000aa x0 : 0000000000000000 [22.096955] Call trace: [22.097505] __flush_work+0x4d8/0x580 [22.098330] flush_delayed_work+0x80/0xb8 [22.099231] fb_deferred_io_cleanup+0x3c/0x130 [22.100217] drm_fbdev_dma_fb_destroy+0x6c/0xe0 [drm_dma_helper] [22.101559] unregister_framebuffer+0x210/0x2f0 [22.102575] drm_fb_helper_unregister_info+0x48/0x60 [22.103683] drm_fbdev_dma_client_unregister+0x4c/0x80 [drm_dma_helper] [22.105147] drm_client_dev_unregister+0x1cc/0x230 [22.106217] drm_dev_unregister+0x58/0x570 [22.107125] apple_drm_unbind+0x50/0x98 [appledrm] [22.108199] component_del+0x1f8/0x3a8 [22.109042] dcp_platform_shutdown+0x24/0x38 [apple_dcp] [22.110357] platform_shutdown+0x70/0x90 [22.111219] device_shutdown+0x368/0x4d8 [22.112095] kernel_restart+0x6c/0x1d0 [22.112946] __arm64_sys_reboot+0x1c8/0x328 [22.113868] invoke_syscall+0x78/0x1a8 [22.114703] do_el0_svc+0x124/0x1a0 [22.115498] el0_svc+0x3c/0xe0 [22.116181] el0t_64_sync_handler+0x70/0xc0 [22.117110] el0t_64_sync+0x190/0x198 [22.117931] ---[end trace 0000000000000000]---			
CVE-2024-50038	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: xtables: avoid NFPROTO_UNSPEC where needed syzbot managed to call xt_cluster match via ebtables: WARNING: CPU: 0 PID: 11 at net/netfilter/xt_cluster.c:72 xt_cluster_mt+0x196/0x780 [..] ebt_do_table+0x174b/0x2a40	2024-10-21	5.5	Medium

		Module registers to NFPROTO_UNSPEC, but it assumes ipv4/ipv6 packet processing. As this is only useful to restrict locally terminating TCP/UDP traffic, register this for ipv4 and ipv6 family only. Pablo points out that this is a general issue, direct users of the set/getsockopt interface can call into targets/matches that were only intended for use with ip(6)tables. Check all UNSPEC matches and targets for similar issues: - matches and targets are fine except if they assume skb_network_header() is valid -- this is only true when called from inet layer: ip(6) stack pulls the ip/ipv6 header into linear data area. - targets that return XT_CONTINUE or other xtables verdicts must be restricted too, they are incompatbile with the ebtables traverser, e.g. EBT_CONTINUE is a completely different value than XT_CONTINUE. Most matches/targets are changed to register for NFPROTO_IPV4/IPV6, as they are provided for use by ip(6)tables. The MARK target is also used by arptables, so register for NFPROTO_ARP too. While at it, bail out if connbytes fails to enable the corresponding conntrack family. This change passes the selftests in iptables.git.			
CVE-2024-50039	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: net/sched: accept TCA_STAB only for root qdisc Most qdiscs maintain their backlog using qdisc_pkt_len(skb) on the assumption it is invariant between the enqueue() and dequeue() handlers.	2024-10-21	5.5	Medium

		Unfortunately syzbot can crash a host rather easily using a TBF + SFQ combination, with an STAB on SFQ [1] We can't support TCA_STAB on arbitrary level, this would require to maintain per-qdisc storage. [1] [88.796496] BUG: kernel NULL pointer dereference, address: 0000000000000000 [88.798611] #PF: supervisor read access in kernel mode [88.799014] #PF: error_code(0x0000) - not-present page [88.799506] PGD 0 P4D 0 [88.799829] Oops: Oops: 0000 [#1] SMP NOPTI [88.800569] CPU: 14 UID: 0 PID: 2053 Comm: b371744477 Not tainted 6.12.0-rc1-virtme #1117 [88.801107] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [88.801779] RIP: 0010:sfq_dequeue (net/sched/sch_sfq.c:272 net/sched/sch_sfq.c:499) sch_sfq [88.802544] Code: 0f b7 50 12 48 8d 04 d5 00 00 00 00 48 89 d6 48 29 d0 48 8b 91 c0 01 00 00 48 c1 e0 03 48 01 c2 66 83 7a 1a 00 7e c0 48 8b 3a <4c> 8b 07 4c 89 02 49 89 50 08 48 c7 47 08 00 00 00 00 48 c7 07 00 All code ======== 0: 0f b7 50 12 movzwl 0x12(%rax),%edx 4: 48 8d 04 d5 00 00 00 lea 0x0(,%rdx,8),%rax b: 00 c: 48 89 d6 mov %rdx,%rsi f: 48 29 d0 sub %rdx,%rax 12: 48 8b 91 c0 01 00 00 mov 0x1c0(%rcx),%rdx 19: 48 c1 e0 03 shl $0x3,%rax 1d: 48 01 c2 add %rax,%rdx 20: 66 83 7a 1a 00 cmpw $0x0,0x1a(%rdx) 25: 7e c0 jle 0xffffffffffffffe7 27: 48 8b 3a mov (%rdx),%rdi 2a:* 4c 8b 07 mov (%rdi),%r8 <-- trapping instruction 2d: 4c 89 02 mov %r8,(%rdx) 30: 49 89 50 08 mov %rdx,0x8(%r8) 34: 48 c7 47 08 00 00 00 movq $0x0,0x8(%rdi) 3b: 00 3c: 48 rex.W 3d: c7 .byte 0xc7			

```
  3e: 07                    (bad)
...

Code starting with the faulting instruction
===========================================
   0: 4c 8b 07              mov    (%rdi),%r8
   3: 4c 89 02              mov    %r8,(%rdx)
   6: 49 89 50 08           mov    %rdx,0x8(%r8)
   a: 48 c7 47 08 00 00 00 movq   $0x0,0x8(%rdi)
  11: 00
  12: 48                    rex.W
  13: c7                    .byte 0xc7
  14: 07                    (bad)
...
[   88.803721] RSP: 0018:ffff9a1f892b7d58 EFLAGS:
00000206
[   88.804032] RAX: 0000000000000000 RBX:
ffff9a1f8420c800 RCX: ffff9a1f8420c800
[   88.804560] RDX: ffff9a1f81bc1440 RSI:
0000000000000000 RDI: 0000000000000000
[   88.805056] RBP: ffffffffc04bb0e0 R08:
0000000000000001 R09: 00000000ff7f9a1f
[   88.805473] R10: 000000000001001b R11:
0000000000009a1f R12: 0000000000000140
[   88.806194] R13: 0000000000000001 R14:
ffff9a1f886df400 R15: ffff9a1f886df4ac
[   88.806734] FS:  00007f445601a740(0000)
GS:ffff9a2e7fd80000(0000)
knlGS:0000000000000000
[   88.807225] CS:  0010 DS: 0000 ES: 0000 CR0:
0000000080050033
[   88.807672] CR2: 0000000000000000 CR3:
000000050cc46000 CR4: 00000000000006f0
[   88.808165] Call Trace:
[   88.808459]  <TASK>
[   88.808710] ? __die
(arch/x86/kernel/dumpstack.c:421
arch/x86/kernel/dumpstack.c:434)
[   88.809261] ? page_fault_oops
(arch/x86/mm/fault.c:715)
[   88.809561] ? exc_page_fault
(./arch/x86/include/asm/irqflags.h:26
./arch/x86/include/asm/irqflags.h:87
./arch/x86/include/asm/irqflags.h:147
arch/x86/mm/fault.c:1489
arch/x86/mm/fault.c:1539)
[   88.809806] ? asm_exc_page_fault
(./arch/x86/include/asm/idtentry.h:623)
[   88.810074] ? sfq_dequeue
(net/sched/sch_sfq.c:272 net/sched/sch_sfq.c:499)
sch_sfq
```

		[88.810411] sfq_reset (net/sched/sch_sfq.c:525) sch_sfq [88.810671] qdisc_reset (./include/linux/skbuff.h:2135 ./include/linux/skbuff.h:2441 ./include/linux/skbuff.h:3304 ./include/linux/skbuff.h:3310 net/sched/sch_g ---truncated---			
CVE-2024-50040	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: igb: Do not bring the device up after non-fatal error Commit 004d25060c78 ("igb: Fix igb_down hung on surprise removal") changed igb_io_error_detected() to ignore non-fatal pcie errors in order to avoid hung task that can happen when igb_down() is called multiple times. This caused an issue when processing transient non-fatal errors. igb_io_resume(), which is called after igb_io_error_detected(), assumes that device is brought down by igb_io_error_detected() if the interface is up. This resulted in panic with stacktrace below. [T3256] igb 0000:09:00.0 haeth0: igb: haeth0 NIC Link is Down [T292] pcieport 0000:00:1c.5: AER: Uncorrected (Non-Fatal) error received: 0000:09:00.0 [T292] igb 0000:09:00.0: PCIe Bus Error: severity=Uncorrected (Non-Fatal), type=Transaction Layer, (Requester ID) [T292] igb 0000:09:00.0: device [8086:1537] error status/mask=00004000/00000000 [T292] igb 0000:09:00.0: [14] CmpltTO [200.105524,009][T292] igb 0000:09:00.0: AER: TLP Header: 00000000 00000000 00000000 00000000 [T292] pcieport 0000:00:1c.5: AER: broadcast error_detected message [T292] igb 0000:09:00.0: Non-correctable non-fatal error reported. [T292] pcieport 0000:00:1c.5: AER: broadcast mmio_enabled message [T292] pcieport 0000:00:1c.5: AER: broadcast resume message [T292] ------------[cut here]------------ [T292] kernel BUG at net/core/dev.c:6539! [T292] invalid opcode: 0000 [#1] PREEMPT SMP [T292] RIP: 0010:napi_enable+0x37/0x40	2024-10-21	5.5	Medium

		[T292] Call Trace: [T292] <TASK> [T292] ? die+0x33/0x90 [T292] ? do_trap+0xdc/0x110 [T292] ? napi_enable+0x37/0x40 [T292] ? do_error_trap+0x70/0xb0 [T292] ? napi_enable+0x37/0x40 [T292] ? napi_enable+0x37/0x40 [T292] ? exc_invalid_op+0x4e/0x70 [T292] ? napi_enable+0x37/0x40 [T292] ? asm_exc_invalid_op+0x16/0x20 [T292] ? napi_enable+0x37/0x40 [T292] igb_up+0x41/0x150 [T292] igb_io_resume+0x25/0x70 [T292] report_resume+0x54/0x70 [T292] ? report_frozen_detected+0x20/0x20 [T292] pci_walk_bus+0x6c/0x90 [T292] ? aer_print_port_info+0xa0/0xa0 [T292] pcie_do_recovery+0x22f/0x380 [T292] aer_process_err_devices+0x110/0x160 [T292] aer_isr+0x1c1/0x1e0 [T292] ? disable_irq_nosync+0x10/0x10 [T292] irq_thread_fn+0x1a/0x60 [T292] irq_thread+0xe3/0x1a0 [T292] ? irq_set_affinity_notifier+0x120/0x120 [T292] ? irq_affinity_notify+0x100/0x100 [T292] kthread+0xe2/0x110 [T292] ? kthread_complete_and_exit+0x20/0x20 [T292] ret_from_fork+0x2d/0x50 [T292] ? kthread_complete_and_exit+0x20/0x20 [T292] ret_from_fork_asm+0x11/0x20 [T292] </TASK> To fix this issue igb_io_resume() checks if the interface is running and the device is not down this means igb_io_error_detected() did not bring the device down and there is no need to bring it up.			
CVE-2024-50041	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix macvlan leak by synchronizing access to mac_filter_hash This patch addresses a macvlan leak issue in the i40e driver caused by concurrent access to vsi->mac_filter_hash. The leak occurs when multiple threads attempt to modify the mac_filter_hash simultaneously, leading to inconsistent state and potential memory leaks.	2024-10-21	5.5	Medium

		To fix this, we now wrap the calls to i40e_del_mac_filter() and zeroing vf->default_lan_addr.addr with spin_lock/unlock_bh(&vsi->mac_filter_hash_lock), ensuring atomic operations and preventing concurrent access. Additionally, we add lockdep_assert_held(&vsi->mac_filter_hash_lock) in i40e_add_mac_filter() to help catch similar issues in the future. Reproduction steps: 1. Spawn VFs and configure port vlan on them. 2. Trigger concurrent macvlan operations (e.g., adding and deleting portvlan and/or mac filters). 3. Observe the potential memory leak and inconsistent state in the mac_filter_hash. This synchronization ensures the integrity of the mac_filter_hash and prevents the described leak.			
CVE-2024-50045	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: netfilter: br_netfilter: fix panic with metadata_dst skb Fix a kernel panic in the br_netfilter module when sending untagged traffic via a VxLAN device. This happens during the check for fragmentation in br_nf_dev_queue_xmit. It is dependent on: 1) the br_netfilter module being loaded; 2) net.bridge.bridge-nf-call-iptables set to 1; 3) a bridge with a VxLAN (single-vxlan-device) netdevice as a bridge port; 4) untagged frames with size higher than the VxLAN MTU forwarded/flooded When forwarding the untagged packet to the VxLAN bridge port, before the netfilter hooks are called, br_handle_egress_vlan_tunnel is called and changes the skb_dst to the tunnel dst. The tunnel_dst is a metadata type of dst, i.e., skb_valid_dst(skb) is false, and metadata-	2024-10-21	5.5	Medium

>dst.dev is NULL.

Then in the br_netfilter hooks, in br_nf_dev_queue_xmit, there's a check for frames that needs to be fragmented: frames with higher MTU than the VxLAN device end up calling br_nf_ip_fragment, which in turns call ip_skb_dst_mtu.

The ip_dst_mtu tries to use the skb_dst(skb) as if it was a valid dst with valid dst->dev, thus the crash.

This case was never supported in the first place, so drop the packet instead.

```
PING 10.0.0.2 (10.0.0.2) from 0.0.0.0 h1-eth0:
2000(2028) bytes of data.
[  176.291791] Unable to handle kernel NULL pointer
dereference at
virtual address 0000000000000110
[  176.292101] Mem abort info:
[  176.292184]   ESR = 0x0000000096000004
[  176.292322]   EC = 0x25: DABT (current EL), IL = 32
bits
[  176.292530]   SET = 0, FnV = 0
[  176.292709]   EA = 0, S1PTW = 0
[  176.292862]   FSC = 0x04: level 0 translation fault
[  176.293013] Data abort info:
[  176.293104]   ISV = 0, ISS = 0x00000004, ISS2 =
0x00000000
[  176.293488]   CM = 0, WnR = 0, TnD = 0, TagAccess
= 0
[  176.293787]   GCS = 0, Overlay = 0, DirtyBit = 0, Xs =
0
[  176.293995] user pgtable: 4k pages, 48-bit VAs,
pgdp=0000000043ef5000
[  176.294166] [0000000000000110]
pgd=0000000000000000,
p4d=0000000000000000
[  176.294827] Internal error: Oops:
0000000096000004 [#1] PREEMPT SMP
[  176.295252] Modules linked in: vxlan
ip6_udp_tunnel udp_tunnel veth
br_netfilter bridge stp llc ipv6 crct10dif_ce
[  176.295923] CPU: 0 PID: 188 Comm: ping Not
tainted
6.8.0-rc3-g5b3fbd61b9d1 #2
[  176.296314] Hardware name: linux,dummy-virt
```

		(DT) [176.296535] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [176.296808] pc : br_nf_dev_queue_xmit+0x390/0x4ec [br_netfilter] [176.297382] lr : br_nf_dev_queue_xmit+0x2ac/0x4ec [br_netfilter] [176.297636] sp : ffff800080003630 [176.297743] x29: ffff800080003630 x28: 0000000000000008 x27: ffff6828c49ad9f8 [176.298093] x26: ffff6828c49ad000 x25: 0000000000000000 x24: 00000000000003e8 [176.298430] x23: 0000000000000000 x22: ffff6828c4960b40 x21: ffff6828c3b16d28 [176.298652] x20: ffff6828c3167048 x19: ffff6828c3b16d00 x18: 0000000000000014 [176.298926] x17: ffffb0476322f000 x16: ffffb7e164023730 x15: 0000000095744632 [176.299296] x14: ffff6828c3f1c880 x13: 0000000000000002 x12: ffffb7e137926a70 [176.299574] x11: 0000000000000001 x10: ffff6828c3f1c898 x9 : 0000000000000000 [176.300049] x8 : ffff6828c49bf070 x7 : 0008460f18d5f20e x6 : f20e0100bebafeca [176.300302] x5 : ffff6828c7f918fe x4 : ffff6828c49bf070 x3 : 0000000000000000 [176.300586] x2 : 0000000000000000 x1 : ffff6828c3c7ad00 x0 : ffff6828c7f918f0 [176.300889] Call trace: [176.301123] br_nf_dev_queue_xmit+0x390/0x4ec [br_netfilter] [176.301411] br_nf_post_routing+0x2a8/0x3e4 [br_netfilter] [176.301703] nf_hook_slow+0x48/0x124 [176.302060] br_forward_finish+0xc8/0xe8 [bridge] [176.302371] br_nf_hook_thresh+0x124/0x134 [br_netfilter] [176.302605] br_nf_forward_finish+0x118/0x22c [br_netfilter] [176.302824]			

		br_nf_forward_ip.part.0+0x264/0x290 [br_netfilter] [176.303136] br_nf_forward+0x2b8/0x4e0 [br_netfilter] [176.303359] nf_hook_slow+0x48/0x124 [176.303 ---truncated---			
CVE-2024-50046	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Prevent NULL-pointer dereference in nfs42_complete_copies() On the node of an NFS client, some files saved in the mountpoint of the NFS server were copied to another location of the same NFS server. Accidentally, the nfs42_complete_copies() got a NULL-pointer dereference crash with the following syslog: [232064.838881] NFSv4: state recovery failed for open file nfs/pvc-12b5200d-cd0f-46a3-b9f0-af8f4fe0ef64.qcow2, error = -116 [232064.839360] NFSv4: state recovery failed for open file nfs/pvc-12b5200d-cd0f-46a3-b9f0-af8f4fe0ef64.qcow2, error = -116 [232066.588183] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000058 [232066.588586] Mem abort info: [232066.588701] ESR = 0x0000000096000007 [232066.588862] EC = 0x25: DABT (current EL), IL = 32 bits [232066.589084] SET = 0, FnV = 0 [232066.589216] EA = 0, S1PTW = 0 [232066.589340] FSC = 0x07: level 3 translation fault [232066.589559] Data abort info: [232066.589683] ISV = 0, ISS = 0x00000007 [232066.589842] CM = 0, WnR = 0 [232066.589967] user pgtable: 64k pages, 48-bit VAs, pgdp=00002000956ff400 [232066.590231] [0000000000000058] pgd=08001100ae100003, p4d=08001100ae100003, pud=08001100ae100003, pmd=08001100b3c00003, pte=0000000000000000 [232066.590757] Internal error: Oops: 96000007 [#1] SMP [232066.590958] Modules linked in: rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace fscache netfs ocfs2_dlmfs ocfs2_stack_o2cb ocfs2_dlm vhost_net vhost vhost_iotlb tap tun	2024-10-21	5.5	Medium

		ipt_rpfilter xt_multiport ip_set_hash_ip ip_set_hash_net xfrm_interface xfrm6_tunnel tunnel4 tunnel6 esp4 ah4 wireguard libcurve25519_generic veth xt_addrtype xt_set nf_conntrack_netlink ip_set_hash_ipportnet ip_set_hash_ipportip ip_set_bitmap_port ip_set_hash_ipport dummy ip_set ip_vs_sh ip_vs_wrr ip_vs_rr ip_vs iptable_filter sch_ingress nfnetlink_cttimeout vport_gre ip_gre ip_tunnel gre vport_geneve geneve vport_vxlan vxlan ip6_udp_tunnel udp_tunnel openvswitch nf_conncount dm_round_robin dm_service_time dm_multipath xt_nat xt_MASQUERADE nft_chain_nat nf_nat xt_mark xt_conntrack xt_comment nft_compat nft_counter nf_tables nfnetlink ocfs2 ocfs2_nodemanager ocfs2_stackglue iscsi_tcp libiscsi_tcp libiscsi scsi_transport_iscsi ipmi_ssif nbd overlay 8021q garp mrp bonding tls rfkill sunrpc ext4 mbcache jbd2 [232066.591052] vfat fat cas_cache cas_disk ses enclosure scsi_transport_sas sg acpi_ipmi ipmi_si ipmi_devintf ipmi_msghandler ip_tables vfio_pci vfio_pci_core vfio_virqfd vfio_iommu_type1 vfio dm_mirror dm_region_hash dm_log dm_mod nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 br_netfilter bridge stp llc fuse xfs libcrc32c ast drm_vram_helper qla2xxx drm_kms_helper syscopyarea crct10dif_ce sysfillrect ghash_ce sysimgblt sha2_ce fb_sys_fops cec sha256_arm64 sha1_ce drm_ttm_helper ttm nvme_fc igb sbsa_gwdt nvme_fabrics drm nvme_core i2c_algo_bit i40e scsi_transport_fc megaraid_sas aes_neon_bs [232066.596953] CPU: 6 PID: 4124696 Comm: 10.253.166.125- Kdump: loaded Not tainted 5.15.131-9.cl9_ocfs2.aarch64 #1 [232066.597356] Hardware name: Great Wall .\x93\x8e...RF6260 V5/GWMSSE2GL1T, BIOS T656FBE_V3.0.18 2024-01-06 [232066.597721] pstate: 20400009 (nzCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [232066.598034] pc : nfs4_reclaim_open_state+0x220/0x800 [nfsv4] [232066.598327] lr : nfs4_reclaim_open_state+0x12c/0x800 [nfsv4] [232066.598595] sp : ffff8000f568fc70 [232066.598731] x29: ffff8000f568fc70 x28: 0000000000001000 x27: ffff21003db33000 [232066.599030] x26: ffff800005521ae0 x25: ffff0100f98fa3f0 x24: 0000000000000001 [232066.599319] x23: ffff800009920008 x22: ffff21003db33040 x21: ffff21003db33050			

		[232066.599628] x20: ffff410172fe9e40 x19: ffff410172fe9e00 x18: 0000000000000000 [232066.599914] x17: 0000000000000000 x16: 0000000000000004 x15: 0000000000000000 [232066.600195] x14: 0000000000000000 x13: ffff800008e685a8 x12: 00000000eac0c6e6 [232066.600498] x11: 00000000000000 ---truncated---			

CVE-2024-50048	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		fbcon: Fix a NULL pointer dereference issue in fbcon_putcs
		
		syzbot has found a NULL pointer dereference bug in fbcon.
		Here is the simplified C reproducer:
		
		struct param {
		uint8_t type;
		struct tiocl_selection ts;
		};
		
		int main()
		{
		struct fb_con2fbmap con2fb;
		struct param param;
		
		int fd = open("/dev/fb1", 0, 0);
		
		con2fb.console = 0x19;
		con2fb.framebuffer = 0;
		ioctl(fd, FBIOPUT_CON2FBMAP, &con2fb);
		
		param.type = 2;
		param.ts.xs = 0; param.ts.ys = 0;
		param.ts.xe = 0; param.ts.ye = 0;
		param.ts.sel_mode = 0;
		
		int fd1 = open("/dev/tty1", O_RDWR, 0);
		ioctl(fd1, TIOCLINUX, &param);
		
		con2fb.console = 1;
		con2fb.framebuffer = 0;
		ioctl(fd, FBIOPUT_CON2FBMAP, &con2fb);
		
		return 0;
		}
		
		After calling ioctl(fd1, TIOCLINUX, &param), the subsequent ioctl(fd, FBIOPUT_CON2FBMAP,
			2024-10-21	5.5	Medium

		&con2fb) causes the kernel to follow a different execution path: set_con2fb_map -> con2fb_init_display -> fbcon_set_disp -> redraw_screen -> hide_cursor -> clear_selection -> highlight -> invert_screen -> do_update_region -> fbcon_putcs -> ops->putcs Since ops->putcs is a NULL pointer, this leads to a kernel panic. To prevent this, we need to call set_blitting_type() within set_con2fb_map() to properly initialize ops->putcs.			
CVE-2024-50049	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointer before dereferencing se [WHAT & HOW] se is null checked previously in the same function, indicating it might be null; therefore, it must be checked when used again. This fixes 1 FORWARD_NULL issue reported by Coverity.	2024-10-21	5.5	Medium
CVE-2024-50058	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: serial: protect uart_port_dtr_rts() in uart_shutdown() too Commit af224ca2df29 (serial: core: Prevent unsafe uart port access, part 3) added few uport == NULL checks. It added one to uart_shutdown(), so the commit assumes, uport can be NULL in there. But right after that protection, there is an unprotected "uart_port_dtr_rts(uport, false);" call. That is invoked only if HUPCL is set, so I assume that is the	2024-10-21	5.5	Medium

		reason why we do not see lots of these reports. Or it cannot be NULL at this point at all for some reason :P. Until the above is investigated, stay on the safe side and move this dereference to the if too. I got this inconsistency from Coverity under CID 1585130. Thanks.			
CVE-2024-50062	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs-srv: Avoid null pointer deref during path establishment For RTRS path establishment, RTRS client initiates and completes con_num of connections. After establishing all its connections, the information is exchanged between the client and server through the info_req message. During this exchange, it is essential that all connections have been established, and the state of the RTRS srv path is CONNECTED. So add these sanity checks, to make sure we detect and abort process in error scenarios to avoid null pointer deref.	2024-10-21	5.5	Medium
CVE-2024-50064	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: zram: free secondary algorithms names We need to kfree() secondary algorithms names when reset zram device that had multi-streams, otherwise we leak memory. [senozhatsky@chromium.org: kfree(NULL) is legal] Link: https://lkml.kernel.org/r/20240917013021.868769-1-senozhatsky@chromium.org	2024-10-21	5.5	Medium
CVE-2024-9677	zyxel - USG FLEX H series uOS firmware	The insufficiently protected credentials vulnerability in the CLI command of the USG FLEX H series uOS firmware version V1.21 and earlier versions could allow an authenticated local attacker to gain privilege escalation by stealing the authentication token of a login administrator. Note that this attack could be	2024-10-22	5.5	Medium

		successful only if the administrator has not logged out.			
CVE-2023-52918	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: media: pci: cx23885: check cx23885_vdev_init() return cx23885_vdev_init() can return a NULL pointer, but that pointer is used in the next line without a check. Add a NULL pointer check and go to the error unwind if it is NULL.	2024-10-22	5.5	Medium
CVE-2023-52919	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nfc: nci: fix possible NULL pointer dereference in send_acknowledge() Handle memory allocation failure from nci_skb_alloc() (calling alloc_skb()) to avoid possible NULL pointer dereference.	2024-10-22	5.5	Medium
CVE-2024-45335	trendmicro - antivirus_one	Trend Micro Antivirus One, version 3.10.4 and below contains a vulnerability that could allow an attacker to use a specifically crafted virus to allow itself to bypass and evade a virus scan detection.	2024-10-22	5.5	Medium
CVE-2024-20274	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Secure Firewall Management Center (FMC) Software, formerly Firepower Management Center Software, could allow an authenticated, remote attacker to inject arbitrary HTML content into a device-generated document. This vulnerability is due to improper validation of user-supplied data. An attacker could exploit this vulnerability by submitting malicious content to an affected device and using the device to generate a document that contains sensitive information. A successful exploit could allow the attacker to alter the standard layout of the device-generated documents, access arbitrary files from the underlying operating system, and conduct server-side request forgery (SSRF) attacks. To successfully exploit this vulnerability, an attacker would need valid credentials for a user account with policy-editing permissions, such as Network Admin, Intrusion	2024-10-23	5.5	Medium

		Admin, or any custom user role with the same capabilities.			
CVE-2024-40810	apple - macos	An out-of-bounds write issue was addressed with improved input validation. This issue is fixed in macOS Sonoma 14.6. An app may be able to cause a coprocessor crash.	2024-10-24	5.5	Medium
CVE-2024-44185	apple - multiple products	The issue was addressed with improved checks. This issue is fixed in tvOS 17.6, visionOS 1.3, Safari 17.6, watchOS 10.6, iOS 17.6 and iPadOS 17.6, macOS Sonoma 14.6. Processing maliciously crafted web content may lead to an unexpected process crash.	2024-10-24	5.5	Medium
CVE-2024-44205	apple - multiple products	A privacy issue was addressed with improved private data redaction for log entries. This issue is fixed in macOS Ventura 13.6.8, macOS Monterey 12.7.6, iOS 16.7.9 and iPadOS 16.7.9, iOS 17.6 and iPadOS 17.6, macOS Sonoma 14.6. A sandboxed app may be able to access sensitive user data in system logs.	2024-10-24	5.5	Medium
CVE-2024-44099	google - android	There is a possible Local bypass of user interaction due to an insecure default value. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47015	google - Android	In ProtocolMiscHwConfigChangeAdapter::GetData() of protocolmiscadapter.cpp, there is a possible out-of-bounds read due to a missing bounds check. This could lead to local information disclosure with baseband firmware compromise required. User Interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47018	google - android	In pmucal_rae_handle_seq_int of flexpmu_cal_rae.c, there is a possible out of bounds read due to a buffer overflow. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47019	google - android	In ProtocolEmbmsSaiListAdapter::Init() of protocolembmsadapter.cpp, there is a possible out of bounds read due to a missing bounds check. This could lead to local information disclosure with baseband firmware compromise required. User Interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47025	google - android	In ppmp_protect_buf of drm_fw.c, there is a possible information disclosure due to a logic error in the code. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47026	google - android	In gsc_gsa_rescue of gsc_gsa.c, there is a possible out of bounds read due to an incorrect bounds check. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium

CVE-2024-47029	google - android	In TrustySharedMemoryManager::GetSharedMemory of ondevice/trusty/trusty_shared_memory_manager.cc, there is a possible out of bounds read due to an incorrect bounds check. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47034	google - android	there is a possible out of bounds read due to a missing bounds check. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	5.5	Medium
CVE-2024-47483	dell - multiple products	Dell Data Lakehouse, version(s) 1.0.0.0 and 1.1.0.0, contain(s) an Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability. An unauthenticated attacker with local access could potentially exploit this vulnerability, leading to Information disclosure.	2024-10-25	5.5	Medium
CVE-2024-20264	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-20269	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-20298	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of	2024-10-23	5.4	Medium

		user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.			
CVE-2024-20300	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-20364	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-20377	cisco - multiple products	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) could allow an authenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack against a user of the interface. This vulnerability is due to the web-based management interface not properly validating user-supplied input. An attacker could exploit this vulnerability by by persuading a user of the interface to click a crafted link. A successful exploit could allow the attacker to execute arbitrary script code in the context of the affected interface or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-20387	cisco - Cisco Firepower	A vulnerability in the web-based management interface of Cisco FMC Software could allow an authenticated, remote attacker to store malicious	2024-10-23	5.4	Medium

	Management Center	content for use in XSS attacks. This vulnerability is due to improper input sanitization in the web-based management interface of Cisco FMC Software. An attacker could exploit this vulnerability by persuading a user to click a malicious link. A successful exploit could allow the attacker to conduct a stored XSS attack on an affected device.			
CVE-2024-20410	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	5.4	Medium
CVE-2024-44206	apple - multiple products	An issue in the handling of URL protocols was addressed with improved logic. This issue is fixed in tvOS 17.6, visionOS 1.3, Safari 17.6, watchOS 10.6, iOS 17.6 and iPadOS 17.6, macOS Sonoma 14.6. A user may be able to bypass some web content restrictions.	2024-10-24	5.4	Medium
CVE-2024-47689	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to don't set SB_RDONLY in f2fs_handle_critical_error() syzbot reports a f2fs bug as below: ------------[cut here]------------ WARNING: CPU: 1 PID: 58 at kernel/rcu/sync.c:177 rcu_sync_dtor+0xcd/0x180 kernel/rcu/sync.c:177 CPU: 1 UID: 0 PID: 58 Comm: kworker/1:2 Not tainted 6.10.0-syzkaller-12562-g1722389b0d86 #0 Workqueue: events destroy_super_work RIP: 0010:rcu_sync_dtor+0xcd/0x180 kernel/rcu/sync.c:177 Call Trace: percpu_free_rwsem+0x41/0x80 kernel/locking/percpu-rwsem.c:42 destroy_super_work+0xec/0x130 fs/super.c:282 process_one_work kernel/workqueue.c:3231 [inline] process_scheduled_works+0xa2c/0x1830 kernel/workqueue.c:3312 worker_thread+0x86d/0xd40 kernel/workqueue.c:3390	2024-10-21	5.3	Medium

		kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 As Christian Brauner pointed out [1]: the root cause is f2fs sets SB_RDONLY flag in internal function, rather than setting the flag covered w/ sb->s_umount semaphore via remount procedure, then below race condition causes this bug: - freeze_super() - sb_wait_write(sb, SB_FREEZE_WRITE) - sb_wait_write(sb, SB_FREEZE_PAGEFAULT) - sb_wait_write(sb, SB_FREEZE_FS) - f2fs_handle_critical_error - sb->s_flags |= SB_RDONLY - thaw_super - thaw_super_locked - sb_rdonly() is true, so it skips sb_freeze_unlock(sb, SB_FREEZE_FS) - deactivate_locked_super Since f2fs has almost the same logic as ext4 [2] when handling critical error in filesystem if it mounts w/ errors=remount-ro option: - set CP_ERROR_FLAG flag which indicates filesystem is stopped - record errors to superblock - set SB_RDONLY falg Once we set CP_ERROR_FLAG flag, all writable interfaces can detect the flag and stop any further updates on filesystem. So, it is safe to not set SB_RDONLY flag, let's remove the logic and keep in line w/ ext4 [3]. [1] https://lore.kernel.org/all/20240729-himbeeren-funknetz-96e62f9c7aee@brauner [2] https://lore.kernel.org/all/20240729132721.hxih6ehigadqf7wx@quack3 [3] https://lore.kernel.org/linux-ext4/20240805201241.27286-1-jack@suse.cz			
CVE-2024-50312	redhat - openshift _containe	A vulnerability was found in GraphQL due to improper access controls on the GraphQL introspection query. This flaw allows unauthorized	2024-10-22	5.3	Medium

	r_platfor m	users to retrieve a comprehensive list of available queries and mutations. Exposure to this flaw increases the attack surface, as it can facilitate the discovery of flaws or errors specific to the application's GraphQL implementation.			
CVE-2024-31880	ibm - Db2 for Linux, UNIX and Windows	IBM Db2 for Linux, UNIX and Windows (includes Db2 Connect Server) 10.5, 11.1, and 11.5 is vulnerable to a denial of service, under specific configurations, as the server may crash when using a specially crafted SQL statement by an authenticated user.	2024-10-23	5.3	Medium
CVE-2024-20388	cisco - multiple products	A vulnerability in the password change feature of Cisco Firepower Management Center (FMC) software could allow an unauthenticated, remote attacker to determine valid user names on an affected device. This vulnerability is due to improper authentication of password update responses. An attacker could exploit this vulnerability by forcing a password reset on an affected device. A successful exploit could allow the attacker to determine valid user names in the unauthenticated response to a forced password reset.	2024-10-23	5.3	Medium
CVE-2024-20493	cisco - multiple products	A vulnerability in the login authentication functionality of the Remote Access SSL VPN feature of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to deny further VPN user authentications for several minutes, resulting in a temporary denial of service (DoS) condition. This vulnerability is due to ineffective handling of memory resources during the authentication process. An attacker could exploit this vulnerability by sending crafted packets, which could cause resource exhaustion of the authentication process. A successful exploit could allow the attacker to deny authentication for Remote Access SSL VPN users for several minutes, resulting in a temporary DoS condition.	2024-10-23	5.3	Medium
CVE-2024-20526	cisco - multiple products	A vulnerability in the SSH server of Cisco Adaptive Security Appliance (ASA) Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition for the SSH server of an affected device. This vulnerability is due to a logic error when an SSH session is established. An attacker could exploit this	2024-10-23	5.3	Medium

		vulnerability by sending crafted SSH messages to an affected device. A successful exploit could allow the attacker to exhaust available SSH resources on the affected device so that new SSH connections to the device are denied, resulting in a DoS condition. Existing SSH connections to the device would continue to function normally. The device must be rebooted manually to recover. However, user traffic would not be impacted and could be managed using a remote application such as Cisco Adaptive Security Device Manager (ASDM).			
CVE-2024-47030	google - Android	Android before 2024-10-05 on Google Pixel devices allows information disclosure in the ACPM component, A-315191818.	2024-10-25	5.1	Medium
CVE-2023-50310	ibm - CICS Transaction Gateway for Multiplatforms	IBM CICS Transaction Gateway for Multiplatforms 9.2 and 9.3 transmits or stores authentication credentials, but it uses an insecure method that is susceptible to unauthorized interception and/or retrieval.	2024-10-23	4.9	Medium
CVE-2024-30159	mitel - micollab	A vulnerability in the web conferencing component of Mitel MiCollab through 9.7.1.110 could allow an authenticated attacker with administrative privileges to conduct a Stored Cross-Site Scripting (XSS) attack due to insufficient validation of user input. A successful exploit could allow an attacker to execute arbitrary scripts.	2024-10-21	4.8	Medium
CVE-2024-30160	mitel - micollab	A vulnerability in the Suite Applications Services component of Mitel MiCollab through 9.7.1.110 could allow an authenticated attacker with administrative privileges to conduct a Stored Cross-Site Scripting (XSS) attack due to insufficient validation of user input. A successful exploit could allow an attacker to execute arbitrary scripts.	2024-10-21	4.8	Medium
CVE-2024-20386	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	4.8	Medium
CVE-2024-20403	cisco - Cisco Firepower	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an authenticated, remote	2024-10-23	4.8	Medium

	Management Center	attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.			
CVE-2024-20409	cisco - Cisco Firepower Management Center	A vulnerability in the web-based management interface of Cisco Firepower Management Center (FMC) Software could allow an unauthenticated, remote attacker to conduct a cross-site scripting (XSS) attack against a user of the interface of an affected device. This vulnerability is due to insufficient validation of user-supplied input by the web-based management interface. An attacker could exploit this vulnerability by inserting crafted input into various data fields in an affected interface. A successful exploit could allow the attacker to execute arbitrary script code in the context of the interface, or access sensitive, browser-based information.	2024-10-23	4.8	Medium

CVE-2024-47679	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved:
		
		vfs: fix race between evice_inodes() and find_inode()&iput()
		
		Hi, all
		
		Recently I noticed a bug[1] in btrfs, after digged it into
		and I believe it'a race in vfs.
		
		Let's assume there's a inode (ie ino 261) with i_count 1 is
		called by iput(), and there's a concurrent thread calling
		generic_shutdown_super().
		
		cpu0: cpu1:
		iput() // i_count is 1
		  ->spin_lock(inode)
		  ->dec i_count to 0
		  ->iput_final() generic_shutdown_super()
		    ->__inode_add_lru() ->evict_inodes()
		      // cause some reason[2] ->if (atomic_read(inode->i_count)) continue;
		      // return before // inode 261 passed the above check
			2024-10-21	4.7	Medium

		// list_lru_add_obj() // and then schedule out
		->spin_unlock()
		// note here: the inode 261
		// was still at sb list and hash list,
		// and I_FREEING|I_WILL_FREE was not been set
		
		btrfs_iget()
		// after some function calls
		->find_inode()
		// found the above inode 261
		->spin_lock(inode)
		// check I_FREEING|I_WILL_FREE
		// and passed
		->__iget()
		->spin_unlock(inode) // schedule back
		->spin_lock(inode)
		// check (I_NEW|I_FREEING|I_WILL_FREE) flags,
		// passed and set I_FREEING
		iput() ->spin_unlock(inode)
		->spin_lock(inode) ->evict()
		// dec i_count to 0
		->iput_final()
		->spin_unlock()
		->evict()
		
		Now, we have two threads simultaneously evicting the same inode, which may trigger the BUG(inode->i_state & I_CLEAR)
		statement both within clear_inode() and iput().
		
		To fix the bug, recheck the inode->i_count after holding i_lock.
		Because in the most scenarios, the first check is valid, and
		the overhead of spin_lock() can be reduced.
		
		If there is any misunderstanding, please let me know, thanks.
		
		[1]: https://lore.kernel.org/linux-btrfs/000000000000eabe1d0619c48986@google.com/
		[2]: The reason might be 1. SB_ACTIVE was removed or 2. mapping_shrinkable()
		return false when I reproduced the bug.			

CVE-2024-49859	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to check atomic_file in f2fs ioctl interfaces	2024-10-21	4.7	Medium

		Some f2fs ioctl interfaces like f2fs_ioc_set_pin_file(), f2fs_move_file_range(), and f2fs_defragment_range() missed to check atomic_write status, which may cause potential race issue, fix it.			
CVE-2024-49998	linux - multiple products	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	2024-10-21	4.7	Medium

<table>
<tr><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td></td><td></td><td>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")</td><td></td><td></td><td></td></tr>
</table>

		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-2024-50006	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: ext4: fix i_data_sem unlock order in ext4_ind_migrate() Fuzzing reports a possible deadlock in jbd2_log_wait_commit. This issue is triggered when an EXT4_IOC_MIGRATE ioctl is set to require synchronous updates because the file descriptor is opened with O_SYNC. This can lead to the jbd2_journal_stop() function calling jbd2_might_wait_for_commit(), potentially causing a deadlock if the EXT4_IOC_MIGRATE call races with a write(2) system call. This problem only arises when CONFIG_PROVE_LOCKING is enabled. In this case, the jbd2_might_wait_for_commit macro locks jbd2_handle in the jbd2_journal_stop function while i_data_sem is locked. This triggers lockdep because the jbd2_journal_start function might also lock the same jbd2_handle simultaneously. Found by Linux Verification Center (linuxtesting.org) with syzkaller. Rule: add	2024-10-21	4.7	Medium
CVE-2022-48989	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: fscache: Fix oops due to race with cookie_lru and use_cookie If a cookie expires from the LRU and the LRU_DISCARD flag is set, but the state machine has not run yet, it's possible	2024-10-21	4.7	Medium

		another thread can call fscache_use_cookie and begin to use it. When the cookie_worker finally runs, it will see the LRU_DISCARD flag set, transition the cookie->state to LRU_DISCARDING, which will then withdraw the cookie. Once the cookie is withdrawn the object is removed the below oops will occur because the object associated with the cookie is now NULL. Fix the oops by clearing the LRU_DISCARD bit if another thread uses the cookie before the cookie_worker runs. BUG: kernel NULL pointer dereference, address: 0000000000000008 ... CPU: 31 PID: 44773 Comm: kworker/u130:1 Tainted: G E 6.0.0-5.dneg.x86_64 #1 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 08/26/2022 Workqueue: events_unbound netfs_rreq_write_to_cache_work [netfs] RIP: 0010:cachefiles_prepare_write+0x28/0x90 [cachefiles] ... Call Trace: netfs_rreq_write_to_cache_work+0x11c/0x320 [netfs] process_one_work+0x217/0x3e0 worker_thread+0x4a/0x3b0 kthread+0xd6/0x100			
CVE-2022-49003	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: nvme: fix SRCU protection of nvme_ns_head list Walking the nvme_ns_head siblings list is protected by the head's srcu in nvme_ns_head_submit_bio() but not nvme_mpath_revalidate_paths(). Removing namespaces from the list also fails to synchronize the srcu. Concurrent scan work can therefore cause use-after-frees. Hold the head's srcu lock in nvme_mpath_revalidate_paths() and	2024-10-21	4.7	Medium

		synchronize with the srcu, not the global RCU, in nvme_ns_remove(). Observed the following panic when making NVMe/RDMA connections with native multipath on the Rocky Linux 8.6 kernel (it seems the upstream kernel has the same race condition). Disassembly shows the faulting instruction is cmp 0x50(%rdx),%rcx; computing capacity != get_capacity(ns->disk). Address 0x50 is dereferenced because ns->disk is NULL. The NULL disk appears to be the result of concurrent scan work freeing the namespace (note the log line in the middle of the panic). [37314.206036] BUG: unable to handle kernel NULL pointer dereference at 0000000000000050 [37314.206036] nvme0n3: detected capacity change from 0 to 11811160064 [37314.299753] PGD 0 P4D 0 [37314.299756] Oops: 0000 [#1] SMP PTI [37314.299759] CPU: 29 PID: 322046 Comm: kworker/u98:3 Kdump: loaded Tainted: G W X --------- - - 4.18.0-372.32.1.el8test86.x86_64 #1 [37314.299762] Hardware name: Dell Inc. PowerEdge R720/0JP31P, BIOS 2.7.0 05/23/2018 [37314.299763] Workqueue: nvme-wq nvme_scan_work [nvme_core] [37314.299783] RIP: 0010:nvme_mpath_revalidate_paths+0x26/0xb0 [nvme_core] [37314.299790] Code: 1f 44 00 00 66 66 66 66 90 55 53 48 8b 5f 50 48 8b 83 c8 c9 00 00 48 8b 13 48 8b 48 50 48 39 d3 74 20 48 8d 42 d0 48 8b 50 20 <48> 3b 4a 50 74 05 f0 80 60 70 ef 48 8b 50 30 48 8d 42 d0 48 39 d3 [37315.058803] RSP: 0018:ffffabe28f913d10 EFLAGS: 00010202 [37315.121316] RAX: ffff927a077da800 RBX: ffff92991dd70000 RCX: 0000000001600000 [37315.206704] RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff92991b719800 [37315.292106] RBP: ffff929a6b70c000 R08: 000000010234cd4a R09: c0000000ffff7fff [37315.377501] R10: 0000000000000001 R11: ffffabe28f913a30 R12: 0000000000000000 [37315.462889] R13: ffff92992716600c R14: ffff929964e6e030 R15: ffff92991dd70000			

		[37315.548286] FS: 0000000000000000(0000) GS:ffff92b87fb80000(0000) knlGS:0000000000000000 [37315.645111] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [37315.713871] CR2: 0000000000000050 CR3: 0000002208810006 CR4: 00000000000606e0 [37315.799267] Call Trace: [37315.828515] nvme_update_ns_info+0x1ac/0x250 [nvme_core] [37315.892075] nvme_validate_or_alloc_ns+0x2ff/0xa00 [nvme_core] [37315.961871] ? __blk_mq_free_request+0x6b/0x90 [37316.015021] nvme_scan_work+0x151/0x240 [nvme_core] [37316.073371] process_one_work+0x1a7/0x360 [37316.121318] ? create_worker+0x1a0/0x1a0 [37316.168227] worker_thread+0x30/0x390 [37316.212024] ? create_worker+0x1a0/0x1a0 [37316.258939] kthread+0x10a/0x120 [37316.297557] ? set_kthread_struct+0x50/0x50 [37316.347590] ret_from_fork+0x35/0x40 [37316.390360] Modules linked in: nvme_rdma nvme_tcp(X) nvme_fabrics nvme_core netconsole iscsi_tcp libiscsi_tcp dm_queue_length dm_service_time nf_conntrack_netlink br_netfilter bridge stp llc overlay nft_chain_nat ipt_MASQUERADE nf_nat xt_addrtype xt_CT nft_counter xt_state xt_conntrack nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 xt_comment xt_multiport nft_compat nf_tables libcrc32c nfnetlink dm_multipath tg3 rpcrdma sunrpc rdma_ucm ib_srpt ib_isert iscsi_target_mod target_core_mod ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm ib_ipoib iw_cm ib_cm intel_rapl_msr iTCO_wdt iTCO_vendor_support dcdbas intel_rapl_common sb_edac x86_pkg_temp_thermal intel_powerclamp coretemp kvm_intel ipmi_ssif kvm irqbypass crct10dif_pclmul crc32_pclmul mlx5_ib ghash_clmulni_intel ib_uverbs rapl intel_cstate intel_uncore ib_core ipmi_si joydev mei_me pcspkr ipmi_devintf mei lpc_ich wmi ipmi_msghandler acpi_power_meter ex ---truncated---			
CVE-2024-47028	google - android	In ffu_flash_pack of ffu.c, there is a possible out of bounds read due to an integer overflow. This could lead to local information disclosure with System execution privileges needed. User interaction is not needed for exploitation.	2024-10-25	4.4	Medium

CVE-2024-43173	ibm - multiple products	IBM Concert 1.0.0 and 1.0.1 vulnerable to attacks that rely on the use of cookies without the SameSite attribute.	2024-10-22	3.7	Low
CVE-2024-47738	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't use rate mask for offchannel TX either Like the commit ab9177d83c04 ("wifi: mac80211: don't use rate mask for scanning"), ignore incorrect settings to avoid no supported rate warning reported by syzbot. The syzbot did bisect and found cause is commit 9df66d5b9f45 ("cfg80211: fix default HE tx bitrate mask in 2G band"), which however corrects bitmask of HE MCS and recognizes correctly settings of empty legacy rate plus HE MCS rate instead of returning -EINVAL. As suggestions [1], follow the change of SCAN TX to consider this case of offchannel TX as well. [1] https://lore.kernel.org/linux-wireless/6ab2dc9c3afe753ca6fdcdd1421e7a1f47e87b84.camel@sipsolutions.net/T/#m2ac2a6d2be06a37c9c47a3d8a44b4f647ed4f024	2024-10-21	3.3	Low
CVE-2024-50044	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: FIX possible deadlock in rfcomm_sk_state_change rfcomm_sk_state_change attempts to use sock_lock so it must never be called with it locked but rfcomm_sock_ioctl always attempt to lock it causing the following trace: == WARNING: possible circular locking dependency detected 6.8.0-syzkaller-08951-gfe46a7dd189e #0 Not tainted -- syz-executor386/5093 is trying to acquire lock: ffff88807c396258 (sk_lock-AF_BLUETOOTH-	2024-10-21	3.3	Low

		BTPROTO_RFCOMM){+.+.}-{0:0}, at: lock_sock include/net/sock.h:1671 [inline] ffff88807c396258 (sk_lock-AF_BLUETOOTH-BTPROTO_RFCOMM){+.+.}-{0:0}, at: rfcomm_sk_state_change+0x5b/0x310 net/bluetooth/rfcomm/sock.c:73 but task is already holding lock: ffff88807badfd28 (&d->lock){+.+.}-{3:3}, at: __rfcomm_dlc_close+0x226/0x6a0 net/bluetooth/rfcomm/core.c:491			
CVE-2024-50057	linux - multiple products	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tipd: Free IRQ only if it was requested before In polling mode, if no IRQ was requested there is no need to free it. Call devm_free_irq() only if client->irq is set. This fixes the warning caused by the tps6598x module removal: WARNING: CPU: 2 PID: 333 at kernel/irq/devres.c:144 devm_free_irq+0x80/0x8c ... ... Call trace: devm_free_irq+0x80/0x8c tps6598x_remove+0x28/0x88 [tps6598x] i2c_device_remove+0x2c/0x9c device_remove+0x4c/0x80 device_release_driver_internal+0x1cc/0x228 driver_detach+0x50/0x98 bus_remove_driver+0x6c/0xbc driver_unregister+0x30/0x60 i2c_del_driver+0x54/0x64 tps6598x_i2c_driver_exit+0x18/0xc3c [tps6598x] __arm64_sys_delete_module+0x184/0x264 invoke_syscall+0x48/0x110 el0_svc_common.constprop.0+0xc8/0xe8 do_el0_svc+0x20/0x2c el0_svc+0x28/0x98 el0t_64_sync_handler+0x13c/0x158 el0t_64_sync+0x190/0x194	2024-10-21	3.3	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.