CVE-2024-58237

net/core/filter.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f1692ee23dcaaddc24ba407b269707ee5df1301f

修改的函数用于判断某个 BPF 辅助函数是否会修改数据包内容，从而导致之前获取的数据包指针失效。新增的代码将 BPF\_FUNC\_tail\_call标记为可能修改数据包内容的函数，即使尾调用本身不直接修改数据包。如果程序中有尾调用，校验器会认为所有数据包指针都可能失效。后续对数据包指针的访问会被拒绝，避免内存错误。

CVE-2024-58099

drivers/net/vmxnet3/vmxnet3\_xdp.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f82eb34fb59a8fb96c19f4f492c20eb774140bb5

计算 DMA 地址时，假设 xdpf->data 的偏移量是固定的，但实际上，XDP BPF 程序可能已修改 xdp->data 的指针位置。动态计算 xdpf->data 的实际偏移量：xdpf->data - xdpf->data\_hard\_start。将静态偏移 VMXNET3\_XDP\_HEADROOM 替换为动态偏移 xdpf->data - (void \*)xdpf。确保 DMA 地址与数据包内容的实际内存布局一致。

CVE-2024-58097

drivers/net/wireless/ath/ath11k/dp\_rx.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b4991fc41745645f8050506f5a8578bd11e6b378

当 buf\_id 无效时，break 会终止整个循环，导致监控环中的后续数据无法处理。硬件可能持续发送相同的无效 buf\_id，引发死循环。使用 goto next\_msdu 跳过无效的 buf\_id，继续处理监控环中的其他数据。避免死循环，同时确保其他有效数据能被正常处理。

CVE-2024-58096

drivers/net/wireless/ath/ath11k/dp\_rx.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b85758e76b6452740fc2a08ced6759af64c0d59a

ath11k\_hal\_srng\_\* 函数需要 srng->lock 保护，但监控模式处理函数未加锁。并发访问会导致 SRNG 状态不一致，触发内核警告。将 mon\_dst\_srng 类型从 void\* 改为 struct hal\_srng\*。在访问 SRNG 前后添加自旋锁（spin\_lock\_bh）。

CVE-2024-58095

fs/jfs/jfs\_extent.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=15469c408af2d7a52fb186a92f2f091b0f13b1fb

extAlloc 和 extRecord 函数在修改文件系统前，未检查挂载是否为只读模式。调用 txBeginAnon() 会初始化一个事务，可能触发对只读存储的非法写入。在 txBeginAnon() 前添加 isReadOnly(ip) 检查。如果文件系统为只读，立即返回 -EIO

CVE-2024-58094

fs/jfs/inode.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f605bc3e162f5c6faa9bd3602ce496053d06a4bb

jfs\_truncate\_nolock() 在截断文件时，未检查文件系统是否为只读模式。如果文件系统为只读，JFS\_SBI(tblk->sb)->log 可能为 NULL，导致后续 txEnd() 操作崩溃。添加 isReadOnly(ip) 检查，与 COMMIT\_Nolink 共用同一逻辑分支。只读模式下调用 xtTruncate() 时使用 COMMIT\_WMAP 标志，避免实际写操作。

CVE-2024-58093

drivers/pci/pcie/aspm.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=cbf937dcadfd571a434f8074d057b32cd14fbea5

链路状态绑定到功能0设备，但允许通过任意功能触发释放。交换机上游端口的parent\_link在下游端口仍需引用时被释放。pciehp默认反向移除放大问题概率。修改后仅当功能0（虚拟P2P桥）移除时释放链路状态。确保所有下游端口处理完毕再释放上游链路。适应pciehp的反向移除机制。

CVE-2024-58092

fs/nfsd/nfs4recover.c

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=de71d4e211eddb670b285a0ea477a299601ce1ca

双重调用链：nfsd4\_client\_tracking\_init()调用check\_for\_legacy\_methods()，两者都会触发client\_tracking\_ops->init()。关键修改：责任分离：将初始化逻辑完全交给nfsd4\_client\_tracking\_init()，状态传递：通过返回-ENOTDIR触发上层统一初始化。

CVE-2024-58091

[drivers/gpu/drm/drm\_fbdev\_dma.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/drm_fbdev_dma.c?id=cdc581169942de3b9e2648cfbd98c5ff9111c2c8)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=cdc581169942de3b9e2648cfbd98c5ff9111c2c8

错误假设：认为所有DMA缓冲区都通过struct page管理。关键改进：影子缓冲机制：为延迟I/O单独分配vmalloc系统内存，通过drm\_fbdev\_dma\_damage\_blit()同步DMA与影子缓冲。操作分离：直接映射模式：继续使用DMA缓冲；延迟I/O模式：强制使用影子缓冲

CVE-2024-58090

[kernel/sched/core.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/kernel/sched/core.c?id=b927c8539f692fb1f9c2f42e6c8ea2d94956f921)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b927c8539f692fb1f9c2f42e6c8ea2d94956f921

错误逻辑：中断状态忽略：cond\_resched()未验证当前中断状态，仅依赖should\_resched()检测。关键修改：通过irqs\_disabled()显式阻止中断禁用环境下的重调度，保持与PREEMPT\_FULL模型的行为一致性

CVE-2024-58088 [kernel/bpf/bpf\_cgrp\_storage.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/kernel/bpf/bpf_cgrp_storage.c?id=fcec95b4ab3e7bc6b2f36e5d59f7e24104ea87f7)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fcec95b4ab3e7bc6b2f36e5d59f7e24104ea87f7

错误逻辑：fentry程序：先获取local\_storage->lock → 尝试获取smap->lock，释放路径：先持有smap->lock → 尝试获取local\_storage->lock。缺失保护：bpf\_cgrp\_storage\_busy计数器未被用于同步，允许fentry程序在存储释放过程中操作相同资源。关键修改：传递bpf\_cgrp\_storage\_busy计数器到释放流程，通过计数器阻止并发访问危险路径

CVE-2024-58086

[drivers/gpu/drm/v3d/v3d\_perfmon.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/v3d/v3d_perfmon.c?id=f8805b12f477bd964e2820a87921c7b58cc2dee3)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f8805b12f477bd964e2820a87921c7b58cc2dee3

核心缺陷：生命周期管理缺失：销毁时未验证目标perfmon是否为当前活动实例，

导致v3d->active\_perfmon成为悬垂指针。竞争条件：其他线程可能通过v3d->active\_perfmon访问已释放内存。关键修改：增加活动实例检查，通过v3d\_perfmon\_stop()安全解除活跃状态

CVE-2024-58085

[security/tomoyo/common.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/security/tomoyo/common.c?id=fe1c021eb03dae0dc9dce55e81f77a60e419a27a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fe1c021eb03dae0dc9dce55e81f77a60e419a27a

核心问题：警告敏感性：内核内存分配器会对"异常大"的分配请求发出警告，但TOMOYO的缓冲区扩展逻辑（每次翻倍）可能触发误报。实际约束：有效策略行通常远小于32KB，超长请求最终会被-ENOMEM拒绝，无需额外长度检查。关键修改：添加\_\_GFP\_NOWARN标志抑制无害警告，保持原有错误处理流程。

CVE-2024-58084

[drivers/firmware/qcom/qcom\_scm.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/firmware/qcom/qcom_scm.c?id=fee921e3c641f64185abee83f9a6e65f0b380682)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fee921e3c641f64185abee83f9a6e65f0b380682

核心缺陷：内存可见性问题：\_\_scm初始化时使用了写屏障（smp\_store\_release()），但读取路径缺少配对的读屏障，可能导致CPU缓存读取旧值。控制流依赖：条件判断\_\_scm ? ...形成控制依赖，但未明确保证内存顺序。关键修改：通过qcom\_scm\_is\_available()引入读屏障（smp\_load\_acquire()），确保\_\_scm的读取在初始化操作之后。

CVE-2024-58083 [include/linux/kvm\_host.h](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/include/linux/kvm_host.h?id=f2f805ada63b536bc192458a7098388286568ad4)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f2f805ada63b536bc192458a7098388286568ad4

核心缺陷：执行顺序错误：array\_index\_nospec()可能将非法索引转换为0,未优先验证i >= num\_vcpus导致可能返回未初始化的vCPU0。生命周期问题：vCPU0可能在创建失败后被释放，但通过此路径仍可访问。关键修改：严格区分"索引越界"和"防推测处理"逻辑，确保非法请求返回NULL而非vCPU0

CVE-2024-58082

[drivers/media/platform/nuvoton/npcm-video.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/media/platform/nuvoton/npcm-video.c?id=c4b7779abc6633677e6edb79e2809f4f61fde157)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=c4b7779abc6633677e6edb79e2809f4f61fde157

核心缺陷：API语义误解：of\_find\_device\_by\_node()失败返回NULL，而非错误指针，IS\_ERR()仅适用于返回ERR\_PTR()的函数。错误传播：错误路径返回0（PTR\_ERR(NULL)），可能被上层误认为成功。关键修改：改用!ece\_pdev检测失败，返回标准的-ENODEV错误码。

CVE-2024-58081

[drivers/clk/mmp/pwr-island.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/mmp/pwr-island.c?id=eca01d5911fb34218d10a58d8d9534b758c8fd0a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eca01d5911fb34218d10a58d8d9534b758c8fd0a

核心缺陷：时序依赖：pm\_genpd\_init()内部调用dev\_set\_name()，依赖genpd.name，但name字段在初始化之后才被赋值。调试FS影响：提交899f44531fe6后，genpd\_debug\_add()改用dev\_name()获取名称，暴露了原有隐患。关键修改：确保genpd.name在pm\_genpd\_init()前有效，保持与其他电源域初始化代码的一致性。

CVE-2024-58080 [drivers/clk/qcom/dispcc-sm6350.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/qcom/dispcc-sm6350.c?id=d4cdb196f182d2fbe336c968228be00d8c3fed05)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d4cdb196f182d2fbe336c968228be00d8c3fed05

核心缺陷：配置不一致：定义了父时钟（bi\_tcxo）但缺少对应的parent\_map，qcom\_find\_src\_index()依赖parent\_map查找父时钟索引。时钟树约束：RCG2（Rate-Controlled Clock Generator）必须同时配置：parent\_data（时钟源描述）parent\_map（硬件寄存器映射）。关键修改：补充缺失的parent\_map，将内联parent\_data改为预定义数组保持一致性

CVE-2024-58078

[drivers/char/misc.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/char/misc.c?id=8b4120b3e060e137eaa8dc76a1c40401088336e5)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=8b4120b3e060e137eaa8dc76a1c40401088336e5

核心缺陷：生命周期不匹配：非动态设备未通过IDA分配，但强制通过IDA释放。范围混淆：动态设备次设备号范围（0-127）与静态范围（128-1048575）处理逻辑割裂。关键修改：统一所有设备号的IDA分配路径。显式区分动态/静态设备号处理

CVE-2024-58077 [sound/soc/soc-pcm.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/sound/soc/soc-pcm.c?id=90778f31efdf44622065ebbe8d228284104bd26f)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=90778f31efdf44622065ebbe8d228284104bd26f

代码逻辑缺陷：soc\_pcm\_ret()函数统一处理所有PCM操作的返回值，对-EINVAL等非致命错误也记录为错误日志。.prepare回调中可能因用户空间非法参数合法返回-EINVAL，但原设计将其与真正的硬件错误同等对待。安全设计缺陷：未区分用户空间输入错误与硬件错误的安全影响层级。修复逻辑：绕过soc\_pcm\_ret()对.prepare回调的通用错误处理，直接返回原始错误码（包括-EINVAL），避免日志记录，在三个关键路径（\_\_soc\_pcm\_prepare/soc\_pcm\_prepare/dpcm\_fe\_dai\_prepare）应用相同修改

CVE-2024-58076 [drivers/clk/qcom/gcc-sm6350.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/qcom/gcc-sm6350.c?id=b6fe13566bf5676b1e3b72d2a06d875733e93ee6)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b6fe13566bf5676b1e3b72d2a06d875733e93ee6

直接原因： gcc\_ufs\_phy\_phy\_aux\_clk\_src和gcc\_usb30\_prim\_mock\_utmi\_clk\_src缺失parent\_map字段定义，当qcom\_find\_src\_index()函数尝试访问未初始化的parent\_map时触发内核崩溃。设计缺陷：违反高通时钟驱动设计规范：所有clk\_rcg2类型时钟必须同时定义parent\_map和parent\_data，原提交(131abae905df)未完整实现时钟父节点映射关系。修复逻辑：明确定义父时钟映射表（gcc\_parent\_map\_3），统一使用结构体数组方式声明parent\_data，确保映射表与数据声明的一致性

CVE-2024-58073 [drivers/gpu/drm/msm/disp/dpu1/dpu\_plane.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/msm/disp/dpu1/dpu_plane.c?id=789384eb1437aed94155dc0eac8a8a6ba1baf578)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=789384eb1437aed94155dc0eac8a8a6ba1baf578

直接原因：pipe->sspp指针在atomic\_print\_state()中被无条件访问，该指针仅在atomic\_check()中初始化（commit 31f7148fd370引入的变更）。架构缺陷：状态打印函数未考虑中间状态有效性，缺乏对管道(pipe)与硬件层(sspp)绑定状态的验证。修复特点：保持与右管道(r\_pipe)相同的防护模式。不影响正常功能路径的性能。

CVE-2024-58071 [drivers/net/team/team\_core.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/team/team_core.c?id=d9bce1310c0e2a55888e3e08c9f69d8377b3a377)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d9bce1310c0e2a55888e3e08c9f69d8377b3a377

直接原因：缺少对网络设备层级关系的检查，允许将下层接口重复添加到team，在team\_device\_event()和team\_add\_slave()中形成锁的循环等待：team\_add\_slave()获取锁 → 触发设备事件通知 → team\_device\_event()尝试重复获取同一锁。架构缺陷：未遵循网络设备层级管理的基本原则，锁机制设计未考虑递归调用场景。修复逻辑：使用netdev\_has\_upper\_dev()检查设备层级关系，提前返回-EBUSY避免后续锁冲突，提供清晰的用户空间错误反馈。

CVE-2024-58070

[kernel/bpf/bpf\_local\_storage.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/kernel/bpf/bpf_local_storage.c?id=c1d398a3af7e59d7fef351c84fed7ebb575d1f1a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=c1d398a3af7e59d7fef351c84fed7ebb575d1f1a

核心问题：PREEMPT\_RT模式下，kmalloc(GFP\_ATOMIC)仍可能触发调度（通过RT spinlock），原有代码未强制使用BPF内存分配器（bpf\_mem\_alloc）处理RT场景。技术细节：bpf\_local\_storage\_update() → bpf\_selem\_alloc() → 在RT内核中尝试kmalloc(GFP\_ATOMIC) → 触发RT spinlock → 调度检查导致警告。修复逻辑：检测PREEMPT\_RT配置状态，无条件启用bpf\_mem\_alloc分配器，保持原生内核的原有选择逻辑。

CVE-2024-58069 [drivers/rtc/rtc-pcf85063.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/rtc/rtc-pcf85063.c?id=e5e06455760f2995b16a176033909347929d1128)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e5e06455760f2995b16a176033909347929d1128

接口不匹配：NVMEM接口支持可变长度缓冲区（可小至1字节），底层regmap\_read()强制要求4字节(unsigned int)存储空间。修复特点：添加中间缓冲区隔离风险，保持NVMEM接口的字节粒度支持，不改变原有功能逻辑。

CVE-2024-58068

[drivers/opp/core.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/opp/core.c?id=ff2def251849133be6076a7c2d427d8eb963c223)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ff2def251849133be6076a7c2d427d8eb963c223

find\_key\_ceil/floor() 通用框架允许为每种 key（freq、clk、volt、bw…）挂接断言函数 assert\_key\_index()，但 bandwidth key 的调用者（dev\_pm\_opp\_find\_bw\*）未挂接任何断言函数，而是传 NULL。结果：index 被直接用于 opp\_table->bandwidth\_table[index]，而 bandwidth\_table 为 NULL 时，index 被当作偏移量访问空指针 +4（即 0x4），触发崩溃。新增 assert\_bandwidth\_index()：判断 opp\_table->path\_count > index，避免越界。修改 dev\_pm\_opp\_find\_bw\_ceil/floor()：将 assert\_bandwidth\_index 作为 assert 回调传入 \_find\_key\_ceil/floor()。代码改动小、无副作用，遵循 OPP 框架已存在的 assert\_key\_index 设计模式。

CVE-2024-58067

[drivers/clk/mmp/clk-pxa1908-mpmu.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/mmp/clk-pxa1908-mpmu.c?id=7def56f841af22e07977e193eea002e085facbdb)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=7def56f841af22e07977e193eea002e085facbdb

API误用：devm\_kzalloc() 是内核内存分配函数，失败时返回 NULL，但原代码错误地假设其返回错误指针（如 ERR\_PTR(-ENOMEM)），并调用 IS\_ERR() 和 PTR\_ERR() 进行检查。逻辑矛盾：IS\_ERR() 仅用于检查错误指针（高位地址），而 NULL 的数值为 0，必然通过检查，导致后续 PTR\_ERR(NULL) 返回无效错误码（通常为 0），掩盖真实的 -ENOMEM 错误。修正检查逻辑：将 IS\_ERR(pxa\_unit) 改为 !pxa\_unit，直接检测 NULL。返回正确错误码：显式返回 -ENOMEM 以反映内存分配失败。一致性改进：补丁与内核其他使用 devm\_kzalloc() 的代码风格保持一致。

CVE-2024-58066

[drivers/clk/mmp/clk-pxa1908-apbcp.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/mmp/clk-pxa1908-apbcp.c?id=3acea81be689b77b3ceac6ff345ff0366734d967)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=3acea81be689b77b3ceac6ff345ff0366734d967

API语义混淆：devm\_kzalloc() 是内核资源托管的内存分配函数，其失败时返回 NULL，但开发者错误地假设其返回错误指针（如 ERR\_PTR(-ENOMEM)），导致误用 IS\_ERR() 和 PTR\_ERR() 进行检查。错误处理失效：当内存分配失败时，原代码会错误地将 NULL 传递给 PTR\_ERR()，返回 0（成功状态码）而非正确的 -ENOMEM，掩盖了真实的错误。修正检查逻辑：将 IS\_ERR(pxa\_unit) 改为显式的 !pxa\_unit 检查，直接检测 NULL。返回正确错误码：明确返回 -ENOMEM 以反映内存分配失败，符合内核错误处理规范。代码一致性：与内核其他使用 devm\_kzalloc() 的模块保持统一风格。

CVE-2024-58065 [drivers/clk/mmp/clk-pxa1908-apbc.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/clk/mmp/clk-pxa1908-apbc.c?id=e5ca5d7b4d7c29246d957dc45d63610584ae3a54)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e5ca5d7b4d7c29246d957dc45d63610584ae3a54

API误用：devm\_kzalloc() 是内核提供的托管内存分配函数，其失败时返回 NULL，但开发者错误地将其与返回错误指针（如 ERR\_PTR(-ENOMEM)）的函数混淆，导致误用 IS\_ERR() 进行检查。错误处理失效：当内存分配失败时，原代码会错误地将 NULL 传递给 PTR\_ERR()，返回 0（表示成功）而非正确的 -ENOMEM，掩盖了真实的错误状态。修正检查逻辑：将 IS\_ERR(pxa\_unit) 改为显式的 !pxa\_unit 检查，直接检测 NULL。正确错误码：明确返回 -ENOMEM 以准确反映内存分配失败。代码一致性：与内核中其他使用 devm\_kzalloc() 的正确模式保持一致。

CVE-2024-58064

[net/wireless/tests/scan.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/wireless/tests/scan.c?id=886271409603956edd09df229dde7442c410a872)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=886271409603956edd09df229dde7442c410a872

内存分配未验证：kunit\_kzalloc()可能因内存不足返回NULL，但原始代码直接解引用返回的指针。测试代码安全性疏忽：虽然是测试代码，但仍需遵循内核安全编程规范。Kunit测试特殊性：测试环境可能模拟内存压力场景，增加了触发概率。添加明确的NULL检查：使用KUNIT\_ASSERT\_NOT\_NULL宏验证分配结果。遵循Kunit最佳实践：确保测试代码本身的健壮性。保持代码一致性：与内核其他Kunit测试用例风格统一

CVE-2024-58063 [drivers/net/wireless/realtek/rtlwifi/pci.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/realtek/rtlwifi/pci.c?id=ee0b0d7baa8a6d42c7988f6e50c8f164cdf3fa47)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ee0b0d7baa8a6d42c7988f6e50c8f164cdf3fa47

资源释放顺序错误：原始代码在init\_sw\_vars失败时直接跳转到fail3标签，跳过了必要的deinit\_sw\_vars调用。无效工作队列访问：rtl\_deinit\_core会销毁工作队列，但在某些错误路径中被错误调用。冗余的pci\_set\_drvdata：PCI核心驱动已处理设备数据清理，重复调用可能导致内存管理问题。调整错误处理路径：新增fail4和fail5标签实现分阶段清理。确保deinit\_sw\_vars在所有错误路径中被调用。移除冗余清理：删除不必要的pci\_set\_drvdata(NULL)调用。严格的反初始化顺序：按照资源分配的逆序进行释放。

CVE-2024-58062

[drivers/net/wireless/intel/iwlwifi/mvm/coex.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/intel/iwlwifi/mvm/coex.c?id=fbb563ad5032a07ac83c746ce5c8de5f25b5ffd0)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fbb563ad5032a07ac83c746ce5c8de5f25b5ffd0

不安全的链路遍历：原始代码使用硬编码循环(IEEE80211\_MLD\_MAX\_NUM\_LINKS)遍历所有可能的链路，而未检查vif->link\_conf[link\_id]是否为NULL。多链路(Multi-Link)操作风险：在802.11be(Wi-Fi 7)多链路设备中，并非所有链路都始终处于活动状态。RCU保护不足：虽然使用了rcu\_dereference\_check，但未先验证指针有效性。使用安全迭代宏：替换为for\_each\_vif\_active\_link，该宏自动过滤非活动链路。简化代码逻辑：移除硬编码循环和显式的RCU解引用检查。保持锁一致性：仍通过lockdep\_is\_held(&mvm->mutex)确保锁安全。

CVE-2024-58061 [net/mac80211/debugfs\_netdev.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/mac80211/debugfs_netdev.c?id=dfe9a043300261afe5eadc07b867a6810c4e999a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=dfe9a043300261afe5eadc07b867a6810c4e999a

协议违反：802.11标准要求MLO设备至少保持一个活动链路（主链路）。不充分的输入验证：ieee80211\_if\_parse\_active\_links()未检查active\_links参数是否为0。调试接口暴露风险：通过debugfs暴露的关键功能缺乏必要的保护。输入验证增强：在kstrtou16()检查后增加!active\_links条件。错误码返回：返回-EINVAL拒绝全链路禁用请求。保持内部断言：保留内部的WARN\_ON用于捕捉编程错误。

CVE-2024-58059 [drivers/media/usb/uvc/uvc\_status.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/media/usb/uvc/uvc_status.c?id=db577ededf3a18b39567fc1a6209f12a0c4a3c52)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=db577ededf3a18b39567fc1a6209f12a0c4a3c52

初始化顺序问题：uvc\_status\_init()与uvc\_status\_unregister()的调用存在时序依赖，但错误路径未正确处理。空指针解引用风险：在dev->status未初始化时调用uvc\_status\_suspend()可能导致内核崩溃。竞态条件：修复的提交c5fe3ed618f9引入的注销逻辑未完全考虑初始化失败场景。添加NULL检查：在uvc\_status\_unregister()入口处验证dev->status有效性。安全返回机制：若状态子系统未初始化，则直接返回而非继续执行注销操作。保持原有逻辑：对已初始化的case保持原有的挂起和输入设备注销流程。

CVE-2024-58058 [fs/ubifs/debug.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/fs/ubifs/debug.c?id=e01b55f261ccc96e347eba4931e4429d080d879d)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e01b55f261ccc96e347eba4931e4429d080d879d

生命周期管理缺陷：slab缓存清理会释放所有znode内存，但未同步更新c->zroot.znode指针状态。调试接口安全缺失：ubifs\_dump\_tnc()未对c->zroot.znode进行NULL检查即直接解引用。内存管理时序问题：缓存释放与调试功能之间存在竞态条件。添加NULL检查：在执行TNC遍历前验证c->zroot.znode有效性。安全失败处理：当TNC树为空时输出明确提示而非崩溃。保持调试功能：对正常情况维持原有的树遍历逻辑。

CVE-2024-58057 [drivers/net/ethernet/intel/idpf/idpf\_main.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/ethernet/intel/idpf/idpf_main.c?id=9a5b021cb8186f1854bac2812bd4f396bb1e881c)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=9a5b021cb8186f1854bac2812bd4f396bb1e881c

CPU绑定风险：原工作队列绑定到特定CPU，无法应对CPU资源竞争。缺乏弹性调度：当绑定CPU被高优先级进程独占时（如通过chrt设置实时优先级），驱动任务无法及时执行。关键路径阻塞：网络驱动的工作队列延迟直接影响数据包处理，可能引发协议栈超时。工作队列模式变更：为所有工作队列添加WQ\_UNBOUND标志：alloc\_workqueue(..., WQ\_UNBOUND | WQ\_MEM\_RECLAIM, ...) 保留内存回收能力：同时保留WQ\_MEM\_RECLAIM标志确保内存压力下的可靠性。NUMA感知：非绑定工作队列仍保持在相同NUMA节点内执行，兼顾性能与弹性。

CVE-2024-58056

[drivers/remoteproc/remoteproc\_core.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/remoteproc/remoteproc_core.c?id=f2013d19b7704cd723ab42664b8d9408ea8cc77c)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f2013d19b7704cd723ab42664b8d9408ea8cc77c

初始化顺序缺陷：rproc->index被初始化为0，但IDA分配可能尚未执行。错误路径处理不当：在rproc\_alloc()的早期错误路径中，put\_device()会触发释放回调，而此时ida\_alloc()可能还未被调用。条件检查失效：rproc\_type\_release()中(rproc->index >= 0)的条件总是成立（因默认值为0），导致错误释放未分配的IDA资源。提前IDA分配：将ida\_alloc()移到rproc\_alloc()的最早可能阶段。确保资源对称性：在错误路径中，只有成功分配的索引才会被释放。保持功能不变：不改变原有成功路径的逻辑。

CVE-2024-58055 [drivers/usb/gadget/function/f\_tcm.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/usb/gadget/function/f_tcm.c?id=f0c33e7d387ccbb6870e73a43c558fefede06614)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f0c33e7d387ccbb6870e73a43c558fefede06614

生命周期管理错误：transport\_send\_check\_condition\_and\_sense()会异步完成命令状态通知，但原始代码立即同步释放命令对象。Kref引用计数问题：未正确处理TARGET\_SCF\_ACK\_KREF引用计数，导致命令对象被过早释放。竞态条件：状态完成回调可能访问已释放的命令对象内存。移除提前释放：删除transport\_generic\_free\_cmd()的显式调用。依赖标准生命周期：让transport\_send\_check\_condition\_and\_sense()的完成回调负责最终释放。保持错误处理：仍通过SCSI检查条件报告错误。

CVE-2024-58054

[drivers/staging/media/max96712/max96712.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/staging/media/max96712/max96712.c?id=ee1b5046d5cd892a0754ab982aeaaad3702083a5)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ee1b5046d5cd892a0754ab982aeaaad3702083a5

指针管理冲突：v4l2\_i2c\_subdev\_init()内部调用i2c\_set\_clientdata()将数据指针重置为v4l2\_subdev结构体，覆盖了驱动原本设置的私有数据指针。生命周期不一致：在probe()中显式设置的priv指针(i2c\_set\_clientdata(client, priv))被后续初始化覆盖。移除路径假设错误：remove()函数假设客户端数据仍指向私有结构，实际却指向了子设备结构。移除冗余设置：删除probe()中的i2c\_set\_clientdata(client, priv)调用。正确指针获取：在remove()中通过container\_of从子设备指针安全获取私有数据。保持功能一致：不改变正常的子设备注册/注销流程。

CVE-2024-58052 [drivers/gpu/drm/amd/pm/powerplay/hwmgr/ppatomctrl.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/amd/pm/powerplay/hwmgr/ppatomctrl.c?id=c47066ed7c8f3b320ef87fa6217a2b8b24e127cc)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=c47066ed7c8f3b320ef87fa6217a2b8b24e127cc

缺失NULL检查：smu\_atom\_get\_data\_table()可能返回NULL，但代码直接解引用其返回的psmu\_info指针。硬件假设风险：虽然Polaris显卡的VBIOS理论上应始终包含该表，但未考虑异常情况（如损坏的VBIOS）。防御性编程不足：关键数据表访问路径缺乏错误处理。添加NULL检查：在解引用psmu\_info前验证指针有效性。返回错误码：当表获取失败时返回-EINVAL。保持功能逻辑：成功路径保持不变。

CVE-2024-58051

[drivers/char/ipmi/ipmb\_dev\_int.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/char/ipmi/ipmb_dev_int.c?id=eb288ab33fd87579789cb331209ff09e988ff4f7)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eb288ab33fd87579789cb331209ff09e988ff4f7

API误用：devm\_kasprintf()失败时返回NULL，但驱动未验证返回值。资源管理缺陷：直接使用未验证的指针作为misc设备名称。防御性编程缺失：关键内存分配路径缺乏错误处理。添加NULL检查：在devm\_kasprintf()后立即验证返回值。返回错误码：内存分配失败时返回-ENOMEM。保持功能一致：成功路径逻辑不变。

CVE-2024-58042 [lib/rhashtable.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/lib/rhashtable.c?id=eb2e58484b838fb4e777ee9721bb9e20e6ca971d)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eb2e58484b838fb4e777ee9721bb9e20e6ca971d

锁顺序违规：原始代码在持有rhashtable桶锁时调用schedule\_work()，而工作队列可能反向获取调度器锁（如rq->lock），形成桶锁 -> wq锁 -> rq锁 -> 桶锁的潜在循环依赖。原子性过度：atomic\_inc()和增长检查本不需在桶锁保护下执行。扩展性限制：严格的锁范围阻碍了rhashtable在调度器等复杂上下文的使用。锁范围优化：将atomic\_inc()和rht\_grow\_above\_75()检查移到桶锁释放后。条件调度：仅当插入成功（ENOENT）且无需表扩容（!new\_tbl）时才触发工作队列。安全保证：通过PTR\_ERR(data) == -ENOENT验证插入成功状态。

CVE-2024-58034 [drivers/memory/tegra/tegra20-emc.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/memory/tegra/tegra20-emc.c?id=e9d07e91de140679eeaf275f47ad154467cb9e05)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e9d07e91de140679eeaf275f47ad154467cb9e05

不安全的节点遍历：原代码使用of\_find\_node\_by\_name()遍历节点，该函数会释放前一个节点的引用，导致引用计数管理混乱。过早释放：在配置不匹配时错误调用of\_node\_put(np)，而此时节点仍可能被后续代码使用。API误用：未遵循设备树节点引用计数的"获取-释放"对称原则。安全遍历API：改用for\_each\_child\_of\_node()宏，自动管理节点引用。精确子节点获取：使用of\_get\_child\_by\_name()替代of\_find\_node\_by\_name()获取子节点。移除危险释放：删除配置不匹配时的错误of\_node\_put(np)调用。

CVE-2024-58022 [drivers/mailbox/mailbox-th1520.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/mailbox/mailbox-th1520.c?id=ecbde88e544ff016fa08bbf2156dc431bb123e9b)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ecbde88e544ff016fa08bbf2156dc431bb123e9b

API语义混淆：devm\_ioremap()失败时返回NULL，但代码错误地假设其返回错误指针（如ERR\_PTR(-ENOMEM)）。错误处理失效：当内存映射失败时，原代码会错误地将NULL传递给IS\_ERR()，导致错误路径未被正确触发。资源泄漏风险：可能掩盖真实的内存映射失败，导致后续操作访问非法地址。修正错误检查：将IS\_ERR(mapped)改为!mapped，直接检测NULL。正确错误码返回：映射失败时返回ERR\_PTR(-ENOMEM)以保持接口一致性。改进错误日志：保留原有的错误消息输出。

CVE-2024-58021 [drivers/hid/hid-winwing.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/hid/hid-winwing.c?id=b99dbdee8a89c44d03ae9830ab19f31e124a3f32)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b99dbdee8a89c44d03ae9830ab19f31e124a3f32

缺失NULL检查：devm\_kasprintf()在内存不足时返回NULL，但驱动直接使用返回值作为LED名称。资源管理缺陷：未考虑内存分配失败路径，违反防御性编程原则。硬件特定：影响WinWing Orion2油门控制器的LED初始化流程。添加NULL检查：在devm\_kasprintf()后立即验证返回值。返回错误码：内存分配失败时返回-ENOMEM。保持功能一致：成功路径逻辑不变。

CVE-2024-58020  [drivers/hid/hid-multitouch.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/hid/hid-multitouch.c?id=aa879ef6d3acf96fa2c7122d0632061d4ea58d48)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=aa879ef6d3acf96fa2c7122d0632061d4ea58d48

缺失NULL检查：devm\_kasprintf()在内存不足时返回NULL，但驱动直接使用返回值作为输入设备名称。资源管理缺陷：未考虑内存分配失败路径，违反防御性编程原则。影响范围：涉及所有使用HID多指触控的设备（如触摸屏、触控板）。添加NULL检查：在devm\_kasprintf()后立即验证返回值。返回错误码：内存分配失败时返回-ENOMEM。条件优化：仅在suffix存在时才执行分配。

CVE-2024-58019

[drivers/gpu/drm/nouveau/nvkm/subdev/gsp/r535.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/nouveau/nvkm/subdev/gsp/r535.c?id=8d9beb4aebc02c4bd09e1d39c9c5f1c68c786dbc)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=8d9beb4aebc02c4bd09e1d39c9c5f1c68c786dbc

消息长度计算错误：原代码未考虑消息头大小，错误计算需要复制的数据长度.。指针推进逻辑缺陷：非回滚情况：错误地将前一条消息体当作下一条消息头，回滚情况：分两次拷贝时未正确计算总页数，导致指针过度推进。边界条件处理不足：未正确处理vGPU启用时的两页消息场景。正确计算：包含消息头（GSP\_MSG\_HDR\_SIZE）的总长度。统一指针推进逻辑：在所有拷贝完成后一次性计算推进量。使用模运算保证循环：rptr = (rptr + pages) % gsp->msgq.cnt

CVE-2024-58018

[drivers/gpu/drm/nouveau/nvkm/subdev/gsp/r535.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/nouveau/nvkm/subdev/gsp/r535.c?id=6b6b75728c86f60c1fc596f0d4542427d0e6065b)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=6b6b75728c86f60c1fc596f0d4542427d0e6065b

缓冲区空间计算错误：原代码仅基于写指针和缓冲区大小计算可用空间，未考虑环形缓冲区的循环特性。竞争条件风险：在等待可用空间时，错误的空间计算可能导致与GSP处理器的读写竞争。数据覆盖风险：可能覆盖GSP正在读取的RPC请求，造成命令解析错误。精确空间计算：引入step变量正确计算当前可写入的页数。安全拷贝策略：基于实际可用空间（step \* GSP\_PAGE\_SIZE）确定拷贝大小。保持功能一致：不改变原有的等待和通知机制。

CVE-2024-58017

[kernel/printk/printk.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/kernel/printk/printk.c?id=dfb7b179741ee09506dc7719d92f9e1cea01f10e)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=dfb7b179741ee09506dc7719d92f9e1cea01f10e

有符号整数溢出：在32位系统上，1 << 31会尝试将0x80000000赋值给有符号整型，这超出了INT\_MAX范围。未定义行为：根据C标准，有符号整数溢出属于未定义行为，可能导致不可预测的结果。平台依赖风险：不同编译器对这种情况的处理可能不一致。强制无符号运算：将1显式转换为u32类型后再进行位移操作。保持语义不变：仍保持2^31的原始设计意图。语法优化：简化类型转换的书写方式。

CVE-2024-58016

[security/safesetid/securityfs.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/security/safesetid/securityfs.c?id=f09ff307c7299392f1c88f763299e24bc99811c7)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f09ff307c7299392f1c88f763299e24bc99811c7

缺少输入验证：handle\_policy\_update()函数未检查用户提供的写入长度，可能触发超大内存分配。内存分配风险：直接使用未经验证的len参数可能导致kmalloc警告或失败。DoS潜在性：恶意用户可能通过写入超大缓冲区消耗系统资源。添加长度检查：在内存分配前验证len < KMALLOC\_MAX\_SIZE。错误码返回：对于过大请求返回-EINVAL。防御性编程：保持原有的内存分配错误检查。

CVE-2024-58015

[drivers/net/wireless/ath/ath12k/debugfs\_htt\_stats.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/ath/ath12k/debugfs_htt_stats.c?id=eb8c0534713865d190856f10bfc97cf0b88475b1)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eb8c0534713865d190856f10bfc97cf0b88475b1

print\_array\_to\_buf\_index() 原型：print\_array\_to\_buf\_index(buf, offset, prefix, start\_idx, arr, len, suffix)其中 len 表示要打印的元素个数，start\_idx=1（宏写死），所以有效下标应为 1 … N-1，共 N-1 个元素。调用者却直接传宏值 ATH12K\_HTT\_TX\_NUM\_\*\_USER\_STATS（= N），导致读取 arr[N]（越界 1 字节）。补丁将 print\_array\_to\_buf\_index() 的 len 参数减 1：ATH12K\_HTT\_TX\_NUM\_AC\_MUMIMO\_USER\_STATS – 1ATH12K\_HTT\_TX\_NUM\_AX\_MUMIMO\_USER\_STATS – 1确保访问下标 1 … N-1，完全落在数组边界内。无功能侧移，仅改变边界值；静态代码缺陷消除，Coverity CID 1600742 / 1600758 关闭。

CVE-2024-58014

[drivers/net/wireless/broadcom/brcm80211/brcmsmac/phy/phy\_n.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/broadcom/brcm80211/brcmsmac/phy/phy_n.c?id=d280a12e9b87819a8a209639d600b48a2d6d65dc)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d280a12e9b87819a8a209639d600b48a2d6d65dc

循环 for (k = 0; k < NPHY\_IQCAL\_NUMGAINS; k++) 仅保证 k < NPHY\_IQCAL\_NUMGAINS，但若所有条目均不满足 if 条件，则 k 退出循环时等于 NPHY\_IQCAL\_NUMGAINS。代码随后直接用 k 作为索引访问 tbl\_iqcal\_gainparams\_nphy[band\_idx][k][1/2/3]，此时已越界 1 个元素。该路径在极端硬件/校准配置下可能触发；静态分析（Smatch/Coverity）标记为“array overrun”。在循环结束后插入：if (WARN\_ON(k == NPHY\_IQCAL\_NUMGAINS)) return;当 k 越界时打印 WARN 并提前返回，避免后续数组访问。WARN\_ON 既作为运行时防护，也帮助开发者收集异常硬件配置。无功能侧移：若 k 未越界，行为与旧代码完全一致。

CVE-2024-58013

[net/bluetooth/mgmt.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/bluetooth/mgmt.c?id=ebb90f23f0ac21044aacf4c61cc5d7841fe99987)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ebb90f23f0ac21044aacf4c61cc5d7841fe99987

mgmt\_pending\_cmd 采用“命令挂起队列 + 完成回调”异步模型。回调路径 mgmt\_remove\_adv\_monitor\_complete() 会调用 mgmt\_pending\_remove() 释放 cmd 对象，但 mgmt\_remove\_adv\_monitor\_sync()（工作队列上下文）后续仍继续访问 cmd->param，出现 use-after-free。缺少在解引用前通过 pending\_find() 重新验证 cmd 是否仍存在于挂起队列。在 mgmt\_remove\_adv\_monitor\_complete() 与 mgmt\_remove\_adv\_monitor\_sync() 开头使用 pending\_find() 检查 cmd 是否仍然有效；若已不存在，直接返回 -ECANCELED，避免后续 cmd->param 解引用。重新排序代码：先检查状态再取 cmd->param，消除竞态窗口。补丁仅增加两条 pending\_find() 调用，无 ABI 或功能变更。

CVE-2024-58010 [fs/binfmt\_flat.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/fs/binfmt_flat.c?id=d17ca8f2dfcf423c439859995910a20e38b86f00)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d17ca8f2dfcf423c439859995910a20e38b86f00

load\_flat\_file() 原校验只检查 text\_len、data\_len、bss\_len、stack\_len、full\_data 的高 4 位（>>28），若任一 ≥ 256 MB 即拒绝。但 relocs 本身未被检查，其值可以 ≥ 0x10000000 / 4（≈ 64 M 项），使得乘法溢出后 full\_data 看起来远小于 256 MB，从而绕过检查。后续执行重定位写入时，真正需要的内存空间（data\_len + relocs \* 4）已超出映射区，产生越界写。将 relocs 加入位掩码校验：if ((text\_len | data\_len | bss\_len | stack\_len | relocs | full\_data) >> 28)，一旦 relocs ≥ 256 MB 或导致 full\_data ≥ 256 MB 立即拒绝加载。补丁简单、无副作用，直接阻断溢出路径。

CVE-2024-58009

[net/bluetooth/l2cap\_sock.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/bluetooth/l2cap_sock.c?id=cf601a24120c674cd7c907ea695f92617af6abd0)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=cf601a24120c674cd7c907ea695f92617af6abd0

l2cap\_sock\_alloc() 的调用点之一（l2cap\_sock\_new\_connection\_cb）传入 sock = NULL。当 l2cap\_chan\_create() 返回 NULL 时，原代码直接 sock->sk = NULL，未检查 sock 有效性。该路径仅在内存不足或内部失败时出现，触发概率低但必然崩溃。在错误处理分支中增加 if (sock) 判断，避免对 NULL 的写操作。补丁保持函数调用顺序不变，确保不会在全局列表中留下半初始化通道，兼顾正确性与简洁性。

CVE-2024-58008

[security/keys/trusted-keys/trusted\_dcp.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/security/keys/trusted-keys/trusted_dcp.c?id=e8d9fab39d1f87b52932646b2f1e7877aa3fc0f4)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e8d9fab39d1f87b52932646b2f1e7877aa3fc0f4

trusted\_dcp\_seal()/trusted\_dcp\_unseal() 原先使用u8 plain\_blob\_key[AES\_KEYSIZE\_128];该数组位于 vmalloc 栈（CONFIG\_VMAP\_STACK=y）时，物理地址不连续，无法直接用于 DMA。sg\_init\_one(&sg, plain\_blob\_key, AES\_KEYSIZE\_128) 生成的 scatterlist 指向 vmalloc 地址；DCP 引擎尝试 DMA-mapping 时失败，后续代码路径解引用无效 dma\_addr 触发 panic。0e28bf61a5f9 修复了密钥泄漏，却未解决 vmalloc 栈不可 DMA 的基本约束。将 plain\_blob\_key 改为 kmalloc 分配：plain\_blob\_key = kmalloc(AES\_KEYSIZE\_128, GFP\_KERNEL);kmalloc 区域保证物理连续且可 DMA。在错误路径和正常出口显式 kfree() 并 memzero\_explicit()，防止泄漏。补丁小、无副作用，完全消除 vmalloc 栈 DMA 不兼容问题。

CVE-2024-58007 [drivers/soc/qcom/socinfo.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/soc/qcom/socinfo.c?id=9c88b3a3fae4d60641c3a45be66269d00eff33cd)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=9c88b3a3fae4d60641c3a45be66269d00eff33cd

边界检查缺陷：原代码使用offsetof()仅检查字段起始位置是否在边界内，未考虑字段长度。硬件兼容性问题：MSM8916的SOCINFO\_VERSION(0,8)不包含serial\_num字段，但驱动仍尝试读取。内存安全违规：实际读取的是socinfo结构体之后的未定义内存区域。正确边界检查：改用offsetofend()确保整个字段在有效范围内。安全数据访问：仅当SOCINFO版本确实包含serial\_num时才读取。保持兼容性：不影响支持该字段的SoC型号。

CVE-2024-58006 [drivers/pci/controller/dwc/pcie-designware-ep.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/pci/controller/dwc/pcie-designware-ep.c?id=b5cacfd067060c75088363ed3e19779078be2755)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b5cacfd067060c75088363ed3e19779078be2755

缺失配置验证：dw\_pcie\_ep\_set\_bar()允许动态更新BAR物理地址，但未检查新size/flags是否匹配现有配置。内存安全风险：若新BAR尺寸小于原配置，主机可能访问超出新映射范围的内存。信任边界突破：违反PCIe规范中BAR配置不可变的要求（除地址外）。严格配置验证：检查新BAR的barno/size/flags必须与原配置完全一致。安全更新逻辑：仅允许更新物理地址，其他属性变更返回-EINVAL。代码注释增强：明确动态更新BAR的限制条件。

CVE-2024-58005

[drivers/char/tpm/eventlog/acpi.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/char/tpm/eventlog/acpi.c?id=a676c0401de59548a5bc1b7aaf98f556ae8ea6db)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=a676c0401de59548a5bc1b7aaf98f556ae8ea6db

大内存分配限制：devm\_kmalloc()底层使用\_\_alloc\_pages()，无法处理16MB级别的连续内存请求（order=12）。资源管理缺陷：原实现依赖设备管理内存自动释放，但无法适应大内存场景。硬件兼容性问题：特定HPE ProLiant设备需要大容量TPM日志缓冲区。内存分配策略升级：使用kvmalloc()替代devm\_kmalloc()，支持非连续大内存，添加devm\_add\_action()确保资源释放。新增释放函数：tpm\_bios\_log\_free()专用于释放日志缓冲区。错误处理增强：完善分配失败时的清理路径。

CVE-2024-58004 [drivers/media/pci/intel/ipu6/ipu6-isys.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/media/pci/intel/ipu6/ipu6-isys.c?id=facb541ff0805314e0b56e508f7d3cbd07af513c)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=facb541ff0805314e0b56e508f7d3cbd07af513c

资源释放缺失：isys\_probe()在错误路径中未调用cpu\_latency\_qos\_remove\_request()。链表一致性破坏：未移除的QoS请求导致后续链表操作（如plist\_add）访问无效内存。内存生命周期问题：已释放的isys结构体仍被QoS子系统引用。正确资源清理：在out\_remove\_pkg\_dir\_shared\_buffer错误路径中添加cpu\_latency\_qos\_remove\_request()调用。保持对称性：与正常移除路径（如驱动卸载）保持一致。防御性编程：确保所有错误路径都清理已分配资源。

CVE-2024-58001

[fs/ocfs2/symlink.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/fs/ocfs2/symlink.c?id=cd3e22b206189cbb4a94229002141e1529f83746)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=cd3e22b206189cbb4a94229002141e1529f83746

锁泄漏：在bh->b\_data读取失败时直接返回，未调用unlock\_page()。错误处理不完整：未遵循"获取资源-使用资源-释放资源"的对称模式。影响范围：涉及快速符号链接（fast symlink）的读取场景。统一错误路径：通过goto out跳转到公共清理代码。确保锁释放：无论读取成功与否都执行unlock\_page()。保持语义不变：正确返回原始错误码（status）。

CVE-2024-58000 [io\_uring/io\_uring.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/io_uring/io_uring.c?id=2a6de94df7bfa76d9850443547e7b3333f63a16a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=2a6de94df7bfa76d9850443547e7b3333f63a16a

未过滤的用户输入：用户控制的偏移量直接用于内核数组访问。推测执行风险：CPU可能基于恶意偏移量进行推测性内存访问。边界检查不足：虽然验证了偏移量+结构体大小不越界，但未防止推测执行期间的越界访问。引入安全索引：使用array\_index\_nospec()限制推测执行的访问范围。正确边界计算：基于ctx->cq\_wait\_size - sizeof(struct io\_uring\_reg\_wait)计算最大安全偏移。保持功能不变：合法访问路径不受影响。

CVE-2024-57998 [io\_uring/io\_uring.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/io_uring/io_uring.c?id=2a6de94df7bfa76d9850443547e7b3333f63a16a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eb6ffa0192ba83ece1a318b956265519c5c7dcec

未防护的用户输入：用户控制的偏移量直接用于内核数组（ctx->cq\_wait\_arg）访问。现代CPU特性：推测执行可能基于恶意偏移预取内存，即使最终检查会拒绝该访问。边界条件特殊：需同时考虑结构体大小（io\_uring\_reg\_wait）和总区域大小（cq\_wait\_size）。推测执行防护：使用array\_index\_nospec()确保CPU不推测越界地址。精确边界计算：cq\_wait\_size - sizeof(io\_uring\_reg\_wait)作为最大安全偏移。防御性编程：保持原有边界检查的同时增加微架构防护。

CVE-2024-57997 [drivers/net/wireless/ath/wcn36xx/main.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/ath/wcn36xx/main.c?id=e95f9c408ff8311f75eeabc8acf34a66670d8815)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e95f9c408ff8311f75eeabc8acf34a66670d8815

原始代码使用 devm\_kmalloc(wcn->dev, n\_channels, GFP\_KERNEL) 分配内存，但错误地仅分配了 n\_channels 个字节，而非 n\_channels \* sizeof(struct wcn36xx\_chan\_survey)。这会导致：内存缓冲区溢出（如果实际写入数据超过分配的大小）。使用未初始化的内存（因分配空间不足，部分结构体成员可能未被覆盖）。根本原因：未考虑结构体大小，直接以通道数量（n\_channels）作为字节数分配内存。修复方式：改用 devm\_kcalloc，明确指定元素数量和每个元素的大小。修复效果：确保分配的内存足够容纳所有 wcn36xx\_chan\_survey 结构体，并初始化为零。

CVE-2024-57996 [net/sched/sch\_sfq.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/sched/sch_sfq.c?id=833e9a1c27b82024db7ff5038a51651f48f05e5e)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=833e9a1c27b82024db7ff5038a51651f48f05e5e

SFQ队列的 limit=1 配置会导致逻辑错误：当第一个数据包入队时，SFQ将其放入队列，qlen=1。当第二个数据包到达时，由于 limit=1，SFQ会丢弃该包，但 qlen 仍保持为 1（尽管队列实际为空）。后续出队操作（sfq\_dec）会错误地递减 qlen，导致 下溢（underflow），进而引发数组越界访问（index=65535）。根本原因：SFQ未正确处理 limit=1 的特殊情况，导致队列状态与 qlen 计数器不一致。内核未在初始化时拒绝 limit=1 的无效配置。修复方式：显式禁止 limit=1 的配置。修复效果：防止用户设置无效的 limit 值。避免队列状态与计数器不一致导致的越界访问。

CVE-2024-57995

[drivers/net/wireless/ath/ath12k/mac.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/ath/ath12k/mac.c?id=f3a95a312419e4f1e992525917da9dbcd247038f)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=f3a95a312419e4f1e992525917da9dbcd247038f

问题代码路径：当 arvif 需要从一个无线设备（radio）切换到另一个时，ath12k\_mac\_unassign\_link\_vif() 会被调用，释放 arvif 的内存。之后，代码检查 arvif->is\_created，此时 arvif 已被释放，导致 释放后使用（UAF）。根本原因：内存管理逻辑错误，未正确处理 arvif 的生命周期。检查 arvif->is\_created 的代码位置不正确，应在重新分配 arvif 之后执行。修复方式：将 if (arvif->is\_created) 检查移到 ath12k\_mac\_assign\_link\_vif() 之后，确保 arvif 已重新分配。修复效果：避免在 arvif 释放后访问其成员。确保逻辑正确性，防止UAF漏洞。

CVE-2024-57993

[drivers/hid/hid-thrustmaster.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/hid/hid-thrustmaster.c?id=e5bcae4212a6a4b4204f46a1b8bcba08909d2007)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e5bcae4212a6a4b4204f46a1b8bcba08909d2007

问题代码路径：驱动直接从USB接口描述符中读取端点地址（bEndpointAddress），但未验证该端点是否为 中断传输（Interrupt） 类型。如果设备返回错误的端点类型（如批量传输Bulk或等时传输Isochronous），后续USB请求可能导致内核警告或崩溃。根本原因：缺乏对USB端点类型的严格校验，导致驱动可能操作不兼容的端点。修复方式：使用 usb\_check\_int\_endpoints() 验证端点是否为中断传输类型。若端点类型不匹配，输出错误日志并提前返回。修复效果：避免操作不支持的端点类型。增强驱动对异常设备的鲁棒性。

CVE-2024-57991

[drivers/net/wireless/realtek/rtw89/chan.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/realtek/rtw89/chan.c?id=e4790b3e314a4814f1680a5dc552031fb199b878)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e4790b3e314a4814f1680a5dc552031fb199b878

问题代码路径：函数 rtw89\_entity\_recalc\_mgnt\_roles() 遍历管理角色列表（active\_list），尝试根据目标模式（target pattern）重新排序。当找到匹配的条目时，会将其移动到列表头部，但仅使用 break 跳出内层循环，外层 list\_for\_each\_entry 仍会继续遍历。在异常情况下（如列表条目多次匹配目标模式），会导致无限循环，触发软死锁。根本原因：循环控制逻辑错误，未完全跳出外层循环。缺乏对列表状态的完整性检查。修复方式：将 break 替换为 goto fill，直接跳出外层循环。修复效果：避免无限循环，确保列表遍历正常终止。保持管理角色列表的正确顺序。

CVE-2024-57990

[drivers/net/wireless/mediatek/mt76/mt7925/mcu.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/mediatek/mt76/mt7925/mcu.c?id=d03b8fe1b518fc2ea2d82588e905f56d80cd64b2)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d03b8fe1b518fc2ea2d82588e905f56d80cd64b2

代码路径：mt7925\_load\_clc()处理来自固件的CLC条目。每个CLC条目都有一个idx字段，用于索引phy->clc[]。问题：原始检查：if (clc->idx > ARRAY\_SIZE(phy->clc))。正确检查：if (clc->idx >= ARRAY\_SIZE(phy->clc))。影响：如果clc->idx == ARRAY\_SIZE(phy->clc)，则会读写缓冲区之后的一个元素。修复方式：将比较运算符从>改为>=，确保严格检查数组边界。修复效果：防止越界访问，确保CLC索引始终在有效范围内。

CVE-2024-57989

[drivers/net/wireless/mediatek/mt76/mt7925/main.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/net/wireless/mediatek/mt76/mt7925/main.c?id=5cd0bd815c8a48862a296df9b30e0ea0da14acd3)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=5cd0bd815c8a48862a296df9b30e0ea0da14acd3

代码路径：函数mt7925\_change\_vif\_links()调用devm\_kzalloc()分配mconf和mlink内存，分配结果未立即检查，直接存入mconfs数组。问题：内存分配失败时返回NULL指针，后续操作可能解引用这些NULL指针导致内核崩溃。修复方式：在分配mconf和mlink后立即添加NULL检查，任一分配失败时返回-ENOMEM错误码。修复效果：防止空指针解引用，在内存不足时优雅地返回错误。

CVE-2024-57988 [drivers/bluetooth/btbcm.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/bluetooth/btbcm.c?id=df2f2d9199e61819cca5da0121dfa4d4cb57000f)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=df2f2d9199e61819cca5da0121dfa4d4cb57000f

问题代码路径：btbcm\_get\_board\_name()函数调用devm\_kstrdup()复制字符串。分配结果未检查，直接传递给strreplace()函数。根本原因：未处理内存分配失败的情况，导致内核可能解引用NULL指针。修复方式：添加对devm\_kstrdup()返回值的NULL检查。分配失败时直接返回NULL，避免后续操作。修复效果：防止空指针解引用，增强代码健壮性。

CVE-2024-57987 [drivers/bluetooth/btrtl.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/bluetooth/btrtl.c?id=3c15082f3567032d196e8760753373332508c2ca)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=3c15082f3567032d196e8760753373332508c2ca

代码路径：btrtl\_setup\_realtek()函数未检查btrtl\_dev->ic\_info是否为NULL，当设备芯片不在维护列表时，ic\_info可能为NULL。问题：直接解引用btrtl\_dev->ic\_info指针，导致内核空指针异常（Oops）。修复方式：添加对btrtl\_dev->ic\_info的NULL检查，只有指针有效时才调用hci\_set\_hw\_info()。修复效果：防止空指针解引用，优雅处理不支持的设备情况。

CVE-2024-57986

[drivers/hid/hid-core.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/hid/hid-core.c?id=ed3d3883476423f337aac0f22c521819b3f1e970)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=ed3d3883476423f337aac0f22c521819b3f1e970

代码路径：hid\_apply\_multiplier()函数处理分辨率乘数（Resolution Multiplier）控制，函数假设总会找到一个逻辑集合（HID\_COLLECTION\_LOGICAL）。问题：当设备描述符中没有定义逻辑集合时，会导致无效指针访问，可能引发内核死循环或崩溃。修复方式：添加显式检查multiplier\_collection->type != HID\_COLLECTION\_LOGICAL，当不满足条件时将指针设为NULL。修复效果：正确处理没有逻辑集合的情况，避免死循环和空指针解引用。

CVE-2024-57985 [drivers/firmware/qcom/qcom\_scm.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/firmware/qcom/qcom_scm.c?id=faf1715798fe72b79e4432ce8c6d03ca69765425)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=faf1715798fe72b79e4432ce8c6d03ca69765425

代码路径：qcom\_scm\_probe()函数在初始化过程中可能因多种原因失败，失败时未清理已赋值的全局\_\_scm变量。问题：其他模块（如TZMEM）会假定\_\_scm可用，可能导致后续\_\_scm\_smc\_call()等函数访问NULL指针。修复方式：在probe失败路径（err标签）添加smp\_store\_release(&\_\_scm, NULL)，使用内存屏障保证可见性。修复效果：确保probe失败时全局状态被正确重置，防止其他模块访问无效指针。

CVE-2024-57984 [drivers/i3c/master/dw-i3c-master.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/i3c/master/dw-i3c-master.c?id=fc84dd3c909a372c0d130f5f84c404717c17eed8)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fc84dd3c909a372c0d130f5f84c404717c17eed8

代码路径：dw\_i3c\_common\_probe()初始化工作队列master->hj\_work，绑定到dw\_i3c\_hj\_work。中断处理程序dw\_i3c\_master\_irq\_handler可能触发dw\_i3c\_hj\_work执行。模块卸载时，dw\_i3c\_common\_remove()直接调用i3c\_master\_unregister(&master->base)释放资源，但未取消工作队列。问题：工作队列可能仍在运行，访问已释放的master->base，导致UAF。修复方式：在dw\_i3c\_common\_remove()中调用cancel\_work\_sync(&master->hj\_work)，确保工作队列完成后再释放资源。修复效果：避免工作队列访问已释放的内存。安全同步资源释放与异步任务。

CVE-2024-57983

[drivers/mailbox/mailbox-th1520.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/mailbox/mailbox-th1520.c?id=db049866943a38bf46a34fa120d526663339d7a5)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=db049866943a38bf46a34fa120d526663339d7a5

代码路径：th1520\_mbox\_suspend\_noirq和th1520\_mbox\_resume\_noirq函数用于保存和恢复MBOX ICU0的中断掩码寄存器。原代码定义数组大小为TH\_1520\_MBOX\_CHANS - 1，但实际需要保存4个寄存器（TH\_1520\_MBOX\_CHANS = 4）。问题：数组大小不足，导致访问intr\_mask[3]时越界，可能破坏相邻内存。修复方式：将数组大小修正为TH\_1520\_MBOX\_CHANS（即4）。修复效果：确保所有4个中断掩码寄存器可被安全存储。消除越界访问风险。

CVE-2024-57981

[drivers/usb/host/xhci-ring.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/usb/host/xhci-ring.c?id=fd8bfaeba4a85b14427899adec0efb3954300653)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fd8bfaeba4a85b14427899adec0efb3954300653

代码路径：命令被提交到命令环的最后一个可用TRB（Transfer Request Block），此时enqueue指针会指向链接TRB（Link TRB）。如果命令被中止，处理中止完成时，dequeue指针会移动到下一个段的第一个TRB。如果没有新命令入队，xhci\_handle\_stopped\_cmd\_ring()会误判仍有待处理命令，尝试调用xhci\_mod\_cmd\_timer()，但cur\_cmd可能为NULL。问题：未检查cur\_cmd是否为NULL，直接传递给xhci\_mod\_cmd\_timer()，导致内核崩溃。修复方式：在调用xhci\_mod\_cmd\_timer()前显式检查cur\_cmd是否为NULL。修复效果：避免空指针解引用。保持命令中止处理的正常逻辑。

CVE-2024-57982 [net/xfrm/xfrm\_state.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/xfrm/xfrm_state.c?id=e952837f3ddb0ff726d5b582aa1aad9aa38d024d)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e952837f3ddb0ff726d5b582aa1aad9aa38d024d

代码路径：rehash操作会先更新state\_bydst指针，再更新state\_hmask，找操作使用旧的hmask值计算哈希，可能导致访问超出新哈希表范围的地址。问题：缺乏对哈希表指针和掩码（hmask）的原子性更新，查找操作可能使用不一致的指针和掩码组合。修复方式：引入xfrm\_hash\_ptrs\_get()函数原子获取哈希表指针和掩码，使用序列计数器（seqlock）确保一致性，添加lockdep断言确保哈希函数仅在持有锁时调用。修复效果：确保查找操作使用一致的指针和掩码，防止越界内存访问。

CVE-2024-57980

[drivers/media/usb/uvc/uvc\_status.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/media/usb/uvc/uvc_status.c?id=d8e63dd7b6683969d3d47c7b8e9635f96d554ad4)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d8e63dd7b6683969d3d47c7b8e9635f96d554ad4

代码路径：uvc\_status\_init()调用kfree(dev->status)释放内存，但未将指针置为NULL。后续uvc\_status\_cleanup()调用kfree(dev->status)时，若dev->status仍指向已释放内存，则导致双重释放。问题：未遵循“释放后置空”的安全编程实践。内存管理逻辑不完整，未处理错误路径的指针状态。修复方式：在kfree(dev->status)后显式设置dev->status = NULL。修复效果：避免uvc\_status\_cleanup()重复释放同一指针。符合“释放后置空”的安全实践。

CVE-2024-57978 [drivers/media/platform/nxp/imx-jpeg/mxc-jpeg.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/media/platform/nxp/imx-jpeg/mxc-jpeg.c?id=fde89fe11b44500bfcb2d405825b69a5df805d19)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=fde89fe11b44500bfcb2d405825b69a5df805d19

代码路径：mxc\_jpeg\_detach\_pm\_domains()函数在清理电源域时，未检查jpeg->pd\_dev[i]是否为错误指针。原代码仅检查jpeg->pd\_dev[i]非NULL，但未处理IS\_ERR()情况。问题：错误指针（如ERR\_PTR(-ENODEV)）被直接传递给pm\_runtime\_suspended()，导致内核Oops。修复方式：使用IS\_ERR\_OR\_NULL()统一检查jpeg->pd\_dev[i]和jpeg->pd\_link[i]。确保错误指针不会被传递给PM运行时函数。修复效果：避免错误指针解引用。代码逻辑更清晰，覆盖所有无效指针情况。

CVE-2024-57976 [fs/btrfs/inode.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/fs/btrfs/inode.c?id=692cf71173bb41395c855acbbbe197d3aedfa5d4)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=692cf71173bb41395c855acbbbe197d3aedfa5d4

代码路径：cow\_file\_range()尝试为文件范围分配空间，可能部分成功（如仅分配前16K）。失败时，原代码立即解锁并清理页面标记（如EXTENT\_DELALLOC），但未正确处理已分配的ordered extents。后续写回操作发现页面仍为脏页但无ordered extents，误入COW修复路径，触发BUG\_ON()。问题：错误处理未保持一致性：部分成功分配的区间未与页面状态同步清理。过早解锁页面，导致并发写回操作观察到不一致状态。修复方式：延迟清理：保持区间锁定直至整个操作完成或完全回滚。分阶段处理：成功区间：标记为PAGE\_SET\_ORDERED，提交后统一清理。失败区间：显式清理EXTENT\_LOCKED | EXTENT\_DELALLOC并终止页面写回。原子性操作：通过extent\_clear\_unlock\_delalloc()统一管理状态。修复效果：避免页面状态与区间管理的竞态条件。确保失败时所有资源（如ordered extents）被正确释放。

CVE-2024-57973 [drivers/infiniband/hw/cxgb4/device.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/infiniband/hw/cxgb4/device.c?id=e53ca458f543aa352d09b484550de173cb9085c2)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=e53ca458f543aa352d09b484550de173cb9085c2

代码路径：copy\_gl\_to\_skb\_pkt()函数计算SKB缓冲区大小时，直接相加用户控制的gl->tot\_len与固定头部长，在32位系统上，若gl->tot\_len接近U32\_MAX，加法可能导致整数溢出。问题：未使用安全的加法运算（如size\_add），可能分配过小缓冲区，导致后续内存越界。修复方式：使用size\_add()宏安全计算缓冲区大小，确保加法运算不会整数溢出。修复效果：消除整数溢出风险，保持原有功能不变。

CVE-2024-57953 [drivers/rtc/rtc-tps6594.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/rtc/rtc-tps6594.c?id=53b0c7b15accb18d15d95c7fe68f61630ebfd1ca)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=53b0c7b15accb18d15d95c7fe68f61630ebfd1ca

代码路径：tps6594\_rtc\_set\_offset()函数中tmp = offset \* TICKS\_PER\_HOUR存在整数溢出风险，offset为long类型（32位系统上最大约20亿），TICKS\_PER\_HOUR原定义为32768 \* 3600（约1.18亿），当offset>18时，32位乘法结果会溢出。问题：未使用足够宽的类型存储中间计算结果，可能导致RTC时间校准错误。修复方式：将TICKS\_PER\_HOUR定义为long long类型（添加LL后缀），确保乘法运算在64位空间进行。修复效果：消除整数溢出风险，保持RTC校准精度。

CVE-2024-57952

[fs/libfs.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/fs/libfs.c?id=b662d858131da9a8a14e68661656989b14dbf113)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=b662d858131da9a8a14e68661656989b14dbf113

问题引入：由commit 64a7ce76fb90引入的修复导致新问题。核心问题：原修复错误地将目录偏移视为单调递增的值而非不透明cookie，在32位系统上，偏移值可能回绕（约几周后），导致dentry2offset(dentry) >= last\_index比较失败，结果使现有目录条目永久从readdir输出中消失（但仍可通过路径查找访问）。修复方式：完全回退有问题的commit 64a7ce76fb90。修复内容：移除offset\_dir\_open操作，恢复原始offset\_iterate\_dir实现（不进行偏移值比较），简化offset\_readdir逻辑。后续计划：准备更安全的修复方案（兼容32位系统）。

CVE-2024-57950

[drivers/gpu/drm/amd/display/dc/dml2/dml21/src/dml2\_core/dml2\_core\_dcn4\_calcs.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/gpu/drm/amd/display/dc/dml2/dml21/src/dml2_core/dml2_core_dcn4_calcs.c?id=c9d6afb4f9c338049662d27d169fba7dd60e337d)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=c9d6afb4f9c338049662d27d169fba7dd60e337d

代码路径：CalculateBytePerPixelAndBlockSizes()函数计算像素字节大小时，原代码将BytePerPixelY和BytePerPixelC初始化为0。问题：这些变量可能被用作除数，若后续逻辑未正确赋值（如未处理的像素格式），可能导致除零错误。修复方式：将BytePerPixelY和BytePerPixelC默认值从0改为1。修复效果：消除除零风险，保持正常功能（1字节/像素是合理默认值）。

CVE-2024-57949  [drivers/irqchip/irq-gic-v3-its.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/irqchip/irq-gic-v3-its.c?id=d7b0e89610dd45ac6cf0d6f99bfa9ccc787db344)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d7b0e89610dd45ac6cf0d6f99bfa9ccc787db344

代码路径：irq\_set\_vcpu\_affinity()调用链进入中断禁用区域（raw\_spin\_lock\_irqsave()），its\_irq\_set\_vcpu\_affinity()使用guard(raw\_spinlock\_irq)错误地重新启用中断，irq\_put\_desc\_unlock()触发警告（因中断本应保持禁用）。问题引入：commit b97e8a2f7130错误替换了原始锁机制。核心问题：在已禁用中断的上下文中错误启用中断，违反嵌套锁/中断状态管理规则。修复方式：将guard(raw\_spinlock\_irq)替换为guard(raw\_spinlock)，保持中断状态不变（不自动启用中断）。修复效果：维持正确的中断禁用上下文，消除竞态条件风险。

CVE-2024-57948

[net/mac802154/iface.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/net/mac802154/iface.c?id=eb09fbeb48709fe66c0d708aed81e910a577a30a)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=eb09fbeb48709fe66c0d708aed81e910a577a30a

竞态场景：CPU0：通过genl删除网络接口（ieee802154\_del\_iface），CPU1：注销硬件设备（ieee802154\_unregister\_hw）。问题本质：两个并行操作尝试修改同一链表（local->interfaces），当接口列表已为空时仍尝试删除节点，导致链表损坏。具体表现：list\_del\_rcu()操作在已删除/空链表上执行，触发内核BUG\_ON（\_\_list\_del\_entry\_valid\_or\_report）。修复方式：在ieee802154\_if\_remove()中添加链表空检查，如果local->interfaces为空则提前返回。修复效果：避免在空链表上执行删除操作，保持RCU和锁的正确使用。

CVE-2024-57946  [drivers/block/virtio\_blk.c](https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/diff/drivers/block/virtio_blk.c?id=d738f3215bb4f88911ff4579780a44960c8e0ca5)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=d738f3215bb4f88911ff4579780a44960c8e0ca5

问题引入：由commit 4ce6e2db00de引入的队列冻结机制。核心问题：原修复在整个挂起上下文中保持队列冻结状态，在挂起上下文中尝试调用bio\_queue\_enter()可能导致死锁，冻结队列会阻塞所有I/O请求，包括可能需要的挂起操作自身。修复方式：修改virtblk\_freeze()：仅临时冻结队列以排空进行中的I/O，立即静默(quiesce)队列，然后解冻队列。修改virtblk\_restore()：取消队列静默状态，不再处理解冻操作（已在冻结时完成）。修复效果：保持排空进行中I/O的能力，避免在挂起上下文中长期冻结队列，消除死锁风险。