

# Full Trivy Vulnerability Report

Image: node:lts  
Scan Time: 2025-04-13T14:35:07Z

## Vulnerability Severity Summary

Severity	Count
LOW	670
CRITICAL	5
HIGH	98
MEDIUM	542

## Detailed Vulnerabilities

**Target: node:lts (debian 12.10)**

**[CVE-2011-3374] It was found that apt-key in apt, all versions, do not correctly valid ... (Severity: LOW)**

Package: apt  
Installed: 2.6.1  
Fixed: %ls(<nil>)

It was found that apt-key in apt, all versions, do not correctly validate gpg keys with the master keyring, leading to a potential man-in-the-middle attack.

More Info: <https://avd.aquasec.com/nvd/cve-2011-3374>

**[TEMP-0841856-B18BAF] [Privilege escalation possible to other user than root] (Severity: LOW)**

Package: bash  
Installed: 5.2.15-2+b7  
Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0841856-B18BAF>

**[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The C++ symbol demangler routine in cplus-dem.c in libiberty, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka libbfd).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

### **[CVE-2018-20673] libiberty: Integer overflow in demangle\_template() function (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %!s(<nil>)

The demangle\_template function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by nm.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

### **[CVE-2018-20712] libiberty: heap-based buffer over-read in d\_expression\_1 (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %!s(<nil>)

A heap-based buffer over-read exists in the function d\_expression\_1 in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by c++filt.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

### **[CVE-2018-9996] binutils: Stack-overflow in libiberty/cplus-dem.c causes crash (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %!s(<nil>)

An issue was discovered in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there are recursive stack frames: demangle\_template\_value\_parm, demangle\_integral\_value, and demangle\_expression.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

### **[CVE-2021-32256] binutils: stack-overflow issue in demangle\_type in rust-demangle.c. (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %!s(<nil>)

An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in demangle\_type in rust-demangle.c.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

### **[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length verdef table (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %!s(<nil>)

A potential heap based buffer overflow was found in \_bfd\_elf\_slurp\_version\_tables() in bfd/elf.c. This may lead to loss of

availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

**[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

**[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

**[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

**[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

#### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function `xstrdup` of the file `liberty/xmalloc.c` of the component `ld`. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function `bfd_malloc` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function `xmemdup` of the file `xmemdup.c` of the component `ld`. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has

been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function xstrdup of the file xstrdup.c of the component ld. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

#### **[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function bfd\_set\_format of the file format.c. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is 8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

#### **[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function \_bfd\_elf\_gc\_mark\_rsec of the file elflink.c of the component ld. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named f9978defb6fab0bd8583942d97c112b0932ac814. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

#### **[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

#### **[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

#### **[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

#### **[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: binutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `931494c9a89558acb36a03a340c01726545eef24`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

#### **[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

### **[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: binutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

### **[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: binutils-common  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The C++ symbol demangler routine in `cplus-dem.c` in `libiberty`, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka `libbfd`).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

### **[CVE-2018-20673] libiberty: Integer overflow in `demangle_template()` function (Severity: LOW)**

Package: binutils-common  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The `demangle_template` function in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by `nm`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

### **[CVE-2018-20712] libiberty: heap-based buffer over-read in `d_expression_1` (Severity: LOW)**

Package: binutils-common  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function `d_expression_1` in `cp-demangle.c` in GNU libiberty, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by `c++filt`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

#### **[CVE-2018-9996] binutils: Stack-overflow in libiberty/cplus-dem.c causes crash (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %s(<nil>)

An issue was discovered in `cplus-dem.c` in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there are recursive stack frames: `demangle_template_value_parm`, `demangle_integral_value`, and `demangle_expression`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

#### **[CVE-2021-32256] binutils: stack-overflow issue in demangle\_type in rust-demangle.c. (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %s(<nil>)

An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in `demangle_type` in `rust-demangle.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

#### **[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-lengthverdef table (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %s(<nil>)

A potential heap based buffer overflow was found in `_bfd_elf_slurp_version_tables()` in `bfd/elf.c`. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

#### **[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %s(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>



### **[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

### **[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

### **[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function xstrdup of the file libiberty/xmalloc.c of the component ld. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function bfd\_malloc of the file libbfd.c of the component ld. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function xmemdup of the file xmemdup.c of the component ld. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function xstrdup of the file xstrdup.c of the component ld. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the

reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

**[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function `bfd_set_format` of the file `format.c`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is `8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

**[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function `_bfd_elf_gc_mark_rsec` of the file `elflink.c` of the component `ld`. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named `f9978defb6fab0bd8583942d97c112b0932ac814`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

**[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

**[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has

been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

**[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

**[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `931494c9a89558acb36a03a340c01726545eef24`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

**[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: binutils-common

Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The C++ symbol demangler routine in `cplus-dem.c` in `libiberty`, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka `libbfd`).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

**[CVE-2018-20673] libiberty: Integer overflow in `demangle_template()` function (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The `demangle_template` function in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by `nm`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

**[CVE-2018-20712] libiberty: heap-based buffer over-read in `d_expression_1` (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function `d_expression_1` in `cp-demangle.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by `c++filt`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

**[CVE-2018-9996] binutils: Stack-overflow in `libiberty/cplus-dem.c` causes crash (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`  
Installed: 2.40-2  
Fixed: %ls(<nil>)

An issue was discovered in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by `libiberty`, and there are recursive stack frames: `demangle_template_value_parm`, `demangle_integral_value`, and `demangle_expression`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

**[CVE-2021-32256] binutils: stack-overflow issue in demangle\_type in rust-demangle.c. (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in demangle\_type in rust-demangle.c.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

**[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length verdef table (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A potential heap based buffer overflow was found in \_bfd\_elf\_slurp\_version\_tables() in bfd/elf.c. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

**[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

**[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

**[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

#### **[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

#### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: `binutils-x86-64-linux-gnu`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function `xstrdup` of the file `liberty/xmalloc.c` of the component `ld`. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: binutils-x86-64-linux-gnu  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function `bfd_malloc` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: binutils-x86-64-linux-gnu  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function `xmemdup` of the file `xmemdup.c` of the component `ld`. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: binutils-x86-64-linux-gnu  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function `xstrdup` of the file `xstrdup.c` of the component `ld`. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

#### **[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function `bfd_set_format` of the file `format.c`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is `8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150`. It is recommended to upgrade the affected component.



More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

**[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function `_bfd_elf_gc_mark_rsec` of the file `elflink.c` of the component `ld`. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named `f9978defb6fab0bd8583942d97c112b0932ac814`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

**[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

**[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

**[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told

to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

**[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %s(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is 931494c9a89558acb36a03a340c01726545eef24. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

**[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %s(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as b425859021d17adf62f06fb904797cf8642986ad. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: binutils-x86-64-linux-gnu

Installed: 2.40-2

Fixed: %s(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named ba6ad3a18cb26b79e0e3b84c39f707535bbc344d. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: bsduutils

Installed: 1:2.38.1-5+deb12u3

Fixed: %s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2016-2781] coreutils: Non-privileged session can escape to the parent session in chroot (Severity: LOW)**

Package: coreutils

Installed: 9.1-1

Fixed: %ls(<nil>)

chroot in GNU coreutils, when used with --userspec, allows local users to escape to the parent session via a crafted TIOCSSTI ioctl call, which pushes characters to the terminal's input buffer.

More Info: <https://avd.aquasec.com/nvd/cve-2016-2781>

**[CVE-2017-18018] coreutils: race condition vulnerability in chown and chgrp (Severity: LOW)**

Package: coreutils

Installed: 9.1-1

Fixed: %ls(<nil>)

In GNU Coreutils through 8.29, chown-core.c in chown and chgrp does not prevent replacement of a plain file with a symlink during use of the POSIX "-R -L" options, which allows local users to modify the ownership of arbitrary files by leveraging a race condition.

More Info: <https://avd.aquasec.com/nvd/cve-2017-18018>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: cpp-12

Installed: 12.2.0-14

Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: cpp-12

Installed: 12.2.0-14

Fixed: %ls(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using

alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2024-2379] curl: QUIC certificate check bypass with wolfSSL (Severity: LOW)**

Package: curl

Installed: 7.88.1-10+deb12u12

Fixed: %ls(<nil>)

libcurl skips the certificate verification for a QUIC connection under certain conditions, when built to use wolfSSL. If told to use an unknown/bad cipher or curve, the error path accidentally skips the verification and returns OK, thus ignoring any certificate problems.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2379>

### **[CVE-2025-0725] libcurl: Buffer Overflow in libcurl via zlib Integer Overflow (Severity: LOW)**

Package: curl

Installed: 7.88.1-10+deb12u12

Fixed: %ls(<nil>)

When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the `CURLOPT\_ACCEPT\_ENCODING` option, **\*\*using zlib 1.2.0.3 or older\*\***, an attacker-controlled integer overflow would make libcurl perform a buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0725>

### **[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: dirmngr

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

### **[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: dirmngr

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: g++-12

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: g++-12

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: gcc-12

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity:**

## LOW)

Package: gcc-12

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: gcc-12-base

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in `demangle_const`, as demonstrated by `nm-new`.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: gcc-12-base

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to

go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2018-1000021] git: client prints server-sent ANSI escape codes to the terminal, allowing for unverified messages to potentially execute arbitrary commands (Severity: LOW)**

Package: git

Installed: 1:2.39.5-0+deb12u2

Fixed: %ls(<nil>)

GIT version 2.15.1 and earlier contains a Input Validation Error vulnerability in Client that can result in problems including messing up terminal configuration to RCE. This attack appear to be exploitable via The user must interact with a malicious git server, (or have their traffic modified in a MITM attack).

More Info: <https://avd.aquasec.com/nvd/cve-2018-1000021>

**[CVE-2022-24975] git: The --mirror option for git leaks secret for deleted content, aka the "GitBleed" (Severity: LOW)**

Package: git

Installed: 1:2.39.5-0+deb12u2

Fixed: %ls(<nil>)

The --mirror documentation for Git through 2.35.1 does not mention the availability of deleted content, aka the "GitBleed" issue. This could present a security risk if information-disclosure auditing processes rely on a clone operation without the --mirror option. Note: This has been disputed by multiple 3rd parties who believe this is an intended feature of the git binary and does not pose a security risk.

More Info: <https://avd.aquasec.com/nvd/cve-2022-24975>

**[CVE-2024-52005] git: The sideband payload is passed unfiltered to the terminal in git (Severity: LOW)**

Package: git

Installed: 1:2.39.5-0+deb12u2

Fixed: %ls(<nil>)

Git is a source code management tool. When cloning from a server (or fetching, or pushing), informational or error messages are transported from the remote Git process to the client via the so-called "sideband channel". These messages will be prefixed with "remote:" and printed directly to the standard error output. Typically, this standard error output is connected to a terminal that understands ANSI escape sequences, which Git did not protect against. Most modern terminals support control sequences that can be used by a malicious actor to hide and misrepresent information, or to mislead the user into executing untrusted scripts. As requested on the git-security mailing list, the patches are under discussion on the public mailing list. Users are advised to update as soon as possible. Users unable to upgrade should avoid recursive clones unless they are from trusted sources.

More Info: <https://avd.aquasec.com/nvd/cve-2024-52005>

**[CVE-2018-1000021] git: client prints server-sent ANSI escape codes to the terminal, allowing for unverified messages to potentially execute arbitrary commands (Severity: LOW)**

Package: git-man

Installed: 1:2.39.5-0+deb12u2

Fixed: %ls(<nil>)

GIT version 2.15.1 and earlier contains a Input Validation Error vulnerability in Client that can result in problems including messing up terminal configuration to RCE. This attack appear to be exploitable via The user must interact with a malicious git server, (or have their traffic modified in a MITM attack).

More Info: <https://avd.aquasec.com/nvd/cve-2018-1000021>

**[CVE-2022-24975] git: The --mirror option for git leaks secret for deleted content, aka the "GitBleed" (Severity: LOW)**

Package: git-man

Installed: 1:2.39.5-0+deb12u2

Fixed: %!s(<nil>)

The --mirror documentation for Git through 2.35.1 does not mention the availability of deleted content, aka the "GitBleed" issue. This could present a security risk if information-disclosure auditing processes rely on a clone operation without the --mirror option. Note: This has been disputed by multiple 3rd parties who believe this is an intended feature of the git binary and does not pose a security risk.

More Info: <https://avd.aquasec.com/nvd/cve-2022-24975>

**[CVE-2024-52005] git: The sideband payload is passed unfiltered to the terminal in git (Severity: LOW)**

Package: git-man

Installed: 1:2.39.5-0+deb12u2

Fixed: %!s(<nil>)

Git is a source code management tool. When cloning from a server (or fetching, or pushing), informational or error messages are transported from the remote Git process to the client via the so-called "sideband channel". These messages will be prefixed with "remote:" and printed directly to the standard error output. Typically, this standard error output is connected to a terminal that understands ANSI escape sequences, which Git did not protect against. Most modern terminals support control sequences that can be used by a malicious actor to hide and misrepresent information, or to mislead the user into executing untrusted scripts. As requested on the git-security mailing list, the patches are under discussion on the public mailing list. Users are advised to update as soon as possible. Users unable to upgrade should avoid recursive clones unless they are from trusted sources.

More Info: <https://avd.aquasec.com/nvd/cve-2024-52005>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gnupg

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gnupg

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)



In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gnupg-l10n

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gnupg-l10n

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gnupg-utils

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gnupg-utils

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets**

**(Severity: LOW)**

Package: gpg

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpg

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpg-agent

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpg-agent

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpg-wks-client

Installed: 2.2.40-1.1

Fixed: %ls(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpg-wks-client

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpg-wks-server

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpg-wks-server

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpgconf

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpgconf

Installed: 2.2.40-1.1

Fixed: %!s(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpgsm  
Installed: 2.2.40-1.1  
Fixed: %!s(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpgsm  
Installed: 2.2.40-1.1  
Fixed: %!s(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2022-3219] gnupg: denial of service issue (resource consumption) using compressed packets (Severity: LOW)**

Package: gpgv  
Installed: 2.2.40-1.1  
Fixed: %!s(<nil>)

GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3219>

**[CVE-2025-30258] gnupg: verification DoS due to a malicious subkey in the keyring (Severity: LOW)**

Package: gpgv  
Installed: 2.2.40-1.1  
Fixed: %!s(<nil>)

In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid backsig or that has incorrect usage flags, the user loses the ability to verify signatures made from certain other signing keys, aka a "verification DoS."

More Info: <https://avd.aquasec.com/nvd/cve-2025-30258>

**[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ...**

**(Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of

service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colorspace.c (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colorspace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: imagemagick

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

**[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: imagemagick-6-common  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: imagemagick-6-common  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: imagemagick-6-common  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: imagemagick-6-common  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: imagemagick-6-common  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: imagemagick-6-common

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: imagemagick-6-common

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: imagemagick-6-common

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: imagemagick-6-common

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>



**[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: imagemagick-6.q16  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: imagemagick-6.q16  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: imagemagick-6.q16  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: imagemagick-6.q16  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: imagemagick-6.q16  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: imagemagick-6.q16

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: imagemagick-6.q16

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: imagemagick-6.q16

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: imagemagick-6.q16

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: krb5-multidev  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: krb5-multidev  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity: LOW)**

Package: krb5-multidev  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: krb5-multidev  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: krb5-multidev  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

**[CVE-2023-6879] aom: heap-buffer-overflow on frame size change (Severity: CRITICAL)**

Package: libaom3

Installed: 3.6.0-1+deb12u1

Fixed: %!s(<nil>)

Increasing the resolution of video frames, while performing a multi-threaded encode, can result in a heap overflow in `av1_loop_restoration_dealloc()`.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6879>

**[CVE-2023-39616] AOMedia v3.0.0 to v3.5.0 was discovered to contain an invalid read mem ... (Severity: HIGH)**

Package: libaom3

Installed: 3.6.0-1+deb12u1

Fixed: %!s(<nil>)

AOMedia v3.0.0 to v3.5.0 was discovered to contain an invalid read memory access via the component `assign_frame_buffer_p` in `av1/common/av1_common_int.h`.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39616>

**[CVE-2011-3374] It was found that apt-key in apt, all versions, do not correctly valid ... (Severity: LOW)**

Package: libapt-pkg6.0

Installed: 2.6.1

Fixed: %!s(<nil>)

It was found that `apt-key` in `apt`, all versions, do not correctly validate `gpg` keys with the master keyring, leading to a potential man-in-the-middle attack.

More Info: <https://avd.aquasec.com/nvd/cve-2011-3374>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libasan8

Installed: 12.2.0-14

Fixed: %!s(<nil>)

`libiberty/rust-demangle.c` in GNU GCC 11.2 allows stack consumption in `demangle_const`, as demonstrated by `nm-new`.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libasan8

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\*** A failure in the `-fstack-protector` feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies

to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libatomic1  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in `demangle_const`, as demonstrated by `nm-new`.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libatomic1  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the `-fstack-protector` feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: libbinutils  
Installed: 2.40-2

Fixed: %ls(<nil>)

The C++ symbol demangler routine in `cplus-dem.c` in `libiberty`, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka `libbfd`).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

#### **[CVE-2018-20673] libiberty: Integer overflow in `demangle_template()` function (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

The `demangle_template` function in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by `nm`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

#### **[CVE-2018-20712] libiberty: heap-based buffer over-read in `d_expression_1` (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function `d_expression_1` in `cp-demangle.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by `c++filt`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

#### **[CVE-2018-9996] binutils: Stack-overflow in `libiberty/cplus-dem.c` causes crash (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by `libiberty`, and there are recursive stack frames: `demangle_template_value_parm`, `demangle_integral_value`, and `demangle_expression`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

#### **[CVE-2021-32256] binutils: stack-overflow issue in `demangle_type` in `rust-demangle.c`. (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in GNU `libiberty`, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in `demangle_type` in `rust-demangle.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

### **[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length verdef table (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A potential heap based buffer overflow was found in `_bfd_elf_slurp_version_tables()` in `bfd/elf.c`. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

### **[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

### **[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

### **[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

### **[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

#### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function `xstrdup` of the file `libiberty/xmalloc.c` of the component `ld`. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: `libbinutils`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function `bfd_malloc` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**



Package: libbinutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function `xmempdup` of the file `xmempdup.c` of the component `ld`. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: libbinutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function `xstrdup` of the file `xstrdup.c` of the component `ld`. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

#### **[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: libbinutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function `bfd_set_format` of the file `format.c`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is 8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

#### **[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: libbinutils  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function `_bfd_elf_gc_mark_rsec` of the file `elflink.c` of the component `ld`. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named f9978defb6fab0bd8583942d97c112b0932ac814. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

#### **[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

#### **[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

#### **[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

#### **[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The

exploit has been disclosed to the public and may be used. The name of the patch is 931494c9a89558acb36a03a340c01726545eef24. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

**[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %s(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: libbinutils

Installed: 2.40-2

Fixed: %s(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libblkid-dev

Installed: 2.38.1-5+deb12u3

Fixed: %s(<nil>)

A flaw was found in the util-linux `chfn` and `chsh` utilities when compiled with Readline support. The Readline library uses an `"INPUTRC"` environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libblkid1

Installed: 2.38.1-5+deb12u3

Fixed: %s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2010-4756] glibc: glob implementation can cause excessive CPU and memory consumption due to crafted glob expressions (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in STAT commands to an FTP daemon, a different vulnerability than CVE-2010-2632.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4756>

**[CVE-2018-20796] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(\{227\})(\\1\\1|t1|\\2537)+' in grep.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20796>

**[CVE-2019-1010022] glibc: stack guard protection bypass (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The component is: nptl. The attack vector is: Exploit stack buffer overflow vulnerability and use this bypass vulnerability to bypass stack guard. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010022>

**[CVE-2019-1010023] glibc: running ldd on malicious ELF leads to code execution because of wrong size computation (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In worst case attacker may evaluate privileges. The component is: libld. The attack vector is: Attacker sends 2 ELF files to victim and asks to run ldd on it. ldd execute code. NOTE: Upstream comments indicate "this is being treated as a non-security bug

and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010023>

**[CVE-2019-1010024] glibc: ASLR bypass using cache of thread stack and heap (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thread stack and heap. The component is: glibc. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010024>

**[CVE-2019-1010025] glibc: information disclosure of heap addresses of pthread\_created thread (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread\_created thread. The component is: glibc. NOTE: the vendor's position is "ASLR bypass itself is not a vulnerability.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010025>

**[CVE-2019-9192] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc-bin

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(\\1\\1)\*' in grep, a different issue than CVE-2018-20796. NOTE: the software maintainer disputes that this is a vulnerability because the behavior occurs only with a crafted pattern

More Info: <https://avd.aquasec.com/nvd/cve-2019-9192>

**[CVE-2010-4756] glibc: glob implementation can cause excessive CPU and memory consumption due to crafted glob expressions (Severity: LOW)**

Package: libc-dev-bin

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in STAT commands to an FTP daemon, a different vulnerability than CVE-2010-2632.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4756>

**[CVE-2018-20796] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc-dev-bin  
Installed: 2.36-9+deb12u10  
Fixed: %!s(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(\227|)(\1\1|t|\2537)+' in grep.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20796>

**[CVE-2019-1010022] glibc: stack guard protection bypass (Severity: LOW)**

Package: libc-dev-bin  
Installed: 2.36-9+deb12u10  
Fixed: %!s(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The component is: nptl. The attack vector is: Exploit stack buffer overflow vulnerability and use this bypass vulnerability to bypass stack guard. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010022>

**[CVE-2019-1010023] glibc: running ldd on malicious ELF leads to code execution because of wrong size computation (Severity: LOW)**

Package: libc-dev-bin  
Installed: 2.36-9+deb12u10  
Fixed: %!s(<nil>)

GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In worst case attacker may evaluate privileges. The component is: libld. The attack vector is: Attacker sends 2 ELF files to victim and asks to run ldd on it. ldd execute code. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010023>

**[CVE-2019-1010024] glibc: ASLR bypass using cache of thread stack and heap (Severity: LOW)**

Package: libc-dev-bin  
Installed: 2.36-9+deb12u10  
Fixed: %!s(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thread stack and heap. The component is: glibc. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010024>

**[CVE-2019-1010025] glibc: information disclosure of heap addresses of pthread\_created thread (Severity: LOW)**

Package: libc-dev-bin  
Installed: 2.36-9+deb12u10  
Fixed: %!s(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread\_created thread. The component is: glibc. NOTE: the vendor's position is "ASLR bypass itself is not a vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010025>

**[CVE-2019-9192] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc-dev-bin

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(!)(\\1\\1)\*' in grep, a different issue than CVE-2018-20796. NOTE: the software maintainer disputes that this is a vulnerability because the behavior occurs only with a crafted pattern

More Info: <https://avd.aquasec.com/nvd/cve-2019-9192>

**[CVE-2010-4756] glibc: glob implementation can cause excessive CPU and memory consumption due to crafted glob expressions (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in STAT commands to an FTP daemon, a different vulnerability than CVE-2010-2632.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4756>

**[CVE-2018-20796] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(!227)(\\1\\1|t1|\\2537)+' in grep.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20796>

**[CVE-2019-1010022] glibc: stack guard protection bypass (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %!s(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The component is: nptl. The attack vector is: Exploit stack buffer overflow vulnerability and use this bypass vulnerability to bypass stack guard. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010022>

**[CVE-2019-1010023] glibc: running ldd on malicious ELF leads to code execution because of wrong size computation (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In worst case attacker may evaluate privileges. The component is: libld. The attack vector is: Attacker sends 2 ELF files to victim and asks to run ldd on it. ldd execute code. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010023>

**[CVE-2019-1010024] glibc: ASLR bypass using cache of thread stack and heap (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thread stack and heap. The component is: glibc. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010024>

**[CVE-2019-1010025] glibc: information disclosure of heap addresses of pthread\_created thread (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread\_created thread. The component is: glibc. NOTE: the vendor's position is "ASLR bypass itself is not a vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010025>

**[CVE-2019-9192] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc6

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(\|)(\\1\\1)\*' in grep, a different issue than CVE-2018-20796. NOTE: the software maintainer disputes that this is a vulnerability because the behavior occurs only with a crafted pattern

More Info: <https://avd.aquasec.com/nvd/cve-2019-9192>



**[CVE-2010-4756] glibc: glob implementation can cause excessive CPU and memory consumption due to crafted glob expressions (Severity: LOW)**

Package: libc6-dev  
Installed: 2.36-9+deb12u10  
Fixed: %ls(<nil>)

The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in STAT commands to an FTP daemon, a different vulnerability than CVE-2010-2632.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4756>

**[CVE-2018-20796] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc6-dev  
Installed: 2.36-9+deb12u10  
Fixed: %ls(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(\{227\})(\{1\}|t1|\{2537\})+' in grep.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20796>

**[CVE-2019-1010022] glibc: stack guard protection bypass (Severity: LOW)**

Package: libc6-dev  
Installed: 2.36-9+deb12u10  
Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The component is: nptl. The attack vector is: Exploit stack buffer overflow vulnerability and use this bypass vulnerability to bypass stack guard. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010022>

**[CVE-2019-1010023] glibc: running ldd on malicious ELF leads to code execution because of wrong size computation (Severity: LOW)**

Package: libc6-dev  
Installed: 2.36-9+deb12u10  
Fixed: %ls(<nil>)

GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In worst case attacker may evaluate privileges. The component is: libld. The attack vector is: Attacker sends 2 ELF files to victim and asks to run ldd on it. ldd execute code. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat.

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010023>

**[CVE-2019-1010024] glibc: ASLR bypass using cache of thread stack and heap (Severity: LOW)**

Package: libc6-dev  
Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thread stack and heap. The component is: glibc. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010024>

**[CVE-2019-1010025] glibc: information disclosure of heap addresses of pthread\_created thread (Severity: LOW)**

Package: libc6-dev

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread\_created thread. The component is: glibc. NOTE: the vendor's position is "ASLR bypass itself is not a vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2019-1010025>

**[CVE-2019-9192] glibc: uncontrolled recursion in function check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c (Severity: LOW)**

Package: libc6-dev

Installed: 2.36-9+deb12u10

Fixed: %ls(<nil>)

In the GNU C Library (aka glibc or libc6) through 2.29, check\_dst\_limits\_calc\_pos\_1 in posix/regexec.c has Uncontrolled Recursion, as demonstrated by '(!)(\\1\\1)\*' in grep, a different issue than CVE-2018-20796. NOTE: the software maintainer disputes that this is a vulnerability because the behavior occurs only with a crafted pattern

More Info: <https://avd.aquasec.com/nvd/cve-2019-9192>

**[CVE-2017-7475] cairo: NULL pointer dereference with a crafted font file (Severity: LOW)**

Package: libcairo-gobject2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the FT\_Load\_Glyph and FT\_Render\_Glyph resulting in an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7475>

**[CVE-2018-18064] cairo: Stack-based buffer overflow via parsing of crafted WebKitGTK+ document (Severity: LOW)**

Package: libcairo-gobject2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKitGTK+ because of the interaction between cairo-rectangular-scan-converter.c (the generate and render\_rows functions) and cairo-image-compositor.c (the \_cairo\_image\_spans\_and\_zero function).

More Info: <https://avd.aquasec.com/nvd/cve-2018-18064>

**[CVE-2019-6461] cairo: assertion problem in \_cairo\_arc\_in\_direction in cairo-arc.c (Severity: LOW)**

Package: libcairo-gobject2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an assertion problem in the function \_cairo\_arc\_in\_direction in the file cairo-arc.c.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6461>

**[CVE-2019-6462] cairo: infinite loop in the function \_arc\_error\_normalized in the file cairo-arc.c (Severity: LOW)**

Package: libcairo-gobject2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an infinite loop in the function \_arc\_error\_normalized in the file cairo-arc.c, related to \_arc\_max\_angle\_for\_tolerance\_normalized.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6462>

**[CVE-2017-7475] cairo: NULL pointer dereference with a crafted font file (Severity: LOW)**

Package: libcairo-script-interpreter2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the FT\_Load\_Glyph and FT\_Render\_Glyph resulting in an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7475>

**[CVE-2018-18064] cairo: Stack-based buffer overflow via parsing of crafted WebKitGTK+ document (Severity: LOW)**

Package: libcairo-script-interpreter2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKitGTK+ because of the interaction between cairo-rectangular-scan-converter.c (the generate and render\_rows functions) and cairo-image-compositor.c (the \_cairo\_image\_spans\_and\_zero function).

More Info: <https://avd.aquasec.com/nvd/cve-2018-18064>

**[CVE-2019-6461] cairo: assertion problem in \_cairo\_arc\_in\_direction in cairo-arc.c (Severity: LOW)**

Package: libcairo-script-interpreter2

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an assertion problem in the function `_cairo_arc_in_direction` in the file `cairo-arc.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6461>

**[CVE-2019-6462] cairo: infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c` (Severity: LOW)**

Package: `libcairo-script-interpreter2`

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c`, related to `_arc_max_angle_for_tolerance_normalized`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6462>

**[CVE-2017-7475] cairo: NULL pointer dereference with a crafted font file (Severity: LOW)**

Package: `libcairo2`

Installed: 1.16.0-7

Fixed: %ls(<nil>)

Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the `FT_Load_Glyph` and `FT_Render_Glyph` resulting in an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7475>

**[CVE-2018-18064] cairo: Stack-based buffer overflow via parsing of crafted WebKitGTK+ document (Severity: LOW)**

Package: `libcairo2`

Installed: 1.16.0-7

Fixed: %ls(<nil>)

cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKitGTK+ because of the interaction between `cairo-rectangular-scan-converter.c` (the `generate` and `render_rows` functions) and `cairo-image-compositor.c` (the `_cairo_image_spans_and_zero` function).

More Info: <https://avd.aquasec.com/nvd/cve-2018-18064>

**[CVE-2019-6461] cairo: assertion problem in `_cairo_arc_in_direction` in `cairo-arc.c` (Severity: LOW)**

Package: `libcairo2`

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an assertion problem in the function `_cairo_arc_in_direction` in the file `cairo-arc.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6461>

**[CVE-2019-6462] cairo: infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c` (Severity: LOW)**

Package: `libcairo2`

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c`, related to `_arc_max_angle_for_tolerance_normalized`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6462>

**[CVE-2017-7475] cairo: NULL pointer dereference with a crafted font file (Severity: LOW)**

Package: libcairo2-dev

Installed: 1.16.0-7

Fixed: %ls(<nil>)

Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the `FT_Load_Glyph` and `FT_Render_Glyph` resulting in an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7475>

**[CVE-2018-18064] cairo: Stack-based buffer overflow via parsing of crafted WebKitGTK+ document (Severity: LOW)**

Package: libcairo2-dev

Installed: 1.16.0-7

Fixed: %ls(<nil>)

cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKitGTK+ because of the interaction between `cairo-rectangular-scan-converter.c` (the `generate` and `render_rows` functions) and `cairo-image-compositor.c` (the `_cairo_image_spans_and_zero` function).

More Info: <https://avd.aquasec.com/nvd/cve-2018-18064>

**[CVE-2019-6461] cairo: assertion problem in `_cairo_arc_in_direction` in `cairo-arc.c` (Severity: LOW)**

Package: libcairo2-dev

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an assertion problem in the function `_cairo_arc_in_direction` in the file `cairo-arc.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6461>

**[CVE-2019-6462] cairo: infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c` (Severity: LOW)**

Package: libcairo2-dev

Installed: 1.16.0-7

Fixed: %ls(<nil>)

An issue was discovered in cairo 1.16.0. There is an infinite loop in the function `_arc_error_normalized` in the file `cairo-arc.c`, related to `_arc_max_angle_for_tolerance_normalized`.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6462>

**[CVE-2025-1390] libcap: pam\_cap: Fix potential configuration parsing error (Severity: MEDIUM)**

Package: libcap2  
Installed: 1:2.66-4  
Fixed: %ls(<nil>)

The PAM module pam\_cap.so of libcap configuration supports group names starting with `@`, during actual parsing, configurations not starting with `@` are incorrectly recognized as group names. This may result in nonintended users being granted an inherited capability set, potentially leading to security risks. Attackers can exploit this vulnerability to achieve local privilege escalation on systems where `/etc/security/capability.conf` is used to configure user inherited privileges by constructing specific usernames.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1390>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libcc1-0  
Installed: 12.2.0-14  
Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libcc1-0  
Installed: 12.2.0-14  
Fixed: %ls(<nil>)

**\*\*DISPUTED\*\***A failure in the `-fstack-protector` feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)

The C++ symbol demangler routine in `cplus-dem.c` in `libiberty`, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka `libbfd`).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

#### **[CVE-2018-20673] libiberty: Integer overflow in `demangle_template()` function (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %ls(<nil>)

The `demangle_template` function in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by `nm`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

#### **[CVE-2018-20712] libiberty: heap-based buffer over-read in `d_expression_1` (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function `d_expression_1` in `cp-demangle.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by `c++filt`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

#### **[CVE-2018-9996] binutils: Stack-overflow in `libiberty/cplus-dem.c` causes crash (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by `libiberty`, and there are recursive stack frames: `demangle_template_value_parm`, `demangle_integral_value`, and `demangle_expression`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

#### **[CVE-2021-32256] binutils: stack-overflow issue in `demangle_type` in `rust-demangle.c`. (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in GNU `libiberty`, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in `demangle_type` in `rust-demangle.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

#### **[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length `verdef` table (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A potential heap based buffer overflow was found in `_bfd_elf_slurp_version_tables()` in `bfd/elf.c`. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

**[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

**[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

**[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

**[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: libctf-nobfd0  
Installed: 2.40-2  
Fixed: %ls(<nil>)



A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

#### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function `xstrdup` of the file `libiberty/xmalloc.c` of the component `ld`. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function `bfd_malloc` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function `xmempdup` of the file `xmempdup.c` of the component `ld`. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function `xstrdup` of the file `xstrdup.c` of the component `ld`. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

#### **[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function `bfd_set_format` of the file `format.c`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is `8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

#### **[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: `libctf-nobfd0`

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function `_bfd_elf_gc_mark_rsec` of the file `elflink.c` of the component `ld`. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named `f9978defb6fab0bd8583942d97c112b0932ac814`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

### **[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

### **[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

### **[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

### **[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is

931494c9a89558acb36a03a340c01726545eef24. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

**[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: libctf-nobfd0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

The C++ symbol demangler routine in `cplus-dem.c` in `libiberty`, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka `libbfd`).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

**[CVE-2018-20673] libiberty: Integer overflow in `demangle_template()` function (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

The `demangle_template` function in `cplus-dem.c` in GNU `libiberty`, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by `nm`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

**[CVE-2018-20712] libiberty: heap-based buffer over-read in d\_expression\_1 (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function `d_expression_1` in `cp-demangle.c` in GNU libiberty, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by `c++filt`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

**[CVE-2018-9996] binutils: Stack-overflow in libiberty/cplus-dem.c causes crash (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in `cplus-dem.c` in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there are recursive stack frames: `demangle_template_value_parm`, `demangle_integral_value`, and `demangle_expression`.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

**[CVE-2021-32256] binutils: stack-overflow issue in demangle\_type in rust-demangle.c. (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in `demangle_type` in `rust-demangle.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

**[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length verdef table (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A potential heap based buffer overflow was found in `_bfd_elf_slurp_version_tables()` in `bfd/elf.c`. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

**[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

**[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

**[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

**[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

**[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be

launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function xstrdup of the file libiberty/xmalloc.c of the component ld. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function bfd\_malloc of the file libbfd.c of the component ld. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function xmemdup of the file xmemdup.c of the component ld. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function xstrdup of the file xstrdup.c of the component ld. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

**[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function bfd\_set\_format of the file format.c. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is 8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

**[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function \_bfd\_elf\_gc\_mark\_rsec of the file elflink.c of the component ld. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named f9978defb6fab0bd8583942d97c112b0932ac814. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

**[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function bfd\_putl64 of the file libbfd.c of the component ld. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is 75086e9de1707281172cc77f178e7949a4414ed0. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

**[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**



Package: libctf0  
Installed: 2.40-2  
Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

#### **[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_eh\_frame memory corruption (Severity: LOW)**

Package: libctf0  
Installed: 2.40-2  
Fixed: %!s(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_eh_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

#### **[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: libctf0  
Installed: 2.40-2  
Fixed: %!s(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `931494c9a89558acb36a03a340c01726545eef24`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

#### **[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: libctf0  
Installed: 2.40-2  
Fixed: %!s(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: libctf0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2024-2379] curl: QUIC certificate check bypass with wolfSSL (Severity: LOW)**

Package: libcurl3-gnutls

Installed: 7.88.1-10+deb12u12

Fixed: %ls(<nil>)

libcurl skips the certificate verification for a QUIC connection under certain conditions, when built to use wolfSSL. If told to use an unknown/bad cipher or curve, the error path accidentally skips the verification and returns OK, thus ignoring any certificate problems.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2379>

**[CVE-2025-0725] libcurl: Buffer Overflow in libcurl via zlib Integer Overflow (Severity: LOW)**

Package: libcurl3-gnutls

Installed: 7.88.1-10+deb12u12

Fixed: %ls(<nil>)

When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the ``CURLOPT_ACCEPT_ENCODING`` option, **\*\*using zlib 1.2.0.3 or older\*\***, an attacker-controlled integer overflow would make libcurl perform a buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0725>

**[CVE-2024-2379] curl: QUIC certificate check bypass with wolfSSL (Severity: LOW)**

Package: libcurl4

Installed: 7.88.1-10+deb12u12

Fixed: %ls(<nil>)

libcurl skips the certificate verification for a QUIC connection under certain conditions, when built to use wolfSSL. If told to use an unknown/bad cipher or curve, the error path accidentally skips the verification and returns OK, thus ignoring any certificate problems.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2379>

**[CVE-2025-0725] libcurl: Buffer Overflow in libcurl via zlib Integer Overflow (Severity: LOW)**

Package: libcurl4

Installed: 7.88.1-10+deb12u12

Fixed: %!s(<nil>)

When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the `CURLOPT\_ACCEPT\_ENCODING` option, **\*\*using zlib 1.2.0.3 or older\*\***, an attacker-controlled integer overflow would make libcurl perform a buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0725>

#### **[CVE-2024-2379] curl: QUIC certificate check bypass with wolfSSL (Severity: LOW)**

Package: libcurl4-openssl-dev

Installed: 7.88.1-10+deb12u12

Fixed: %!s(<nil>)

libcurl skips the certificate verification for a QUIC connection under certain conditions, when built to use wolfSSL. If told to use an unknown/bad cipher or curve, the error path accidentally skips the verification and returns OK, thus ignoring any certificate problems.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2379>

#### **[CVE-2025-0725] libcurl: Buffer Overflow in libcurl via zlib Integer Overflow (Severity: LOW)**

Package: libcurl4-openssl-dev

Installed: 7.88.1-10+deb12u12

Fixed: %!s(<nil>)

When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the `CURLOPT\_ACCEPT\_ENCODING` option, **\*\*using zlib 1.2.0.3 or older\*\***, an attacker-controlled integer overflow would make libcurl perform a buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0725>

#### **[CVE-2023-32570] VideoLAN dav1d before 1.2.0 has a thread\_task.c race condition that ca ... (Severity: MEDIUM)**

Package: libdav1d6

Installed: 1.0.0-2+deb12u1

Fixed: %!s(<nil>)

VideoLAN dav1d before 1.2.0 has a thread\_task.c race condition that can lead to an application crash, related to dav1d\_decode\_frame\_exit.

More Info: <https://avd.aquasec.com/nvd/cve-2023-32570>

#### **[CVE-2023-51792] Buffer Overflow vulnerability in libde265 v1.0.12 allows a local attac ... (Severity: MEDIUM)**

Package: libde265-0

Installed: 1.0.11-1+deb12u2

Fixed: %!s(<nil>)

Buffer Overflow vulnerability in libde265 v1.0.12 allows a local attacker to cause a denial of service via the allocation size exceeding the maximum supported size of 0x10000000000.

More Info: <https://avd.aquasec.com/nvd/cve-2023-51792>

**[CVE-2024-38949] Heap Buffer Overflow vulnerability in Libde265 v1.0.15 allows attacker ... (Severity: MEDIUM)**

Package: libde265-0

Installed: 1.0.11-1+deb12u2

Fixed: %!s(<nil>)

Heap Buffer Overflow vulnerability in Libde265 v1.0.15 allows attackers to crash the application via crafted payload to display444as420 function at sdl.cc

More Info: <https://avd.aquasec.com/nvd/cve-2024-38949>

**[CVE-2024-38950] Heap Buffer Overflow vulnerability in Libde265 v1.0.15 allows attacker ... (Severity: MEDIUM)**

Package: libde265-0

Installed: 1.0.11-1+deb12u2

Fixed: %!s(<nil>)

Heap Buffer Overflow vulnerability in Libde265 v1.0.15 allows attackers to crash the application via crafted payload to \_\_interceptor\_memcpy function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38950>

**[CVE-2021-46310] An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows at ... (Severity: MEDIUM)**

Package: libdjvulibre-dev

Installed: 3.5.28-2+b1

Fixed: %!s(<nil>)

An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46310>

**[CVE-2021-46312] An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in all ... (Severity: MEDIUM)**

Package: libdjvulibre-dev

Installed: 3.5.28-2+b1

Fixed: %!s(<nil>)

An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46312>

**[CVE-2021-46310] An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows at ... (Severity: MEDIUM)**

Package: libdjvulibre-text

Installed: 3.5.28-2

Fixed: %ls(<nil>)

An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46310>

**[CVE-2021-46312] An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in all ... (Severity: MEDIUM)**

Package: libdjvulibre-text

Installed: 3.5.28-2

Fixed: %ls(<nil>)

An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46312>

**[CVE-2021-46310] An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows at ... (Severity: MEDIUM)**

Package: libdjvulibre21

Installed: 3.5.28-2+b1

Fixed: %ls(<nil>)

An issue was discovered IW44Image.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46310>

**[CVE-2021-46312] An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in all ... (Severity: MEDIUM)**

Package: libdjvulibre21

Installed: 3.5.28-2+b1

Fixed: %ls(<nil>)

An issue was discovered IW44EncodeCodec.cpp in djvulibre 3.5.28 in allows attackers to cause a denial of service via divide by zero.

More Info: <https://avd.aquasec.com/nvd/cve-2021-46312>

**[CVE-2024-25260] elfutils: global-buffer-overflow exists in the function ebl\_machine\_flag\_name in eblmachineflagname.c (Severity: LOW)**

Package: libelf1

Installed: 0.188-2.1

Fixed: %ls(<nil>)

elfutils v0.189 was discovered to contain a NULL pointer dereference via the handle\_verdef() function at readelf.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25260>

**[CVE-2025-1352] elfutils: GNU elfutils eu-readelf libdw\_alloc.c \_\_libdw\_thread\_tail memory corruption**

## **(Severity: LOW)**

Package: libelf1

Installed: 0.188-2.1

Fixed: %!s(<nil>)

A vulnerability has been found in GNU elfutils 0.192 and classified as critical. This vulnerability affects the function `__libdw_thread_tail` in the library `libdw_alloc.c` of the component `eu-readelf`. The manipulation of the argument `w` leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `2636426a091bd6c6f7f02e49ab20d4cdc6bfc753`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1352>

## **[CVE-2025-1365] elfutils: GNU elfutils eu-readelf readelf.c process\_symtab buffer overflow (Severity: LOW)**

Package: libelf1

Installed: 0.188-2.1

Fixed: %!s(<nil>)

A vulnerability, which was classified as critical, was found in GNU elfutils 0.192. This affects the function `process_symtab` of the file `readelf.c` of the component `eu-readelf`. The manipulation of the argument `D/a` leads to buffer overflow. Local access is required to approach this attack. The exploit has been disclosed to the public and may be used. The identifier of the patch is `5e5c0394d82c53e97750fe7b18023e6f84157b81`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1365>

## **[CVE-2025-1371] elfutils: GNU elfutils eu-read readelf.c handle\_dynamic\_symtab null pointer dereference (Severity: LOW)**

Package: libelf1

Installed: 0.188-2.1

Fixed: %!s(<nil>)

A vulnerability has been found in GNU elfutils 0.192 and classified as problematic. This vulnerability affects the function `handle_dynamic_symtab` of the file `readelf.c` of the component `eu-read`. The manipulation leads to null pointer dereference. Attacking locally is a requirement. The exploit has been disclosed to the public and may be used. The patch is identified as `b38e562a4c907e08171c76b8b2def8464d5a104a`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1371>

## **[CVE-2025-1372] elfutils: GNU elfutils eu-readelf readelf.c print\_string\_section buffer overflow (Severity: LOW)**

Package: libelf1

Installed: 0.188-2.1

Fixed: %!s(<nil>)

A vulnerability was found in GNU elfutils 0.192. It has been declared as critical. Affected by this vulnerability is the function `dump_data_section/print_string_section` of the file `readelf.c` of the component `eu-readelf`. The manipulation of the argument `z/x` leads to buffer overflow. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The identifier of the patch is `73db9d2021cab9e23fd734b0a76a612d52a6f1db`. It is

recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1372>

**[CVE-2025-1376] elfutils: GNU elfutils eu-strip elf\_strptr.c elf\_strptr denial of service (Severity: LOW)**

Package: libelf1  
Installed: 0.188-2.1  
Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in GNU elfutils 0.192. This vulnerability affects the function `elf_strptr` in the library `/libelf/elf_strptr.c` of the component `eu-strip`. The manipulation leads to denial of service. It is possible to launch the attack on the local host. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `b16f441cca0a4841050e3215a9f120a6d8aea918`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1376>

**[CVE-2025-1377] elfutils: GNU elfutils eu-strip strip.c gelf\_getsymshndx denial of service (Severity: LOW)**

Package: libelf1  
Installed: 0.188-2.1  
Fixed: %!s(<nil>)

A vulnerability, which was classified as problematic, has been found in GNU elfutils 0.192. This issue affects the function `gelf_getsymshndx` of the file `strip.c` of the component `eu-strip`. The manipulation leads to denial of service. The attack needs to be approached locally. The exploit has been disclosed to the public and may be used. The identifier of the patch is `fbf1df9ca286de3323ae541973b08449f8d03aba`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1377>

**[CVE-2023-52425] expat: parsing large tokens can trigger a denial of service (Severity: HIGH)**

Package: libexpat1  
Installed: 2.5.0-1+deb12u1  
Fixed: %!s(<nil>)

`libexpat` through 2.5.0 allows a denial of service (resource consumption) because many full reparsings are required in the case of a large token for which multiple buffer fills are needed.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52425>

**[CVE-2024-8176] libexpat: expat: Improper Restriction of XML Entity Expansion Depth in libexpat (Severity: HIGH)**

Package: libexpat1  
Installed: 2.5.0-1+deb12u1  
Fixed: %!s(<nil>)

A stack overflow vulnerability exists in the `libexpat` library due to the way it handles recursive entity expansion in XML documents. When parsing an XML document with deeply nested entity references, `libexpat` can be forced to recurse indefinitely, exhausting the stack space and causing a crash. This issue could lead to denial of service (DoS) or, in some cases, exploitable memory corruption, depending on the environment and library usage.

More Info: <https://avd.aquasec.com/nvd/cve-2024-8176>

### **[CVE-2024-50602] libexpat: expat: DoS via XML\_ResumeParser (Severity: MEDIUM)**

Package: libexpat1

Installed: 2.5.0-1+deb12u1

Fixed: %!s(<nil>)

An issue was discovered in libexpat before 2.6.4. There is a crash within the XML\_ResumeParser function because XML\_StopParser can stop/suspend an unstarted parser.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50602>

### **[CVE-2023-52426] expat: recursive XML entity expansion vulnerability (Severity: LOW)**

Package: libexpat1

Installed: 2.5.0-1+deb12u1

Fixed: %!s(<nil>)

libexpat through 2.5.0 allows recursive XML Entity Expansion if XML\_DTD is undefined at compile time.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52426>

### **[CVE-2024-28757] expat: XML Entity Expansion (Severity: LOW)**

Package: libexpat1

Installed: 2.5.0-1+deb12u1

Fixed: %!s(<nil>)

libexpat through 2.6.1 allows an XML Entity Expansion attack when there is isolated use of external parsers (created via XML\_ExternalEntityParserCreate).

More Info: <https://avd.aquasec.com/nvd/cve-2024-28757>

### **[CVE-2023-52425] expat: parsing large tokens can trigger a denial of service (Severity: HIGH)**

Package: libexpat1-dev

Installed: 2.5.0-1+deb12u1

Fixed: %!s(<nil>)

libexpat through 2.5.0 allows a denial of service (resource consumption) because many full reparsings are required in the case of a large token for which multiple buffer fills are needed.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52425>

### **[CVE-2024-8176] libexpat: expat: Improper Restriction of XML Entity Expansion Depth in libexpat (Severity: HIGH)**

Package: libexpat1-dev

Installed: 2.5.0-1+deb12u1

Fixed: %!s(<nil>)

A stack overflow vulnerability exists in the libexpat library due to the way it handles recursive entity expansion in XML documents. When parsing an XML document with deeply nested entity references, libexpat can be forced to recurse indefinitely, exhausting the stack space and causing a crash. This issue could lead to denial of service (DoS) or, in some cases, exploitable memory corruption, depending on the environment and library usage.

More Info: <https://avd.aquasec.com/nvd/cve-2024-8176>



### **[CVE-2024-50602] libexpat: expat: DoS via XML\_ResumeParser (Severity: MEDIUM)**

Package: libexpat1-dev  
Installed: 2.5.0-1+deb12u1  
Fixed: %!s(<nil>)

An issue was discovered in libexpat before 2.6.4. There is a crash within the XML\_ResumeParser function because XML\_StopParser can stop/suspend an unstarted parser.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50602>

### **[CVE-2023-52426] expat: recursive XML entity expansion vulnerability (Severity: LOW)**

Package: libexpat1-dev  
Installed: 2.5.0-1+deb12u1  
Fixed: %!s(<nil>)

libexpat through 2.5.0 allows recursive XML Entity Expansion if XML\_DTD is undefined at compile time.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52426>

### **[CVE-2024-28757] expat: XML Entity Expansion (Severity: LOW)**

Package: libexpat1-dev  
Installed: 2.5.0-1+deb12u1  
Fixed: %!s(<nil>)

libexpat through 2.6.1 allows an XML Entity Expansion attack when there is isolated use of external parsers (created via XML\_ExternalEntityParserCreate).

More Info: <https://avd.aquasec.com/nvd/cve-2024-28757>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libgcc-12-dev  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libgcc-12-dev  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized

local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libgcc-s1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libgcc-s1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2024-2236] libgcrypt: vulnerable to Marvin Attack (Severity: MEDIUM)**

Package: libgcrypt20

Installed: 1.10.1-3

Fixed: %!s(<nil>)

A timing-based side-channel flaw was found in libgcrypt's RSA implementation. This issue may allow a remote attacker

to initiate a Bleichenbacher-style attack, which can lead to the decryption of RSA ciphertexts.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2236>

**[CVE-2018-6829] libgcrypt: ElGamal implementation doesn't have semantic security due to incorrectly encoded plaintexts possibly allowing to obtain sensitive information (Severity: LOW)**

Package: libgcrypt20

Installed: 1.10.1-3

Fixed: %ls(<nil>)

cipher/elgamal.c in Libgcrypt through 1.8.2, when used to encrypt messages directly, improperly encodes plaintexts, which allows attackers to obtain sensitive information by reading ciphertext data (i.e., it does not have semantic security in face of a ciphertext-only attack). The Decisional Diffie-Hellman (DDH) assumption does not hold for Libgcrypt's ElGamal implementation.

More Info: <https://avd.aquasec.com/nvd/cve-2018-6829>

**[CVE-2012-0039] glib2: hash table collisions CPU usage DoS (Severity: LOW)**

Package: libglib2.0-0

Installed: 2.74.6-2+deb12u5

Fixed: %ls(<nil>)

GLib 2.31.8 and earlier, when the `g_str_hash` function is used, computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table. NOTE: this issue may be disputed by the vendor; the existence of the `g_str_hash` function is not a vulnerability in the library, because callers of `g_hash_table_new` and `g_hash_table_new_full` can specify an arbitrary hash function that is appropriate for the application.

More Info: <https://avd.aquasec.com/nvd/cve-2012-0039>

**[CVE-2025-3360] glibc: GLib prior to 2.82.5 is vulnerable to integer overflow and buffer under-read when parsing a very long invalid ISO 8601 timestamp with `g_date_time_new_from_iso8601()`. (Severity: LOW)**

Package: libglib2.0-0

Installed: 2.74.6-2+deb12u5

Fixed: %ls(<nil>)

A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 timestamp with the `g_date_time_new_from_iso8601()` function.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3360>

**[CVE-2012-0039] glib2: hash table collisions CPU usage DoS (Severity: LOW)**

Package: libglib2.0-bin

Installed: 2.74.6-2+deb12u5

Fixed: %ls(<nil>)

GLib 2.31.8 and earlier, when the `g_str_hash` function is used, computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table. NOTE: this issue may be disputed by the

vendor; the existence of the `g_str_hash` function is not a vulnerability in the library, because callers of `g_hash_table_new` and `g_hash_table_new_full` can specify an arbitrary hash function that is appropriate for the application.

More Info: <https://avd.aquasec.com/nvd/cve-2012-0039>

**[CVE-2025-3360] glibc: GLib prior to 2.82.5 is vulnerable to integer overflow and buffer under-read when parsing a very long invalid ISO 8601 timestamp with `g_date_time_new_from_iso8601()`. (Severity: LOW)**

Package: `libglib2.0-bin`  
Installed: 2.74.6-2+deb12u5  
Fixed: %ls(<nil>)

A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 timestamp with the `g_date_time_new_from_iso8601()` function.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3360>

**[CVE-2012-0039] glib2: hash table collisions CPU usage DoS (Severity: LOW)**

Package: `libglib2.0-data`  
Installed: 2.74.6-2+deb12u5  
Fixed: %ls(<nil>)

GLib 2.31.8 and earlier, when the `g_str_hash` function is used, computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table. NOTE: this issue may be disputed by the vendor; the existence of the `g_str_hash` function is not a vulnerability in the library, because callers of `g_hash_table_new` and `g_hash_table_new_full` can specify an arbitrary hash function that is appropriate for the application.

More Info: <https://avd.aquasec.com/nvd/cve-2012-0039>

**[CVE-2025-3360] glibc: GLib prior to 2.82.5 is vulnerable to integer overflow and buffer under-read when parsing a very long invalid ISO 8601 timestamp with `g_date_time_new_from_iso8601()`. (Severity: LOW)**

Package: `libglib2.0-data`  
Installed: 2.74.6-2+deb12u5  
Fixed: %ls(<nil>)

A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 timestamp with the `g_date_time_new_from_iso8601()` function.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3360>

**[CVE-2012-0039] glib2: hash table collisions CPU usage DoS (Severity: LOW)**

Package: `libglib2.0-dev`  
Installed: 2.74.6-2+deb12u5  
Fixed: %ls(<nil>)

GLib 2.31.8 and earlier, when the `g_str_hash` function is used, computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU

consumption) via crafted input to an application that maintains a hash table. NOTE: this issue may be disputed by the vendor; the existence of the `g_str_hash` function is not a vulnerability in the library, because callers of `g_hash_table_new` and `g_hash_table_new_full` can specify an arbitrary hash function that is appropriate for the application.

More Info: <https://avd.aquasec.com/nvd/cve-2012-0039>

**[CVE-2025-3360] glibc: GLib prior to 2.82.5 is vulnerable to integer overflow and buffer under-read when parsing a very long invalid ISO 8601 timestamp with `g_date_time_new_from_iso8601()`. (Severity: LOW)**

Package: `libglib2.0-dev`  
Installed: 2.74.6-2+deb12u5  
Fixed: %s(<nil>)

A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 timestamp with the `g_date_time_new_from_iso8601()` function.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3360>

**[CVE-2012-0039] glib2: hash table collisions CPU usage DoS (Severity: LOW)**

Package: `libglib2.0-dev-bin`  
Installed: 2.74.6-2+deb12u5  
Fixed: %s(<nil>)

GLib 2.31.8 and earlier, when the `g_str_hash` function is used, computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table. NOTE: this issue may be disputed by the vendor; the existence of the `g_str_hash` function is not a vulnerability in the library, because callers of `g_hash_table_new` and `g_hash_table_new_full` can specify an arbitrary hash function that is appropriate for the application.

More Info: <https://avd.aquasec.com/nvd/cve-2012-0039>

**[CVE-2025-3360] glibc: GLib prior to 2.82.5 is vulnerable to integer overflow and buffer under-read when parsing a very long invalid ISO 8601 timestamp with `g_date_time_new_from_iso8601()`. (Severity: LOW)**

Package: `libglib2.0-dev-bin`  
Installed: 2.74.6-2+deb12u5  
Fixed: %s(<nil>)

A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 timestamp with the `g_date_time_new_from_iso8601()` function.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3360>

**[CVE-2011-3389] HTTPS: block-wise chosen-plaintext attack against SSL/TLS (BEAST) (Severity: LOW)**

Package: `libgnutls30`  
Installed: 3.7.9-2+deb12u4  
Fixed: %s(<nil>)

The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other products, encrypts data by using CBC mode with chained initialization vectors, which allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.

More Info: <https://avd.aquasec.com/nvd/cve-2011-3389>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libgomp1

Installed: 12.2.0-14

Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libgomp1

Installed: 12.2.0-14

Fixed: %ls(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2017-13716] binutils: Memory leak with the C++ symbol demangler routine in libiberty (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

The C++ symbol demangler routine in cplus-dem.c in libiberty, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka libbfd).

More Info: <https://avd.aquasec.com/nvd/cve-2017-13716>

### **[CVE-2018-20673] libiberty: Integer overflow in demangle\_template() function (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

The demangle\_template function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by nm.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20673>

### **[CVE-2018-20712] libiberty: heap-based buffer over-read in d\_expression\_1 (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

A heap-based buffer over-read exists in the function d\_expression\_1 in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31.1. A crafted input can cause segmentation faults, leading to denial-of-service, as demonstrated by c++filt.

More Info: <https://avd.aquasec.com/nvd/cve-2018-20712>

### **[CVE-2018-9996] binutils: Stack-overflow in libiberty/cplus-dem.c causes crash (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there are recursive stack frames: demangle\_template\_value\_parm, demangle\_integral\_value, and demangle\_expression.

More Info: <https://avd.aquasec.com/nvd/cve-2018-9996>

### **[CVE-2021-32256] binutils: stack-overflow issue in demangle\_type in rust-demangle.c. (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.36. It is a stack-overflow issue in demangle\_type in rust-demangle.c.

More Info: <https://avd.aquasec.com/nvd/cve-2021-32256>

### **[CVE-2023-1972] binutils: Illegal memory access when accessing a zero-length verdef table (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

A potential heap based buffer overflow was found in `_bfd_elf_slurp_version_tables()` in `bfd/elf.c`. This may lead to loss of availability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1972>

**[CVE-2024-53589] binutils: objdump: buffer Overflow in the BFD library's handling of tekhex format files (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

GNU objdump 2.43 is vulnerable to Buffer Overflow in the BFD (Binary File Descriptor) library's handling of tekhex format files.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53589>

**[CVE-2024-57360] binutils: nm: potential segmentation fault when displaying symbols without version info (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

<https://www.gnu.org/software/binutils/> nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57360>

**[CVE-2025-0840] binutils: GNU Binutils objdump.c disassemble\_bytes stack-based overflow (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function `disassemble_bytes` of the file `binutils/objdump.c`. The manipulation of the argument `buf` leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. The identifier of the patch is `baac6c221e9d69335bf41366a1c7d87d8ab2f893`. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0840>

**[CVE-2025-1147] binutils: GNU Binutils nm nm.c internal\_strlen buffer overflow (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function `__sanitizer::internal_strlen` of the file `binutils/nm.c` of the component `nm`. The manipulation of the argument `const` leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The



exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1147>

#### **[CVE-2025-1148] binutils: GNU Binutils ld ldelfgen.c link\_order\_scan memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function `link_order_scan` of the file `ld/ldelfgen.c` of the component `ld`. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1148>

#### **[CVE-2025-1149] binutils: GNU Binutils ld xmalloc.c xstrdup memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been classified as problematic. This affects the function `xstrdup` of the file `libiberty/xmalloc.c` of the component `ld`. The manipulation leads to memory leak. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1149>

#### **[CVE-2025-1150] binutils: GNU Binutils ld libbfd.c bfd\_malloc memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function `bfd_malloc` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise `ld`. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1150>

#### **[CVE-2025-1151] binutils: GNU Binutils ld xmemdup.c xmemdup memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function

xmempdup of the file xmempdup.c of the component ld. The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1151>

#### **[CVE-2025-1152] binutils: GNU Binutils ld xstrdup.c xstrdup memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function xstrdup of the file xstrdup.c of the component ld. The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."

More Info: <https://avd.aquasec.com/nvd/cve-2025-1152>

#### **[CVE-2025-1153] binutils: GNU Binutils format.c bfd\_set\_format memory corruption (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function bfd\_set\_format of the file format.c. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is 8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150. It is recommended to upgrade the affected component.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1153>

#### **[CVE-2025-1176] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec heap-based overflow (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %!s(<nil>)

A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function \_bfd\_elf\_gc\_mark\_rsec of the file elflink.c of the component ld. The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named f9978defb6fab0bd8583942d97c112b0932ac814. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1176>

#### **[CVE-2025-1178] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function `bfd_putl64` of the file `libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is `75086e9de1707281172cc77f178e7949a4414ed0`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1178>

#### **[CVE-2025-1179] binutils: GNU Binutils ld libbfd.c bfd\_putl64 memory corruption (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability was found in GNU Binutils 2.43. It has been rated as critical. Affected by this issue is the function `bfd_putl64` of the file `bfd/libbfd.c` of the component `ld`. The manipulation leads to memory corruption. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer explains, that "[t]his bug has been fixed at some point between the 2.43 and 2.44 releases".

More Info: <https://avd.aquasec.com/nvd/cve-2025-1179>

#### **[CVE-2025-1180] binutils: GNU Binutils ld elf-eh-frame.c \_bfd\_elf\_write\_section\_elf\_frame memory corruption (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as problematic has been found in GNU Binutils 2.43. This affects the function `_bfd_elf_write_section_elf_frame` of the file `bfd/elf-eh-frame.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to initiate the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1180>

#### **[CVE-2025-1181] binutils: GNU Binutils ld elflink.c \_bfd\_elf\_gc\_mark\_rsec memory corruption (Severity: LOW)**

Package: `libgprofng0`

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function `_bfd_elf_gc_mark_rsec` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is `931494c9a89558acb36a03a340c01726545eef24`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1181>

**[CVE-2025-1182] binutils: GNU Binutils ld elflink.c bfd\_elf\_reloc\_symbol\_deleted\_p memory corruption (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function `bfd_elf_reloc_symbol_deleted_p` of the file `bfd/elflink.c` of the component `ld`. The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as `b425859021d17adf62f06fb904797cf8642986ad`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1182>

**[CVE-2025-3198] binutils: GNU Binutils objdump bucomm.c display\_info memory leak (Severity: LOW)**

Package: libgprofng0

Installed: 2.40-2

Fixed: %ls(<nil>)

A vulnerability has been found in GNU Binutils 2.43/2.44 and classified as problematic. Affected by this vulnerability is the function `display_info` of the file `binutils/bucomm.c` of the component `objdump`. The manipulation leads to memory leak. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The patch is named `ba6ad3a18cb26b79e0e3b84c39f707535bbc344d`. It is recommended to apply a patch to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-3198>

**[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libgssapi-krb5-2

Installed: 1.20.1-2+deb12u2

Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in `/krb5/src/kdc/ndr.c`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

**[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libgssapi-krb5-2

Installed: 1.20.1-2+deb12u2

Fixed: %ls(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause `kadmind` to write beyond the end of the mapped region for the `iprop` log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

**[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity: LOW)**

Package: libgssapi-krb5-2

Installed: 1.20.1-2+deb12u2

Fixed: %ls(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

**[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libgssapi-krb5-2  
Installed: 1.20.1-2+deb12u2  
Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

**[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libgssapi-krb5-2  
Installed: 1.20.1-2+deb12u2  
Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

**[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libgssrpc4  
Installed: 1.20.1-2+deb12u2  
Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

**[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libgssrpc4  
Installed: 1.20.1-2+deb12u2  
Fixed: %ls(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

**[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity: LOW)**

Package: libgssrpc4  
Installed: 1.20.1-2+deb12u2  
Fixed: %ls(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

#### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libgssrpc4

Installed: 1.20.1-2+deb12u2

Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

#### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libgssrpc4

Installed: 1.20.1-2+deb12u2

Fixed: %ls(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

#### **[CVE-2023-25193] harfbuzz: allows attackers to trigger $O(n^2)$ growth via consecutive marks (Severity: HIGH)**

Package: libharfbuzz0b

Installed: 6.0.0+dfsg-3

Fixed: %ls(<nil>)

hb-ot-layout-gsubpos.hh in HarfBuzz through 6.0.0 allows attackers to trigger  $O(n^2)$  growth via consecutive marks during the process of looking back for base glyphs when attaching marks.

More Info: <https://avd.aquasec.com/nvd/cve-2023-25193>

#### **[CVE-2025-29482] Buffer Overflow vulnerability in libheif 1.19.7 allows a local attacker to execute arbitrary code via the SAO (Sample Adaptive Offset) processing of libde265. (Severity: MEDIUM)**

Package: libheif1

Installed: 1.15.1-1+deb12u1

Fixed: %ls(<nil>)

Buffer Overflow vulnerability in libheif 1.19.7 allows a local attacker to execute arbitrary code via the SAO (Sample Adaptive Offset) processing of libde265.

More Info: <https://avd.aquasec.com/nvd/cve-2025-29482>

#### **[CVE-2023-49463] libheif v1.17.5 was discovered to contain a segmentation violation via ... (Severity: LOW)**

Package: libheif1

Installed: 1.15.1-1+deb12u1

Fixed: %!s(<nil>)

libheif v1.17.5 was discovered to contain a segmentation violation via the function find\_exif\_tag at /libheif/exif.cc.

More Info: <https://avd.aquasec.com/nvd/cve-2023-49463>

**[CVE-2024-25269] libheif <= 1.17.6 contains a memory leak in the function JpegEncoder:: ... (Severity: LOW)**

Package: libheif1

Installed: 1.15.1-1+deb12u1

Fixed: %!s(<nil>)

libheif <= 1.17.6 contains a memory leak in the function JpegEncoder::Encode. This flaw allows an attacker to cause a denial of service attack.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25269>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libitm1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libitm1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2020-36325] jansson: out-of-bounds read in json\_loads() due to a parsing error (Severity: LOW)**

Package: libjansson4

Installed: 2.14-2

Fixed: %!s(<nil>)

An issue was discovered in Jansson through 2.13.1. Due to a parsing error in json\_loads, there's an out-of-bounds read-access bug. NOTE: the vendor reports that this only occurs when a programmer fails to follow the API specification

More Info: <https://avd.aquasec.com/nvd/cve-2020-36325>

**[CVE-2017-9937] libtiff: memory malloc failure in tif\_jbig.c could cause DOS. (Severity: LOW)**

Package: libjbig-dev

Installed: 2.1-6.1

Fixed: %!s(<nil>)

In LibTIFF 4.0.8, there is a memory malloc failure in tif\_jbig.c. A crafted TIFF document can lead to an abort resulting in a remote denial of service attack.

More Info: <https://avd.aquasec.com/nvd/cve-2017-9937>

**[CVE-2017-9937] libtiff: memory malloc failure in tif\_jbig.c could cause DOS. (Severity: LOW)**

Package: libjbig0

Installed: 2.1-6.1

Fixed: %!s(<nil>)

In LibTIFF 4.0.8, there is a memory malloc failure in tif\_jbig.c. A crafted TIFF document can lead to an abort resulting in a remote denial of service attack.

More Info: <https://avd.aquasec.com/nvd/cve-2017-9937>

**[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libk5crypto3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

**[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libk5crypto3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

**[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**



## LOW)

Package: libk5crypto3  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libk5crypto3  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libk5crypto3  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkadm5clnt-mit12  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkadm5clnt-mit12  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkadm5clnt-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkadm5clnt-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkadm5clnt-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkadm5srv-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkadm5srv-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkadm5srv-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkadm5srv-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkadm5srv-mit12

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkdb5-10

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkdb5-10

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkdb5-10

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkdb5-10

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkdb5-10

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkrb5-3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkrb5-3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkrb5-3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkrb5-3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkrb5-3

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkrb5-dev

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkrb5-dev

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkrb5-dev

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkrb5-dev

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkrb5-dev

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2024-26462] krb5: Memory leak at /krb5/src/kdc/ndr.c (Severity: MEDIUM)**

Package: libkrb5support0

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26462>

### **[CVE-2025-24528] krb5: overflow when calculating ulog block size (Severity: MEDIUM)**

Package: libkrb5support0

Installed: 1.20.1-2+deb12u2

Fixed: %!s(<nil>)

A flaw was found in krb5. With incremental propagation enabled, an authenticated attacker can cause kadmind to write beyond the end of the mapped region for the iprop log file. This issue can trigger a process crash and lead to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24528>

### **[CVE-2018-5709] krb5: integer overflow in dbentry->n\_key\_data in kadmin/dbutil/dump.c (Severity:**

## LOW)

Package: libkrb5support0  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n\_key\_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.

More Info: <https://avd.aquasec.com/nvd/cve-2018-5709>

### **[CVE-2024-26458] krb5: Memory leak at /krb5/src/lib/rpc/pmap\_rmt.c (Severity: LOW)**

Package: libkrb5support0  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/rpc/pmap\_rmt.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26458>

### **[CVE-2024-26461] krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c (Severity: LOW)**

Package: libkrb5support0  
Installed: 1.20.1-2+deb12u2  
Fixed: %!s(<nil>)

Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26461>

### **[CVE-2025-29070] A heap buffer overflow vulnerability has been identified in thesmooth2 ... (Severity: LOW)**

Package: liblcms2-2  
Installed: 2.14-2  
Fixed: %!s(<nil>)

A heap buffer overflow vulnerability has been identified in thesmooth2() in cmsgamma.c in lcms2-2.16 which allows a remote attacker to cause a denial of service. NOTE: the Supplier disputes this because "this is not exploitable as this function is never called on normal color management, is there only as a helper for low-level programming and investigation."

More Info: <https://avd.aquasec.com/nvd/cve-2025-29070>

### **[CVE-2025-29070] A heap buffer overflow vulnerability has been identified in thesmooth2 ... (Severity: LOW)**

Package: liblcms2-dev  
Installed: 2.14-2  
Fixed: %!s(<nil>)

A heap buffer overflow vulnerability has been identified in thesmooth2() in cmsgamma.c in lcms2-2.16 which allows a remote attacker to cause a denial of service. NOTE: the Supplier disputes this because "this is not exploitable as this

function is never called on normal color management, is there only as a helper for low-level programming and investigation."

More Info: <https://avd.aquasec.com/nvd/cve-2025-29070>

**[CVE-2023-2953] openldap: null pointer dereference in ber\_memalloc\_x function (Severity: HIGH)**

Package: libldap-2.5-0

Installed: 2.5.13+dfsg-5

Fixed: %ls(<nil>)

A vulnerability was found in openldap. This security flaw causes a null pointer dereference in ber\_memalloc\_x() function.

More Info: <https://avd.aquasec.com/nvd/cve-2023-2953>

**[CVE-2015-3276] openldap: incorrect multi-keyword mode cipherstring parsing (Severity: LOW)**

Package: libldap-2.5-0

Installed: 2.5.13+dfsg-5

Fixed: %ls(<nil>)

The nss\_parse\_ciphers function in libraries/libldap/tls\_m.c in OpenLDAP does not properly parse OpenSSL-style multi-keyword mode cipher strings, which might cause a weaker than intended cipher to be used and allow remote attackers to have unspecified impact via unknown vectors.

More Info: <https://avd.aquasec.com/nvd/cve-2015-3276>

**[CVE-2017-14159] openldap: Privilege escalation via PID file manipulation (Severity: LOW)**

Package: libldap-2.5-0

Installed: 2.5.13+dfsg-5

Fixed: %ls(<nil>)

slapd in OpenLDAP 2.4.45 and earlier creates a PID file after dropping privileges to a non-root account, which might allow local users to kill arbitrary processes by leveraging access to this non-root account for PID file modification before a root script executes a "kill `cat /pathname`" command, as demonstrated by openldap-initscript.

More Info: <https://avd.aquasec.com/nvd/cve-2017-14159>

**[CVE-2017-17740] openldap: contrib/slapd-modules/nops/nops.c attempts to free stack buffer allowing remote attackers to cause a denial of service (Severity: LOW)**

Package: libldap-2.5-0

Installed: 2.5.13+dfsg-5

Fixed: %ls(<nil>)

contrib/slapd-modules/nops/nops.c in OpenLDAP through 2.4.45, when both the nops module and the memberof overlay are enabled, attempts to free a buffer that was allocated on the stack, which allows remote attackers to cause a denial of service (slapd crash) via a member MODDN operation.

More Info: <https://avd.aquasec.com/nvd/cve-2017-17740>

**[CVE-2020-15719] openldap: Certificate validation incorrectly matches name against CN-ID (Severity: LOW)**

Package: libldap-2.5-0



Installed: 2.5.13+dfsg-5  
Fixed: %!s(<nil>)

libldap in certain third-party OpenLDAP packages has a certificate-validation flaw when the third-party package is asserting RFC6125 support. It considers CN even when there is a non-matching subjectAltName (SAN). This is fixed in, for example, openldap-2.4.46-10.el8 in Red Hat Enterprise Linux.

More Info: <https://avd.aquasec.com/nvd/cve-2020-15719>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: liblsan0  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: liblsan0  
Installed: 12.2.0-14  
Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-6-arch-config  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

#### **[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

#### **[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

#### **[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

#### **[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

#### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-6-arch-config

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main

JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-6-headers

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail,

which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

#### **[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

#### **[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

#### **[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

#### **[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

#### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickcore-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

#### **[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

#### **[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

#### **[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

#### **[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

#### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**



Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-6.q16-6-extra  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-6.q16-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-6.q16-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-6.q16-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-6.q16-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickcore-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickcore-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickcore-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickcore-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickcore-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickcore-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickcore-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickcore-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickcore-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickcore-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

#### **[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

#### **[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

#### **[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

#### **[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

#### **[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickwand-6-headers  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickwand-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickwand-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickwand-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickwand-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickwand-6.q16-6

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**



Package: libmagickwand-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickwand-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickwand-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickwand-6.q16-6  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %ls(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

#### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

#### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

#### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickwand-6.q16-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

#### **[CVE-2005-0406] A design flaw in image processing software that modifies JPEG images m ... (Severity: LOW)**

Package: libmagickwand-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A design flaw in image processing software that modifies JPEG images might not modify the original EXIF thumbnail, which could lead to an information leak of potentially sensitive visual information that had been removed from the main JPEG image.

More Info: <https://avd.aquasec.com/nvd/cve-2005-0406>

**[CVE-2008-3134] GraphicsMagick/ImageMagick: multiple crash or DoS issues (Severity: LOW)**

Package: libmagickwand-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

Multiple unspecified vulnerabilities in GraphicsMagick before 1.2.4 allow remote attackers to cause a denial of service (crash, infinite loop, or memory consumption) via (a) unspecified vectors in the (1) AVI, (2) AVS, (3) DCM, (4) EPT, (5) FITS, (6) MTV, (7) PALM, (8) RLA, and (9) TGA decoder readers; and (b) the GetImageCharacteristics function in magick/image.c, as reachable from a crafted (10) PNG, (11) JPEG, (12) BMP, or (13) TIFF file.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3134>

**[CVE-2016-8678] ImageMagick: Heap-buffer overflow in IsPixelMonochrome (Severity: LOW)**

Package: libmagickwand-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The IsPixelMonochrome function in MagickCore/pixel-accessor.h in ImageMagick 7.0.3.0 allows remote attackers to cause a denial of service (out-of-bounds read and crash) via a crafted file. NOTE: the vendor says "This is a Q64 issue and we do not support Q64."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8678>

**[CVE-2017-11754] ImageMagick: Memory leak in WritePICONImage function (Severity: LOW)**

Package: libmagickwand-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an OpenPixelCache call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11754>

**[CVE-2017-11755] ImageMagick: Memory leak in WritePICONImage function via mishandled AcquireSemaphoreInfo call (Severity: LOW)**

Package: libmagickwand-dev

Installed: 8:6.9.11.60+dfsg-1.6+deb12u2

Fixed: %!s(<nil>)

The WritePICONImage function in coders/xpm.c in ImageMagick 7.0.6-4 allows remote attackers to cause a denial of service (memory leak) via a crafted file that is mishandled in an AcquireSemaphoreInfo call.

More Info: <https://avd.aquasec.com/nvd/cve-2017-11755>

**[CVE-2017-7275] ImageMagick: Memory allocation failure in AcquireMagickMemory (incomplete fix for CVE-2016-8866) (Severity: LOW)**

Package: libmagickwand-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

The ReadPCXImage function in coders/pcx.c in ImageMagick 7.0.4.9 allows remote attackers to cause a denial of service (attempted large memory allocation and application crash) via a crafted file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-8862 and CVE-2016-8866.

More Info: <https://avd.aquasec.com/nvd/cve-2017-7275>

### **[CVE-2018-15607] ImageMagick: CPU Exhaustion via crafted input file (Severity: LOW)**

Package: libmagickwand-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

In ImageMagick 7.0.8-11 Q16, a tiny input file 0x50 0x36 0x36 0x36 0x36 0x4c 0x36 0x38 0x36 0x36 0x36 0x36 0x36 0x1f 0x35 0x50 0x00 can result in a hang of several minutes during which CPU and memory resources are consumed until ultimately an attempted large memory allocation fails. Remote attackers could leverage this vulnerability to cause a denial of service via a crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2018-15607>

### **[CVE-2021-20311] ImageMagick: Division by zero in sRGBTransformImage() in MagickCore/colospace.c (Severity: LOW)**

Package: libmagickwand-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A flaw was found in ImageMagick in versions before 7.0.11, where a division by zero in sRGBTransformImage() in the MagickCore/colospace.c may trigger undefined behavior via a crafted image file that is submitted by an attacker processed by an application using ImageMagick. The highest threat from this vulnerability is to system availability.

More Info: <https://avd.aquasec.com/nvd/cve-2021-20311>

### **[CVE-2023-34152] ImageMagick: RCE (shell command injection) vulnerability in OpenBlob with --enable-pipes configured (Severity: LOW)**

Package: libmagickwand-dev  
Installed: 8:6.9.11.60+dfsg-1.6+deb12u2  
Fixed: %!s(<nil>)

A vulnerability was found in ImageMagick. This security flaw cause a remote code execution vulnerability in OpenBlob with --enable-pipes configured.

More Info: <https://avd.aquasec.com/nvd/cve-2023-34152>

### **[CVE-2023-52969] mariadb: MariaDB Server Crash Due to Empty Backtrace Log (Severity: MEDIUM)**

Package: libmariadb-dev  
Installed: 1:10.11.11-0+deb12u1  
Fixed: %!s(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, and 11.0 through 11.0.\* can sometimes crash with an empty backtrace log. This may be related to make\_aggr\_tables\_info and optimize\_stage2.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52969>

**[CVE-2023-52970] mariadb: MariaDB Server Crash via Item\_direct\_view\_ref (Severity: MEDIUM)**

Package: libmariadb-dev

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, 11.0 through 11.0.\*, and 11.1 through 11.4.\* crashes in Item\_direct\_view\_ref::derived\_field\_transformer\_for\_where.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52970>

**[CVE-2023-52971] mariadb: MariaDB Server Crash (Severity: MEDIUM)**

Package: libmariadb-dev

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.10 through 10.11.\* and 11.0 through 11.4.\* crashes in JOIN::fix\_allSplittings\_in\_plan.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52971>

**[CVE-2023-52969] mariadb: MariaDB Server Crash Due to Empty Backtrace Log (Severity: MEDIUM)**

Package: libmariadb-dev-compat

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, and 11.0 through 11.0.\* can sometimes crash with an empty backtrace log. This may be related to make\_aggr\_tables\_info and optimize\_stage2.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52969>

**[CVE-2023-52970] mariadb: MariaDB Server Crash via Item\_direct\_view\_ref (Severity: MEDIUM)**

Package: libmariadb-dev-compat

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, 11.0 through 11.0.\*, and 11.1 through 11.4.\* crashes in Item\_direct\_view\_ref::derived\_field\_transformer\_for\_where.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52970>

**[CVE-2023-52971] mariadb: MariaDB Server Crash (Severity: MEDIUM)**

Package: libmariadb-dev-compat

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.10 through 10.11.\* and 11.0 through 11.4.\* crashes in JOIN::fix\_allSplittings\_in\_plan.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52971>

**[CVE-2023-52969] mariadb: MariaDB Server Crash Due to Empty Backtrace Log (Severity: MEDIUM)**

Package: libmariadb3  
Installed: 1:10.11.11-0+deb12u1  
Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, and 11.0 through 11.0.\* can sometimes crash with an empty backtrace log. This may be related to make\_aggr\_tables\_info and optimize\_stage2.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52969>

**[CVE-2023-52970] mariadb: MariaDB Server Crash via Item\_direct\_view\_ref (Severity: MEDIUM)**

Package: libmariadb3  
Installed: 1:10.11.11-0+deb12u1  
Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, 11.0 through 11.0.\*, and 11.1 through 11.4.\* crashes in Item\_direct\_view\_ref::derived\_field\_transformer\_for\_where.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52970>

**[CVE-2023-52971] mariadb: MariaDB Server Crash (Severity: MEDIUM)**

Package: libmariadb3  
Installed: 1:10.11.11-0+deb12u1  
Fixed: %ls(<nil>)

MariaDB Server 10.10 through 10.11.\* and 11.0 through 11.4.\* crashes in JOIN::fix\_allSplittings\_in\_plan.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52971>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libmount-dev  
Installed: 2.38.1-5+deb12u3  
Fixed: %ls(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libmount1  
Installed: 2.38.1-5+deb12u3  
Fixed: %ls(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libncurses-dev

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libncurses5-dev

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libncurses6

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libncursesw5-dev

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libncursesw6

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-5841] OpenEXR: Heap Overflow in Scanline Deep Data Parsing (Severity: CRITICAL)**

Package: libopenexr-3-1-30

Installed: 3.1.5-5

Fixed: %!s(<nil>)



Due to a failure in validating the number of scanline samples of a OpenEXR file containing deep scanline data, Academy Software Foundation OpenEXR image parsing library version 3.2.1 and prior is susceptible to a heap-based buffer overflow vulnerability. This issue was resolved as of versions v3.2.2 and v3.1.12 of the affected library.

More Info: <https://avd.aquasec.com/nvd/cve-2023-5841>

**[CVE-2017-14988] OpenEXR: Excessive memory allocation in Header::readfrom (Severity: LOW)**

Package: libopenexr-3-1-30

Installed: 3.1.5-5

Fixed: %!s(<nil>)

Header::readfrom in IlmImf/ImfHeader.cpp in OpenEXR 2.2.0 allows remote attackers to cause a denial of service (excessive memory allocation) via a crafted file that is accessed with the ImfOpenInputFile function in IlmImf/ImfCRgbaFile.cpp. NOTE: The maintainer and multiple third parties believe that this vulnerability isn't valid

More Info: <https://avd.aquasec.com/nvd/cve-2017-14988>

**[CVE-2024-31047] An issue in Academy Software Foundation openexr v.3.2.3 and before all ... (Severity: UNKNOWN)**

Package: libopenexr-3-1-30

Installed: 3.1.5-5

Fixed: %!s(<nil>)

An issue in Academy Software Foundation openexr v.3.2.3 and before allows a local attacker to cause a denial of service (DoS) via the convert function of exrmultipart.cpp.

More Info: <https://avd.aquasec.com/nvd/cve-2024-31047>

**[CVE-2023-5841] OpenEXR: Heap Overflow in Scanline Deep Data Parsing (Severity: CRITICAL)**

Package: libopenexr-dev

Installed: 3.1.5-5

Fixed: %!s(<nil>)

Due to a failure in validating the number of scanline samples of a OpenEXR file containing deep scanline data, Academy Software Foundation OpenEXR image parsing library version 3.2.1 and prior is susceptible to a heap-based buffer overflow vulnerability. This issue was resolved as of versions v3.2.2 and v3.1.12 of the affected library.

More Info: <https://avd.aquasec.com/nvd/cve-2023-5841>

**[CVE-2017-14988] OpenEXR: Excessive memory allocation in Header::readfrom (Severity: LOW)**

Package: libopenexr-dev

Installed: 3.1.5-5

Fixed: %!s(<nil>)

Header::readfrom in IlmImf/ImfHeader.cpp in OpenEXR 2.2.0 allows remote attackers to cause a denial of service (excessive memory allocation) via a crafted file that is accessed with the ImfOpenInputFile function in IlmImf/ImfCRgbaFile.cpp. NOTE: The maintainer and multiple third parties believe that this vulnerability isn't valid

More Info: <https://avd.aquasec.com/nvd/cve-2017-14988>

**[CVE-2024-31047] An issue in Academy Software Foundation openexr v.3.2.3 and before all ... (Severity: UNKNOWN)**

Package: libopenexr-dev  
Installed: 3.1.5-5  
Fixed: %ls(<nil>)

An issue in Academy Software Foundation openexr v.3.2.3 and before allows a local attacker to cause a denial of service (DoS) via the convert function of exrmultipart.cpp.

More Info: <https://avd.aquasec.com/nvd/cve-2024-31047>

**[CVE-2023-39327] openjpeg: Malicious files can cause the program to enter a large loop (Severity: MEDIUM)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

A flaw was found in OpenJPEG. Maliciously constructed pictures can cause the program to enter a large loop and continuously print warning messages on the terminal.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39327>

**[CVE-2023-39328] openjpeg: denial of service via crafted image file (Severity: MEDIUM)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

A vulnerability was found in OpenJPEG similar to CVE-2019-6988. This flaw allows an attacker to bypass existing protections and cause an application crash through a maliciously crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39328>

**[CVE-2023-39329] openjpeg: Resource exhaustion will occur in the opj\_t1\_decode\_cblks function in the tcd.c (Severity: MEDIUM)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

A flaw was found in OpenJPEG. A resource exhaustion can occur in the opj\_t1\_decode\_cblks function in tcd.c through a crafted image file, causing a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39329>

**[CVE-2016-10505] openjpeg: NULL pointer dereference in imagetopnm function in convert.c (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

NULL pointer dereference vulnerabilities in the imagetopnm function in convert.c, sycc444\_to\_rgb function in color.c, color\_esycc\_to\_rgb function in color.c, and sycc422\_to\_rgb function in color.c in OpenJPEG before 2.2.0 allow remote attackers to cause a denial of service (application crash) via crafted j2k files.

More Info: <https://avd.aquasec.com/nvd/cve-2016-10505>

**[CVE-2016-9113] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

There is a NULL pointer dereference in function imagetobmp of convertbmp.c:980 of OpenJPEG 2.1.2. image->comps[0].data is not assigned a value after initialization(NULL). Impact is Denial of Service.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9113>

**[CVE-2016-9114] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

There is a NULL Pointer Access in function imagetopnm of convert.c:1943(jp2) of OpenJPEG 2.1.2. image->comps[compno].data is not assigned a value after initialization(NULL). Impact is Denial of Service.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9114>

**[CVE-2016-9115] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

Heap Buffer Over-read in function imagetotga of convert.c(jp2):942 in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9115>

**[CVE-2016-9116] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

NULL Pointer Access in function imagetopnm of convert.c:2226(jp2) in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9116>

**[CVE-2016-9117] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

NULL Pointer Access in function imagetopnm of convert.c(jp2):1289 in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9117>

**[CVE-2016-9580] openjpeg2: Integer overflow in tftoimage causes heap buffer overflow (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

An integer overflow vulnerability was found in tftoimage function in openjpeg 2.1.2, resulting in heap buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9580>

**[CVE-2016-9581] openjpeg2: Infinite loop in tftoimage resulting into heap buffer overflow in convert\_32s\_C1P1 (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

An infinite loop vulnerability in tftoimage that results in heap buffer overflow in convert\_32s\_C1P1 was found in openjpeg 2.1.2.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9581>

**[CVE-2017-17479] openjpeg: Stack-buffer overflow in the pgxtoimage function (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

In OpenJPEG 2.3.0, a stack-based buffer overflow was discovered in the pgxtoimage function in jpwl/convert.c. The vulnerability causes an out-of-bounds write, which may lead to remote denial of service or possibly remote code execution.

More Info: <https://avd.aquasec.com/nvd/cve-2017-17479>

**[CVE-2018-16375] openjpeg: Heap-based buffer overflow in pnmtoimage function in bin/jpwl/convert.c (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

An issue was discovered in OpenJPEG 2.3.0. Missing checks for header\_info.height and header\_info.width in the function pnmtoimage in bin/jpwl/convert.c can lead to a heap-based buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2018-16375>

**[CVE-2018-16376] openjpeg: Heap-based buffer overflow in function t2\_encode\_packet in src/lib/openmj2/t2.c (Severity: LOW)**

Package: libopenjp2-7  
Installed: 2.5.0-2+deb12u1  
Fixed: %ls(<nil>)

An issue was discovered in OpenJPEG 2.3.0. A heap-based buffer overflow was discovered in the function t2\_encode\_packet in lib/openmj2/t2.c. The vulnerability causes an out-of-bounds write, which may lead to remote denial

of service or possibly unspecified other impact.

More Info: <https://avd.aquasec.com/nvd/cve-2018-16376>

**[CVE-2018-20846] openjpeg: out-of-bounds read in functions pi\_next\_lrcp, pi\_next\_rlcp, pi\_next\_rpcl, pi\_next\_pcrl, pi\_next\_rpcl, and pi\_next\_cpcl in openmj2/pi.c leads to denial of service (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

Out-of-bounds accesses in the functions pi\_next\_lrcp, pi\_next\_rlcp, pi\_next\_rpcl, pi\_next\_pcrl, pi\_next\_rpcl, and pi\_next\_cpcl in openmj2/pi.c in OpenJPEG through 2.3.0 allow remote attackers to cause a denial of service (application crash).

More Info: <https://avd.aquasec.com/nvd/cve-2018-20846>

**[CVE-2019-6988] openjpeg: DoS via memory exhaustion in opj\_decompress (Severity: LOW)**

Package: libopenjp2-7

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

An issue was discovered in OpenJPEG 2.3.0. It allows remote attackers to cause a denial of service (attempted excessive memory allocation) in opj\_calloc in openjp2/opj\_malloc.c, when called from opj\_tcd\_init\_tile in openjp2/tcd.c, as demonstrated by the 64-bit opj\_decompress.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6988>

**[CVE-2023-39327] openjpeg: Malicious files can cause the program to enter a large loop (Severity: MEDIUM)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

A flaw was found in OpenJPEG. Maliciously constructed pictures can cause the program to enter a large loop and continuously print warning messages on the terminal.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39327>

**[CVE-2023-39328] openjpeg: denial of service via crafted image file (Severity: MEDIUM)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

A vulnerability was found in OpenJPEG similar to CVE-2019-6988. This flaw allows an attacker to bypass existing protections and cause an application crash through a maliciously crafted file.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39328>

**[CVE-2023-39329] openjpeg: Resource exhaustion will occur in the opj\_t1\_decode\_cblks function in the tcd.c (Severity: MEDIUM)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

A flaw was found in OpenJPEG. A resource exhaustion can occur in the `opj_t1_decode_cblks` function in `tcd.c` through a crafted image file, causing a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39329>

**[CVE-2016-10505] openjpeg: NULL pointer dereference in imagetopnm function in convert.c (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

NULL pointer dereference vulnerabilities in the `imagetopnm` function in `convert.c`, `sycc444_to_rgb` function in `color.c`, `color_esycc_to_rgb` function in `color.c`, and `sycc422_to_rgb` function in `color.c` in OpenJPEG before 2.2.0 allow remote attackers to cause a denial of service (application crash) via crafted j2k files.

More Info: <https://avd.aquasec.com/nvd/cve-2016-10505>

**[CVE-2016-9113] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

There is a NULL pointer dereference in function `imagetobmp` of `convertbmp.c:980` of OpenJPEG 2.1.2. `image->comps[0].data` is not assigned a value after initialization(NULL). Impact is Denial of Service.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9113>

**[CVE-2016-9114] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

There is a NULL Pointer Access in function `imagetopnm` of `convert.c:1943(jp2)` of OpenJPEG 2.1.2. `image->comps[compno].data` is not assigned a value after initialization(NULL). Impact is Denial of Service.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9114>

**[CVE-2016-9115] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

Heap Buffer Over-read in function `imagetotga` of `convert.c(jp2):942` in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9115>

**[CVE-2016-9116] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

NULL Pointer Access in function imagetopnm of convert.c:2226(jp2) in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9116>

#### **[CVE-2016-9117] openjpeg2: Multiple security issues (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

NULL Pointer Access in function imagetopnm of convert.c(jp2):1289 in OpenJPEG 2.1.2. Impact is Denial of Service. Someone must open a crafted j2k file.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9117>

#### **[CVE-2016-9580] openjpeg2: Integer overflow in tifoimage causes heap buffer overflow (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

An integer overflow vulnerability was found in tifoimage function in openjpeg 2.1.2, resulting in heap buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9580>

#### **[CVE-2016-9581] openjpeg2: Infinite loop in tifoimage resulting into heap buffer overflow in convert\_32s\_C1P1 (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

An infinite loop vulnerability in tifoimage that results in heap buffer overflow in convert\_32s\_C1P1 was found in openjpeg 2.1.2.

More Info: <https://avd.aquasec.com/nvd/cve-2016-9581>

#### **[CVE-2017-17479] openjpeg: Stack-buffer overflow in the pgxtoimage function (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %ls(<nil>)

In OpenJPEG 2.3.0, a stack-based buffer overflow was discovered in the pgxtoimage function in jpwl/convert.c. The vulnerability causes an out-of-bounds write, which may lead to remote denial of service or possibly remote code execution.

More Info: <https://avd.aquasec.com/nvd/cve-2017-17479>

#### **[CVE-2018-16375] openjpeg: Heap-based buffer overflow in pnmtimage function in bin/jpwl/convert.c**

**(Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %!s(<nil>)

An issue was discovered in OpenJPEG 2.3.0. Missing checks for header\_info.height and header\_info.width in the function pnmtioimage in bin/jpwl/convert.c can lead to a heap-based buffer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2018-16375>

**[CVE-2018-16376] openjpeg: Heap-based buffer overflow in function t2\_encode\_packet in src/lib/openmj2/t2.c (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %!s(<nil>)

An issue was discovered in OpenJPEG 2.3.0. A heap-based buffer overflow was discovered in the function t2\_encode\_packet in lib/openmj2/t2.c. The vulnerability causes an out-of-bounds write, which may lead to remote denial of service or possibly unspecified other impact.

More Info: <https://avd.aquasec.com/nvd/cve-2018-16376>

**[CVE-2018-20846] openjpeg: out-of-bounds read in functions pi\_next\_rlcp, pi\_next\_rlcp, pi\_next\_rpcl, pi\_next\_pcl, pi\_next\_rpcl, and pi\_next\_cpcl in openmj2/pi.c leads to denial of service (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %!s(<nil>)

Out-of-bounds accesses in the functions pi\_next\_rlcp, pi\_next\_rlcp, pi\_next\_rpcl, pi\_next\_pcl, pi\_next\_rpcl, and pi\_next\_cpcl in openmj2/pi.c in OpenJPEG through 2.3.0 allow remote attackers to cause a denial of service (application crash).

More Info: <https://avd.aquasec.com/nvd/cve-2018-20846>

**[CVE-2019-6988] openjpeg: DoS via memory exhaustion in opj\_decompress (Severity: LOW)**

Package: libopenjp2-7-dev

Installed: 2.5.0-2+deb12u1

Fixed: %!s(<nil>)

An issue was discovered in OpenJPEG 2.3.0. It allows remote attackers to cause a denial of service (attempted excessive memory allocation) in opj\_calloc in openjp2/opj\_malloc.c, when called from opj\_tcd\_init\_tile in openjp2/tcd.c, as demonstrated by the 64-bit opj\_decompress.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6988>

**[CVE-2024-10041] pam: libpam: Libpam vulnerable to read hashed password (Severity: MEDIUM)**

Package: libpam-modules

Installed: 1.5.2-6+deb12u1

Fixed: %!s(<nil>)

A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the victim



program to execute by sending characters to its standard input (stdin). As this occurs, the attacker can train the branch predictor to execute an ROP chain speculatively. This flaw could result in leaked passwords, such as those found in /etc/shadow while performing authentications.

More Info: <https://avd.aquasec.com/nvd/cve-2024-10041>

**[CVE-2024-22365] pam: allowing unprivileged user to block another user namespace (Severity: MEDIUM)**

Package: libpam-modules  
Installed: 1.5.2-6+deb12u1  
Fixed: %ls(<nil>)

linux-pam (aka Linux PAM) before 1.6.0 allows attackers to cause a denial of service (blocked login process) via mkfifo because the openat call (for protect\_dir) lacks O\_DIRECTORY.

More Info: <https://avd.aquasec.com/nvd/cve-2024-22365>

**[CVE-2024-10041] pam: libpam: Libpam vulnerable to read hashed password (Severity: MEDIUM)**

Package: libpam-modules-bin  
Installed: 1.5.2-6+deb12u1  
Fixed: %ls(<nil>)

A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the victim program to execute by sending characters to its standard input (stdin). As this occurs, the attacker can train the branch predictor to execute an ROP chain speculatively. This flaw could result in leaked passwords, such as those found in /etc/shadow while performing authentications.

More Info: <https://avd.aquasec.com/nvd/cve-2024-10041>

**[CVE-2024-22365] pam: allowing unprivileged user to block another user namespace (Severity: MEDIUM)**

Package: libpam-modules-bin  
Installed: 1.5.2-6+deb12u1  
Fixed: %ls(<nil>)

linux-pam (aka Linux PAM) before 1.6.0 allows attackers to cause a denial of service (blocked login process) via mkfifo because the openat call (for protect\_dir) lacks O\_DIRECTORY.

More Info: <https://avd.aquasec.com/nvd/cve-2024-22365>

**[CVE-2024-10041] pam: libpam: Libpam vulnerable to read hashed password (Severity: MEDIUM)**

Package: libpam-runtime  
Installed: 1.5.2-6+deb12u1  
Fixed: %ls(<nil>)

A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the victim program to execute by sending characters to its standard input (stdin). As this occurs, the attacker can train the branch predictor to execute an ROP chain speculatively. This flaw could result in leaked passwords, such as those found in /etc/shadow while performing authentications.

More Info: <https://avd.aquasec.com/nvd/cve-2024-10041>

**[CVE-2024-22365] pam: allowing unprivileged user to block another user namespace (Severity: MEDIUM)**

Package: libpam-runtime  
Installed: 1.5.2-6+deb12u1  
Fixed: %!s(<nil>)

linux-pam (aka Linux PAM) before 1.6.0 allows attackers to cause a denial of service (blocked login process) via mkfifo because the openat call (for protect\_dir) lacks O\_DIRECTORY.

More Info: <https://avd.aquasec.com/nvd/cve-2024-22365>

**[CVE-2024-10041] pam: libpam: Libpam vulnerable to read hashed password (Severity: MEDIUM)**

Package: libpam0g  
Installed: 1.5.2-6+deb12u1  
Fixed: %!s(<nil>)

A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the victim program to execute by sending characters to its standard input (stdin). As this occurs, the attacker can train the branch predictor to execute an ROP chain speculatively. This flaw could result in leaked passwords, such as those found in /etc/shadow while performing authentications.

More Info: <https://avd.aquasec.com/nvd/cve-2024-10041>

**[CVE-2024-22365] pam: allowing unprivileged user to block another user namespace (Severity: MEDIUM)**

Package: libpam0g  
Installed: 1.5.2-6+deb12u1  
Fixed: %!s(<nil>)

linux-pam (aka Linux PAM) before 1.6.0 allows attackers to cause a denial of service (blocked login process) via mkfifo because the openat call (for protect\_dir) lacks O\_DIRECTORY.

More Info: <https://avd.aquasec.com/nvd/cve-2024-22365>

**[CVE-2023-31484] perl: CPAN.pm does not verify TLS certificates when downloading distributions over HTTPS (Severity: HIGH)**

Package: libperl5.36  
Installed: 5.36.0-7+deb12u1  
Fixed: %!s(<nil>)

CPAN.pm before 2.35 does not verify TLS certificates when downloading distributions over HTTPS.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31484>

**[CVE-2011-4116] perl: File::Temp insecure temporary file handling (Severity: LOW)**

Package: libperl5.36  
Installed: 5.36.0-7+deb12u1  
Fixed: %!s(<nil>)

\_is\_safe in the File::Temp module for Perl does not properly handle symlinks.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4116>

**[CVE-2023-31486] http-tiny: insecure TLS cert default (Severity: LOW)**

Package: libperl5.36

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default TLS configuration where users must opt in to verify certificates.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31486>

**[CVE-2023-37769] stress-test master commit e4c878 was discovered to contain a FPE vulne ... (Severity: LOW)**

Package: libpixman-1-0

Installed: 0.42.2-1

Fixed: %!s(<nil>)

stress-test master commit e4c878 was discovered to contain a FPE vulnerability via the component combine\_inner at /pixman-combine-float.c.

More Info: <https://avd.aquasec.com/nvd/cve-2023-37769>

**[CVE-2023-37769] stress-test master commit e4c878 was discovered to contain a FPE vulne ... (Severity: LOW)**

Package: libpixman-1-dev

Installed: 0.42.2-1

Fixed: %!s(<nil>)

stress-test master commit e4c878 was discovered to contain a FPE vulnerability via the component combine\_inner at /pixman-combine-float.c.

More Info: <https://avd.aquasec.com/nvd/cve-2023-37769>

**[CVE-2021-4214] libpng: hardcoded value leads to heap-overflow (Severity: LOW)**

Package: libpng-dev

Installed: 1.6.39-2

Fixed: %!s(<nil>)

A heap overflow flaw was found in libpngs' pngimage.c program. This flaw allows an attacker with local network access to pass a specially crafted PNG file to the pngimage utility, causing an application to crash, leading to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2021-4214>

**[CVE-2021-4214] libpng: hardcoded value leads to heap-overflow (Severity: LOW)**

Package: libpng16-16

Installed: 1.6.39-2

Fixed: %!s(<nil>)

A heap overflow flaw was found in libpngs' pngimage.c program. This flaw allows an attacker with local network access

to pass a specially crafted PNG file to the pngimage utility, causing an application to crash, leading to a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2021-4214>

#### **[CVE-2023-4016] procps: ps buffer overflow (Severity: LOW)**

Package: libproc2-0

Installed: 2:4.0.2-3

Fixed: %ls(<nil>)

Under some circumstances, this weakness allows a user who has access to run the `ps` utility on a machine, the ability to write almost unlimited amounts of unfiltered data into the process heap.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4016>

#### **[CVE-2025-0938] python: cpython: URL parser allowed square brackets in domain names (Severity: MEDIUM)**

Package: libpython3.11-minimal

Installed: 3.11.2-6+deb12u5

Fixed: %ls(<nil>)

The Python standard library functions `urllib.parse.urlsplit` and `urlparse` accepted domain names that included square brackets which isn't valid according to RFC 3986. Square brackets are only meant to be used as delimiters for specifying IPv6 and IPvFuture hosts in URLs. This could result in differential parsing across the Python URL parser and other specification-compliant URL parsers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0938>

#### **[CVE-2025-1795] python: Mishandling of comma during folding and unicode-encoding of email headers (Severity: LOW)**

Package: libpython3.11-minimal

Installed: 3.11.2-6+deb12u5

Fixed: %ls(<nil>)

During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-encoded then the separator itself is also unicode-encoded. Expected behavior is that the separating comma remains a plain comma. This can result in the address header being misinterpreted by some mail servers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1795>

#### **[CVE-2025-0938] python: cpython: URL parser allowed square brackets in domain names (Severity: MEDIUM)**

Package: libpython3.11-stdlib

Installed: 3.11.2-6+deb12u5

Fixed: %ls(<nil>)

The Python standard library functions `urllib.parse.urlsplit` and `urlparse` accepted domain names that included square brackets which isn't valid according to RFC 3986. Square brackets are only meant to be used as delimiters for specifying IPv6 and IPvFuture hosts in URLs. This could result in differential parsing across the Python URL parser and other specification-compliant URL parsers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0938>

**[CVE-2025-1795] python: Mishandling of comma during folding and unicode-encoding of email headers (Severity: LOW)**

Package: libpython3.11-stdlib

Installed: 3.11.2-6+deb12u5

Fixed: %ls(<nil>)

During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-encoded then the separator itself is also unicode-encoded. Expected behavior is that the separating comma remains a plain comma. This can result in the address header being misinterpreted by some mail servers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1795>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libquadmath0

Installed: 12.2.0-14

Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libquadmath0

Installed: 12.2.0-14

Fixed: %ls(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libsmartcols1

Installed: 2.38.1-5+deb12u3

Fixed: %!s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

#### **[CVE-2025-29088] sqlite: Denial of Service in SQLite (Severity: MEDIUM)**

Package: libsqlite3-0

Installed: 3.40.1-2+deb12u1

Fixed: %!s(<nil>)

An issue in sqlite v.3.49.0 allows an attacker to cause a denial of service via the SQLITE\_DBCONFIG\_LOOKASIDE component

More Info: <https://avd.aquasec.com/nvd/cve-2025-29088>

#### **[CVE-2021-45346] sqlite: crafted SQL query allows a malicious user to obtain sensitive information (Severity: LOW)**

Package: libsqlite3-0

Installed: 3.40.1-2+deb12u1

Fixed: %!s(<nil>)

A Memory Leak vulnerability exists in SQLite Project SQLite3 3.35.1 and 3.37.0 via maliciously crafted SQL Queries (made via editing the Database File), it is possible to query a record, and leak subsequent bytes of memory that extend beyond the record, which could let a malicious user obtain sensitive information. NOTE: The developer disputes this as a vulnerability stating that If you give SQLite a corrupted database file and submit a query against the database, it might read parts of the database that you did not intend or expect.

More Info: <https://avd.aquasec.com/nvd/cve-2021-45346>

#### **[CVE-2025-29088] sqlite: Denial of Service in SQLite (Severity: MEDIUM)**

Package: libsqlite3-dev

Installed: 3.40.1-2+deb12u1

Fixed: %!s(<nil>)

An issue in sqlite v.3.49.0 allows an attacker to cause a denial of service via the SQLITE\_DBCONFIG\_LOOKASIDE component

More Info: <https://avd.aquasec.com/nvd/cve-2025-29088>

#### **[CVE-2021-45346] sqlite: crafted SQL query allows a malicious user to obtain sensitive information (Severity: LOW)**

Package: libsqlite3-dev

Installed: 3.40.1-2+deb12u1

Fixed: %!s(<nil>)

A Memory Leak vulnerability exists in SQLite Project SQLite3 3.35.1 and 3.37.0 via maliciously crafted SQL Queries (made via editing the Database File), it is possible to query a record, and leak subsequent bytes of memory that extend

beyond the record, which could let a malicious user obtain sensitive information. NOTE: The developer disputes this as a vulnerability stating that If you give SQLite a corrupted database file and submit a query against the database, it might read parts of the database that you did not intend or expect.

More Info: <https://avd.aquasec.com/nvd/cve-2021-45346>

**[CVE-2024-13176] openssl: Timing side-channel in ECDSA signature computation (Severity: MEDIUM)**

Package: libssl-dev  
Installed: 3.0.15-1~deb12u1  
Fixed: %ls(<nil>)

Issue summary: A timing side-channel which could potentially allow recovering the private key exists in the ECDSA signature computation.

Impact summary: A timing side-channel in ECDSA signature computations could allow recovering the private key by an attacker. However, measuring the timing would require either local access to the signing application or a very fast network connection with low latency.

There is a timing signal of around 300 nanoseconds when the top word of the inverted ECDSA nonce value is zero. This can happen with significant probability only for some of the supported elliptic curves. In particular the NIST P-521 curve is affected. To be able to measure this leak, the attacker process must either be located in the same physical computer or must have a very fast network connection with low latency. For that reason the severity of this vulnerability is Low.

The FIPS modules in 3.4, 3.3, 3.2, 3.1 and 3.0 are affected by this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-13176>

**[CVE-2024-13176] openssl: Timing side-channel in ECDSA signature computation (Severity: MEDIUM)**

Package: libssl3  
Installed: 3.0.15-1~deb12u1  
Fixed: %ls(<nil>)

Issue summary: A timing side-channel which could potentially allow recovering the private key exists in the ECDSA signature computation.

Impact summary: A timing side-channel in ECDSA signature computations could allow recovering the private key by an attacker. However, measuring the timing would require either local access to the signing application or a very fast network connection with low latency.

There is a timing signal of around 300 nanoseconds when the top word of the inverted ECDSA nonce value is zero. This can happen with significant probability only for some of the supported elliptic curves. In particular the NIST P-521 curve is affected. To be able to measure this leak, the attacker process must either be located in the same physical computer or must have a very fast network connection with low latency. For that reason the severity of this vulnerability is Low.

The FIPS modules in 3.4, 3.3, 3.2, 3.1 and 3.0 are affected by this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-13176>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libstdc++-12-dev

Installed: 12.2.0-14

Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libstdc++-12-dev

Installed: 12.2.0-14

Fixed: %ls(<nil>)

**\*\*DISPUTED\*\*** A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libstdc++6

Installed: 12.2.0-14

Fixed: %ls(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

**[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libstdc++6



Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\*** A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using `alloca()`. The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

**[CVE-2013-4392] systemd: TOCTOU race condition when updating file permissions and SELinux security contexts (Severity: LOW)**

Package: libsystemd0

Installed: 252.36-1~deb12u1

Fixed: %!s(<nil>)

systemd, when updating file permissions, allows local users to change the permissions and SELinux security contexts for arbitrary files via a symlink attack on unspecified files.

More Info: <https://avd.aquasec.com/nvd/cve-2013-4392>

**[CVE-2023-31437] An issue was discovered in systemd 253. An attacker can modify a sealed log file (Severity: LOW)**

Package: libsystemd0

Installed: 252.36-1~deb12u1

Fixed: %!s(<nil>)

An issue was discovered in systemd 253. An attacker can modify a sealed log file such that, in some views, not all existing and sealed log messages are displayed. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31437>

**[CVE-2023-31438] An issue was discovered in systemd 253. An attacker can truncate a sealed log file (Severity: LOW)**

Package: libsystemd0

Installed: 252.36-1~deb12u1

Fixed: %!s(<nil>)

An issue was discovered in systemd 253. An attacker can truncate a sealed log file and then resume log sealing such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that

any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31438>

**[CVE-2023-31439] An issue was discovered in systemd 253. An attacker can modify the con ... (Severity: LOW)**

Package: libsystemd0

Installed: 252.36-1~deb12u1

Fixed: %ls(<nil>)

An issue was discovered in systemd 253. An attacker can modify the contents of past events in a sealed log file and then adjust the file such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31439>

**[CVE-2023-52355] libtiff: TIFFRasterScanlineSize64 produce too-big size and could cause OOM (Severity: HIGH)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An out-of-memory flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFRasterScanlineSize64() API. This flaw allows a remote attacker to cause a denial of service via a crafted input with a size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52355>

**[CVE-2023-6277] libtiff: Out-of-memory in TIFFOpen via a craft file (Severity: MEDIUM)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An out-of-memory flaw was found in libtiff. Passing a crafted tiff file to TIFFOpen() API may allow a remote attacker to cause a denial of service via a craft input with size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6277>

**[CVE-2017-16232] libtiff: Memory leaks in tif\_open.c, tif\_lzw.c, and tif\_aux.c (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF 4.0.8 has multiple memory leak vulnerabilities, which allow attackers to cause a denial of service (memory consumption), as demonstrated by tif\_open.c, tif\_lzw.c, and tif\_aux.c. NOTE: Third parties were unable to reproduce the issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-16232>

**[CVE-2017-17973] libtiff: heap-based use after free in tiff2pdf.c:t2p\_writeproc (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.8, there is a heap-based use-after-free in the t2p\_writeproc function in tiff2pdf.c. NOTE: there is a third-party report of inability to reproduce this issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-17973>

### **[CVE-2017-5563] libtiff: Heap-buffer overflow in LZWEncode tif\_lzw.c (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF version 4.0.7 is vulnerable to a heap-based buffer over-read in tif\_lzw.c resulting in DoS or code execution via a crafted bmp image to tools/bmp2tiff.

More Info: <https://avd.aquasec.com/nvd/cve-2017-5563>

### **[CVE-2017-9117] libtiff: Heap-based buffer over-read in bmp2tiff (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.6 and possibly other versions, the program processes BMP images without verifying that biWidth and biHeight in the bitmap-information header match the actual input, as demonstrated by a heap-based buffer over-read in bmp2tiff. NOTE: mentioning bmp2tiff does not imply that the activation point is in the bmp2tiff.c file (which was removed before the 4.0.7 release).

More Info: <https://avd.aquasec.com/nvd/cve-2017-9117>

### **[CVE-2018-10126] libtiff: NULL pointer dereference in the jpeg\_fdct\_16x16 function in jfdctint.c (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

ijg-libjpeg before 9d, as used in tiff2pdf (from LibTIFF) and other products, does not check for a NULL pointer at a certain place in jpeg\_fdct\_16x16 in jfdctint.c.

More Info: <https://avd.aquasec.com/nvd/cve-2018-10126>

### **[CVE-2022-1210] tiff: Malicious file leads to a denial of service in TIFF File Handler (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

A vulnerability classified as problematic was found in LibTIFF 4.3.0. Affected by this vulnerability is the TIFF File Handler of tiff2ps. Opening a malicious file leads to a denial of service. The attack can be launched remotely but requires user interaction. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2022-1210>

### **[CVE-2023-1916] libtiff: out-of-bounds read in extractImageSection() in tools/tiffcrop.c (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

A flaw was found in tiffcrop, a program distributed by the libtiff package. A specially crafted tiff file can lead to an out-of-bounds read in the extractImageSection function in tools/tiffcrop.c, resulting in a denial of service and limited information disclosure. This issue affects libtiff versions 4.x.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1916>

### **[CVE-2023-3164] libtiff: heap-buffer-overflow in extractImageSection() (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

A heap-buffer-overflow vulnerability was found in LibTIFF, in extractImageSection() at tools/tiffcrop.c:7916 and tools/tiffcrop.c:7801. This flaw allows attackers to cause a denial of service via a crafted tiff file.

More Info: <https://avd.aquasec.com/nvd/cve-2023-3164>

### **[CVE-2023-6228] libtiff: heap-based buffer overflow in cpStripToTile() in tools/tiffcp.c (Severity: LOW)**

Package: libtiff-dev

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An issue was found in the tiffcp utility distributed by the libtiff package where a crafted TIFF file on processing may cause a heap-based buffer overflow leads to an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6228>

### **[CVE-2023-52355] libtiff: TIFFRasterScanlineSize64 produce too-big size and could cause OOM (Severity: HIGH)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An out-of-memory flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFRasterScanlineSize64() API. This flaw allows a remote attacker to cause a denial of service via a crafted input with a size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52355>

### **[CVE-2023-6277] libtiff: Out-of-memory in TIFFOpen via a craft file (Severity: MEDIUM)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An out-of-memory flaw was found in libtiff. Passing a crafted tiff file to TIFFOpen() API may allow a remote attacker to cause a denial of service via a craft input with size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6277>

**[CVE-2017-16232] libtiff: Memory leaks in tif\_open.c, tif\_lzw.c, and tif\_aux.c (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF 4.0.8 has multiple memory leak vulnerabilities, which allow attackers to cause a denial of service (memory consumption), as demonstrated by tif\_open.c, tif\_lzw.c, and tif\_aux.c. NOTE: Third parties were unable to reproduce the issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-16232>

**[CVE-2017-17973] libtiff: heap-based use after free in tiff2pdf.c:t2p\_writeproc (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.8, there is a heap-based use-after-free in the t2p\_writeproc function in tiff2pdf.c. NOTE: there is a third-party report of inability to reproduce this issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-17973>

**[CVE-2017-5563] libtiff: Heap-buffer overflow in LZWEncode tif\_lzw.c (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF version 4.0.7 is vulnerable to a heap-based buffer over-read in tif\_lzw.c resulting in DoS or code execution via a crafted bmp image to tools/bmp2tiff.

More Info: <https://avd.aquasec.com/nvd/cve-2017-5563>

**[CVE-2017-9117] libtiff: Heap-based buffer over-read in bmp2tiff (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.6 and possibly other versions, the program processes BMP images without verifying that biWidth and biHeight in the bitmap-information header match the actual input, as demonstrated by a heap-based buffer over-read in bmp2tiff. NOTE: mentioning bmp2tiff does not imply that the activation point is in the bmp2tiff.c file (which was removed before the 4.0.7 release).

More Info: <https://avd.aquasec.com/nvd/cve-2017-9117>

**[CVE-2018-10126] libtiff: NULL pointer dereference in the jpeg\_fdct\_16x16 function in jfdctint.c (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

ijg-libjpeg before 9d, as used in tiff2pdf (from LibTIFF) and other products, does not check for a NULL pointer at a

certain place in jpeg\_fdct\_16x16 in jfdctint.c.

More Info: <https://avd.aquasec.com/nvd/cve-2018-10126>

**[CVE-2022-1210] tiff: Malicious file leads to a denial of service in TIFF File Handler (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in LibTIFF 4.3.0. Affected by this vulnerability is the TIFF File Handler of tiff2ps. Opening a malicious file leads to a denial of service. The attack can be launched remotely but requires user interaction. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2022-1210>

**[CVE-2023-1916] libtiff: out-of-bounds read in extractImageSection() in tools/tiffcrop.c (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A flaw was found in tiffcrop, a program distributed by the libtiff package. A specially crafted tiff file can lead to an out-of-bounds read in the extractImageSection function in tools/tiffcrop.c, resulting in a denial of service and limited information disclosure. This issue affects libtiff versions 4.x.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1916>

**[CVE-2023-3164] libtiff: heap-buffer-overflow in extractImageSection() (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A heap-buffer-overflow vulnerability was found in LibTIFF, in extractImageSection() at tools/tiffcrop.c:7916 and tools/tiffcrop.c:7801. This flaw allows attackers to cause a denial of service via a crafted tiff file.

More Info: <https://avd.aquasec.com/nvd/cve-2023-3164>

**[CVE-2023-6228] libtiff: heap-based buffer overflow in cpStripToTile() in tools/tiffcp.c (Severity: LOW)**

Package: libtiff6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

An issue was found in the tiffcp utility distributed by the libtiff package where a crafted TIFF file on processing may cause a heap-based buffer overflow leads to an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6228>

**[CVE-2023-52355] libtiff: TIFFRasterScanlineSize64 produce too-big size and could cause OOM (Severity: HIGH)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

An out-of-memory flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFRasterScanlineSize64() API. This flaw allows a remote attacker to cause a denial of service via a crafted input with a size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52355>

**[CVE-2023-6277] libtiff: Out-of-memory in TIFFOpen via a craft file (Severity: MEDIUM)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

An out-of-memory flaw was found in libtiff. Passing a crafted tiff file to TIFFOpen() API may allow a remote attacker to cause a denial of service via a craft input with size smaller than 379 KB.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6277>

**[CVE-2017-16232] libtiff: Memory leaks in tif\_open.c, tif\_lzw.c, and tif\_aux.c (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF 4.0.8 has multiple memory leak vulnerabilities, which allow attackers to cause a denial of service (memory consumption), as demonstrated by tif\_open.c, tif\_lzw.c, and tif\_aux.c. NOTE: Third parties were unable to reproduce the issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-16232>

**[CVE-2017-17973] libtiff: heap-based use after free in tiff2pdf.c:t2p\_writeproc (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.8, there is a heap-based use-after-free in the t2p\_writeproc function in tiff2pdf.c. NOTE: there is a third-party report of inability to reproduce this issue

More Info: <https://avd.aquasec.com/nvd/cve-2017-17973>

**[CVE-2017-5563] libtiff: Heap-buffer overflow in LZWEncode tif\_lzw.c (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

LibTIFF version 4.0.7 is vulnerable to a heap-based buffer over-read in tif\_lzw.c resulting in DoS or code execution via a crafted bmp image to tools/bmp2tiff.

More Info: <https://avd.aquasec.com/nvd/cve-2017-5563>

**[CVE-2017-9117] libtiff: Heap-based buffer over-read in bmp2tiff (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %ls(<nil>)

In LibTIFF 4.0.6 and possibly other versions, the program processes BMP images without verifying that biWidth and biHeight in the bitmap-information header match the actual input, as demonstrated by a heap-based buffer over-read in bmp2tiff. NOTE: mentioning bmp2tiff does not imply that the activation point is in the bmp2tiff.c file (which was removed before the 4.0.7 release).

More Info: <https://avd.aquasec.com/nvd/cve-2017-9117>

**[CVE-2018-10126] libtiff: NULL pointer dereference in the jpeg\_fdct\_16x16 function in jfdctint.c (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

ijg-libjpeg before 9d, as used in tiff2pdf (from LibTIFF) and other products, does not check for a NULL pointer at a certain place in jpeg\_fdct\_16x16 in jfdctint.c.

More Info: <https://avd.aquasec.com/nvd/cve-2018-10126>

**[CVE-2022-1210] tiff: Malicious file leads to a denial of service in TIFF File Handler (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A vulnerability classified as problematic was found in LibTIFF 4.3.0. Affected by this vulnerability is the TIFF File Handler of tiff2ps. Opening a malicious file leads to a denial of service. The attack can be launched remotely but requires user interaction. The exploit has been disclosed to the public and may be used.

More Info: <https://avd.aquasec.com/nvd/cve-2022-1210>

**[CVE-2023-1916] libtiff: out-of-bounds read in extractImageSection() in tools/tiffcrop.c (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A flaw was found in tiffcrop, a program distributed by the libtiff package. A specially crafted tiff file can lead to an out-of-bounds read in the extractImageSection function in tools/tiffcrop.c, resulting in a denial of service and limited information disclosure. This issue affects libtiff versions 4.x.

More Info: <https://avd.aquasec.com/nvd/cve-2023-1916>

**[CVE-2023-3164] libtiff: heap-buffer-overflow in extractImageSection() (Severity: LOW)**

Package: libtiffxx6

Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

A heap-buffer-overflow vulnerability was found in LibTIFF, in extractImageSection() at tools/tiffcrop.c:7916 and tools/tiffcrop.c:7801. This flaw allows attackers to cause a denial of service via a crafted tiff file.

More Info: <https://avd.aquasec.com/nvd/cve-2023-3164>

**[CVE-2023-6228] libtiff: heap-based buffer overflow in cpStripToTile() in tools/tiffcp.c (Severity: LOW)**

Package: libtiffxx6



Installed: 4.5.0-6+deb12u2

Fixed: %!s(<nil>)

An issue was found in the tiffcp utility distributed by the libtiff package where a crafted TIFF file on processing may cause a heap-based buffer overflow leads to an application crash.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6228>

### **[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: libtinfo6

Installed: 6.4-4

Fixed: %!s(<nil>)

NCurses v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle\_const (Severity: LOW)**

Package: libtsan2

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libtsan2

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2022-27943] binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in**

### **demangle\_const (Severity: LOW)**

Package: libubsan1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle\_const, as demonstrated by nm-new.

More Info: <https://avd.aquasec.com/nvd/cve-2022-27943>

### **[CVE-2023-4039] gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 (Severity: LOW)**

Package: libubsan1

Installed: 12.2.0-14

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\***A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using alloca(). The stack-protector operates as intended for statically-sized local variables.

The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4039>

### **[CVE-2013-4392] systemd: TOCTOU race condition when updating file permissions and SELinux security contexts (Severity: LOW)**

Package: libudev1

Installed: 252.36-1~deb12u1

Fixed: %!s(<nil>)

systemd, when updating file permissions, allows local users to change the permissions and SELinux security contexts for arbitrary files via a symlink attack on unspecified files.

More Info: <https://avd.aquasec.com/nvd/cve-2013-4392>

### **[CVE-2023-31437] An issue was discovered in systemd 253. An attacker can modify a sealed log file such that, in some views, not all (Severity: LOW)**

Package: libudev1

Installed: 252.36-1~deb12u1

Fixed: %!s(<nil>)

An issue was discovered in systemd 253. An attacker can modify a sealed log file such that, in some views, not all

existing and sealed log messages are displayed. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31437>

**[CVE-2023-31438] An issue was discovered in systemd 253. An attacker can truncate a sea ... (Severity: LOW)**

Package: libudev1

Installed: 252.36-1~deb12u1

Fixed: %ls(<nil>)

An issue was discovered in systemd 253. An attacker can truncate a sealed log file and then resume log sealing such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31438>

**[CVE-2023-31439] An issue was discovered in systemd 253. An attacker can modify the con ... (Severity: LOW)**

Package: libudev1

Installed: 252.36-1~deb12u1

Fixed: %ls(<nil>)

An issue was discovered in systemd 253. An attacker can modify the contents of past events in a sealed log file and then adjust the file such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."

More Info: <https://avd.aquasec.com/nvd/cve-2023-31439>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: libuuid1

Installed: 2.38.1-5+deb12u3

Fixed: %ls(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2007-3476] libgd Denial of service by corrupted GIF images (Severity: LOW)**

Package: libwmf-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

Array index error in gd\_gif\_in.c in the GD Graphics Library (libgd) before 2.0.35 allows user-assisted remote attackers to cause a denial of service (crash and heap corruption) via large color index values in crafted image data, which results in a segmentation fault.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3476>

**[CVE-2007-3477] gd: arc drawing functions can consume large amount of CPU time (Severity: LOW)**

Package: libwmf-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

The (a) imagearc and (b) imagefilledarc functions in GD Graphics Library (libgd) before 2.0.35 allow attackers to cause a denial of service (CPU consumption) via a large (1) start or (2) end angle degree value.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3477>

**[CVE-2007-3996] php multiple integer overflows in gd (Severity: LOW)**

Package: libwmf-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

Multiple integer overflows in libgd in PHP before 5.2.4 allow remote attackers to cause a denial of service (application crash) and possibly execute arbitrary code via a large (1) srcW or (2) srcH value to the (a) gdImageCopyResized function, or a large (3) sy (height) or (4) sx (width) value to the (b) gdImageCreate or the (c) gdImageCreateTrueColor function.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3996>

**[CVE-2009-3546] gd: insufficient input validation in \_gdGetColors() (Severity: LOW)**

Package: libwmf-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

The \_gdGetColors function in gd\_gd.c in PHP 5.2.11 and 5.3.x before 5.3.1, and the GD Graphics Library 2.x, does not properly verify a certain colorsTotal structure member, which might allow remote attackers to conduct buffer overflow or buffer over-read attacks via a crafted GD file, a different vulnerability than CVE-2009-3293. NOTE: some of these details are obtained from third party information.

More Info: <https://avd.aquasec.com/nvd/cve-2009-3546>

**[TEMP-0601525-BEBB65] [libgd2: gdImageColorTransparent can write outside buffer] (Severity: LOW)**

Package: libwmf-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0601525-BEBB65>

**[CVE-2007-3476] libgd Denial of service by corrupted GIF images (Severity: LOW)**

Package: libwmf-dev

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

Array index error in gd\_gif\_in.c in the GD Graphics Library (libgd) before 2.0.35 allows user-assisted remote attackers to cause a denial of service (crash and heap corruption) via large color index values in crafted image data, which results in

a segmentation fault.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3476>

**[CVE-2007-3477] gd: arc drawing functions can consume large amount of CPU time (Severity: LOW)**

Package: libwmf-dev

Installed: 0.2.12-5.1

Fixed: %!s(<nil>)

The (a) imagearc and (b) imagefilledarc functions in GD Graphics Library (libgd) before 2.0.35 allow attackers to cause a denial of service (CPU consumption) via a large (1) start or (2) end angle degree value.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3477>

**[CVE-2007-3996] php multiple integer overflows in gd (Severity: LOW)**

Package: libwmf-dev

Installed: 0.2.12-5.1

Fixed: %!s(<nil>)

Multiple integer overflows in libgd in PHP before 5.2.4 allow remote attackers to cause a denial of service (application crash) and possibly execute arbitrary code via a large (1) srcW or (2) srcH value to the (a) gdImageCopyResized function, or a large (3) sy (height) or (4) sx (width) value to the (b) gdImageCreate or the (c) gdImageCreateTrueColor function.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3996>

**[CVE-2009-3546] gd: insufficient input validation in \_gdGetColors() (Severity: LOW)**

Package: libwmf-dev

Installed: 0.2.12-5.1

Fixed: %!s(<nil>)

The \_gdGetColors function in gd\_gd.c in PHP 5.2.11 and 5.3.x before 5.3.1, and the GD Graphics Library 2.x, does not properly verify a certain colorsTotal structure member, which might allow remote attackers to conduct buffer overflow or buffer over-read attacks via a crafted GD file, a different vulnerability than CVE-2009-3293. NOTE: some of these details are obtained from third party information.

More Info: <https://avd.aquasec.com/nvd/cve-2009-3546>

**[TEMP-0601525-BEBB65] [libgd2: gdImageColorTransparent can write outside buffer] (Severity: LOW)**

Package: libwmf-dev

Installed: 0.2.12-5.1

Fixed: %!s(<nil>)

%!s(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0601525-BEBB65>

**[CVE-2007-3476] libgd Denial of service by corrupted GIF images (Severity: LOW)**

Package: libwmflite-0.2-7

Installed: 0.2.12-5.1

Fixed: %!s(<nil>)

Array index error in gd\_gif\_in.c in the GD Graphics Library (libgd) before 2.0.35 allows user-assisted remote attackers to cause a denial of service (crash and heap corruption) via large color index values in crafted image data, which results in a segmentation fault.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3476>

#### **[CVE-2007-3477] gd: arc drawing functions can consume large amount of CPU time (Severity: LOW)**

Package: libwmflite-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

The (a) imagearc and (b) imagefilledarc functions in GD Graphics Library (libgd) before 2.0.35 allow attackers to cause a denial of service (CPU consumption) via a large (1) start or (2) end angle degree value.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3477>

#### **[CVE-2007-3996] php multiple integer overflows in gd (Severity: LOW)**

Package: libwmflite-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

Multiple integer overflows in libgd in PHP before 5.2.4 allow remote attackers to cause a denial of service (application crash) and possibly execute arbitrary code via a large (1) srcW or (2) srcH value to the (a) gdImageCopyResized function, or a large (3) sy (height) or (4) sx (width) value to the (b) gdImageCreate or the (c) gdImageCreateTrueColor function.

More Info: <https://avd.aquasec.com/nvd/cve-2007-3996>

#### **[CVE-2009-3546] gd: insufficient input validation in \_gdGetColors() (Severity: LOW)**

Package: libwmflite-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

The \_gdGetColors function in gd\_gd.c in PHP 5.2.11 and 5.3.x before 5.3.1, and the GD Graphics Library 2.x, does not properly verify a certain colorsTotal structure member, which might allow remote attackers to conduct buffer overflow or buffer over-read attacks via a crafted GD file, a different vulnerability than CVE-2009-3293. NOTE: some of these details are obtained from third party information.

More Info: <https://avd.aquasec.com/nvd/cve-2009-3546>

#### **[TEMP-0601525-BEBB65] [libgd2: gdImageColorTransparent can write outside buffer] (Severity: LOW)**

Package: libwmflite-0.2-7

Installed: 0.2.12-5.1

Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0601525-BEBB65>

#### **[CVE-2024-25062] libxml2: use-after-free in XMLReader (Severity: HIGH)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

An issue was discovered in libxml2 before 2.11.7 and 2.12.x before 2.12.5. When using the XML Reader interface with DTD validation and XInclude expansion enabled, processing crafted XML documents can lead to an xmlValidatePopElement use-after-free.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25062>

### **[CVE-2024-56171] libxml2: Use-After-Free in libxml2 (Severity: HIGH)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a use-after-free in xmlSchemaDCFillNodeTables and xmlSchemaBubbleIDCNodeTables in xmlschemas.c. To exploit this, a crafted XML document must be validated against an XML schema with certain identity constraints, or a crafted XML schema must be used.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56171>

### **[CVE-2025-24928] libxml2: Stack-based buffer overflow in xmlSnprintfElements of libxml2 (Severity: HIGH)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a stack-based buffer overflow in xmlSnprintfElements in valid.c. To exploit this, DTD validation must occur for an untrusted document or untrusted DTD. NOTE: this is similar to CVE-2017-9047.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24928>

### **[CVE-2025-27113] libxml2: NULL Pointer Dereference in libxml2 xmlPatMatch (Severity: HIGH)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a NULL pointer dereference in xmlPatMatch in pattern.c.

More Info: <https://avd.aquasec.com/nvd/cve-2025-27113>

### **[CVE-2022-49043] libxml: use-after-free in xmlXIncludeAddNode (Severity: MEDIUM)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

xmlXIncludeAddNode in xinclude.c in libxml2 before 2.11.0 has a use-after-free.

More Info: <https://avd.aquasec.com/nvd/cve-2022-49043>

### **[CVE-2023-39615] libxml2: crafted xml can cause global buffer overflow (Severity: MEDIUM)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

Xmlsoft Libxml2 v2.11.0 was discovered to contain an out-of-bounds read via the xmlSAX2StartElement() function at /libxml2/SAX2.c. This vulnerability allows attackers to cause a Denial of Service (DoS) via supplying a crafted XML file. NOTE: the vendor's position is that the product does not support the legacy SAX1 interface with custom callbacks; there is a crash even without crafted input.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39615>

#### **[CVE-2023-45322] libxml2: use-after-free in xmlUnlinkNode() in tree.c (Severity: MEDIUM)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

libxml2 through 2.11.5 has a use-after-free that can only occur after a certain memory allocation fails. This occurs in xmlUnlinkNode in tree.c. NOTE: the vendor's position is "I don't think these issues are critical enough to warrant a CVE ID ... because an attacker typically can't control when memory allocations fail."

More Info: <https://avd.aquasec.com/nvd/cve-2023-45322>

#### **[CVE-2025-32414] libxml2: Out-of-Bounds Read in libxml2 (Severity: MEDIUM)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

In libxml2 before 2.13.8 and 2.14.x before 2.14.2, out-of-bounds memory access can occur in the Python API (Python bindings) because of an incorrect return value. This occurs in xmlPythonFileRead and xmlPythonFileReadRaw because of a difference between bytes and characters.

More Info: <https://avd.aquasec.com/nvd/cve-2025-32414>

#### **[CVE-2024-34459] libxml2: buffer over-read in xmlHTMLPrintFileContext in xmllint.c (Severity: LOW)**

Package: libxml2

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

An issue was discovered in xmllint (from libxml2) before 2.11.8 and 2.12.x before 2.12.7. Formatting error messages with xmllint --htmlout can result in a buffer over-read in xmlHTMLPrintFileContext in xmllint.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-34459>

#### **[CVE-2024-25062] libxml2: use-after-free in XMLReader (Severity: HIGH)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

An issue was discovered in libxml2 before 2.11.7 and 2.12.x before 2.12.5. When using the XML Reader interface with DTD validation and XInclude expansion enabled, processing crafted XML documents can lead to an xmlValidatePopElement use-after-free.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25062>



### **[CVE-2024-56171] libxml2: Use-After-Free in libxml2 (Severity: HIGH)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a use-after-free in xmlSchemaDCFillNodeTables and xmlSchemaBubbleDCNodeTables in xmlschemas.c. To exploit this, a crafted XML document must be validated against an XML schema with certain identity constraints, or a crafted XML schema must be used.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56171>

### **[CVE-2025-24928] libxml2: Stack-based buffer overflow in xmlSnprintfElements of libxml2 (Severity: HIGH)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a stack-based buffer overflow in xmlSnprintfElements in valid.c. To exploit this, DTD validation must occur for an untrusted document or untrusted DTD. NOTE: this is similar to CVE-2017-9047.

More Info: <https://avd.aquasec.com/nvd/cve-2025-24928>

### **[CVE-2025-27113] libxml2: NULL Pointer Dereference in libxml2 xmlPatMatch (Severity: HIGH)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a NULL pointer dereference in xmlPatMatch in pattern.c.

More Info: <https://avd.aquasec.com/nvd/cve-2025-27113>

### **[CVE-2022-49043] libxml: use-after-free in xmlXIncludeAddNode (Severity: MEDIUM)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

xmlXIncludeAddNode in xinclude.c in libxml2 before 2.11.0 has a use-after-free.

More Info: <https://avd.aquasec.com/nvd/cve-2022-49043>

### **[CVE-2023-39615] libxml2: crafted xml can cause global buffer overflow (Severity: MEDIUM)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %!s(<nil>)

Xmlsoft Libxml2 v2.11.0 was discovered to contain an out-of-bounds read via the xmlSAX2StartElement() function at /libxml2/SAX2.c. This vulnerability allows attackers to cause a Denial of Service (DoS) via supplying a crafted XML file. NOTE: the vendor's position is that the product does not support the legacy SAX1 interface with custom callbacks; there is a crash even without crafted input.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39615>

#### **[CVE-2023-45322] libxml2: use-after-free in xmlUnlinkNode() in tree.c (Severity: MEDIUM)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

libxml2 through 2.11.5 has a use-after-free that can only occur after a certain memory allocation fails. This occurs in xmlUnlinkNode in tree.c. NOTE: the vendor's position is "I don't think these issues are critical enough to warrant a CVE ID ... because an attacker typically can't control when memory allocations fail."

More Info: <https://avd.aquasec.com/nvd/cve-2023-45322>

#### **[CVE-2025-32414] libxml2: Out-of-Bounds Read in libxml2 (Severity: MEDIUM)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

In libxml2 before 2.13.8 and 2.14.x before 2.14.2, out-of-bounds memory access can occur in the Python API (Python bindings) because of an incorrect return value. This occurs in xmlPythonFileRead and xmlPythonFileReadRaw because of a difference between bytes and characters.

More Info: <https://avd.aquasec.com/nvd/cve-2025-32414>

#### **[CVE-2024-34459] libxml2: buffer over-read in xmlHTMLPrintFileContext in xmllint.c (Severity: LOW)**

Package: libxml2-dev

Installed: 2.9.14+dfsg-1.3~deb12u1

Fixed: %ls(<nil>)

An issue was discovered in xmllint (from libxml2) before 2.11.8 and 2.12.x before 2.12.7. Formatting error messages with xmllint --htmlout can result in a buffer over-read in xmlHTMLPrintFileContext in xmllint.c.

More Info: <https://avd.aquasec.com/nvd/cve-2024-34459>

#### **[CVE-2015-9019] libxslt: math.random() in xslt uses unseeded randomness (Severity: LOW)**

Package: libxslt1-dev

Installed: 1.1.35-1+deb12u1

Fixed: %ls(<nil>)

In libxslt 1.1.29 and earlier, the EXSLT math.random function was not initialized with a random seed during startup, which could cause usage of this function to produce predictable outputs.

More Info: <https://avd.aquasec.com/nvd/cve-2015-9019>

#### **[CVE-2015-9019] libxslt: math.random() in xslt uses unseeded randomness (Severity: LOW)**

Package: libxslt1.1

Installed: 1.1.35-1+deb12u1

Fixed: %ls(<nil>)

In libxslt 1.1.29 and earlier, the EXSLT math.random function was not initialized with a random seed during startup, which could cause usage of this function to produce predictable outputs.

More Info: <https://avd.aquasec.com/nvd/cve-2015-9019>

**[CVE-2013-7445] kernel: memory exhaustion via crafted Graphics Execution Manager (GEM) objects (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

The Direct Rendering Manager (DRM) subsystem in the Linux kernel through 4.x mishandles requests for Graphics Execution Manager (GEM) objects, which allows context-dependent attackers to cause a denial of service (memory consumption) via an application that processes graphics data, as demonstrated by JavaScript code that creates many CANVAS elements for rendering by Chrome or Firefox.

More Info: <https://avd.aquasec.com/nvd/cve-2013-7445>

**[CVE-2019-19449] kernel: mounting a crafted f2fs filesystem image can lead to slab-out-of-bounds read access in f2fs\_build\_segment\_manager in fs/f2fs/segment.c (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel 5.0.21, mounting a crafted f2fs filesystem image can lead to slab-out-of-bounds read access in f2fs\_build\_segment\_manager in fs/f2fs/segment.c, related to init\_min\_max\_mtime in fs/f2fs/segment.c (because the second argument to get\_seg\_entry is not validated).

More Info: <https://avd.aquasec.com/nvd/cve-2019-19449>

**[CVE-2019-19814] kernel: out-of-bounds write in \_\_remove\_dirty\_segment in fs/f2fs/segment.c (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel 5.0.21, mounting a crafted f2fs filesystem image can cause \_\_remove\_dirty\_segment slab-out-of-bounds write access because an array is bounded by the number of dirty types (8) but the array index can exceed this.

More Info: <https://avd.aquasec.com/nvd/cve-2019-19814>

**[CVE-2021-3847] kernel: low-privileged user privileges escalation (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An unauthorized access to the execution of the setuid file with capabilities flaw in the Linux kernel OverlayFS subsystem was found in the way user copying a capable file from a nosuid mount into another mount. A local user could use this flaw to escalate their privileges on the system.

More Info: <https://avd.aquasec.com/nvd/cve-2021-3847>

**[CVE-2021-3864] kernel: descendant's dumpable setting with certain SUID binaries (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

A flaw was found in the way the dumpable flag setting was handled when certain SUID binaries executed its descendants. The prerequisite is a SUID binary that sets real UID equal to effective UID, and real GID equal to effective GID. The descendant will then have a dumpable value set to 1. As a result, if the descendant process crashes and core\_pattern is set to a relative value, its core dump is stored in the current directory with uid:gid permissions. An unprivileged local user with eligible root SUID binary could use this flaw to place core dumps into root-owned directories, potentially resulting in escalation of privileges.

More Info: <https://avd.aquasec.com/nvd/cve-2021-3864>

### **[CVE-2023-52452] kernel: bpf: Fix accesses to uninit stack slots (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Fix accesses to uninit stack slots

Privileged programs are supposed to be able to read uninitialized stack memory (ever since 6715df8d5) but, before this patch, these accesses were permitted inconsistently. In particular, accesses were permitted above state->allocated\_stack, but not below it. In other words, if the stack was already "large enough", the access was permitted, but otherwise the access was rejected instead of being allowed to "grow the stack". This undesired rejection was happening in two places:

- in check\_stack\_slot\_within\_bounds()
- in check\_stack\_range\_initialized()

This patch arranges for these accesses to be permitted. A bunch of tests that were relying on the old rejection had to change; all of them were changed to add also run\_unprivileged, in which case the old behavior persists. One tests couldn't be updated - global\_func16 - because it can't run unprivileged for other reasons.

This patch also fixes the tracking of the stack size for variable-offset reads. This second fix is bundled in the same commit as the first one because they're inter-related. Before this patch, writes to the stack using registers containing a variable offset (as opposed to registers with fixed, known values) were not properly contributing to the function's needed stack size. As a result, it was possible for a program to verify, but then to attempt to read out-of-bounds data at runtime because a too small stack had been allocated for it.

Each function tracks the size of the stack it needs in bpf\_subprog\_info.stack\_depth, which is maintained by update\_stack\_depth(). For regular memory accesses, check\_mem\_access() was calling update\_state\_depth() but it was passing in only the fixed part of the offset register, ignoring the variable offset. This was incorrect; the minimum possible value of that register should be used instead.

This tracking is now fixed by centralizing the tracking of stack size in `grow_stack_state()`, and by lifting the calls to `grow_stack_state()` to `check_stack_access_within_bounds()` as suggested by Andrii. The code is now simpler and more convincingly tracks the correct maximum stack size. `check_stack_range_initialized()` can now rely on enough stack having been allocated for the access; this helps with the fix for the first issue.

A few tests were changed to also check the stack depth computation. The one that fails without this patch is `verifier_var_off:stack_write_priv_vs_unpriv`.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52452>

### **[CVE-2023-52586] kernel: drm/msm/dpu: Add mutex lock in control vblank irq (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/msm/dpu: Add mutex lock in control vblank irq

Add a mutex lock to control vblank irq to synchronize vblank enable/disable operations happening from different threads to prevent race conditions while registering/unregistering the vblank irq callback.

v4: -Removed `vblank_ctl_lock` from `dpu_encoder_virt`, so it is only a parameter of `dpu_encoder_phys`.

-Switch from atomic `refcnt` to a simple int counter as mutex has now been added

v3: Mistakenly did not change wording in last version. It is done now.

v2: Slightly changed wording of commit message

Patchwork: <https://patchwork.freedesktop.org/patch/571854/>

More Info: <https://avd.aquasec.com/nvd/cve-2023-52586>

### **[CVE-2023-52624] kernel: drm/amd/display: Wake DMCUB before executing GPINT commands (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Wake DMCUB before executing GPINT commands

[Why]

DMCUB can be in idle when we attempt to interface with the HW through the GPINT mailbox resulting in a system hang.

[How]

Add `dc_wake_and_execute_gpint()` to wrap the wake, execute, sleep sequence.

If the GPINT executes successfully then DMCUB will be put back into sleep after the optional response is returned.

It functions similar to the inbox command interface.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52624>

## **[CVE-2023-52751] kernel: smb: client: fix use-after-free in smb2\_query\_info\_compound() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: fix use-after-free in `smb2_query_info_compound()`

The following UAF was triggered when running `fstests generic/072` with KASAN enabled against Windows Server 2022 and mount options `'multichannel,max_channels=2,vers=3.1.1,mfsymlinks,noperm'`

BUG: KASAN: slab-use-after-free in `smb2_query_info_compound+0x423/0x6d0 [cifs]`  
Read of size 8 at addr `ffff888014941048` by task `xfs_io/27534`

CPU: 0 PID: 27534 Comm: xfs\_io Not tainted 6.6.0-rc7 #1  
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS  
rel-1.16.2-3-gd478f380-rebuilt.opensuse.org 04/01/2014

Call Trace:

```
dump_stack_lvl+0x4a/0x80
print_report+0xcf/0x650
? srso_alias_return_thunk+0x5/0x7f
? srso_alias_return_thunk+0x5/0x7f
? __phys_addr+0x46/0x90
kasan_report+0xda/0x110
? smb2_query_info_compound+0x423/0x6d0 [cifs]
? smb2_query_info_compound+0x423/0x6d0 [cifs]
smb2_query_info_compound+0x423/0x6d0 [cifs]
? __pfx_smb2_query_info_compound+0x10/0x10 [cifs]
? srso_alias_return_thunk+0x5/0x7f
? __stack_depot_save+0x39/0x480
? kasan_save_stack+0x33/0x60
? kasan_set_track+0x25/0x30
? ____kasan_slab_free+0x126/0x170
smb2_queryfs+0xc2/0x2c0 [cifs]
? __pfx_smb2_queryfs+0x10/0x10 [cifs]
? __pfx__lock_acquire+0x10/0x10
smb311_queryfs+0x210/0x220 [cifs]
? __pfx_smb311_queryfs+0x10/0x10 [cifs]
? srso_alias_return_thunk+0x5/0x7f
```

? \_\_lock\_acquire+0x480/0x26c0  
? lock\_release+0x1ed/0x640  
? srso\_alias\_return\_thunk+0x5/0x7f  
? do\_raw\_spin\_unlock+0x9b/0x100  
cifs\_statfs+0x18c/0x4b0 [cifs]  
statfs\_by\_dentry+0x9b/0xf0  
fd\_statfs+0x4e/0xb0  
\_\_do\_sys\_fstatfs+0x7f/0xe0  
? \_\_pfx\_\_do\_sys\_fstatfs+0x10/0x10  
? srso\_alias\_return\_thunk+0x5/0x7f  
? lockdep\_hardirqs\_on\_prepare+0x136/0x200  
? srso\_alias\_return\_thunk+0x5/0x7f  
do\_syscall\_64+0x3f/0x90  
entry\_SYSCALL\_64\_after\_hwframe+0x6e/0xd8

Allocated by task 27534:

kasan\_save\_stack+0x33/0x60  
kasan\_set\_track+0x25/0x30  
\_\_kasan\_kmalloc+0x8f/0xa0  
open\_cached\_dir+0x71b/0x1240 [cifs]  
smb2\_query\_info\_compound+0x5c3/0x6d0 [cifs]  
smb2\_queryfs+0xc2/0x2c0 [cifs]  
smb311\_queryfs+0x210/0x220 [cifs]  
cifs\_statfs+0x18c/0x4b0 [cifs]  
statfs\_by\_dentry+0x9b/0xf0  
fd\_statfs+0x4e/0xb0  
\_\_do\_sys\_fstatfs+0x7f/0xe0  
do\_syscall\_64+0x3f/0x90  
entry\_SYSCALL\_64\_after\_hwframe+0x6e/0xd8

Freed by task 27534:

kasan\_save\_stack+0x33/0x60  
kasan\_set\_track+0x25/0x30  
kasan\_save\_free\_info+0x2b/0x50  
\_\_kasan\_slab\_free+0x126/0x170  
slab\_free\_freelist\_hook+0xd0/0x1e0  
\_\_kmem\_cache\_free+0x9d/0x1b0  
open\_cached\_dir+0xff5/0x1240 [cifs]  
smb2\_query\_info\_compound+0x5c3/0x6d0 [cifs]  
smb2\_queryfs+0xc2/0x2c0 [cifs]

This is a race between open\_cached\_dir() and cached\_dir\_lease\_break() where the cache entry for the open directory handle receives a lease break while creating it. And before returning from open\_cached\_dir(), we put the last reference of the new @cfid because of !@cfid->has\_lease.

Besides the UAF, while running xfstests a lot of missed lease breaks have been noticed in tests that run several concurrent statfs(2) calls on those cached fids

CIFS: VFS: \\w22-root1.gandalf.test No task to wake, unknown frame...

CIFS: VFS: \\w22-root1.gandalf.test Cmd: 18 Err: 0x0 Flags: 0x1...

CIFS: VFS: \\w22-root1.gandalf.test smb buf 00000000715bfe83 len 108  
CIFS: VFS: Dump pending requests:  
CIFS: VFS: \\w22-root1.gandalf.test No task to wake, unknown frame...  
CIFS: VFS: \\w22-root1.gandalf.test Cmd: 18 Err: 0x0 Flags: 0x1...  
CIFS: VFS: \\w22-root1.gandalf.test smb buf 000000005aa7316e len 108  
...

To fix both, in `open_cached_dir()` ensure that `@cfid->has_lease` is set right before sending out compounded request so that any potential lease break will be get processed by demultiplex thread while we're still caching `@cfid`. And, if open failed for some reason, re-check `@cfid->has_lease` to decide whether or not put lease reference.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52751>

### **[CVE-2024-21803] kernel: bluetooth: use-after-free vulnerability in af\_bluetooth.c (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

Use After Free vulnerability in Linux Linux kernel kernel on Linux, x86, ARM (bluetooth modules) allows Local Execution of Code. This vulnerability is associated with program files [https://gitee.Com/anolis/cloud-kernel/blob/devel-5.10/net/bluetooth/af\\_bluetooth.C](https://gitee.Com/anolis/cloud-kernel/blob/devel-5.10/net/bluetooth/af_bluetooth.C).

This issue affects Linux kernel: from v2.6.12-rc2 before v6.8-rc1.

More Info: <https://avd.aquasec.com/nvd/cve-2024-21803>

### **[CVE-2024-25742] hw: amd: Instruction raise #VC exception at exit (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel before 6.9, an untrusted hypervisor can inject virtual interrupt 29 (#VC) at any point in time and can trigger its handler. This affects AMD SEV-SNP and AMD SEV-ES.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25742>

### **[CVE-2024-25743] hw: amd: Instruction raise #VC exception at exit (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel through 6.9, an untrusted hypervisor can inject virtual interrupts 0 and 14 at any point in time and can trigger the SIGFPE signal handler in userspace applications. This affects AMD SEV-SNP and AMD SEV-ES.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25743>

### **[CVE-2024-26669] kernel: net/sched: flower: Fix chain template offload (Severity: HIGH)**

Package: linux-libc-dev



Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/sched: flower: Fix chain template offload

When a qdisc is deleted from a net device the stack instructs the underlying driver to remove its flow offload callback from the associated filter block using the 'FLOW\_BLOCK\_UNBIND' command. The stack then continues to replay the removal of the filters in the block for this driver by iterating over the chains in the block and invoking the 'reoffload' operation of the classifier being used. In turn, the classifier in its 'reoffload' operation prepares and emits a 'FLOW\_CLS\_DESTROY' command for each filter.

However, the stack does not do the same for chain templates and the underlying driver never receives a 'FLOW\_CLS\_TMPLT\_DESTROY' command when a qdisc is deleted. This results in a memory leak [1] which can be reproduced using [2].

Fix by introducing a 'tmplt\_reoffload' operation and have the stack invoke it with the appropriate arguments as part of the replay. Implement the operation in the sole classifier that supports chain templates (flower) by emitting the 'FLOW\_CLS\_TMPLT\_{CREATE,DESTROY}' command based on whether a flow offload callback is being bound to a filter block or being unbound from one.

As far as I can tell, the issue happens since cited commit which reordered `tcf_block_offload_unbind()` before `tcf_block_flush_all_chains()` in `__tcf_block_put()`. The order cannot be reversed as the filter block is expected to be freed after flushing all the chains.

[1]

unreferenced object 0xffff888107e28800 (size 2048):

comm "tc", pid 1079, jiffies 4294958525 (age 3074.287s)

hex dump (first 32 bytes):

```
b1 a6 7c 11 81 88 ff ff e0 5b b3 10 81 88 ff ff ..|.....[.....
01 00 00 00 00 00 00 00 e0 aa b0 84 ff ff ff ff .....
```

backtrace:

```
[<ffffffffff81c06a68>] __kmem_cache_alloc_node+0x1e8/0x320
[<ffffffffff81ab374e>] __kmalloc+0x4e/0x90
[<ffffffffff832aec6d>] mlxsw_sp_acl_ruleset_get+0x34d/0x7a0
[<ffffffffff832bc195>] mlxsw_sp_flower_tmplt_create+0x145/0x180
[<ffffffffff832b2e1a>] mlxsw_sp_flow_block_cb+0x1ea/0x280
[<ffffffffff83a10613>] tc_setup_cb_call+0x183/0x340
[<ffffffffff83a9f85a>] fl_tmplt_create+0x3da/0x4c0
[<ffffffffff83a22435>] tc_ctl_chain+0xa15/0x1170
[<ffffffffff838a863c>] rtnetlink_rcv_msg+0x3cc/0xed0
[<ffffffffff83ac87f0>] netlink_rcv_skb+0x170/0x440
[<ffffffffff83ac6270>] netlink_unicast+0x540/0x820
[<ffffffffff83ac6e28>] netlink_sendmsg+0x8d8/0xda0
[<ffffffffff83793def>] ____sys_sendmsg+0x30f/0xa80
```

```
[<ffffff8379d29a>] __sys_sendmsg+0x13a/0x1e0
[<ffffff8379d50c>] __sys_sendmsg+0x11c/0x1f0
[<ffffff843b9ce0>] do_syscall_64+0x40/0xe0
unreferenced object 0xffff88816d2c0400 (size 1024):
 comm "tc", pid 1079, jiffies 4294958525 (age 3074.287s)
 hex dump (first 32 bytes):
 40 00 00 00 00 00 00 00 57 f6 38 be 00 00 00 00 @.....W.8.....
 10 04 2c 6d 81 88 ff ff 10 04 2c 6d 81 88 ff ff ..,m.....,m....
backtrace:
[<ffffff81c06a68>] __kmem_cache_alloc_node+0x1e8/0x320
[<ffffff81ab36c1>] __kmalloc_node+0x51/0x90
[<ffffff81a8ed96>] kvmalloc_node+0xa6/0x1f0
[<ffffff82827d03>] bucket_table_alloc.isra.0+0x83/0x460
[<ffffff82828d2b>] rhashtable_init+0x43b/0x7c0
[<ffffff832aed48>] mlxsw_sp_acl_ruleset_get+0x428/0x7a0
[<ffffff832bc195>] mlxsw_sp_flow_tmplt_create+0x145/0x180
[<ffffff832b2e1a>] mlxsw_sp_flow_block_cb+0x1ea/0x280
[<ffffff83a10613>] tc_setup_cb_call+0x183/0x340
[<ffffff83a9f85a>] fl_tmplt_create+0x3da/0x4c0
[<ffffff83a22435>] tc_ctl_chain+0xa15/0x1170
[<ffffff838a863c>] rtnetlink_rcv_msg+0x3cc/0xed0
[<ffffff83ac87f0>] netlink_rcv_skb+0x170/0x440
[<ffffff83ac6270>] netlink_unicast+0x540/0x820
[<ffffff83ac6e28>] netlink_sendmsg+0x8d8/0xda0
[<ffffff83793def>] __sys_sendmsg+0x30f/0xa80
```

```
[2]
# tc qdisc add dev swp1 clsact
# tc chain add dev swp1 ingress proto ip chain 1 flower dst_ip 0.0.0.0/32
# tc qdisc del dev
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26669>

## **[CVE-2024-26739] kernel: net/sched: act\_mirred: don't override retval if we already lost the skb (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/sched: act\_mirred: don't override retval if we already lost the skb

If we're redirecting the skb, and haven't called tcf\_mirred\_forward(), yet, we need to tell the core to drop the skb by setting the retcode to SHOT. If we have called tcf\_mirred\_forward(), however, the skb is out of our hands and returning SHOT will lead to UaF.

Move the retval override to the error path which actually need it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26739>

**[CVE-2024-26836] kernel: platform/x86: think-lmi: Fix password opcode ordering for workstations (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

platform/x86: think-lmi: Fix password opcode ordering for workstations

The Lenovo workstations require the password opcode to be run before the attribute value is changed (if Admin password is enabled).

Tested on some Thinkpads to confirm they are OK with this order too.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26836>

**[CVE-2024-26913] kernel: drm/amd/display: Fix dcn35 8k30 Underflow/Corruption Issue (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix dcn35 8k30 Underflow/Corruption Issue

[why]

odm calculation is missing for pipe split policy determination and cause Underflow/Corruption issue.

[how]

Add the odm calculation.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26913>

**[CVE-2024-26930] kernel: scsi: qla2xxx: Fix double free of the ha->vp\_map pointer (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: qla2xxx: Fix double free of the ha->vp\_map pointer

Coverity scan reported potential risk of double free of the pointer ha->vp\_map. ha->vp\_map was freed in qla2x00\_mem\_alloc(), and again freed in function qla2x00\_mem\_free(ha).

Assign NULL to vp\_map and kfree take care of NULL.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26930>

## [CVE-2024-26944] kernel: btrfs: zoned: fix use-after-free in do\_zone\_finish() (Severity: HIGH)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: zoned: fix use-after-free in do\_zone\_finish()

Shinichiro reported the following use-after-free triggered by the device replace operation in fstests btrfs/070.

BTRFS info (device nullb1): scrub: finished on devid 1 with status: 0

=====

BUG: KASAN: slab-use-after-free in do\_zone\_finish+0x91a/0xb90 [btrfs]

Read of size 8 at addr ffff8881543c8060 by task btrfs-cleaner/3494007

CPU: 0 PID: 3494007 Comm: btrfs-cleaner Tainted: G W 6.8.0-rc5-kts #1

Hardware name: Supermicro Super Server/X11SPi-TF, BIOS 3.3 02/21/2020

Call Trace:

<TASK>

dump\_stack\_lvl+0x5b/0x90

print\_report+0xcf/0x670

? \_\_virt\_addr\_valid+0x200/0x3e0

kasan\_report+0xd8/0x110

? do\_zone\_finish+0x91a/0xb90 [btrfs]

? do\_zone\_finish+0x91a/0xb90 [btrfs]

do\_zone\_finish+0x91a/0xb90 [btrfs]

btrfs\_delete\_unused\_bgs+0x5e1/0x1750 [btrfs]

? \_\_pfx\_btrfs\_delete\_unused\_bgs+0x10/0x10 [btrfs]

? btrfs\_put\_root+0x2d/0x220 [btrfs]

? btrfs\_clean\_one\_deleted\_snapshot+0x299/0x430 [btrfs]

cleaner\_kthread+0x21e/0x380 [btrfs]

? \_\_pfx\_cleaner\_kthread+0x10/0x10 [btrfs]

kthread+0x2e3/0x3c0

? \_\_pfx\_kthread+0x10/0x10

ret\_from\_fork+0x31/0x70

? \_\_pfx\_kthread+0x10/0x10

ret\_from\_fork\_asm+0x1b/0x30

</TASK>

Allocated by task 3493983:

kasan\_save\_stack+0x33/0x60

kasan\_save\_track+0x14/0x30

\_\_kasan\_kmalloc+0xaa/0xb0

btrfs\_alloc\_device+0xb3/0x4e0 [btrfs]

device\_list\_add.constprop.0+0x993/0x1630 [btrfs]

btrfs\_scan\_one\_device+0x219/0x3d0 [btrfs]

btrfs\_control\_ioctl+0x26e/0x310 [btrfs]

\_\_x64\_sys\_ioctl+0x134/0x1b0

do\_syscall\_64+0x99/0x190

entry\_SYSCALL\_64\_after\_hwframe+0x6e/0x76

Freed by task 3494056:

```
kasan_save_stack+0x33/0x60
kasan_save_track+0x14/0x30
kasan_save_free_info+0x3f/0x60
poison_slab_object+0x102/0x170
__kasan_slab_free+0x32/0x70
kfree+0x11b/0x320
btrfs_rm_dev_replace_free_srcdev+0xca/0x280 [btrfs]
btrfs_dev_replace_finishing+0xd7e/0x14f0 [btrfs]
btrfs_dev_replace_by_ioctl+0x1286/0x25a0 [btrfs]
btrfs_ioctl+0xb27/0x57d0 [btrfs]
__x64_sys_ioctl+0x134/0x1b0
do_syscall_64+0x99/0x190
entry_SYSCALL_64_after_hwframe+0x6e/0x76
```

The buggy address belongs to the object at ffff8881543c8000

which belongs to the cache kmalloc-1k of size 1024

The buggy address is located 96 bytes inside of

freed 1024-byte region [ffff8881543c8000, ffff8881543c8400)

The buggy address belongs to the physical page:

page:00000000fe2c1285 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x1543c8

head:00000000fe2c1285 order:3 entire\_mapcount:0 nr\_pages\_mapped:0 pincount:0

flags: 0x17fffc0000840(slab|head|node=0|zone=2|lastcpupid=0x1ffff)

page\_type: 0xfffffff()

raw: 0017fffc0000840 ffff888100042dc0 ffffea0019e8f200 dead000000000002

raw: 0000000000000000 000000000100010 00000001ffffff 0000000000000000

page dumped because: kasan: bad access detected

Memory state around the buggy address:

ffff8881543c7f00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

ffff8881543c7f80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

>ffff8881543c8000: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb

^

ffff8881543c8080: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb

ffff8881543c8100: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb

This UAF happens because we're accessing stale zone information of a  
already removed btrfs\_device in do\_zone\_finish().

The sequence of events is as follows:

btrfs\_dev\_replace\_start

btrfs\_scrub\_dev

btrfs\_dev\_replace\_finishing

btrfs\_dev\_replace\_update\_device\_in\_mapping\_tree <-- devices replaced

btrfs\_rm\_dev\_replace\_free\_srcdev

btrfs\_free\_device <-- device freed

cleaner\_kthread

btrfs\_delete\_unused\_bgs

btrfs\_zone\_finish

do\_zone\_finish <-- refers the freed device

The reason for this is that we're using a  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-26944>

## **[CVE-2024-26982] kernel: Squashfs: check the inode number is not the invalid value of zero (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

Squashfs: check the inode number is not the invalid value of zero

Syskiller has produced an out of bounds access in fill\_meta\_index().

That out of bounds access is ultimately caused because the inode has an inode number with the invalid value of zero, which was not checked.

The reason this causes the out of bounds access is due to following sequence of events:

1. Fill\_meta\_index() is called to allocate (via empty\_meta\_index()) and fill a metadata index. It however suffers a data read error and aborts, invalidating the newly returned empty metadata index. It does this by setting the inode number of the index to zero, which means unused (zero is not a valid inode number).
2. When fill\_meta\_index() is subsequently called again on another read operation, locate\_meta\_index() returns the previous index because it matches the inode number of 0. Because this index has been returned it is expected to have been filled, and because it hasn't been, an out of bounds access is performed.

This patch adds a sanity check which checks that the inode number is not zero when the inode is created and returns -EINVAL if it is.

[phillip@squashfs.org.uk: whitespace fix]

Link: <https://lkml.kernel.org/r/20240409204723.446925-1-phillip@squashfs.org.uk>

More Info: <https://avd.aquasec.com/nvd/cve-2024-26982>

## **[CVE-2024-27042] kernel: drm/amdgpu: Fix potential out-of-bounds access in amdgpu\_discovery\_reg\_base\_init() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Fix potential out-of-bounds access in 'amdgpu\_discovery\_reg\_base\_init()'

The issue arises when the array 'adev->vcn.vcn\_config' is accessed before checking if the index 'adev->vcn.num\_vcn\_inst' is within the bounds of the array.

The fix involves moving the bounds check before the array access. This ensures that 'adev->vcn.num\_vcn\_inst' is within the bounds of the array before it is used as an index.

Fixes the below:  
drivers/gpu/drm/amd/amdgpu/amdgpu\_discovery.c:1289 amdgpu\_discovery\_reg\_base\_init() error: testing array offset 'adev->vcn.num\_vcn\_inst' after use.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27042>

**[CVE-2024-35866] kernel: smb: client: fix potential UAF in cifs\_dump\_full\_key() (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

smb: client: fix potential UAF in cifs\_dump\_full\_key()

Skip sessions that are being teared down (status == SES\_EXITING) to avoid UAF.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35866>

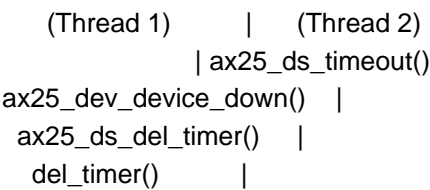
**[CVE-2024-35887] kernel: ax25: fix use-after-free bugs caused by ax25\_ds\_del\_timer (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ax25: fix use-after-free bugs caused by ax25\_ds\_del\_timer

When the ax25 device is detaching, the ax25\_dev\_device\_down() calls ax25\_ds\_del\_timer() to cleanup the slave\_timer. When the timer handler is running, the ax25\_ds\_del\_timer() that calls del\_timer() in it will return directly. As a result, the use-after-free bugs could happen, one of the scenarios is shown below:



```
ax25_dev_put() //FREE |
| ax25_dev-> //USE
```

In order to mitigate bugs, when the device is detaching, use `timer_shutdown_sync()` to stop the timer.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35887>

**[CVE-2024-35929] kernel: rcu/nocb: Fix WARN\_ON\_ONCE() in the rcu\_nocb\_bypass\_lock() (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

rcu/nocb: Fix WARN\_ON\_ONCE() in the rcu\_nocb\_bypass\_lock()

For the kernels built with `CONFIG_RCU_NOCB_CPU_DEFAULT_ALL=y` and `CONFIG_RCU_LAZY=y`, the following scenarios will trigger `WARN_ON_ONCE()` in the `rcu_nocb_bypass_lock()` and `rcu_nocb_wait_contended()` functions:

CPU2	CPU11
kthread	
rcu_nocb_cb_kthread	ksys_write
rcu_do_batch	vfs_write
rcu_torture_timer_cb	proc_sys_write
__kmem_cache_free	proc_sys_call_handler
kmemleak_free	drop_caches_sysctl_handler
delete_object_full	drop_slab
__delete_object	shrink_slab
put_object	lazy_rcu_shrink_scan
call_rcu	rcu_nocb_flush_bypass
__call_rcu_commn	rcu_nocb_bypass_lock
	raw_spin_trylock(&rdp->nocb_bypass_lock) fail
	atomic_inc(&rdp->nocb_lock_contended);
rcu_nocb_wait_contended	WARN_ON_ONCE(smp_processor_id() != rdp->cpu);
WARN_ON_ONCE(atomic_read(&rdp->nocb_lock_contended))	
_ _ _ _ _ _ _ _ _ _ same rdp and rdp->cpu != 11 _ _ _ _ _ _ _ _ _ _	

Reproduce this bug with "echo 3 > /proc/sys/vm/drop\_caches".

This commit therefore uses `rcu_nocb_try_flush_bypass()` instead of `rcu_nocb_flush_bypass()` in `lazy_rcu_shrink_scan()`. If the `nocb_bypass` queue is being flushed, then `rcu_nocb_try_flush_bypass` will return directly.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35929>

**[CVE-2024-36013] kernel: Bluetooth: L2CAP: Fix slab-use-after-free in l2cap\_connect() (Severity: HIGH)**

Package: linux-libc-dev



Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: L2CAP: Fix slab-use-after-free in l2cap\_connect()

Extend a critical section to prevent chan from early freeing.

Also make the l2cap\_connect() return type void. Nothing is using the returned value but it is ugly to return a potentially freed pointer.

Making it void will help with backports because earlier kernels did use the return value. Now the compile will break for kernels where this patch is not a complete fix.

Call stack summary:

[use]

l2cap\_bredr\_sig\_cmd

l2cap\_connect

OE mutex\_lock(&conn->chan\_lock);

, chan = pchan->ops->new\_connection(pchan); <- alloc chan

, \_\_l2cap\_chan\_add(conn, chan);

, l2cap\_chan\_hold(chan);

, list\_add(&chan->list, &conn->chan\_l); ... (1)

mutex\_unlock(&conn->chan\_lock);

chan->conf\_state ... (4) <- use after free

[free]

l2cap\_conn\_del

OE mutex\_lock(&conn->chan\_lock);

, foreach chan in conn->chan\_l: ... (2)

, l2cap\_chan\_put(chan);

, l2cap\_chan\_destroy

, kfree(chan) ... (3) <- chan freed

mutex\_unlock(&conn->chan\_lock);

=====

BUG: KASAN: slab-use-after-free in instrument\_atomic\_read

include/linux/instrumented.h:68 [inline]

BUG: KASAN: slab-use-after-free in \_test\_bit

include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline]

BUG: KASAN: slab-use-after-free in l2cap\_connect+0xa67/0x11a0

net/bluetooth/l2cap\_core.c:4260

Read of size 8 at addr ffff88810bf040a0 by task kworker/u3:1/311

More Info: <https://avd.aquasec.com/nvd/cve-2024-36013>

**[CVE-2024-36921] kernel: wifi: iwlmwifi: mvm: guard against invalid STA ID on removal (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: iwlmwifi: mvm: guard against invalid STA ID on removal

Guard against invalid station IDs in iwl\_mvm\_mld\_rm\_sta\_id as that would result in out-of-bounds array accesses. This prevents issues should the driver get into a bad state during error handling.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36921>

### **[CVE-2024-38570] kernel: gfs2: Fix potential glock use-after-free on unmount (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

gfs2: Fix potential glock use-after-free on unmount

When a DLM lockspace is released and there are still locks in that lockspace, DLM will unlock those locks automatically. Commit fb6791d100d1b started exploiting this behavior to speed up filesystem unmount: gfs2 would simply free glocks it didn't want to unlock and then release the lockspace. This didn't take the bast callbacks for asynchronous lock contention notifications into account, which remain active until a lock is unlocked or its lockspace is released.

To prevent those callbacks from accessing deallocated objects, put the glocks that should not be unlocked on the sd\_dead\_glocks list, release the lockspace, and only then free those glocks.

As an additional measure, ignore unexpected ast and bast callbacks if the receiving glock is dead.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38570>

### **[CVE-2024-38630] kernel: watchdog: cpu5wdt.c: Fix use-after-free bug caused by cpu5wdt\_trigger (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

watchdog: cpu5wdt.c: Fix use-after-free bug caused by cpu5wdt\_trigger

When the cpu5wdt module is removing, the origin code uses del\_timer() to de-activate the timer. If the timer handler is running, del\_timer() could not stop it and will return directly. If the port region is released by release\_region() and then the timer handler cpu5wdt\_trigger() calls outb() to write into the region that is released, the use-after-free bug will happen.

Change `del_timer()` to `timer_shutdown_sync()` in order that the timer handler could be finished before the port region is released.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38630>

### **[CVE-2024-39479] kernel: drm/i915/hwmon: Get rid of devm (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/i915/hwmon: Get rid of devm

When both hwmon and hwmon drvdata (on which hwmon depends) are device managed resources, the expectation, on device unbind, is that hwmon will be released before drvdata. However, in i915 there are two separate code paths, which both release either drvdata or hwmon and either can be released before the other. These code paths (for device unbind) are as follows (see also the bug referenced below):

Call Trace:

```
release_nodes+0x11/0x70
devres_release_group+0xb2/0x110
component_unbind_all+0x8d/0xa0
component_del+0xa5/0x140
intel_pxp_tee_component_fini+0x29/0x40 [i915]
intel_pxp_fini+0x33/0x80 [i915]
i915_driver_remove+0x4c/0x120 [i915]
i915_pci_remove+0x19/0x30 [i915]
pci_device_remove+0x32/0xa0
device_release_driver_internal+0x19c/0x200
unbind_store+0x9c/0xb0
```

and

Call Trace:

```
release_nodes+0x11/0x70
devres_release_all+0x8a/0xc0
device_unbind_cleanup+0x9/0x70
device_release_driver_internal+0x1c1/0x200
unbind_store+0x9c/0xb0
```

This means that in i915, if use devm, we cannot guarantee that hwmon will always be released before drvdata. Which means that we have a uaf if hwmon sysfs is accessed when drvdata has been released but hwmon hasn't.

The only way out of this seems to be to get rid of devm\_ and release/free everything explicitly during device unbind.

v2: Change commit message and other minor code changes

v3: Cleanup from i915\_hwmon\_register on error (Armin Wolf)

v4: Eliminate potential static analyzer warning (Rodrigo)

Eliminate fetch\_and\_zero (Jani)  
v5: Restore previous logic for ddat\_gt->hwmon\_dev error return (Andi)

More Info: <https://avd.aquasec.com/nvd/cve-2024-39479>

### **[CVE-2024-39508] kernel: io\_uring/io-wq: Use set\_bit() and test\_bit() at worker->flags (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

io\_uring/io-wq: Use set\_bit() and test\_bit() at worker->flags

Utilize set\_bit() and test\_bit() on worker->flags within io\_uring/io-wq to address potential data races.

The structure io\_worker->flags may be accessed through various data paths, leading to concurrency issues. When KCSAN is enabled, it reveals data races occurring in io\_worker\_handle\_work and io\_wq\_activate\_free\_worker functions.

BUG: KCSAN: data-race in io\_worker\_handle\_work / io\_wq\_activate\_free\_worker  
write to 0xffff8885c4246404 of 4 bytes by task 49071 on cpu 28:  
io\_worker\_handle\_work (io\_uring/io-wq.c:434 io\_uring/io-wq.c:569)  
io\_wq\_worker (io\_uring/io-wq.c:?)  
<snip>

read to 0xffff8885c4246404 of 4 bytes by task 49024 on cpu 5:  
io\_wq\_activate\_free\_worker (io\_uring/io-wq.c:? io\_uring/io-wq.c:285)  
io\_wq\_enqueue (io\_uring/io-wq.c:947)  
io\_queue\_iowq (io\_uring/io\_uring.c:524)  
io\_req\_task\_submit (io\_uring/io\_uring.c:1511)  
io\_handle\_tw\_list (io\_uring/io\_uring.c:1198)  
<snip>

Line numbers against commit 18daea77cca6 ("Merge tag 'for-linus' of git://git.kernel.org/pub/scm/virt/kvm/kvm").

These races involve writes and reads to the same memory location by different tasks running on different CPUs. To mitigate this, refactor the code to use atomic operations such as set\_bit(), test\_bit(), and clear\_bit() instead of basic "and" and "or" operations. This ensures thread-safe manipulation of worker flags.

Also, move `create\_index` to avoid holes in the structure.

More Info: <https://avd.aquasec.com/nvd/cve-2024-39508>

### **[CVE-2024-41013] kernel: xfs: don't walk off the end of a directory data block (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

xfs: don't walk off the end of a directory data block

This adds sanity checks for xfs\_dir2\_data\_unused and xfs\_dir2\_data\_entry to make sure don't stray beyond valid memory region. Before patching, the loop simply checks that the start offset of the dup and dep is within the range. So in a crafted image, if last entry is xfs\_dir2\_data\_unused, we can change dup->length to dup->length-1 and leave 1 byte of space. In the next traversal, this space will be considered as dup or dep. We may encounter an out of bound read when accessing the fixed members.

In the patch, we make sure that the remaining bytes large enough to hold an unused entry before accessing xfs\_dir2\_data\_unused and xfs\_dir2\_data\_unused is XFS\_DIR2\_DATA\_ALIGN byte aligned. We also make sure that the remaining bytes large enough to hold a dirent with a single-byte name before accessing xfs\_dir2\_data\_entry.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41013>

### **[CVE-2024-42162] kernel: gve: Account for stopped queues when reading NIC stats (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

gve: Account for stopped queues when reading NIC stats

We now account for the fact that the NIC might send us stats for a subset of queues. Without this change, gve\_get\_ethtool\_stats might make an invalid access on the priv->stats\_report->stats array.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42162>

### **[CVE-2024-44941] kernel: f2fs: fix to cover read extent cache access with lock (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to cover read extent cache access with lock

syzbot reports a f2fs bug as below:

BUG: KASAN: slab-use-after-free in sanity\_check\_extent\_cache+0x370/0x410 fs/f2fs/extent\_cache.c:46  
Read of size 4 at addr ffff8880739ab220 by task syz-executor200/5097

CPU: 0 PID: 5097 Comm: syz-executor200 Not tainted 6.9.0-rc6-syzkaller #0

Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/27/2024

Call Trace:

<TASK>

```
__dump_stack lib/dump_stack.c:88 [inline]
dump_stack_lvl+0x241/0x360 lib/dump_stack.c:114
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
sanity_check_extent_cache+0x370/0x410 fs/f2fs/extent_cache.c:46
do_read_inode fs/f2fs/inode.c:509 [inline]
f2fs_iget+0x33e1/0x46e0 fs/f2fs/inode.c:560
f2fs_nfs_get_inode+0x74/0x100 fs/f2fs/super.c:3237
generic_fh_to_dentry+0x9f/0xf0 fs/libfs.c:1413
exportfs_decode_fh_raw+0x152/0x5f0 fs/exportfs/expfs.c:444
exportfs_decode_fh+0x3c/0x80 fs/exportfs/expfs.c:584
do_handle_to_path fs/fhandle.c:155 [inline]
handle_to_path fs/fhandle.c:210 [inline]
do_handle_open+0x495/0x650 fs/fhandle.c:226
do_syscall_x64 arch/x86/entry/common.c:52 [inline]
do_syscall_64+0xf5/0x240 arch/x86/entry/common.c:83
entry_SYSCALL_64_after_hwframe+0x77/0x7f
```

We missed to cover `sanity_check_extent_cache()` w/ extent cache lock, so, below race case may happen, result in use after free issue.

```
- f2fs_iget
- do_read_inode
- f2fs_init_read_extent_tree
: add largest extent entry in to cache
  - shrink
  - f2fs_shrink_read_extent_tree
  - __shrink_extent_tree
  - __detach_extent_node
: drop largest extent entry
- sanity_check_extent_cache
: access et->largest w/o lock
```

let's refactor `sanity_check_extent_cache()` to avoid extent cache access and call it before `f2fs_init_read_extent_tree()` to fix this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44941>

### **[CVE-2024-44942] kernel: f2fs: fix to do sanity check on F2FS\_INLINE\_DATA flag in inode during GC (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to do sanity check on F2FS\_INLINE\_DATA flag in inode during GC

syzbot reports a f2fs bug as below:

-----[ cut here ]-----

kernel BUG at fs/f2fs/inline.c:258!

CPU: 1 PID: 34 Comm: kworker/u8:2 Not tainted 6.9.0-rc6-syzkaller-00012-g9e4bc4bcae01 #0

RIP: 0010:f2fs\_write\_inline\_data+0x781/0x790 fs/f2fs/inline.c:258

Call Trace:

f2fs\_write\_single\_data\_page+0xb65/0x1d60 fs/f2fs/data.c:2834  
f2fs\_write\_cache\_pages fs/f2fs/data.c:3133 [inline]  
\_\_f2fs\_write\_data\_pages fs/f2fs/data.c:3288 [inline]  
f2fs\_write\_data\_pages+0x1efe/0x3a90 fs/f2fs/data.c:3315  
do\_writepages+0x35b/0x870 mm/page-writeback.c:2612  
\_\_writeback\_single\_inode+0x165/0x10b0 fs/fs-writeback.c:1650  
writeback\_sb\_inodes+0x905/0x1260 fs/fs-writeback.c:1941  
wb\_writeback+0x457/0xce0 fs/fs-writeback.c:2117  
wb\_do\_writeback fs/fs-writeback.c:2264 [inline]  
wb\_workfn+0x410/0x1090 fs/fs-writeback.c:2304  
process\_one\_work kernel/workqueue.c:3254 [inline]  
process\_scheduled\_works+0xa12/0x17c0 kernel/workqueue.c:3335  
worker\_thread+0x86d/0xd70 kernel/workqueue.c:3416  
kthread+0x2f2/0x390 kernel/kthread.c:388  
ret\_from\_fork+0x4d/0x80 arch/x86/kernel/process.c:147  
ret\_from\_fork\_asm+0x1a/0x30 arch/x86/entry/entry\_64.S:244

The root cause is: inline\_data inode can be fuzzed, so that there may be valid blkaddr in its direct node, once f2fs triggers background GC to migrate the block, it will hit f2fs\_bug\_on() during dirty page writeback.

Let's add sanity check on F2FS\_INLINE\_DATA flag in inode during GC, so that, it can forbid migrating inline\_data inode's data block for fixing.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44942>

### **[CVE-2024-44951] kernel: serial: sc16is7xx: fix TX fifo corruption (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

serial: sc16is7xx: fix TX fifo corruption

Sometimes, when a packet is received on channel A at almost the same time as a packet is about to be transmitted on channel B, we observe with a logic analyzer that the received packet on channel A is transmitted on channel B. In other words, the Tx buffer data on channel B is corrupted with data from channel A.

The problem appeared since commit 4409df5866b7 ("serial: sc16is7xx: change EFR lock to operate on each channels"), which changed the EFR locking to operate on each channel instead of chip-wise.

This commit has introduced a regression, because the EFR lock is used not only to protect the EFR registers access, but also, in a very obscure and undocumented way, to protect access to the data buffer, which is shared by the Tx and Rx handlers, but also by each channel of the IC.

Fix this regression first by switching to `kfifo_out_linear_ptr()` in `sc16is7xx_handle_tx()` to eliminate the need for a shared Rx/Tx buffer.

Secondly, replace the chip-wise Rx buffer with a separate Rx buffer for each channel.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44951>

### **[CVE-2024-46774] kernel: powerpc/rtas: Prevent Spectre v1 gadget construction in sys\_rtas() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

powerpc/rtas: Prevent Spectre v1 gadget construction in sys\_rtas()

Smatch warns:

```
arch/powerpc/kernel/rtas.c:1932 __do_sys_rtas() warn: potential  
spectre issue 'args.args' [r] (local cap)
```

The 'nargs' and 'nret' locals come directly from a user-supplied buffer and are used as indexes into a small stack-based array and as inputs to `copy_to_user()` after they are subject to bounds checks.

Use `array_index_nospec()` after the bounds checks to clamp these values for speculative execution.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46774>

### **[CVE-2024-46786] kernel: fscache: delete fscache\_cookie\_lru\_timer when fscache exits to avoid UAF (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fscache: delete fscache\_cookie\_lru\_timer when fscache exits to avoid UAF

The `fscache_cookie_lru_timer` is initialized when the fscache module is inserted, but is not deleted when the fscache module is removed.

If `timer_reduce()` is called before removing the fscache module, the `fscache_cookie_lru_timer` will be added to the timer list of



the current cpu. Afterwards, a use-after-free will be triggered in the softIRQ after removing the fscache module, as follows:

```
=====
BUG: unable to handle page fault for address: fffffbfff803c9e9
PF: supervisor read access in kernel mode
PF: error_code(0x0000) - not-present page
PGD 21f6ea067 P4D 21f6ea067 PUD 21f6e6067 PMD 110a7c067 PTE 0
Oops: Oops: 0000 [#1] PREEMPT SMP KASAN PTI
CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Tainted: G W 6.11.0-rc3 #855
Tainted: [W]=WARN
RIP: 0010:__run_timer_base.part.0+0x254/0x8a0
Call Trace:
<IRQ>
tmigr_handle_remote_up+0x627/0x810
__walk_groups.isra.0+0x47/0x140
tmigr_handle_remote+0x1fa/0x2f0
handle_softirqs+0x180/0x590
irq_exit_rcu+0x84/0xb0
sysvec_apic_timer_interrupt+0x6e/0x90
</IRQ>
<TASK>
asm_sysvec_apic_timer_interrupt+0x1a/0x20
RIP: 0010:default_idle+0xf/0x20
default_idle_call+0x38/0x60
do_idle+0x2b5/0x300
cpu_startup_entry+0x54/0x60
start_secondary+0x20d/0x280
common_startup_64+0x13e/0x148
</TASK>
Modules linked in: [last unloaded: netfs]
=====
```

Therefore delete fscache\_cookie\_lru\_timer when removing the fscache module.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46786>

**[CVE-2024-46811] kernel: drm/amd/display: Fix index may exceed array range within fpu\_update\_bw\_bounding\_box (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix index may exceed array range within fpu\_update\_bw\_bounding\_box

**[Why]**

Coverity reports OVERRUN warning. soc.num\_states could be 40. But array range of bw\_params->clk\_table.entries is 8.

**[How]**

Assert if soc.num\_states greater than 8.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46811>

**[CVE-2024-46813] kernel: drm/amd/display: Check link\_index before accessing dc->links[] (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Check link\_index before accessing dc->links[]

[WHY & HOW]

dc->links[] has max size of MAX\_LINKS and NULL is return when trying to access with out-of-bound index.

This fixes 3 OVERRUN and 1 RESOURCE\_LEAK issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46813>

**[CVE-2024-47691] kernel: f2fs: fix to avoid use-after-free in f2fs\_stop\_gc\_thread() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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;

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47691>

### **[CVE-2024-49928] kernel: wifi: rtw89: avoid reading out of bounds when loading TX power FW elements (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49928>

### **[CVE-2024-50029] kernel: Bluetooth: hci\_conn: Fix UAF in hci\_enhanced\_setup\_sync (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

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

? 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

More Info: <https://avd.aquasec.com/nvd/cve-2024-50029>

## **[CVE-2024-50063] kernel: bpf: Prevent tail call between progs attached to different hooks (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Prevent tail call between progs attached to different hooks

bpf progs can be attached to kernel functions, and the attached functions can take different parameters or return different return values. If prog attached to one kernel function tail calls prog attached to another kernel function, the ctx access or return value verification could be bypassed.

For example, if prog1 is attached to func1 which takes only 1 parameter and prog2 is attached to func2 which takes two parameters. Since verifier assumes the bpf ctx passed to prog2 is constructed based on func2's prototype, verifier allows prog2 to access the second parameter from the bpf ctx passed to it. The problem is that verifier does not prevent prog1 from passing its bpf ctx to prog2 via tail call. In this case, the bpf ctx passed to prog2 is constructed from func1 instead of func2, that is, the assumption for ctx access verification is bypassed.

Another example, if BPF LSM prog1 is attached to hook file\_alloc\_security, and BPF LSM prog2 is attached to hook bpf\_lsm\_audit\_rule\_known. Verifier knows the return value rules for these two hooks, e.g. it is legal for bpf\_lsm\_audit\_rule\_known to return positive number 1, and it is illegal for file\_alloc\_security to return positive number. So verifier allows prog2 to return positive number 1, but does not allow prog1 to return positive number. The problem is that verifier does not prevent prog1 from calling prog2 via tail call. In this case, prog2's return value 1 will be used as the return value for prog1's hook file\_alloc\_security. That is, the return value rule is bypassed.

This patch adds restriction for tail call to prevent such bypasses.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50063>

**[CVE-2024-50112] kernel: x86/lam: Disable ADDRESS\_MASKING in most cases (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

x86/lam: Disable ADDRESS\_MASKING in most cases

Linear Address Masking (LAM) has a weakness related to transient execution as described in the SLAM paper[1]. Unless Linear Address Space Separation (LASS) is enabled this weakness may be exploitable.

Until kernel adds support for LASS[2], only allow LAM for COMPILE\_TEST, or when speculation mitigations have been disabled at compile time, otherwise keep LAM disabled.

There are no processors in market that support LAM yet, so currently nobody is affected by this issue.

[1] SLAM: [https://download.vusec.net/papers/slam\\_sp24.pdf](https://download.vusec.net/papers/slam_sp24.pdf)

[2] LASS: <https://lore.kernel.org/lkml/20230609183632.48706-1-alexander.shishkin@linux.intel.com/>

[ dhansen: update SPECULATION\_MITIGATIONS -> CPU\_MITIGATIONS ]

More Info: <https://avd.aquasec.com/nvd/cve-2024-50112>

**[CVE-2024-50217] kernel: btrfs: fix use-after-free of block device file in \_\_btrfs\_free\_extra\_devids() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: fix use-after-free of block device file in \_\_btrfs\_free\_extra\_devids()

Mounting btrfs from two images (which have the same one fsid and two different dev\_uuids) in certain executing order may trigger an UAF for variable 'device->bdev\_file' in \_\_btrfs\_free\_extra\_devids(). And following are the details:

1. Attach image\_1 to loop0, attach image\_2 to loop1, and scan btrfs devices by ioctl(BTRFS\_IOC\_SCAN\_DEV):

```
    / btrfs_device_1 â†’ loop0
fs_device
```

```

    \ btrfs_device_2 â†' loop1
2. mount /dev/loop0 /mnt
   btrfs_open_devices
   btrfs_device_1->bdev_file = btrfs_get_bdev_and_sb(loop0)
   btrfs_device_2->bdev_file = btrfs_get_bdev_and_sb(loop1)
   btrfs_fill_super
   open_ctree
   fail: btrfs_close_devices // -ENOMEM
   btrfs_close_bdev(btrfs_device_1)
       fput(btrfs_device_1->bdev_file)
       // btrfs_device_1->bdev_file is freed
   btrfs_close_bdev(btrfs_device_2)
       fput(btrfs_device_2->bdev_file)

3. mount /dev/loop1 /mnt
   btrfs_open_devices
   btrfs_get_bdev_and_sb(&bdev_file)
   // EIO, btrfs_device_1->bdev_file is not assigned,
   // which points to a freed memory area
   btrfs_device_2->bdev_file = btrfs_get_bdev_and_sb(loop1)
   btrfs_fill_super
   open_ctree
   btrfs_free_extra_devids
   if (btrfs_device_1->bdev_file)
       fput(btrfs_device_1->bdev_file) // UAF !

```

Fix it by setting 'device->bdev\_file' as 'NULL' after closing the btrfs\_device in btrfs\_close\_one\_device().

More Info: <https://avd.aquasec.com/nvd/cve-2024-50217>

## **[CVE-2024-50226] kernel: cxl/port: Fix use-after-free, permit out-of-order decoder shutdown (Severity: HIGH)**

Package: linux-libc-dev  
 Installed: 6.1.129-1  
 Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

cxl/port: Fix use-after-free, permit out-of-order decoder shutdown

In support of investigating an initialization failure report [1], cxl\_test was updated to register mock memory-devices after the mock root-port/bus device had been registered. That led to cxl\_test crashing with a use-after-free bug with the following signature:

```

cxl_port_attach_region: cxl region3: cxl_host_bridge.0:port3 decoder3.0 add: mem0:decoder7.0 @ 0 next:
cxl_switch_uport.0 nr_eps: 1 nr_targets: 1
cxl_port_attach_region: cxl region3: cxl_host_bridge.0:port3 decoder3.0 add: mem4:decoder14.0 @ 1 next:
cxl_switch_uport.0 nr_eps: 2 nr_targets: 1
cxl_port_setup_targets: cxl region3: cxl_switch_uport.0:port6 target[0] = cxl_switch_dport.0 for mem0:decoder7.0 @ 0
1) cxl_port_setup_targets: cxl region3: cxl_switch_uport.0:port6 target[1] = cxl_switch_dport.4 for mem4:decoder14.0 @

```

1

```
[..]
cxld_unregister: cxl decoder14.0:
cxl_region_decode_reset: cxl_region region3:
mock_decoder_reset: cxl_port port3: decoder3.0 reset
2) mock_decoder_reset: cxl_port port3: decoder3.0: out of order reset, expected decoder3.1
cxl_endpoint_decoder_release: cxl decoder14.0:
[..]
cxld_unregister: cxl decoder7.0:
3) cxl_region_decode_reset: cxl_region region3:
Oops: general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6bc3: 0000 [#1] PREEMPT SMP
PTI
[..]
RIP: 0010:to_cxl_port+0x8/0x60 [cxl_core]
[..]
Call Trace:
<TASK>
cxl_region_decode_reset+0x69/0x190 [cxl_core]
cxl_region_detach+0xe8/0x210 [cxl_core]
cxl_decoder_kill_region+0x27/0x40 [cxl_core]
cxld_unregister+0x5d/0x60 [cxl_core]
```

At 1) a region has been established with 2 endpoint decoders (7.0 and 14.0). Those endpoints share a common switch-decoder in the topology (3.0). At teardown, 2), decoder14.0 is the first to be removed and hits the "out of order reset case" in the switch decoder. The effect though is that region3 cleanup is aborted leaving it in-tact and referencing decoder14.0. At 3) the second attempt to teardown region3 trips over the stale decoder14.0 object which has long since been deleted.

The fix here is to recognize that the CXL specification places no mandate on in-order shutdown of switch-decoders, the driver enforces in-order allocation, and hardware enforces in-order commit. So, rather than fail and leave objects dangling, always remove them.

In support of making `cxl_region_decode_reset()` always succeed, `cxl_region_invalidate_memregion()` failures are turned into warnings. Crashing the kernel is ok there since system integrity is at risk if caches cannot be managed around physical address mutation events like CXL region destruction.

A new `device_for_each_child_reverse_from()` is added to cleanup `port->commit_end` after all dependent decoders have been disabled. In other words if decoders are allocated 0->1->2 and disabled 1->2->0 then `port->commit_end` only decrements from 2 after 2 has been disabled, and it decrements all the way to zero since 1 was disabled previously.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50226>

### **[CVE-2024-50246] kernel: fs/ntfs3: Add rough attr alloc\_size check (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1



Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

fs/ntfs3: Add rough attr alloc\_size check

More Info: <https://avd.aquasec.com/nvd/cve-2024-50246>

**[CVE-2024-53068] kernel: firmware: arm\_scmi: Fix slab-use-after-free in scmi\_bus\_notifier() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

firmware: arm\_scmi: Fix slab-use-after-free in scmi\_bus\_notifier()

The scmi\_dev->name is released prematurely in \_\_scmi\_device\_destroy(), which causes slab-use-after-free when accessing scmi\_dev->name in scmi\_bus\_notifier(). So move the release of scmi\_dev->name to scmi\_device\_release() to avoid slab-use-after-free.

```
| BUG: KASAN: slab-use-after-free in strncmp+0xe4/0xec
| Read of size 1 at addr fffff80a482bcc0 by task swapper/0/1
|
| CPU: 1 PID: 1 Comm: swapper/0 Not tainted 6.6.38-debug #1
| Hardware name: Qualcomm Technologies, Inc. SA8775P Ride (DT)
| Call trace:
| dump_backtrace+0x94/0x114
| show_stack+0x18/0x24
| dump_stack_lvl+0x48/0x60
| print_report+0xf4/0x5b0
| kasan_report+0xa4/0xec
| __asan_report_load1_noabort+0x20/0x2c
| strncmp+0xe4/0xec
| scmi_bus_notifier+0x5c/0x54c
| notifier_call_chain+0xb4/0x31c
| blocking_notifier_call_chain+0x68/0x9c
| bus_notify+0x54/0x78
| device_del+0x1bc/0x840
| device_unregister+0x20/0xb4
| __scmi_device_destroy+0xac/0x280
| scmi_device_destroy+0x94/0xd0
| scmi_chan_setup+0x524/0x750
| scmi_probe+0x7fc/0x1508
| platform_probe+0xc4/0x19c
| really_probe+0x32c/0x99c
| __driver_probe_device+0x15c/0x3c4
| driver_probe_device+0x5c/0x170
| __driver_attach+0x1c8/0x440
| bus_for_each_dev+0xf4/0x178
```

| driver\_attach+0x3c/0x58  
| bus\_add\_driver+0x234/0x4d4  
| driver\_register+0xf4/0x3c0  
| \_\_platform\_driver\_register+0x60/0x88  
| scmi\_driver\_init+0xb0/0x104  
| do\_one\_initcall+0xb4/0x664  
| kernel\_init\_freeable+0x3c8/0x894  
| kernel\_init+0x24/0x1e8  
| ret\_from\_fork+0x10/0x20  
|  
| Allocated by task 1:  
| kasan\_save\_stack+0x2c/0x54  
| kasan\_set\_track+0x2c/0x40  
| kasan\_save\_alloc\_info+0x24/0x34  
| \_\_kasan\_kmalloc+0xa0/0xb8  
| \_\_kmalloc\_node\_track\_caller+0x6c/0x104  
| kstrdup+0x48/0x84  
| kstrdup\_const+0x34/0x40  
| \_\_scmi\_device\_create.part.0+0x8c/0x408  
| scmi\_device\_create+0x104/0x370  
| scmi\_chan\_setup+0x2a0/0x750  
| scmi\_probe+0x7fc/0x1508  
| platform\_probe+0xc4/0x19c  
| really\_probe+0x32c/0x99c  
| \_\_driver\_probe\_device+0x15c/0x3c4  
| driver\_probe\_device+0x5c/0x170  
| \_\_driver\_attach+0x1c8/0x440  
| bus\_for\_each\_dev+0xf4/0x178  
| driver\_attach+0x3c/0x58  
| bus\_add\_driver+0x234/0x4d4  
| driver\_register+0xf4/0x3c0  
| \_\_platform\_driver\_register+0x60/0x88  
| scmi\_driver\_init+0xb0/0x104  
| do\_one\_initcall+0xb4/0x664  
| kernel\_init\_freeable+0x3c8/0x894  
| kernel\_init+0x24/0x1e8  
| ret\_from\_fork+0x10/0x20  
|  
| Freed by task 1:  
| kasan\_save\_stack+0x2c/0x54  
| kasan\_set\_track+0x2c/0x40  
| kasan\_save\_free\_info+0x38/0x5c  
| \_\_kasan\_slab\_free+0xe8/0x164  
| \_\_kmem\_cache\_free+0x11c/0x230  
| kfree+0x70/0x130  
| kfree\_const+0x20/0x40  
| \_\_scmi\_device\_destroy+0x70/0x280  
| scmi\_device\_destroy+0x94/0xd0  
| scmi\_chan\_setup+0x524/0x750  
| scmi\_probe+0x7fc/0x1508  
| platform\_probe+0xc4/0x19c  
| really\_probe+0x32c/0x99c  
| \_\_driver\_probe\_device+0x15c/0x3c4

| driver\_probe\_device+0x5c/0x170  
| \_\_driver\_attach+0x1c8/0x440  
| bus\_for\_each\_dev+0xf4/0x178  
| driver\_attach+0x3c/0x58  
| bus\_add\_driver+0x234/0x4d4  
| driver\_register+0xf4/0x3c0  
| \_\_platform\_driver\_register+0x60/0x88  
| scmi\_driver\_init+0xb0/0x104  
| do\_one\_initcall+0xb4/0x664  
| kernel\_init\_freeable+0x3c8/0x894  
| kernel\_init+0x24/0x1e8  
| ret\_from\_fork+0x10/0x20

More Info: <https://avd.aquasec.com/nvd/cve-2024-53068>

**[CVE-2024-53108] kernel: drm/amd/display: Adjust VSDB parser for replay feature (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Adjust VSDB parser for replay feature

At some point, the IEEE ID identification for the replay check in the AMD EDID was added. However, this check causes the following out-of-bounds issues when using KASAN:

```
[ 27.804016] BUG: KASAN: slab-out-of-bounds in amdgpu_dm_update_freesync_caps+0xefa/0x17a0 [amdgpu]
[ 27.804788] Read of size 1 at addr ffff8881647fdb00 by task systemd-udevd/383
```

...

```
[ 27.821207] Memory state around the buggy address:
[ 27.821215] ffff8881647fda00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[ 27.821224] ffff8881647fda80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[ 27.821234] >ffff8881647fdb00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[ 27.821243]                ^
[ 27.821250] ffff8881647fdb80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[ 27.821259] ffff8881647fdc00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[ 27.821268] =====
```

This is caused because the ID extraction happens outside of the range of the edid lenght. This commit addresses this issue by considering the amd\_vsdb\_block size.

(cherry picked from commit b7e381b1ccd5e778e3d9c44c669ad38439a861d8)

More Info: <https://avd.aquasec.com/nvd/cve-2024-53108>

**[CVE-2024-53133] kernel: drm/amd/display: Handle dml allocation failure to avoid crash (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Handle dml allocation failure to avoid crash

[Why]

In the case where a dml allocation fails for any reason, the current state's dml contexts would no longer be valid. Then subsequent calls `dc_state_copy_internal` would shallow copy invalid memory and if the new state was released, a double free would occur.

[How]

Reset dml pointers in `new_state` to NULL and avoid invalid pointer

(cherry picked from commit `bcafdc61529a48f6f06355d78eb41b3aeda5296c`)

More Info: <https://avd.aquasec.com/nvd/cve-2024-53133>

### **[CVE-2024-53166] kernel: block, bfq: fix bfqq uaf in bfq\_limit\_depth() (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

block, bfq: fix bfqq uaf in `bfq_limit_depth()`

Set new allocated bfqq to bic or remove freed bfqq from bic are both protected by `bfqd->lock`, however `bfq_limit_depth()` is deferencing bfqq from bic without the lock, this can lead to UAF if the `io_context` is shared by multiple tasks.

For example, test bfq with `io_uring` can trigger following UAF in v6.6:

=====  
BUG: KASAN: slab-use-after-free in `bfqq_group+0x15/0x50`

Call Trace:

<TASK>

`dump_stack_lvl+0x47/0x80`  
`print_address_description.constprop.0+0x66/0x300`  
`print_report+0x3e/0x70`  
`kasan_report+0xb4/0xf0`  
`bfqq_group+0x15/0x50`  
`bfqq_request_over_limit+0x130/0x9a0`  
`bfq_limit_depth+0x1b5/0x480`  
`__blk_mq_alloc_requests+0x2b5/0xa00`  
`blk_mq_get_new_requests+0x11d/0x1d0`

blk\_mq\_submit\_bio+0x286/0xb00  
submit\_bio\_noacct\_nocheck+0x331/0x400  
\_\_block\_write\_full\_folio+0x3d0/0x640  
writepage\_cb+0x3b/0xc0  
write\_cache\_pages+0x254/0x6c0  
write\_cache\_pages+0x254/0x6c0  
do\_writepages+0x192/0x310  
filemap\_fdatawrite\_wbc+0x95/0xc0  
\_\_filemap\_fdatawrite\_range+0x99/0xd0  
filemap\_write\_and\_wait\_range.part.0+0x4d/0xa0  
blkdev\_read\_iter+0xef/0x1e0  
io\_read+0x1b6/0x8a0  
io\_issue\_sqe+0x87/0x300  
io\_wq\_submit\_work+0xeb/0x390  
io\_worker\_handle\_work+0x24d/0x550  
io\_wq\_worker+0x27f/0x6c0  
ret\_from\_fork\_asm+0x1b/0x30  
</TASK>

Allocated by task 808602:

kasan\_save\_stack+0x1e/0x40  
kasan\_set\_track+0x21/0x30  
\_\_kasan\_slab\_alloc+0x83/0x90  
kmem\_cache\_alloc\_node+0x1b1/0x6d0  
bfq\_get\_queue+0x138/0xfa0  
bfq\_get\_bfqq\_handle\_split+0xe3/0x2c0  
bfq\_init\_rq+0x196/0xbb0  
bfq\_insert\_request.isra.0+0xb5/0x480  
bfq\_insert\_requests+0x156/0x180  
blk\_mq\_insert\_request+0x15d/0x440  
blk\_mq\_submit\_bio+0x8a4/0xb00  
submit\_bio\_noacct\_nocheck+0x331/0x400  
\_\_blkdev\_direct\_IO\_async+0x2dd/0x330  
blkdev\_write\_iter+0x39a/0x450  
io\_write+0x22a/0x840  
io\_issue\_sqe+0x87/0x300  
io\_wq\_submit\_work+0xeb/0x390  
io\_worker\_handle\_work+0x24d/0x550  
io\_wq\_worker+0x27f/0x6c0  
ret\_from\_fork+0x2d/0x50  
ret\_from\_fork\_asm+0x1b/0x30

Freed by task 808589:

kasan\_save\_stack+0x1e/0x40  
kasan\_set\_track+0x21/0x30  
kasan\_save\_free\_info+0x27/0x40  
\_\_kasan\_slab\_free+0x126/0x1b0  
kmem\_cache\_free+0x10c/0x750  
bfq\_put\_queue+0x2dd/0x770  
\_\_bfq\_insert\_request.isra.0+0x155/0x7a0  
bfq\_insert\_request.isra.0+0x122/0x480  
bfq\_insert\_requests+0x156/0x180  
blk\_mq\_dispatch\_plug\_list+0x528/0x7e0

```
blk_mq_flush_plug_list.part.0+0xe5/0x590
__blk_flush_plug+0x3b/0x90
blk_finish_plug+0x40/0x60
do_writepages+0x19d/0x310
filemap_fdatawrite_wbc+0x95/0xc0
__filemap_fdatawrite_range+0x99/0xd0
filemap_write_and_wait_range.part.0+0x4d/0xa0
blkdev_read_iter+0xef/0x1e0
io_read+0x1b6/0x8a0
io_issue_sqe+0x87/0x300
io_wq_submit_work+0xeb/0x390
io_worker_handle_work+0x24d/0x550
io_wq_worker+0x27f/0x6c0
ret_from_fork+0x2d/0x50
ret_from_fork_asm+0x1b/0x30
```

Fix the problem by protecting bic\_to\_bfqq() with bfqd->lock.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53166>

**[CVE-2024-53168] kernel: sunrpc: fix one UAF issue caused by sunrpc kernel tcp socket (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

sunrpc: fix one UAF issue caused by sunrpc kernel tcp socket

BUG: KASAN: slab-use-after-free in tcp\_write\_timer\_handler+0x156/0x3e0  
Read of size 1 at addr ffff88811f322cd by task swapper/0/0

CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.0-rc4-dirty #7  
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1  
Call Trace:

```
<IRQ>
dump_stack_lvl+0x68/0xa0
print_address_description.constprop.0+0x2c/0x3d0
print_report+0xb4/0x270
kasan_report+0xbd/0xf0
tcp_write_timer_handler+0x156/0x3e0
tcp_write_timer+0x66/0x170
call_timer_fn+0xfb/0x1d0
__run_timers+0x3f8/0x480
run_timer_softirq+0x9b/0x100
handle_softirqs+0x153/0x390
__irq_exit_rcu+0x103/0x120
irq_exit_rcu+0xe/0x20
sysvec_apic_timer_interrupt+0x76/0x90
</IRQ>
<TASK>
```

```
asm_sysvec_apic_timer_interrupt+0x1a/0x20
RIP: 0010:default_idle+0xf/0x20
Code: 4c 01 c7 4c 29 c2 e9 72 ff ff 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 f3 0f 1e fa 66 90 0f 00 2d 33 f8 25 00 fb f4 <fa> c3 cc cc cc
cc 66 66 2e 0f 1f 84 00 00 00 00 00 90 90 90 90 90
RSP: 0018:fffffffa2007e28 EFLAGS: 00000242
RAX: 00000000000f3b31 RBX: 1fffffff4400fc7 RCX: ffffffffa09c3196
RDX: 0000000000000000 RSI: 0000000000000000 RDI: fffffff9f00590f
RBP: 0000000000000000 R08: 0000000000000001 R09: ffffed102360835d
R10: ffff88811b041aeb R11: 0000000000000001 R12: 0000000000000000
R13: ffffffffa202d7c0 R14: 0000000000000000 R15: 00000000000147d0
default_idle_call+0x6b/0xa0
cpuidle_idle_call+0x1af/0x1f0
do_idle+0xbc/0x130
cpu_startup_entry+0x33/0x40
rest_init+0x11f/0x210
start_kernel+0x39a/0x420
x86_64_start_reservations+0x18/0x30
x86_64_start_kernel+0x97/0xa0
common_startup_64+0x13e/0x141
</TASK>
```

Allocated by task 595:

```
kasan_save_stack+0x24/0x50
kasan_save_track+0x14/0x30
__kasan_slab_alloc+0x87/0x90
kmem_cache_alloc_noprof+0x12b/0x3f0
copy_net_ns+0x94/0x380
create_new_namespaces+0x24c/0x500
unshare_nsproxy_namespaces+0x75/0xf0
ksys_unshare+0x24e/0x4f0
__x64_sys_unshare+0x1f/0x30
do_syscall_64+0x70/0x180
entry_SYSCALL_64_after_hwframe+0x76/0x7e
```

Freed by task 100:

```
kasan_save_stack+0x24/0x50
kasan_save_track+0x14/0x30
kasan_save_free_info+0x3b/0x60
__kasan_slab_free+0x54/0x70
kmem_cache_free+0x156/0x5d0
cleanup_net+0x5d3/0x670
process_one_work+0x776/0xa90
worker_thread+0x2e2/0x560
kthread+0x1a8/0x1f0
ret_from_fork+0x34/0x60
ret_from_fork_asm+0x1a/0x30
```

Reproduction script:

```
mkdir -p /mnt/nfsshare
mkdir -p /mnt/nfs/netns_1
mkfs.ext4 /dev/sdb
```

```
mount /dev/sdb /mnt/nfsshare
systemctl restart nfs-server
chmod 777 /mnt/nfsshare
exportfs -i -o rw,no_root_squash */mnt/nfsshare
```

```
ip netns add netns_1
ip link add name veth_1_peer type veth peer veth_1
ifconfig veth_1_peer 11.11.0.254 up
ip link set veth_1 netns netns_1
ip netns exec netns_1 ifconfig veth_1 11.11.0.1
```

```
ip netns exec netns_1 /root/iptables -A OUTPUT -d 11.11.0.254 -p tcp \
--tcp-flags FIN FIN -j DROP
```

(note: In my environment, a DESTROY\_CLIENTID operation is always sent immediately, breaking the nfs tcp connection.)

```
ip netns exec netns_1 timeout -s 9 300 mount -t nfs -o proto=tcp,vers=4.1 \
11.11.0.254:/mnt/nfsshare /mnt/nfs/netns_1
```

```
ip netns del netns_1
```

The reason here is that the tcp socket in netns\_1 (nfs side) has been shutdown and closed (done in xs\_destroy), but the FIN message (with ack) is discarded, and the nfsd side keeps sending retransmission messages. As a result, when the tcp sock in netns\_1 processes the received message, it sends the message (FIN message) in the sending queue, and the tcp timer is re-established. When the network namespace is deleted, the net structure accessed by tcp's timer handler function causes problems.

To fix this problem, let's hold netns refcnt for the tcp kernel socket as done in other modules. This is an ugly hack which can easily be backported to earlier kernels. A proper fix which cleans up the interfaces will follow, but may not be so easy to backport.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53168>

### **[CVE-2024-53179] kernel: smb: client: fix use-after-free of signing key (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: fix use-after-free of signing key

Customers have reported use-after-free in @ses->auth\_key.response with SMB2.1 + sign mounts which occurs due to following race:

task A	task B
cifs_mount()	
dfs_mount_share()	
get_session()	
cifs_mount_get_session()	cifs_send_recv()



```
cifs_get_smb_ses()      compound_send_recv()
cifs_setup_session()    smb2_setup_request()
kfree_sensitive()       smb2_calc_signature()
                        crypto_shash_setkey() *UAF*
```

Fix this by ensuring that we have a valid @ses->auth\_key.response by checking whether @ses->ses\_status is SES\_GOOD or SES\_EXITING with @ses->ses\_lock held. After commit 24a9799aa8ef ("smb: client: fix UAF in smb2\_reconnect\_server()"), we made sure to call ->logoff() only when @ses was known to be good (e.g. valid ->auth\_key.response), so it's safe to access signing key when @ses->ses\_status == SES\_EXITING.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53179>

### **[CVE-2024-53203] kernel: usb: typec: fix potential array underflow in ucsi\_ccg\_sync\_control() (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

usb: typec: fix potential array underflow in ucsi\_ccg\_sync\_control()

The "command" variable can be controlled by the user via debugfs. The worry is that if con\_index is zero then "&uc->ucsi->connector[con\_index - 1]" would be an array underflow.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53203>

### **[CVE-2024-53216] kernel: nfsd: release svc\_expkey/svc\_export with rcu\_work (Severity: HIGH)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nfsd: release svc\_expkey/svc\_export with rcu\_work

The last reference for `cache\_head` can be reduced to zero in `c\_show` and `e\_show` (using `rcu\_read\_lock` and `rcu\_read\_unlock`). Consequently, `svc\_export\_put` and `expkey\_put` will be invoked, leading to two issues:

1. The `svc\_export\_put` will directly free ex\_uuid. However, `e\_show`/`c\_show` will access `ex\_uuid` after `cache\_put`, which can trigger a use-after-free issue, shown below.

=====

BUG: KASAN: slab-use-after-free in svc\_export\_show+0x362/0x430 [nfsd]  
Read of size 1 at addr ff11000010fdc120 by task cat/870

CPU: 1 UID: 0 PID: 870 Comm: cat Not tainted 6.12.0-rc3+ #1  
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS  
1.16.1-2.fc37 04/01/2014

Call Trace:

<TASK>

dump\_stack\_lvl+0x53/0x70  
print\_address\_description.constprop.0+0x2c/0x3a0  
print\_report+0xb9/0x280  
kasan\_report+0xae/0xe0  
svc\_export\_show+0x362/0x430 [nfsd]  
c\_show+0x161/0x390 [sunrpc]  
seq\_read\_iter+0x589/0x770  
seq\_read+0x1e5/0x270  
proc\_reg\_read+0xe1/0x140  
vfs\_read+0x125/0x530  
ksys\_read+0xc1/0x160  
do\_syscall\_64+0x5f/0x170  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Allocated by task 830:

kasan\_save\_stack+0x20/0x40  
kasan\_save\_track+0x14/0x30  
\_\_kasan\_kmalloc+0x8f/0xa0  
\_\_kmalloc\_node\_track\_caller\_noprof+0x1bc/0x400  
kmemdup\_noprof+0x22/0x50  
svc\_export\_parse+0x8a9/0xb80 [nfsd]  
cache\_do\_downcall+0x71/0xa0 [sunrpc]  
cache\_write\_procfs+0x8e/0xd0 [sunrpc]  
proc\_reg\_write+0xe1/0x140  
vfs\_write+0x1a5/0x6d0  
ksys\_write+0xc1/0x160  
do\_syscall\_64+0x5f/0x170  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Freed by task 868:

kasan\_save\_stack+0x20/0x40  
kasan\_save\_track+0x14/0x30  
kasan\_save\_free\_info+0x3b/0x60  
\_\_kasan\_slab\_free+0x37/0x50  
kfree+0xf3/0x3e0  
svc\_export\_put+0x87/0xb0 [nfsd]  
cache\_purge+0x17f/0x1f0 [sunrpc]  
nfsd\_destroy\_serv+0x226/0x2d0 [nfsd]  
nfsd\_svc+0x125/0x1e0 [nfsd]  
write\_threads+0x16a/0x2a0 [nfsd]  
nfsctl\_transaction\_write+0x74/0xa0 [nfsd]  
vfs\_write+0x1a5/0x6d0  
ksys\_write+0xc1/0x160  
do\_syscall\_64+0x5f/0x170  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

2. We cannot sleep while using `rcu\_read\_lock`/`rcu\_read\_unlock`.  
However, `svc\_export\_put`/`expkey\_put` will call path\_put, which

subsequently triggers a sleeping operation due to the following  
`dput`.

```
=====
WARNING: suspicious RCU usage
5.10.0-dirty #141 Not tainted
-----
...
Call Trace:
dump_stack+0x9a/0xd0
__might_sleep+0x231/0x240
dput+0x39/0x600
path_put+0x1b/0x30
svc_export_put+0x17/0x80
e_show+0x1c9/0x200
seq_read_iter+0x63f/0x7c0
seq_read+0x226/0x2d0
vfs_read+0x113/0x2c0
ksys_read+0xc9/0x170
do_syscall_64+0x33/0x40
entry_SYSCALL_64_after_hwframe+0x67/0xd1
```

Fix these issues by using `rcu\_work` to help release  
`svc\_expkey`/`svc\_export`. This approach allows for an asynchronous  
context to invoke `path\_put` and also facilitates the freeing of  
`uuid/exp/key` after an RCU grace period.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53216>

### **[CVE-2024-56538] kernel: drm: zynqmp\_kms: Unplug DRM device before removal (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm: zynqmp\_kms: Unplug DRM device before removal

Prevent userspace accesses to the DRM device from causing  
use-after-frees by unplugging the device before we remove it. This  
causes any further userspace accesses to result in an error without  
further calls into this driver's internals.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56538>

### **[CVE-2024-56775] kernel: drm/amd/display: Fix handling of plane refcount (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix handling of plane refcount

[Why]

The mechanism to backup and restore plane states doesn't maintain refcount, which can cause issues if the refcount of the plane changes in between backup and restore operations, such as memory leaks if the refcount was supposed to go down, or double frees / invalid memory accesses if the refcount was supposed to go up.

[How]

Cache and re-apply current refcount when restoring plane states.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56775>

**[CVE-2024-56784] kernel: drm/amd/display: Adding array index check to prevent memory corruption (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Adding array index check to prevent memory corruption

[Why & How]

Array indices out of bound caused memory corruption. Adding checks to ensure that array index stays in bound.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56784>

**[CVE-2024-57900] kernel: ila: serialize calls to nf\_register\_net\_hooks() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ila: serialize calls to nf\_register\_net\_hooks()

syzbot found a race in ila\_add\_mapping() [1]

commit 031ae72825ce ("ila: call nf\_unregister\_net\_hooks() sooner")  
attempted to fix a similar issue.

Looking at the syzbot repro, we have concurrent ILA\_CMD\_ADD commands.

Add a mutex to make sure at most one thread is calling nf\_register\_net\_hooks().

[1]

BUG: KASAN: slab-use-after-free in rht\_key\_hashfn include/linux/rhashtable.h:159 [inline]

BUG: KASAN: slab-use-after-free in \_\_rhashtable\_lookup.constprop.0+0x426/0x550 include/linux/rhashtable.h:604

Read of size 4 at addr ffff888028f40008 by task dhcpcd/5501

CPU: 1 UID: 0 PID: 5501 Comm: dhcpcd Not tainted 6.13.0-rc4-syzkaller-00054-gd6ef8b40d075 #0  
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024  
Call Trace:

<IRQ>

```
__dump_stack lib/dump_stack.c:94 [inline]
dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120
print_address_description mm/kasan/report.c:378 [inline]
print_report+0xc3/0x620 mm/kasan/report.c:489
kasan_report+0xd9/0x110 mm/kasan/report.c:602
rht_key_hashfn include/linux/rhashtable.h:159 [inline]
__rhashtable_lookup.constprop.0+0x426/0x550 include/linux/rhashtable.h:604
rhashtable_lookup include/linux/rhashtable.h:646 [inline]
rhashtable_lookup_fast include/linux/rhashtable.h:672 [inline]
ila_lookup_wildcards net/ipv6/ila/ila_xlat.c:127 [inline]
ila_xlat_addr net/ipv6/ila/ila_xlat.c:652 [inline]
ila_nf_input+0x1ee/0x620 net/ipv6/ila/ila_xlat.c:185
nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline]
nf_hook_slow+0xbb/0x200 net/netfilter/core.c:626
nf_hook.constprop.0+0x42e/0x750 include/linux/netfilter.h:269
NF_HOOK include/linux/netfilter.h:312 [inline]
ipv6_rcv+0xa4/0x680 net/ipv6/ip6_input.c:309
__netif_receive_skb_one_core+0x12e/0x1e0 net/core/dev.c:5672
__netif_receive_skb+0x1d/0x160 net/core/dev.c:5785
process_backlog+0x443/0x15f0 net/core/dev.c:6117
__napi_poll.constprop.0+0xb7/0x550 net/core/dev.c:6883
napi_poll net/core/dev.c:6952 [inline]
net_rx_action+0xa94/0x1010 net/core/dev.c:7074
handle_softirqs+0x213/0x8f0 kernel/softirq.c:561
__do_softirq kernel/softirq.c:595 [inline]
invoke_softirq kernel/softirq.c:435 [inline]
__irq_exit_rcu+0x109/0x170 kernel/softirq.c:662
irq_exit_rcu+0x9/0x30 kernel/softirq.c:678
instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline]
sysvec_apic_timer_interrupt+0xa4/0xc0 arch/x86/kernel/apic/apic.c:1049
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-57900>

### **[CVE-2024-57982] kernel: xfrm: state: fix out-of-bounds read during lookup (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

xfrm: state: fix out-of-bounds read during lookup

lookup and resize can run in parallel.

The xfrm\_state\_hash\_generation seqlock ensures a retry, but the hash functions can observe a hmask value that is too large for the new hlist array.

rehash does:

```
rcu_assign_pointer(net->xfrm.state_bydst, ndst) [..]  
net->xfrm.state_hmask = nhashmask;
```

While state lookup does:

```
h = xfrm_dst_hash(net, daddr, saddr, tmpl->reqid, encap_family);  
hlist_for_each_entry_rcu(x, net->xfrm.state_bydst + h, bydst) {
```

This is only safe in case the update to state\_bydst is larger than net->xfrm.xfrm\_state\_hmask (or if the lookup function gets serialized via state spinlock again).

Fix this by prefetching state\_hmask and the associated pointers.

The xfrm\_state\_hash\_generation seqlock retry will ensure that the pointer and the hmask will be consistent.

The existing helpers, like xfrm\_dst\_hash(), are now unsafe for RCU side, add lockdep assertions to document that they are only safe for insert side.

xfrm\_state\_lookup\_byaddr() uses the spinlock rather than RCU.

AFAICS this is an oversight from back when state lookup was converted to RCU, this lock should be replaced with RCU in a future patch.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57982>

## **[CVE-2024-57984] kernel: i3c: dw: Fix use-after-free in dw\_i3c\_master driver due to race condition (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

i3c: dw: Fix use-after-free in dw\_i3c\_master driver due to race condition

In dw\_i3c\_common\_probe, &master->hj\_work is bound with dw\_i3c\_hj\_work. And dw\_i3c\_master\_irq\_handler can call dw\_i3c\_master\_irq\_handle\_ibis function to start the work.

If we remove the module which will call dw\_i3c\_common\_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
	dw_i3c_hj_work
dw_i3c_common_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 `dw_i3c_common_remove`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57984>

### **[CVE-2024-58002] kernel: media: uvcvideo: Remove dangling pointers (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: uvcvideo: Remove dangling pointers

When an async control is written, we copy a pointer to the file handle that started the operation. That pointer will be used when the device is done. Which could be anytime in the future.

If the user closes that file descriptor, its structure will be freed, and there will be one dangling pointer per pending async control, that the driver will try to use.

Clean all the dangling pointers during `release()`.

To avoid adding a performance penalty in the most common case (no async operation), a counter has been introduced with some logic to make sure that it is properly handled.

More Info: <https://avd.aquasec.com/nvd/cve-2024-58002>

### **[CVE-2025-21702] kernel: pfifo\_tail\_enqueue: Drop new packet when `sch->limit == 0` (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

pfifo\_tail\_enqueue: Drop new packet when `sch->limit == 0`

Expected behaviour:

In case we reach scheduler's limit, `pfifo_tail_enqueue()` will drop a packet in scheduler's queue and decrease scheduler's `qlen` by one. Then, `pfifo_tail_enqueue()` enqueue new packet and increase scheduler's `qlen` by one. Finally, `pfifo_tail_enqueue()` return ``NET_XMIT_CN`` status code.

Weird behaviour:

In case we set ``sch->limit == 0`` and trigger `pfifo_tail_enqueue()` on a

scheduler that has no packet, the 'drop a packet' step will do nothing.  
This means the scheduler's qlen still has value equal 0.  
Then, we continue to enqueue new packet and increase scheduler's qlen by one. In summary, we can leverage pfifo\_tail\_enqueue() to increase qlen by one and return `NET\_XMIT\_CN` status code.

The problem is:

Let's say we have two qdiscs: Qdisc\_A and Qdisc\_B.

- Qdisc\_A's type must have '->graft()' function to create parent/child relationship.  
Let's say Qdisc\_A's type is `hfsc`. Enqueue packet to this qdisc will trigger `hfsc\_enqueue`.
- Qdisc\_B's type is pfifo\_head\_drop. Enqueue packet to this qdisc will trigger `pfifo\_tail\_enqueue`.
- Qdisc\_B is configured to have `sch->limit == 0`.
- Qdisc\_A is configured to route the enqueued's packet to Qdisc\_B.

Enqueue packet through Qdisc\_A will lead to:

- hfsc\_enqueue(Qdisc\_A) -> pfifo\_tail\_enqueue(Qdisc\_B)
- Qdisc\_B->q.qlen += 1
- pfifo\_tail\_enqueue() return `NET\_XMIT\_CN`
- hfsc\_enqueue() check for `NET\_XMIT\_SUCCESS` and see `NET\_XMIT\_CN` => hfsc\_enqueue() don't increase qlen of Qdisc\_A.

The whole process lead to a situation where Qdisc\_A->q.qlen == 0 and Qdisc\_B->q.qlen == 1.

Replace 'hfsc' with other type (for example: 'drr') still lead to the same problem.

This violate the design where parent's qlen should equal to the sum of its childrens'qlen.

Bug impact: This issue can be used for user->kernel privilege escalation when it is reachable.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21702>

### **[CVE-2025-21855] kernel: ibmvnic: Don't reference skb after sending to VIOS (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ibmvnic: Don't reference skb after sending to VIOS

Previously, after successfully flushing the xmit buffer to VIOS,  
the tx\_bytes stat was incremented by the length of the skb.

It is invalid to access the skb memory after sending the buffer to the VIOS because, at any point after sending, the VIOS can trigger an interrupt to free this memory. A race between reading skb->len and freeing the skb is possible (especially during LPM) and will result in use-after-free:

=====

BUG: KASAN: slab-use-after-free in ibmvnic\_xmit+0x75c/0x1808 [ibmvnic]

Read of size 4 at addr c00000024eb48a70 by task hxecom/14495

<...>

Call Trace:

[c000000118f66cf0] [c0000000018cba6c] dump\_stack\_lvl+0x84/0xe8 (unreliable)

[c000000118f66d20] [c0000000006f0080] print\_report+0x1a8/0x7f0



```
[c000000118f66df0] [c0000000006f08f0] kasan_report+0x128/0x1f8
[c000000118f66f00] [c0000000006f2868] __asan_load4+0xac/0xe0
[c000000118f66f20] [c0080000046eac84] ibmvnic_xmit+0x75c/0x1808 [ibmvnic]
[c000000118f67340] [c0000000014be168] dev_hard_start_xmit+0x150/0x358
<...>
```

Freed by task 0:

```
kasan_save_stack+0x34/0x68
kasan_save_track+0x2c/0x50
kasan_save_free_info+0x64/0x108
__kasan_mempool_poison_object+0x148/0x2d4
napi_skb_cache_put+0x5c/0x194
net_tx_action+0x154/0x5b8
handle_softirqs+0x20c/0x60c
do_softirq_own_stack+0x6c/0x88
<...>
```

The buggy address belongs to the object at c00000024eb48a00 which belongs to the cache skbuff\_head\_cache of size 224

=====

More Info: <https://avd.aquasec.com/nvd/cve-2025-21855>

### **[CVE-2025-21858] kernel: geneve: Fix use-after-free in geneve\_find\_dev(). (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

geneve: Fix use-after-free in geneve\_find\_dev().

syzkaller reported a use-after-free in geneve\_find\_dev() [0] without repro.

geneve\_configure() links struct geneve\_dev.next to net\_generic(net, geneve\_net\_id)->geneve\_list.

The net here could differ from dev\_net(dev) if IFLA\_NET\_NS\_PID, IFLA\_NET\_NS\_FD, or IFLA\_TARGET\_NETNSID is set.

When dev\_net(dev) is dismantled, geneve\_exit\_batch\_rtnl() finally calls unregister\_netdevice\_queue() for each dev in the netns, and later the dev is freed.

However, its geneve\_dev.next is still linked to the backend UDP socket netns.

Then, use-after-free will occur when another geneve dev is created in the netns.

Let's call geneve\_dellink() instead in geneve\_destroy\_tunnels().

[0]:

BUG: KASAN: slab-use-after-free in geneve\_find\_dev drivers/net/geneve.c:1295 [inline]

BUG: KASAN: slab-use-after-free in geneve\_configure+0x234/0x858 drivers/net/geneve.c:1343  
Read of size 2 at addr ffff000054d6ee24 by task syz.1.4029/13441

CPU: 1 UID: 0 PID: 13441 Comm: syz.1.4029 Not tainted 6.13.0-g0ad9617c78ac #24  
dc35ca22c79fb82e8e7bc5c9c9adafea898b1e3d

Hardware name: linux,dummy-virt (DT)

Call trace:

```
show_stack+0x38/0x50 arch/arm64/kernel/stacktrace.c:466 (C)
__dump_stack lib/dump_stack.c:94 [inline]
dump_stack_lvl+0xbc/0x108 lib/dump_stack.c:120
print_address_description mm/kasan/report.c:378 [inline]
print_report+0x16c/0x6f0 mm/kasan/report.c:489
kasan_report+0xc0/0x120 mm/kasan/report.c:602
__asan_report_load2_noabort+0x20/0x30 mm/kasan/report_generic.c:379
geneve_find_dev drivers/net/geneve.c:1295 [inline]
geneve_configure+0x234/0x858 drivers/net/geneve.c:1343
geneve_newlink+0xb8/0x128 drivers/net/geneve.c:1634
rtnl_newlink_create+0x23c/0x868 net/core/rtnetlink.c:3795
__rtnl_newlink net/core/rtnetlink.c:3906 [inline]
rtnl_newlink+0x1054/0x1630 net/core/rtnetlink.c:4021
rtnetlink_rcv_msg+0x61c/0x918 net/core/rtnetlink.c:6911
netlink_rcv_skb+0x1dc/0x398 net/netlink/af_netlink.c:2543
rtnetlink_rcv+0x34/0x50 net/core/rtnetlink.c:6938
netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline]
netlink_unicast+0x618/0x838 net/netlink/af_netlink.c:1348
netlink_sendmsg+0x5fc/0x8b0 net/netlink/af_netlink.c:1892
sock_sendmsg_nosec net/socket.c:713 [inline]
__sock_sendmsg net/socket.c:728 [inline]
__sys_sendmsg+0x410/0x6f8 net/socket.c:2568
__sys_sendmsg+0x178/0x1d8 net/socket.c:2622
__sys_sendmsg net/socket.c:2654 [inline]
__do_sys_sendmsg net/socket.c:2659 [inline]
__se_sys_sendmsg net/socket.c:2657 [inline]
__arm64_sys_sendmsg+0x12c/0x1c8 net/socket.c:2657
__invoke_syscall arch/arm64/kernel/syscall.c:35 [inline]
invoke_syscall+0x90/0x278 arch/arm64/kernel/syscall.c:49
el0_svc_common+0x13c/0x250 arch/arm64/kernel/syscall.c:132
do_el0_svc+0x54/0x70 arch/arm64/kernel/syscall.c:151
el0_svc+0x4c/0xa8 arch/arm64/kernel/entry-common.c:744
el0t_64_sync_handler+0x78/0x108 arch/arm64/kernel/entry-common.c:762
el0t_64_sync+0x198/0x1a0 arch/arm64/kernel/entry.S:600
```

Allocated by task 13247:

```
kasan_save_stack mm/kasan/common.c:47 [inline]
kasan_save_track+0x30/0x68 mm/kasan/common.c:68
kasan_save_alloc_info+0x44/0x58 mm/kasan/generic.c:568
poison_kmalloc_redzone mm/kasan/common.c:377 [inline]
__kasan_kmalloc+0x84/0xa0 mm/kasan/common.c:394
kasan_kmalloc include/linux/kasan.h:260 [inline]
__do_kmalloc_node mm/slub.c:4298 [inline]
__kmalloc_node_noprof+0x2a0/0x560 mm/slub.c:4304
__kvmalloc_node_noprof+0x9c/0x230 mm/util.c:645
alloc_netdev_mqs+0xb8/0x11a0 net/core/dev.c:11470
```

```
rtnl_create_link+0x2b8/0xb50 net/core/rtnetlink.c:3604
rtnl_newlink_create+0x19c/0x868 net/core/rtnetlink.c:3780
__rtnl_newlink net/core/rtnetlink.c:3906 [inline]
rtnl_newlink+0x1054/0x1630 net/core/rtnetlink.c:4021
rtnetlink_rcv_msg+0x61c/0x918 net/core/rtnetlink.c:6911
netlink_rcv_skb+0x1dc/0x398 net/netlink/af_netlink.c:2543
rtnetlink_rcv+0x34/0x50 net/core/rtnetlink.c:6938
netlink_unicast_kernel net/netlink/af_n
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21858>

### **[CVE-2025-21863] kernel: io\_uring: prevent opcode speculation (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

io\_uring: prevent opcode speculation

sqe->opcode is used for different tables, make sure we sanitise it against speculations.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21863>

### **[CVE-2025-21919] kernel: sched/fair: Fix potential memory corruption in child\_cfs\_rq\_on\_list (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

sched/fair: Fix potential memory corruption in child\_cfs\_rq\_on\_list

child\_cfs\_rq\_on\_list attempts to convert a 'prev' pointer to a cfs\_rq. This 'prev' pointer can originate from struct rq's leaf\_cfs\_rq\_list, making the conversion invalid and potentially leading to memory corruption. Depending on the relative positions of leaf\_cfs\_rq\_list and the task group (tg) pointer within the struct, this can cause a memory fault or access garbage data.

The issue arises in list\_add\_leaf\_cfs\_rq, where both cfs\_rq->leaf\_cfs\_rq\_list and rq->leaf\_cfs\_rq\_list are added to the same leaf list. Also, rq->tmp\_alone\_branch can be set to rq->leaf\_cfs\_rq\_list.

This adds a check `if (prev == &rq->leaf\_cfs\_rq\_list)` after the main conditional in child\_cfs\_rq\_on\_list. This ensures that the container\_of operation will convert a correct cfs\_rq struct.

This check is sufficient because only cfs\_rqs on the same CPU are added

to the list, so verifying the 'prev' pointer against the current rq's list head is enough.

Fixes a potential memory corruption issue that due to current struct layout might not be manifesting as a crash but could lead to unpredictable behavior when the layout changes.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21919>

### **[CVE-2025-21920] kernel: vlan: enforce underlying device type (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

vlan: enforce underlying device type

Currently, VLAN devices can be created on top of non-ethernet devices.

Besides the fact that it doesn't make much sense, this also causes a bug which leaks the address of a kernel function to usermode.

When creating a VLAN device, we initialize GARP (`garp_init_applicant`) and MRP (`mrp_init_applicant`) for the underlying device.

As part of the initialization process, we add the multicast address of each applicant to the underlying device, by calling `dev_mc_add`.

`__dev_mc_add` uses `dev->addr_len` to determine the length of the new multicast address.

This causes an out-of-bounds read if `dev->addr_len` is greater than 6, since the multicast addresses provided by GARP and MRP are only 6 bytes long.

This behaviour can be reproduced using the following commands:

```
ip tunnel add gretest mode ip6gre local ::1 remote ::2 dev lo
ip l set up dev gretest
ip link add link gretest name vlantest type vlan id 100
```

Then, the following command will display the address of `garp_pdu_rcv`:

```
ip maddr show | grep 01:80:c2:00:00:21
```

Fix the bug by enforcing the type of the underlying device during VLAN device initialization.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21920>

### **[CVE-2025-21926] kernel: net: gso: fix ownership in `__udp_gso_segment` (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net: gso: fix ownership in \_\_udp\_gso\_segment

In \_\_udp\_gso\_segment the skb destructor is removed before segmenting the skb but the socket reference is kept as-is. This is an issue if the original skb is later orphaned as we can hit the following bug:

kernel BUG at ./include/linux/skbuff.h:3312! (skb\_orphan)

RIP: 0010:ip\_rcv\_core+0x8b2/0xca0

Call Trace:

ip\_rcv+0xab/0x6e0

\_\_netif\_receive\_skb\_one\_core+0x168/0x1b0

process\_backlog+0x384/0x1100

\_\_napi\_poll.constprop.0+0xa1/0x370

net\_rx\_action+0x925/0xe50

The above can happen following a sequence of events when using OpenVSwitch, when an OVS\_ACTION\_ATTR\_USERSPACE action precedes an OVS\_ACTION\_ATTR\_OUTPUT action:

1. OVS\_ACTION\_ATTR\_USERSPACE is handled (in do\_execute\_actions): the skb goes through queue\_gso\_packets and then \_\_udp\_gso\_segment, where its destructor is removed.
2. The segments' data are copied and sent to userspace.
3. OVS\_ACTION\_ATTR\_OUTPUT is handled (in do\_execute\_actions) and the same original skb is sent to its path.
4. If it later hits skb\_orphan, we hit the bug.

Fix this by also removing the reference to the socket in \_\_udp\_gso\_segment.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21926>

### **[CVE-2025-21927] kernel: nvme-tcp: fix potential memory corruption in nvme\_tcp\_recv\_pdu() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nvme-tcp: fix potential memory corruption in nvme\_tcp\_recv\_pdu()

nvme\_tcp\_recv\_pdu() doesn't check the validity of the header length.

When header digests are enabled, a target might send a packet with an invalid header length (e.g. 255), causing nvme\_tcp\_verify\_hdgst()

to access memory outside the allocated area and cause memory corruptions

by overwriting it with the calculated digest.

Fix this by rejecting packets with an unexpected header length.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21927>

**[CVE-2025-21928] kernel: HID: intel-ish-hid: Fix use-after-free issue in ishtp\_hid\_remove() (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

HID: intel-ish-hid: Fix use-after-free issue in ishtp\_hid\_remove()

The system can experience a random crash a few minutes after the driver is removed. This issue occurs due to improper handling of memory freeing in the ishtp\_hid\_remove() function.

The function currently frees the ``driver_data`` directly within the loop that destroys the HID devices, which can lead to accessing freed memory. Specifically, ``hid_destroy_device()`` uses ``driver_data`` when it calls ``hid_ishtp_set_feature()`` to power off the sensor, so freeing ``driver_data`` beforehand can result in accessing invalid memory.

This patch resolves the issue by storing the ``driver_data`` in a temporary variable before calling ``hid_destroy_device()``, and then freeing the ``driver_data`` after the device is destroyed.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21928>

**[CVE-2025-21934] kernel: rapidio: fix an API misuse when rio\_add\_net() fails (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

rapidio: fix an API misuse when rio\_add\_net() fails

rio\_add\_net() calls device\_register() and fails when device\_register() fails. Thus, put\_device() should be used rather than kfree(). Add "mport->net = NULL;" to avoid a use after free issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21934>

**[CVE-2025-21945] kernel: ksmbd: fix use-after-free in smb2\_lock (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix use-after-free in smb2\_lock

If smb\_lock->zero\_len has value, ->list of smb\_lock is not delete and flock is old one. It will cause use-after-free on error handling routine.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21945>

**[CVE-2025-21979] kernel: wifi: cfg80211: cancel wiphy\_work before freeing wiphy (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

wifi: cfg80211: cancel wiphy\_work before freeing wiphy

A wiphy\_work can be queued from the moment the wiphy is allocated and initialized (i.e. wiphy\_new\_nm). When a wiphy\_work is queued, the rdev::wiphy\_work is getting queued.

If wiphy\_free is called before the rdev::wiphy\_work had a chance to run, the wiphy memory will be freed, and then when it eventually gets to run it'll use invalid memory.

Fix this by canceling the work before freeing the wiphy.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21979>

**[CVE-2025-21991] kernel: x86/microcode/AMD: Fix out-of-bounds on systems with CPU-less NUMA nodes (Severity: HIGH)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

x86/microcode/AMD: Fix out-of-bounds on systems with CPU-less NUMA nodes

Currently, load\_microcode\_amd() iterates over all NUMA nodes, retrieves their CPU masks and unconditionally accesses per-CPU data for the first CPU of each mask.

According to Documentation/admin-guide/mm/numaperf.rst:

"Some memory may share the same node as a CPU, and others are provided as memory only nodes."

Therefore, some node CPU masks may be empty and wouldn't have a "first CPU".

On a machine with far memory (and therefore CPU-less NUMA nodes):

- `cpumask_of_node(nid)` is 0
- `cpumask_first(0)` is `CONFIG_NR_CPUS`
- `cpu_data(CONFIG_NR_CPUS)` accesses the `cpu_info` per-CPU array at an index that is 1 out of bounds

This does not have any security implications since flashing microcode is a privileged operation but I believe this has reliability implications by potentially corrupting memory while flashing a microcode update.

When booting with `CONFIG_UBSAN_BOUNDS=y` on an AMD machine that flashes a microcode update. I get the following splat:

```
UBSAN: array-index-out-of-bounds in arch/x86/kernel/cpu/microcode/amd.c:X:Y
index 512 is out of range for type 'unsigned long[512]'
```

```
[...]
```

Call Trace:

```
dump_stack
__ubsan_handle_out_of_bounds
load_microcode_amd
request_microcode_amd
reload_store
kernfs_fop_write_iter
vfs_write
ksys_write
do_syscall_64
entry_SYSCALL_64_after_hwframe
```

Change the loop to go over only NUMA nodes which have CPUs before determining whether the first CPU on the respective node needs microcode update.

[ bp: Message commit message, fix typo. ]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21991>

### **[CVE-2025-21993] kernel: iscsi\_ibft: Fix UBSAN shift-out-of-bounds warning in `ibft_attr_show_nic()` (Severity: HIGH)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

`iscsi_ibft`: Fix UBSAN shift-out-of-bounds warning in `ibft_attr_show_nic()`

When performing an iSCSI boot using IPv6, `iscsistart` still reads the `/sys/firmware/ibft/ethernetX/subnet-mask` entry. Since the IPv6 prefix length is 64, this causes the shift exponent to become negative, triggering a UBSAN warning. As the concept of a subnet mask does not apply to IPv6, the value is set to `~0` to suppress the warning message.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21993>



## [CVE-2025-21999] kernel: proc: fix UAF in proc\_get\_inode() (Severity: HIGH)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

proc: fix UAF in proc\_get\_inode()

Fix race between rmmmod and /proc/XXX's inode instantiation.

The bug is that pde->proc\_ops don't belong to /proc, it belongs to a module, therefore dereferencing it after /proc entry has been registered is a bug unless use\_pde/unuse\_pde() pair has been used.

use\_pde/unuse\_pde can be avoided (2 atomic ops!) because pde->proc\_ops never changes so information necessary for inode instantiation can be saved \_before\_ proc\_register() in PDE itself and used later, avoiding pde->proc\_ops->... dereference.

```
rmmod                lookup
sys_delete_module
    proc_lookup_de
    pde_get(de);
    proc_get_inode(dir->i_sb, de);
mod->exit()
proc_remove
    remove_proc_subtree
    proc_entry_rundown(de);
free_module(mod);

    if (S_ISREG(inode->i_mode))
    if (de->proc_ops->proc_read_iter)
--> As module is already freed, will trigger UAF
```

BUG: unable to handle page fault for address: fffffbfff80a702b

PGD 817fc4067 P4D 817fc4067 PUD 817fc0067 PMD 102ef4067 PTE 0

Oops: Oops: 0000 [#1] PREEMPT SMP KASAN PTI

CPU: 26 UID: 0 PID: 2667 Comm: ls Tainted: G

Hardware name: QEMU Standard PC (i440FX + PIIX, 1996)

RIP: 0010:proc\_get\_inode+0x302/0x6e0

RSP: 0018:ffff88811c837998 EFLAGS: 00010a06

RAX: dffffc0000000000 RBX: ffffffff0538140 RCX: 0000000000000007

RDX: 1ffffbfff80a702b RSI: 0000000000000001 RDI: ffffffff0538158

RBP: ffff8881299a6000 R08: 0000000067bbe1e5 R09: 1ffff11023906f20

R10: ffffffff560ca07 R11: ffffffff2b43a58 R12: ffff888105bb78f0

R13: ffff888100518048 R14: ffff8881299a6004 R15: 0000000000000001

FS: 00007f95b9686840(0000) GS:ffff8883af100000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: fffffbfff80a702b CR3: 0000000117dd2000 CR4: 000000000000006f0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

Call Trace:

<TASK>

proc\_lookup\_de+0x11f/0x2e0

\_\_lookup\_slow+0x188/0x350

walk\_component+0x2ab/0x4f0

path\_lookupat+0x120/0x660

filename\_lookup+0x1ce/0x560

vfs\_statx+0xac/0x150

\_\_do\_sys\_newstat+0x96/0x110

do\_syscall\_64+0x5f/0x170

entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

[adobriyan@gmail.com: don't do 2 atomic ops on the common path]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21999>

**[CVE-2019-15213] kernel: use-after-free caused by malicious USB device in drivers/media/usb/dvb-usb/dvb-usb-init.c (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in the Linux kernel before 5.2.3. There is a use-after-free caused by a malicious USB device in the drivers/media/usb/dvb-usb/dvb-usb-init.c driver.

More Info: <https://avd.aquasec.com/nvd/cve-2019-15213>

**[CVE-2019-16089] kernel: Improper return check in nbd\_genl\_status function in drivers/block/nbd.c (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in the Linux kernel through 5.2.13. nbd\_genl\_status in drivers/block/nbd.c does not check the nla\_nest\_start\_noflag return value.

More Info: <https://avd.aquasec.com/nvd/cve-2019-16089>

**[CVE-2019-20794] kernel: task processes not being properly ended could lead to resource exhaustion (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in the Linux kernel 4.18 through 5.6.11 when unprivileged user namespaces are allowed. A user can create their own PID namespace, and mount a FUSE filesystem. Upon interaction with this FUSE filesystem, if the userspace component is terminated via a kill of the PID namespace's pid 1, it will result in a hung task, and resources being permanently locked up until system reboot. This can result in resource exhaustion.

More Info: <https://avd.aquasec.com/nvd/cve-2019-20794>

**[CVE-2020-14304] kernel: ethtool when reading eeprom of device could lead to memory leak**

**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A memory disclosure flaw was found in the Linux kernel's ethernet drivers, in the way it read data from the EEPROM of the device. This flaw allows a local user to read uninitialized values from the kernel memory. The highest threat from this vulnerability is to confidentiality.

More Info: <https://avd.aquasec.com/nvd/cve-2020-14304>

**[CVE-2020-36694] kernel: netfilter: use-after-free in the packet processing context (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in netfilter in the Linux kernel before 5.10. There can be a use-after-free in the packet processing context, because the per-CPU sequence count is mishandled during concurrent iptables rules replacement. This could be exploited with the CAP\_NET\_ADMIN capability in an unprivileged namespace. NOTE: cc00bca was reverted in 5.12.

More Info: <https://avd.aquasec.com/nvd/cve-2020-36694>

**[CVE-2021-47658] kernel: drm/amd/pm: fix a potential gpu\_metrics\_table memory leak (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/pm: fix a potential gpu\_metrics\_table memory leak

Memory is allocated for gpu\_metrics\_table in renoir\_init\_smc\_tables(), but not freed in int smu\_v12\_0\_fini\_smc\_tables(). Free it!

More Info: <https://avd.aquasec.com/nvd/cve-2021-47658>

**[CVE-2023-0597] kernel: x86/mm: Randomize per-cpu entry area (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A flaw possibility of memory leak in the Linux kernel cpu\_entry\_area mapping of X86 CPU data to memory was found in the way user can guess location of exception stack(s) or other important data. A local user could use this flaw to get access to some important data with expected location in memory.

More Info: <https://avd.aquasec.com/nvd/cve-2023-0597>

**[CVE-2023-21264] In multiple functions of mem\_protect.c, there is a possible way to acc ... (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In multiple functions of mem\_protect.c, there is a possible way to access hypervisor memory due to a memory access check in the wrong place. This could lead to local escalation of privilege with System execution privileges needed. User interaction is not needed for exploitation.

More Info: <https://avd.aquasec.com/nvd/cve-2023-21264>

**[CVE-2023-23005] kernel: incorrect check for error case in the memory\_tier\_init (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel before 6.2, mm/memory-tiers.c misinterprets the alloc\_memory\_type return value (expects it to be NULL in the error case, whereas it is actually an error pointer). NOTE: this is disputed by third parties because there are no realistic cases in which a user can cause the alloc\_memory\_type error case to be reached.

More Info: <https://avd.aquasec.com/nvd/cve-2023-23005>

**[CVE-2023-31082] kernel: sleeping function called from an invalid context in gsmlid\_write (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in drivers/tty/n\_gsm.c in the Linux kernel 6.2. There is a sleeping function called from an invalid context in gsmlid\_write, which will block the kernel. Note: This has been disputed by 3rd parties as not a valid vulnerability.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31082>

**[CVE-2023-3397] kernel: slab-use-after-free Write in txEnd due to race condition (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

A race condition occurred between the functions lmLogClose and txEnd in JFS, in the Linux Kernel, executed in different threads. This flaw allows a local attacker with normal user privileges to crash the system or leak internal kernel information.

More Info: <https://avd.aquasec.com/nvd/cve-2023-3397>

**[CVE-2023-37454] kernel: udf: use-after-free write in udf\_close\_lvid (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.4.2. A crafted UDF filesystem image causes a use-after-free write

operation in the `udf_put_super` and `udf_close_lvid` functions in `fs/udf/super.c`. NOTE: the [suse.com](https://suse.com) reference has a different perspective about this.

More Info: <https://avd.aquasec.com/nvd/cve-2023-37454>

**[CVE-2023-4010] kernel: usb: hcd: malformed USB descriptor leads to infinite loop in `usb_giveback_urb()` (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A flaw was found in the USB Host Controller Driver framework in the Linux kernel. The `usb_giveback_urb` function has a logic loophole in its implementation. Due to the inappropriate judgment condition of the `goto` statement, the function cannot return under the input of a specific malformed descriptor file, so it falls into an endless loop, resulting in a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4010>

**[CVE-2023-4133] kernel: cxgb4: use-after-free in `ch_flower_stats_cb()` (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A use-after-free vulnerability was found in the `cxgb4` driver in the Linux kernel. The bug occurs when the `cxgb4` device is detaching due to a possible rearming of the `flower_stats_timer` from the work queue. This flaw allows a local user to crash the system, causing a denial of service condition.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4133>

**[CVE-2023-52485] kernel: drm/amd/display: Wake DMCUB before sending a command cause deadlock (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

`drm/amd/display: Wake DMCUB before sending a command`

[Why]

We can hang in place trying to send commands when the DMCUB isn't powered on.

[How]

For functions that execute within a DC context or DC lock we can wrap the direct calls to `dm_execute_dmub_cmd/list` with code that exits idle power optimizations and reallows once we're done with the command submission on success.

For DM direct submissions the DM will need to manage the enter/exit sequencing manually.

We cannot invoke a DMCUB command directly within the DM execution helper or we can deadlock.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52485>

**[CVE-2023-52590] kernel: ocfs2: Avoid touching renamed directory if parent does not change (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ocfs2: Avoid touching renamed directory if parent does not change

The VFS will not be locking moved directory if its parent does not change. Change ocfs2 rename code to avoid touching renamed directory if its parent does not change as without locking that can corrupt the filesystem.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52590>

**[CVE-2023-52591] kernel: reiserfs: Avoid touching renamed directory if parent does not change (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

reiserfs: Avoid touching renamed directory if parent does not change

The VFS will not be locking moved directory if its parent does not change. Change reiserfs rename code to avoid touching renamed directory if its parent does not change as without locking that can corrupt the filesystem.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52591>

**[CVE-2023-52596] kernel: sysctl: Fix out of bounds access for empty sysctl registers (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

sysctl: Fix out of bounds access for empty sysctl registers

When registering tables to the sysctl subsystem there is a check to see if header is a permanently empty directory (used for mounts). This check evaluates the first element of the ctl\_table. This results in an out of

bounds evaluation when registering empty directories.

The function `register_sysctl_mount_point` now passes a `ctl_table` of size 1 instead of size 0. It now relies solely on the type to identify a permanently empty register.

Make sure that the `ctl_table` has at least one element before testing for permanent emptiness.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52596>

## **[CVE-2023-52625] kernel: drm/amd/display: Refactor DMCUB enter/exit idle interface (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Refactor DMCUB enter/exit idle interface

### **[Why]**

We can hang in place trying to send commands when the DMCUB isn't powered on.

### **[How]**

We need to exit out of the idle state prior to sending a command, but the process that performs the exit also invokes a command itself.

Fixing this issue involves the following:

1. Using a software state to track whether or not we need to start the process to exit idle or notify idle.

It's possible for the hardware to have exited an idle state without driver knowledge, but entering one is always restricted to a driver allow - which makes the SW state vs HW state mismatch issue purely one of optimization, which should seldomly be hit, if at all.

2. Refactor any instances of exit/notify idle to use a single wrapper that maintains this SW state.

This works similar to `dc_allow_idle_optimizations`, but works at the DMCUB level and makes sure the state is marked prior to any notify/exit idle so we don't enter an infinite loop.

3. Make sure we exit out of idle prior to sending any commands or waiting for DMCUB idle.

This patch takes care of 1/2. A future patch will take care of wrapping DMCUB command submission with calls to this new interface.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52625>

**[CVE-2023-52629] kernel: sh: push-switch: Reorder cleanup operations to avoid use-after-free bug (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

sh: push-switch: Reorder cleanup operations to avoid use-after-free bug

The original code puts `flush_work()` before `timer_shutdown_sync()` in `switch_drv_remove()`. Although we use `flush_work()` to stop the worker, it could be rescheduled in `switch_timer()`. As a result, a use-after-free bug can occur. The details are shown below:

(cpu 0)		(cpu 1)
<code>switch_drv_remove()</code>		
<code>flush_work()</code>		
...		<code>switch_timer // timer</code>
		<code>schedule_work(&amp;psw-&gt;work)</code>
<code>timer_shutdown_sync()</code>		
...		<code>switch_work_handler // worker</code>
<code>kfree(psw) // free</code>		
		<code>psw-&gt;state = 0 // use</code>

This patch puts `timer_shutdown_sync()` before `flush_work()` to mitigate the bugs. As a result, the worker and timer will be stopped safely before the deallocate operations.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52629>

**[CVE-2023-52648] kernel: drm/vmwgfx: Unmap the surface before resetting it on a plane state (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/vmwgfx: Unmap the surface before resetting it on a plane state

Switch to a new plane state requires unreferencing of all held surfaces. In the work required for mob cursors the mapped surfaces started being cached but the variable indicating whether the surface is currently mapped was not being reset. This leads to crashes as the duplicated state, incorrectly, indicates the that surface is mapped even when no surface is present. That's because after unreferencing the surface it's perfectly possible for the plane to be backed by a bo instead of a surface.



Reset the surface mapped flag when unreferencing the plane state surface  
to fix null derefs in cleanup. Fixes crashes in KDE KWin 6.0 on Wayland:

Oops: 0000 [#1] PREEMPT SMP PTI

CPU: 4 PID: 2533 Comm: kwin\_wayland Not tainted 6.7.0-rc3-vmwgfx #2

Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020

RIP: 0010:vmw\_du\_cursor\_plane\_cleanup\_fb+0x124/0x140 [vmwgfx]

Code: 00 00 00 75 3a 48 83 c4 10 5b 5d c3 cc cc cc cc 48 8b b3 a8 00 00 00 48 c7 c7 99 90 43 c0 e8 93 c5 db ca 48 8b  
83 a8 00 00 00 <48> 8b 78 28 e8 e3 f>

RSP: 0018:ffffb6b98216fa80 EFLAGS: 00010246

RAX: 0000000000000000 RBX: ffff969d84cdcb00 RCX: 0000000000000027

RDX: 0000000000000000 RSI: 0000000000000001 RDI: ffff969e75f21600

RBP: ffff969d4143dc50 R08: 0000000000000000 R09: ffffb6b98216f920

R10: 0000000000000003 R11: ffff969e7feb3b10 R12: 0000000000000000

R13: 0000000000000000 R14: 000000000000027b R15: ffff969d49c9fc00

FS: 00007f1e8f1b4180(0000) GS:ffff969e75f00000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 0000000000000028 CR3: 0000000104006004 CR4: 00000000003706f0

Call Trace:

<TASK>

? \_\_die+0x23/0x70

? page\_fault\_oops+0x171/0x4e0

? exc\_page\_fault+0x7f/0x180

? asm\_exc\_page\_fault+0x26/0x30

? vmw\_du\_cursor\_plane\_cleanup\_fb+0x124/0x140 [vmwgfx]

drm\_atomic\_helper\_cleanup\_planes+0x9b/0xc0

commit\_tail+0xd1/0x130

drm\_atomic\_helper\_commit+0x11a/0x140

drm\_atomic\_commit+0x97/0xd0

? \_\_pfx\_\_drm\_printfn\_info+0x10/0x10

drm\_atomic\_helper\_update\_plane+0xf5/0x160

drm\_mode\_cursor\_universal+0x10e/0x270

drm\_mode\_cursor\_common+0x102/0x230

? \_\_pfx\_drm\_mode\_cursor2\_ioctl+0x10/0x10

drm\_ioctl\_kernel+0xb2/0x110

drm\_ioctl+0x26d/0x4b0

? \_\_pfx\_drm\_mode\_cursor2\_ioctl+0x10/0x10

? \_\_pfx\_drm\_ioctl+0x10/0x10

vmw\_generic\_ioctl+0xa4/0x110 [vmwgfx]

\_\_x64\_sys\_ioctl+0x94/0xd0

do\_syscall\_64+0x61/0xe0

? \_\_x64\_sys\_ioctl+0xaf/0xd0

? syscall\_exit\_to\_user\_mode+0x2b/0x40

? do\_syscall\_64+0x70/0xe0

? \_\_x64\_sys\_ioctl+0xaf/0xd0

? syscall\_exit\_to\_user\_mode+0x2b/0x40

? do\_syscall\_64+0x70/0xe0

? exc\_page\_fault+0x7f/0x180

entry\_SYSCALL\_64\_after\_hwframe+0x6e/0x76

RIP: 0033:0x7f1e93f279ed

Code: 04 25 28 00 00 00 48 89 45 c8 31 c0 48 8d 45 10 c7 45 b0 10 00 00 00 48 89 45 b8 48 8d 45 d0 48 89 45 c0 b8  
10 00 00 00 0f 05 <89> c2 3d 00 f0 ff f>

RSP: 002b:00007ffca0faf600 EFLAGS: 00000246 ORIG\_RAX: 0000000000000010

RAX: ffffffffda RBX: 000055db876ed2c0 RCX: 00007f1e93f279ed  
RDX: 00007fca0faf6c0 RSI: 00000000c02464bb RDI: 0000000000000015  
RBP: 00007fca0faf650 R08: 000055db87184010 R09: 0000000000000007  
R10: 000055db886471a0 R11: 0000000000000246 R12: 00007fca0faf6c0  
R13: 00000000c02464bb R14: 0000000000000015 R15: 00007fca0faf790

</TASK>

Modules linked in: snd\_seq\_dummy snd\_hrtimer nf\_conntrack\_netbios\_ns nf\_conntrack\_broadcast nft\_fib\_inet  
nft\_fib\_ipv4 nft\_fib\_ipv6 nft\_fib nft\_reject\_inet

CR2: 0000000000000028

---[ end trace 0000000000000000 ]---

RIP: 0010:vmw\_du\_cursor\_plane\_cleanup\_fb+0x124/0x140 [vmwgfx]

Code: 00 00 00 75 3a 48 83 c4 10 5b 5d c3 cc cc cc cc 48 8b b3 a8 00 00 00 48 c7 c7 99 90 43 c0 e8 93 c5 db ca 48 8b  
83 a8 00 00 00 <48> 8b 78 28 e8 e3 f>

RSP: 0018:ffffb6b98216fa80 EFLAGS: 00010246

RAX: 0000000000000000 RBX: ffff969d84cdcb00 RCX: 0000000000000027

RDX: 0000000000000000 RSI: 0000000000000001 RDI: ffff969e75f21600

RBP: ffff969d4143

---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2023-52648>

### **[CVE-2023-52653] kernel: SUNRPC: fix a memleak in gss\_import\_v2\_context (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

SUNRPC: fix a memleak in gss\_import\_v2\_context

The ctx->mech\_used.data allocated by kmemdup is not freed in neither  
gss\_import\_v2\_context nor its caller gss\_krb5\_import\_sec\_context,  
which frees ctx on error.

Thus, this patch reform the last call of gss\_import\_v2\_context to the  
gss\_krb5\_import\_ctx\_v2, preventing the memleak while keeping the return  
formation.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52653>

### **[CVE-2023-52658] kernel: Revert "net/mlx5: Block entering switchdev mode with ns inconsistency"; (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Revert "net/mlx5: Block entering switchdev mode with ns inconsistency"

This reverts commit 662404b24a4c4d839839ed25e3097571f5938b9b.

The revert is required due to the suspicion it is not good for anything

and cause crash.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52658>

**[CVE-2023-52671] kernel: drm/amd/display: Fix hang/underflow when transitioning to ODM4:1 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix hang/underflow when transitioning to ODM4:1

**[Why]**

Under some circumstances, disabling an OPTC and attempting to reclaim its OPP(s) for a different OPTC could cause a hang/underflow due to OPPs not being properly disconnected from the disabled OPTC.

**[How]**

Ensure that all OPPs are unassigned from an OPTC when it gets disabled.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52671>

**[CVE-2023-52673] kernel: drm/amd/display: Fix a debugfs null pointer error (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix a debugfs null pointer error

**[WHY & HOW]**

Check whether get\_subvp\_en() callback exists before calling it.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52673>

**[CVE-2023-52676] kernel: bpf: Guard stack limits against 32bit overflow (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Guard stack limits against 32bit overflow

This patch promotes the arithmetic around checking stack bounds to be done in the 64-bit domain, instead of the current 32bit. The arithmetic implies adding together a 64-bit register with a int offset. The register was checked to be below 1<<29 when it was variable, but not when it was fixed. The offset either comes from an instruction (in which

case it is 16 bit), from another register (in which case the caller checked it to be below  $1 \ll 29$  [1]), or from the size of an argument to a kfunc (in which case it can be a u32 [2]). Between the register being inconsistently checked to be below  $1 \ll 29$ , and the offset being up to an u32, it appears that we were open to overflowing the `int`s which were currently used for arithmetic.

[1]

<https://github.com/torvalds/linux/blob/815fb87b753055df2d9e50f6cd80eb10235fe3e9/kernel/bpf/verifier.c#L7494-L7498>

[2] <https://github.com/torvalds/linux/blob/815fb87b753055df2d9e50f6cd80eb10235fe3e9/kernel/bpf/verifier.c#L11904>

More Info: <https://avd.aquasec.com/nvd/cve-2023-52676>

### **[CVE-2023-52761] kernel: riscv: VMAP\_STACK overflow detection thread-safe (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

riscv: VMAP\_STACK overflow detection thread-safe

commit 31da94c25aea ("riscv: add VMAP\_STACK overflow detection") added support for CONFIG\_VMAP\_STACK. If overflow is detected, CPU switches to `shadow\_stack` temporarily before switching finally to per-cpu `overflow\_stack`.

If two CPUs/harts are racing and end up in overflowing kernel stack, one or both will end up corrupting each other state because `shadow\_stack` is not per-cpu. This patch optimizes per-cpu overflow stack switch by directly picking per-cpu `overflow\_stack` and gets rid of `shadow\_stack`.

Following are the changes in this patch

- Defines an asm macro to obtain per-cpu symbols in destination register.
- In entry.S, when overflow is detected, per-cpu overflow stack is located using per-cpu asm macro. Computing per-cpu symbol requires a temporary register. x31 is saved away into CSR\_SCRATCH (CSR\_SCRATCH is anyways zero since we're in kernel).

Please see Links for additional relevant discussion and alternative solution.

Tested by `echo EXHAUST\_STACK > /sys/kernel/debug/provoke-crash/DIRECT`  
Kernel crash log below

Insufficient stack space to handle exception!/debug/provoke-crash/DIRECT  
Task stack: [0xff20000010a98000..0xff20000010a9c000]  
Overflow stack: [0xff600001f7d98370..0xff600001f7d99370]  
CPU: 1 PID: 205 Comm: bash Not tainted 6.1.0-rc2-00001-g328a1f96f7b9 #34  
Hardware name: riscv-virtio,qemu (DT)  
epc : \_\_memset+0x60/0xfc

```
ra : recursive_loop+0x48/0xc6 [lkdtm]
epc : ffffffff808de0e4 ra : ffffffff0163a752 sp : ff20000010a97e80
gp : ffffffff815c0330 tp : ff600000820ea280 t0 : ff20000010a97e88
t1 : 000000000000002e t2 : 3233206874706564 s0 : ff20000010a982b0
s1 : 0000000000000012 a0 : ff20000010a97e88 a1 : 0000000000000000
a2 : 0000000000000400 a3 : ff20000010a98288 a4 : 0000000000000000
a5 : 0000000000000000 a6 : ffffffff43f0 a7 : 00007fffffffff
s2 : ff20000010a97e88 s3 : ffffffff01644680 s4 : ff20000010a9be90
s5 : ff600000842ba6c0 s6 : 00aaaaaac29e42b0 s7 : 00ffffff0aa3684
s8 : 00aaaaaac2978040 s9 : 0000000000000065 s10: 00ffffff8a7cad10
s11: 00ffffff8a76a4e0 t3 : ffffffff815dbaf4 t4 : ffffffff815dbaf4
t5 : ffffffff815dbab8 t6 : ff20000010a9bb48
status: 0000000200000120 badaddr: ff20000010a97e88 cause: 000000000000000f
Kernel panic - not syncing: Kernel stack overflow
CPU: 1 PID: 205 Comm: bash Not tainted 6.1.0-rc2-00001-g328a1f96f7b9 #34
Hardware name: riscv-virtio,qemu (DT)
Call Trace:
[<ffffffff80006754>] dump_backtrace+0x30/0x38
[<ffffffff808de798>] show_stack+0x40/0x4c
[<ffffffff808ea2a8>] dump_stack_lvl+0x44/0x5c
[<ffffffff808ea2d8>] dump_stack+0x18/0x20
[<ffffffff808dec06>] panic+0x126/0x2fe
[<ffffffff800065ea>] walk_stackframe+0x0/0xf0
[<ffffffff0163a752>] recursive_loop+0x48/0xc6 [lkdtm]
SMP: stopping secondary CPUs
---[ end Kernel panic - not syncing: Kernel stack overflow ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2023-52761>

## **[CVE-2023-52770] kernel: f2fs: split initial and dynamic conditions for extent\_cache (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: split initial and dynamic conditions for extent\_cache

Let's allocate the extent\_cache tree without dynamic conditions to avoid a missing condition causing a panic as below.

```
# create a file w/ a compressed flag
# disable the compression
# panic while updating extent_cache
```

F2FS-fs (dm-64): Swapfile: last extent is not aligned to section  
F2FS-fs (dm-64): Swapfile (3) is not align to section: 1) creat(), 2) ioctl(F2FS\_IOC\_SET\_PIN\_FILE), 3) falloca(2097152 \* N)  
Adding 124996k swap on ./swap-file. Priority:0 extents:2 across:17179494468k

=====

BUG: KASAN: null-ptr-deref in instrument\_atomic\_read\_write out/common/include/linux/instrumented.h:101 [inline]

BUG: KASAN: null-ptr-deref in atomic\_try\_cmpxchg\_acquire  
out/common/include/asm-generic/atomic-instrumented.h:705 [inline]  
BUG: KASAN: null-ptr-deref in queued\_write\_lock out/common/include/asm-generic/qrwlock.h:92 [inline]  
BUG: KASAN: null-ptr-deref in \_\_raw\_write\_lock out/common/include/linux/rwlock\_api\_smp.h:211 [inline]  
BUG: KASAN: null-ptr-deref in \_raw\_write\_lock+0x5a/0x110 out/common/kernel/locking/spinlock.c:295  
Write of size 4 at addr 0000000000000030 by task syz-executor154/3327

CPU: 0 PID: 3327 Comm: syz-executor154 Tainted: G O 5.10.185 #1  
Hardware name: emulation qemu-x86/qemu-x86, BIOS 2023.01-21885-gb3cc1cd24d 01/01/2023

Call Trace:

\_\_dump\_stack out/common/lib/dump\_stack.c:77 [inline]  
dump\_stack\_lvl+0x17e/0x1c4 out/common/lib/dump\_stack.c:118  
\_\_kasan\_report+0x16c/0x260 out/common/mm/kasan/report.c:415  
kasan\_report+0x51/0x70 out/common/mm/kasan/report.c:428  
kasan\_check\_range+0x2f3/0x340 out/common/mm/kasan/generic.c:186  
\_\_kasan\_check\_write+0x14/0x20 out/common/mm/kasan/shadow.c:37  
instrument\_atomic\_read\_write out/common/include/linux/instrumented.h:101 [inline]  
atomic\_try\_cmpxchg\_acquire out/common/include/asm-generic/atomic-instrumented.h:705 [inline]  
queued\_write\_lock out/common/include/asm-generic/qrwlock.h:92 [inline]  
\_\_raw\_write\_lock out/common/include/linux/rwlock\_api\_smp.h:211 [inline]  
\_raw\_write\_lock+0x5a/0x110 out/common/kernel/locking/spinlock.c:295  
\_\_drop\_extent\_tree+0xdf/0x2f0 out/common/fs/f2fs/extent\_cache.c:1155  
f2fs\_drop\_extent\_tree+0x17/0x30 out/common/fs/f2fs/extent\_cache.c:1172  
f2fs\_insert\_range out/common/fs/f2fs/file.c:1600 [inline]  
f2fs\_fallocate+0x19fd/0x1f40 out/common/fs/f2fs/file.c:1764  
vfs\_fallocate+0x514/0x9b0 out/common/fs/open.c:310  
ksys\_fallocate out/common/fs/open.c:333 [inline]  
\_\_do\_sys\_fallocate out/common/fs/open.c:341 [inline]  
\_\_se\_sys\_fallocate out/common/fs/open.c:339 [inline]  
\_\_x64\_sys\_fallocate+0xb8/0x100 out/common/fs/open.c:339  
do\_syscall\_64+0x35/0x50 out/common/arch/x86/entry/common.c:46

More Info: <https://avd.aquasec.com/nvd/cve-2023-52770>

## **[CVE-2023-52771] kernel: cxl/port: Fix delete\_endpoint() vs parent unregistration race (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

cxl/port: Fix delete\_endpoint() vs parent unregistration race

The CXL subsystem, at cxl\_mem ->probe() time, establishes a lineage of ports (struct cxl\_port objects) between an endpoint and the root of a CXL topology. Each port including the endpoint port is attached to the cxl\_port driver.

Given that setup, it follows that when either any port in that lineage goes through a cxl\_port ->remove() event, or the memdev goes through a cxl\_mem ->remove() event. The hierarchy below the removed port, or the

entire hierarchy if the memdev is removed needs to come down.

The `delete_endpoint()` callback is careful to check whether it is being called to tear down the hierarchy, or if it is only being called to teardown the memdev because an ancestor port is going through `->remove()`.

That care needs to take the `device_lock()` of the endpoint's parent. Which requires 2 bugs to be fixed:

1/ A reference on the parent is needed to prevent use-after-free scenarios like this signature:

```
BUG: spinlock bad magic on CPU#0, kworker/u56:0/11
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS edk2-20230524-3.fc38 05/24/2023
Workqueue: cxl_port detach_memdev [cxl_core]
RIP: 0010:spin_bug+0x65/0xa0
Call Trace:
  do_raw_spin_lock+0x69/0xa0
  __mutex_lock+0x695/0xb80
  delete_endpoint+0xad/0x150 [cxl_core]
  devres_release_all+0xb8/0x110
  device_unbind_cleanup+0xe/0x70
  device_release_driver_internal+0x1d2/0x210
  detach_memdev+0x15/0x20 [cxl_core]
  process_one_work+0x1e3/0x4c0
  worker_thread+0x1dd/0x3d0
```

2/ In the case of RCH topologies, the parent device that needs to be locked is not always `@port->dev` as returned by `cxl_mem_find_port()`, use `endpoint->dev.parent` instead.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52771>

### **[CVE-2023-52797] kernel: drivers: perf: Check find\_first\_bit() return value (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drivers: perf: Check `find_first_bit()` return value

We must check the return value of `find_first_bit()` before using the return value as an index array since it happens to overflow the array and then panic:

```
[ 107.318430] Kernel BUG [#1]
[ 107.319434] CPU: 3 PID: 1238 Comm: kill Tainted: G      E      6.6.0-rc6ubuntu-defconfig #2
[ 107.319465] Hardware name: riscv-virtio,qemu (DT)
[ 107.319551] epc : pmu_sbi_ovf_handler+0x3a4/0x3ae
[ 107.319840] ra : pmu_sbi_ovf_handler+0x52/0x3ae
[ 107.319868] epc : ffffffff80a0a77c ra : ffffffff80a0a42a sp : ffffaf83fecda350
```

```
[ 107.319884] gp : ffffffff823961a8 tp : ffffaf8083db1dc0 t0 : ffffaf83fecda480
[ 107.319899] t1 : ffffffff80cafe62 t2 : 000000000000ff00 s0 : ffffaf83fecda520
[ 107.319921] s1 : ffffaf83fecda380 a0 : 00000018fca29df0 a1 : ffffffff
[ 107.319936] a2 : 0000000001073734 a3 : 0000000000000004 a4 : 0000000000000000
[ 107.319951] a5 : 0000000000000040 a6 : 000000001d1c8774 a7 : 0000000000504d55
[ 107.319965] s2 : ffffffff82451f10 s3 : ffffffff82724e70 s4 : 000000000000003f
[ 107.319980] s5 : 0000000000000011 s6 : ffffaf8083db27c0 s7 : 0000000000000000
[ 107.319995] s8 : 0000000000000001 s9 : 00007fffb45d6558 s10: 00007fffb45d81a0
[ 107.320009] s11: ffffaf7fff600000 t3 : 0000000000000004 t4 : 0000000000000000
[ 107.320023] t5 : ffffaf7f80000000 t6 : ffffaf8000000000
[ 107.320037] status: 0000000200000100 badaddr: 0000000000000000 cause: 0000000000000003
[ 107.320081] [<ffffff80a0a77c>] pmu_sbi_ovf_handler+0x3a4/0x3ae
[ 107.320112] [<ffffff800b42d0>] handle_percpu_devid_irq+0x9e/0x1a0
[ 107.320131] [<ffffff800ad92c>] generic_handle_domain_irq+0x28/0x36
[ 107.320148] [<ffffff8065f9f8>] riscv_intc_irq+0x36/0x4e
[ 107.320166] [<ffffff80caf4a0>] handle_riscv_irq+0x54/0x86
[ 107.320189] [<ffffff80cb0036>] do_irq+0x64/0x96
[ 107.320271] Code: 85a6 855e b097 ff7f 80e7 9220 b709 9002 4501 bbd9 (9002) 6097
[ 107.320585] ---[ end trace 0000000000000000 ]---
[ 107.320704] Kernel panic - not syncing: Fatal exception in interrupt
[ 107.320775] SMP: stopping secondary CPUs
[ 107.321219] Kernel Offset: 0x0 from 0xfffffff800000000
[ 107.333051] ---[ end Kernel panic - not syncing: Fatal exception in interrupt ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2023-52797>

### **[CVE-2023-52857] kernel: drm/mediatek: Fix coverity issue with unintentional integer overflow (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/mediatek: Fix coverity issue with unintentional integer overflow

1. Instead of multiplying 2 variable of different types. Change to assign a value of one variable and then multiply the other variable.
2. Add a int variable for multiplier calculation instead of calculating different types multiplier with dma\_addr\_t variable directly.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52857>

### **[CVE-2023-52888] kernel: media: mediatek: vcodec: Only free buffer VA that is not NULL (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:



media: mediatek: vcodec: Only free buffer VA that is not NULL

In the MediaTek vcodec driver, while mtk\_vcodec\_mem\_free() is mostly called only when the buffer to free exists, there are some instances that didn't do the check and triggered warnings in practice.

We believe those checks were forgotten unintentionally. Add the checks back to fix the warnings.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52888>

**[CVE-2023-52920] kernel: bpf: support non-r10 register spill/fill to/from stack in precision tracking (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: support non-r10 register spill/fill to/from stack in precision tracking

Use instruction (jump) history to record instructions that performed register spill/fill to/from stack, regardless if this was done through read-only r10 register, or any other register after copying r10 into it \*and\* potentially adjusting offset.

To make this work reliably, we push extra per-instruction flags into instruction history, encoding stack slot index (spi) and stack frame number in extra 10 bit flags we take away from prev\_idx in instruction history. We don't touch idx field for maximum performance, as it's checked most frequently during backtracking.

This change removes basically the last remaining practical limitation of precision backtracking logic in BPF verifier. It fixes known deficiencies, but also opens up new opportunities to reduce number of verified states, explored in the subsequent patches.

There are only three differences in selftests' BPF object files according to veristat, all in the positive direction (less states).

File	Program	Insns (A)	Insns (B)	Insns (DIFF)	States (A)	States (B)	States (DIFF)
test_cls_redirect_dynptr.bpf.linked3.o	cls_redirect	2987	2864	-123 (-4.12%)	240	231	-9 (-3.75%)
xdp_synproxy_kern.bpf.linked3.o	syncookie_tc	82848	82661	-187 (-0.23%)	5107	5073	-34 (-0.67%)
xdp_synproxy_kern.bpf.linked3.o	syncookie_xdp	85116	84964	-152 (-0.18%)	5162	5130	-32 (-0.62%)

Note, I avoided renaming jmp\_history to more generic insn\_hist to minimize number of lines changed and potential merge conflicts between bpf and bpf-next trees.

Notice also `cur_hist_entry` pointer reset to `NULL` at the beginning of instruction verification loop. This pointer avoids the problem of relying on last jump history entry's `insn_idx` to determine whether we already have entry for current instruction or not. It can happen that we added jump history entry because current instruction is `_jmp_point()`, but also we need to add instruction flags for stack access. In this case, we don't want to entries, so we need to reuse last added entry, if it is present.

Relying on `insn_idx` comparison has the same ambiguity problem as the one that was fixed recently in [0], so we avoid that.

[0] <https://patchwork.kernel.org/project/netdevbpf/patch/20231110002638.4168352-3-andrii@kernel.org/>

More Info: <https://avd.aquasec.com/nvd/cve-2023-52920>

### **[CVE-2023-52927] kernel: netfilter: allow exp not to be removed in `nf_ct_find_expectation` (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

netfilter: allow exp not to be removed in `nf_ct_find_expectation`

Currently `nf_conntrack_in()` calling `nf_ct_find_expectation()` will remove the exp from the hash table. However, in some scenario, we expect the exp not to be removed when the created ct will not be confirmed, like in OVS and TC conntrack in the following patches.

This patch allows exp not to be removed by setting `IPS_CONFIRMED` in the status of the `tmpl`.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52927>

### **[CVE-2023-6039] kernel: use-after-free in `drivers/net/usb/lan78xx.c` in `lan78xx_disconnect` (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A use-after-free flaw was found in `lan78xx_disconnect` in `drivers/net/usb/lan78xx.c` in the network sub-component, `net/usb/lan78xx` in the Linux Kernel. This flaw allows a local attacker to crash the system when the LAN78XX USB device detaches.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6039>

### **[CVE-2023-6240] kernel: Marvin vulnerability side-channel leakage in the RSA decryption operation (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A Marvin vulnerability side-channel leakage was found in the RSA decryption operation in the Linux Kernel. This issue may allow a network attacker to decrypt ciphertexts or forge signatures, limiting the services that use that private key.

More Info: <https://avd.aquasec.com/nvd/cve-2023-6240>

**[CVE-2024-2193] hw: Spectre-SRC that is Speculative Race Conditions (SRCs) for synchronization primitives similar like Spectre V1 with possibility to bypass software features (e.g., IPIs, high-precision timers, etc) (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A Speculative Race Condition (SRC) vulnerability that impacts modern CPU architectures supporting speculative execution (related to Spectre V1) has been disclosed. An unauthenticated attacker can exploit this vulnerability to disclose arbitrary data from the CPU using race conditions to access the speculative executable code paths.

More Info: <https://avd.aquasec.com/nvd/cve-2024-2193>

**[CVE-2024-24855] kernel: Race condition in lpfc\_unregister\_fcf\_rescan() in scsi/lpfc/lpfc\_hbadisc.c (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

A race condition was found in the Linux kernel's scsi device driver in lpfc\_unregister\_fcf\_rescan() function. This can result in a null pointer dereference issue, possibly leading to a kernel panic or denial of service issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-24855>

**[CVE-2024-24864] A race condition was found in the Linux kernel's media/dvb-core in dvb ... (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A race condition was found in the Linux kernel's media/dvb-core in dvbdmx\_write() function. This can result in a null pointer dereference issue, possibly leading to a kernel panic or denial of service issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-24864>

### **[CVE-2024-25740] kernel: memory leak in ubi driver (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

A memory leak flaw was found in the UBI driver in drivers/mtd/ubi/attach.c in the Linux kernel through 6.7.4 for UBI\_IOCATT, because kobj->name is not released.

More Info: <https://avd.aquasec.com/nvd/cve-2024-25740>

### **[CVE-2024-26618] hw: arm64/sme: Always exit sme\_alloc() early with existing storage (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

arm64/sme: Always exit sme\_alloc() early with existing storage

When sme\_alloc() is called with existing storage and we are not flushing we will always allocate new storage, both leaking the existing storage and corrupting the state. Fix this by separating the checks for flushing and for existing storage as we do for SVE.

Callers that reallocate (eg, due to changing the vector length) should call sme\_free() themselves.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26618>

### **[CVE-2024-26647] kernel: drm/amd/display: Fix late dereference 'dsc' check in 'link\_set\_dsc\_pps\_packet()' (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix late derefrence 'dsc' check in 'link\_set\_dsc\_pps\_packet()'

In link\_set\_dsc\_pps\_packet(), 'struct display\_stream\_compressor \*dsc' was dereferenced in a DC\_LOGGER\_INIT(dsc->ctx->logger); before the 'dsc' NULL pointer check.

Fixes the below:

drivers/gpu/drm/amd/amdgpu/./display/dc/link/link\_dpms.c:905 link\_set\_dsc\_pps\_packet() warn: variable dereferenced before check 'dsc' (see line 903)

More Info: <https://avd.aquasec.com/nvd/cve-2024-26647>

### **[CVE-2024-26648] kernel: NULL check in edp\_setup\_replay() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix variable deferencing before NULL check in edp\_setup\_replay()

In edp\_setup\_replay(), 'struct dc \*dc' & 'struct dmub\_replay \*replay'  
was dereferenced before the pointer 'link' & 'replay' NULL check.

Fixes the below:

drivers/gpu/drm/amd/amdgpu/./display/dc/link/protocols/link\_edp\_panel\_control.c:947 edp\_setup\_replay() warn:  
variable dereferenced before check 'link' (see line 933)

More Info: <https://avd.aquasec.com/nvd/cve-2024-26648>

### **[CVE-2024-26656] kernel: drm/amdgpu: use-after-free vulnerability (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: fix use-after-free bug

The bug can be triggered by sending a single amdgpu\_gem\_userptr\_ioctl  
to the AMDGPU DRM driver on any ASICs with an invalid address and size.

The bug was reported by Joonkyo Jung <joonkyoj@yonsei.ac.kr>.

For example the following code:

```
static void Syzkaller1(int fd)
{
    struct drm_amdgpu_gem_userptr arg;
    int ret;

    arg.addr = 0xffffffff0000;
    arg.size = 0x80000000; /*2 Gb*/
    arg.flags = 0x7;
    ret = drmIoctl(fd, 0xc1186451/*amdgpu_gem_userptr_ioctl*/, &arg);
}
```

Due to the address and size are not valid there is a failure in  
amdgpu\_hmm\_register->mmu\_interval\_notifier\_insert->\_\_mmu\_interval\_notifier\_insert->  
check\_shl\_overflow, but we even the amdgpu\_hmm\_register failure we still call  
amdgpu\_hmm\_unregister into amdgpu\_gem\_object\_free which causes access to a bad address.  
The following stack is below when the issue is reproduced when Kazan is enabled:

```
[ +0.000014] Hardware name: ASUS System Product Name/ROG STRIX B550-F GAMING (WI-FI), BIOS 1401
12/03/2020
[ +0.000009] RIP: 0010:mmu_interval_notifier_remove+0x327/0x340
[ +0.000017] Code: ff ff 49 89 44 24 08 48 b8 00 01 00 00 00 00 ad de 4c 89 f7 49 89 47 40 48 83 c0 22 49 89 47 48 e8
ce d1 2d 01 e9 32 ff ff ff <0f> 0b e9 16 ff ff ff 4c 89 ef e8 fa 14 b3 ff e9 36 ff ff ff e8 80
```

```
[ +0.000014] RSP: 0018:ffffc90002657988 EFLAGS: 00010246
[ +0.000013] RAX: 0000000000000000 RBX: 1ffff920004caf35 RCX: ffffffff8160565b
[ +0.000011] RDX: dffffc0000000000 RSI: 0000000000000004 RDI: ffff8881a9f78260
[ +0.000010] RBP: fffffc90002657a70 R08: 0000000000000001 R09: fffff520004caf25
[ +0.000010] R10: 0000000000000003 R11: ffffffff8161d1d6 R12: ffff88810e988c00
[ +0.000010] R13: ffff888126fb5a00 R14: ffff88810e988c0c R15: ffff8881a9f78260
[ +0.000011] FS: 00007ff9ec848540(0000) GS:ffff8883cc880000(0000) knlGS:0000000000000000
[ +0.000012] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ +0.000010] CR2: 000055b3f7e14328 CR3: 00000001b5770000 CR4: 0000000000350ef0
[ +0.000010] Call Trace:
[ +0.000006] <TASK>
[ +0.000007] ? show_regs+0x6a/0x80
[ +0.000018] ? __warn+0xa5/0x1b0
[ +0.000019] ? mmu_interval_notifier_remove+0x327/0x340
[ +0.000018] ? report_bug+0x24a/0x290
[ +0.000022] ? handle_bug+0x46/0x90
[ +0.000015] ? exc_invalid_op+0x19/0x50
[ +0.000016] ? asm_exc_invalid_op+0x1b/0x20
[ +0.000017] ? kasan_save_stack+0x26/0x50
[ +0.000017] ? mmu_interval_notifier_remove+0x23b/0x340
[ +0.000019] ? mmu_interval_notifier_remove+0x327/0x340
[ +0.000019] ? mmu_interval_notifier_remove+0x23b/0x340
[ +0.000020] ? __pfx_mmu_interval_notifier_remove+0x10/0x10
[ +0.000017] ? kasan_save_alloc_info+0x1e/0x30
[ +0.000018] ? srso_return_thunk+0x5/0x5f
[ +0.000014] ? __kasan_kmalloc+0xb1/0xc0
[ +0.000018] ? srso_return_thunk+0x5/0x5f
[ +0.000013] ? __kasan_check_read+0x11/0x20
[ +0.000020] amdgpu_hmm_unregister+0x34/0x50 [amdgpu]
[ +0.004695] amdgpu_gem_object_free+0x66/0xa0 [amdgpu]
[ +0.004534] ? __pfx_amdgpu_gem_object_free+0x10/0x10 [amdgpu]
[ +0.004291] ? do_syscall_64+0x5f/0xe0
[ +0.000023] ? srso_return_thunk+0x5/0x5f
[ +0.000017] drm_gem_object_free+0x3b/0x50 [drm]
[ +0.000489] amdgpu_gem_userptr_ioctl+0x306/0x500 [amdgpu]
[ +0.004295] ? __pfx_amdgpu_gem_userptr_ioctl+0x10/0x10 [amdgpu]
[ +0.004270] ? srso_return_thunk+0x5/0x5f
[ +0.000014] ? __this_cpu_preempt_check+0x13/0x20
[ +0.000015] ? srso_return_thunk+0x5/0x5f
[ +0.000013] ? sysvec_apic_timer_interrupt+0x57/0xc0
[ +0.000020] ? srso_return_thunk+0x5/0x5f
[ +0.000014] ? asm_sysvec_apic_timer_interrupt+0x1b/0x20
[ +0.000022] ? drm_ioctl_kernel+0x17b/0x1f0 [drm]
[ +0.000496] ? __pfx_amdgpu_gem_userptr_ioctl+0x10/0x10 [amdgpu]
[ +0.004272] ? drm_ioctl_kernel+0x190/0x1f0 [drm]
[ +0.000492] drm_ioctl_kernel+0x140/0x1f0 [drm]
[ +0.000497] ? __pfx_amdgpu_gem_userptr_ioctl+0x10/0x10 [amdgpu]
[ +0.004297] ? __pfx_drm_ioctl_kernel+0x10/0x10 [d
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26656>

**[CVE-2024-26661] kernel: drm/amd/display: Add NULL test for 'timing generator' in 'dcn21\_set\_pipe()'**

**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add NULL test for 'timing generator' in 'dcn21\_set\_pipe()'

In "u32 otg\_inst = pipe\_ctx->stream\_res.tg->inst;"  
pipe\_ctx->stream\_res.tg could be NULL, it is relying on the caller to  
ensure the tg is not NULL.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26661>

**[CVE-2024-26662] kernel: drm/amd/display: 'panel\_cntl' could be null in 'dcn21\_set\_backlight\_level()'**  
**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix 'panel\_cntl' could be null in 'dcn21\_set\_backlight\_level()'

'panel\_cntl' structure used to control the display panel could be null,  
dereferencing it could lead to a null pointer access.

Fixes the below:

drivers/gpu/drm/amd/amdgpu/../display/dc/hwss/dcn21/dcn21\_hwseq.c:269 dcn21\_set\_backlight\_level() error: we  
previously assumed 'panel\_cntl' could be null (see line 250)

More Info: <https://avd.aquasec.com/nvd/cve-2024-26662>

**[CVE-2024-26670] kernel: arm64: entry: fix ARM64\_WORKAROUND\_SPECULATIVE\_UNPRIV\_LOAD**  
**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

arm64: entry: fix ARM64\_WORKAROUND\_SPECULATIVE\_UNPRIV\_LOAD

Currently the ARM64\_WORKAROUND\_SPECULATIVE\_UNPRIV\_LOAD workaround isn't  
quite right, as it is supposed to be applied after the last explicit  
memory access, but is immediately followed by an LDR.

The ARM64\_WORKAROUND\_SPECULATIVE\_UNPRIV\_LOAD workaround is used to  
handle Cortex-A520 erratum 2966298 and Cortex-A510 erratum 3117295,  
which are described in:

\* <https://developer.arm.com/documentation/SDEN2444153/0600/?lang=en>

\* <https://developer.arm.com/documentation/SDEN1873361/1600/?lang=en>

In both cases the workaround is described as:

| If pagetable isolation is disabled, the context switch logic in the  
| kernel can be updated to execute the following sequence on affected  
| cores before exiting to EL0, and after all explicit memory accesses:

- |
- | 1. A non-shareable TLBI to any context and/or address, including  
| unused contexts or addresses, such as a `TLBI VALE1 Xzr`.
  - |
  - | 2. A DSB NSH to guarantee completion of the TLBI.

The important part being that the TLBI+DSB must be placed "after all explicit memory accesses".

Unfortunately, as-implemented, the TLBI+DSB is immediately followed by an LDR, as we have:

```
| alternative_if ARM64_WORKAROUND_SPECULATIVE_UNPRIV_LOAD
| tlbi vale1, xzr
| dsb nsh
| alternative_else_nop_endif
| alternative_if_not ARM64_UNMAP_KERNEL_AT_EL0
| ldr lr, [sp, #S_LR]
| add sp, sp, #PT_REGS_SIZE // restore sp
| eret
| alternative_else_nop_endif
|
| [ ... KPTI exception return path ... ]
```

This patch fixes this by reworking the logic to place the TLBI+DSB immediately before the ERET, after all explicit memory accesses.

The ERET is currently in a separate alternative block, and alternatives cannot be nested. To account for this, the alternative block for ARM64\_UNMAP\_KERNEL\_AT\_EL0 is replaced with a single alternative branch to skip the KPTI logic, with the new shape of the logic being:

```
| alternative_insn "b .L_skip_tramp_exit_\\@", nop, ARM64_UNMAP_KERNEL_AT_EL0
| [ ... KPTI exception return path ... ]
|.L_skip_tramp_exit_\\@:
|
| ldr lr, [sp, #S_LR]
| add sp, sp, #PT_REGS_SIZE // restore sp
|
| alternative_if ARM64_WORKAROUND_SPECULATIVE_UNPRIV_LOAD
| tlbi vale1, xzr
| dsb nsh
| alternative_else_nop_endif
| eret
```



The new structure means that the workaround is only applied when KPTI is not in use; this is fine as noted in the documented implications of the erratum:

| Pagetable isolation between EL0 and higher level ELs prevents the  
| issue from occurring.

... and as per the workaround description quoted above, the workaround is only necessary "If pagetable isolation is disabled".

More Info: <https://avd.aquasec.com/nvd/cve-2024-26670>

**[CVE-2024-26672] kernel: drm/amdgpu: variable 'mca\_funcs' dereferenced before NULL check in 'amdgpu\_mca\_smu\_get\_mca\_entry()' (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Fix variable 'mca\_funcs' dereferenced before NULL check in 'amdgpu\_mca\_smu\_get\_mca\_entry()'

Fixes the below:

drivers/gpu/drm/amd/amdgpu/amdgpu\_mca.c:377 amdgpu\_mca\_smu\_get\_mca\_entry() warn: variable dereferenced before check 'mca\_funcs' (see line 368)

```
357 int amdgpu_mca_smu_get_mca_entry(struct amdgpu_device *adev,  
    enum amdgpu_mca_error_type type,  
358     int idx, struct mca_bank_entry *entry)  
359 {  
360     const struct amdgpu_mca_smu_funcs *mca_funcs =  
        adev->mca.mca_funcs;  
361     int count;  
362  
363     switch (type) {  
364     case AMDGPU_MCA_ERROR_TYPE_UE:  
365         count = mca_funcs->max_ue_count;
```

mca\_funcs is dereferenced here.

```
366         break;  
367     case AMDGPU_MCA_ERROR_TYPE_CE:  
368         count = mca_funcs->max_ce_count;
```

mca\_funcs is dereferenced here.

```
369         break;  
370     default:  
371         return -EINVAL;  
372     }  
373
```

```
374     if (idx >= count)
375         return -EINVAL;
376
377     if (mca_funcs && mca_funcs->mca_get_mca_entry)
        ~~~~~
```

Checked too late!

More Info: <https://avd.aquasec.com/nvd/cve-2024-26672>

**[CVE-2024-26677] kernel: rxrpc: Fix delayed ACKs to not set the reference serial number (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

rxrpc: Fix delayed ACKs to not set the reference serial number

Fix the construction of delayed ACKs to not set the reference serial number as they can't be used as an RTT reference.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26677>

**[CVE-2024-26691] kernel: KVM: arm64: Fix circular locking dependency (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

KVM: arm64: Fix circular locking dependency

The rule inside kvm enforces that the vcpu->mutex is taken \*inside\* kvm->lock. The rule is violated by the pkvm\_create\_hyp\_vm() which acquires the kvm->lock while already holding the vcpu->mutex lock from kvm\_vcpu\_ioctl(). Avoid the circular locking dependency altogether by protecting the hyp vm handle with the config\_lock, much like we already do for other forms of VM-scoped data.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26691>

**[CVE-2024-26719] kernel: nouveau: offload fence uevents work to workqueue (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nouveau: offload fence uevents work to workqueue

This should break the deadlock between the fctx lock and the irq lock.

This offloads the processing off the work from the irq into a workqueue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26719>

**[CVE-2024-26740] kernel: net/sched: act\_mirred: use the backlog for mirrored ingress (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/sched: act\_mirred: use the backlog for mirrored ingress

The test Davide added in commit ca22da2fbd69 ("act\_mirred: use the backlog for nested calls to mirrored ingress") hangs our testing VMs every 10 or so runs, with the familiar tcp\_v4\_rcv -> tcp\_v4\_rcv deadlock reported by lockdep.

The problem as previously described by Davide (see Link) is that if we reverse flow of traffic with the redirect (egress -> ingress) we may reach the same socket which generated the packet. And we may still be holding its socket lock. The common solution to such deadlocks is to put the packet in the Rx backlog, rather than run the Rx path inline. Do that for all egress -> ingress reversals, not just once we started to nest mirrored calls.

In the past there was a concern that the backlog indirection will lead to loss of error reporting / less accurate stats. But the current workaround does not seem to address the issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26740>

**[CVE-2024-26756] kernel: md: Don't register sync\_thread for reshape directly (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

md: Don't register sync\_thread for reshape directly

Currently, if reshape is interrupted, then reassemble the array will register sync\_thread directly from pers->run(), in this case 'MD\_RECOVERY\_RUNNING' is set directly, however, there is no guarantee that md\_do\_sync() will be executed, hence stop\_sync\_thread() will hang because 'MD\_RECOVERY\_RUNNING' can't be cleared.

Last patch make sure that md\_do\_sync() will set MD\_RECOVERY\_DONE, however, following hang can still be triggered by dm-raid test

shell/lvconvert-raid-reshape.sh occasionally:

```
[root@fedora ~]# cat /proc/1982/stack
[<0>] stop_sync_thread+0x1ab/0x270 [md_mod]
[<0>] md_frozen_sync_thread+0x5c/0xa0 [md_mod]
[<0>] raid_presuspend+0x1e/0x70 [dm_raid]
[<0>] dm_table_presuspend_targets+0x40/0xb0 [dm_mod]
[<0>] __dm_destroy+0x2a5/0x310 [dm_mod]
[<0>] dm_destroy+0x16/0x30 [dm_mod]
[<0>] dev_remove+0x165/0x290 [dm_mod]
[<0>] ctl_ioctl+0x4bb/0x7b0 [dm_mod]
[<0>] dm_ctl_ioctl+0x11/0x20 [dm_mod]
[<0>] vfs_ioctl+0x21/0x60
[<0>] __x64_sys_ioctl+0xb9/0xe0
[<0>] do_syscall_64+0xc6/0x230
[<0>] entry_SYSCALL_64_after_hwframe+0x6c/0x74
```

Meanwhile mddev->recovery is:

```
MD_RECOVERY_RUNNING |
MD_RECOVERY_INTR |
MD_RECOVERY_RESHAPE |
MD_RECOVERY_FROZEN
```

Fix this problem by remove the code to register sync\_thread directly from raid10 and raid5. And let md\_check\_recovery() to register sync\_thread.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26756>

**[CVE-2024-26757] kernel: md: Don't ignore read-only array in md\_check\_recovery() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

md: Don't ignore read-only array in md\_check\_recovery()

Usually if the array is not read-write, md\_check\_recovery() won't register new sync\_thread in the first place. And if the array is read-write and sync\_thread is registered, md\_set\_readonly() will unregister sync\_thread before setting the array read-only. md/raid follow this behavior hence there is no problem.

After commit f52f5c71f3d4 ("md: fix stopping sync thread"), following hang can be triggered by test shell/integrity-caching.sh:

1) array is read-only. dm-raid update super block:

```
rs_update_sbs
```

```
ro = mddev->ro
```

```
mddev->ro = 0
```

-> set array read-write  
md\_update\_sb

2) register new sync thread concurrently.

3) dm-raid set array back to read-only:

rs\_update\_sbs  
mddev->ro = ro

4) stop the array:

raid\_dtr  
md\_stop  
stop\_sync\_thread  
set\_bit(MD\_RECOVERY\_INTR, &mddev->recovery);  
md\_wakeup\_thread\_directly(mddev->sync\_thread);  
wait\_event(..., !test\_bit(MD\_RECOVERY\_RUNNING, &mddev->recovery))

5) sync thread done:

md\_do\_sync  
set\_bit(MD\_RECOVERY\_DONE, &mddev->recovery);  
md\_wakeup\_thread(mddev->thread);

6) daemon thread can't unregister sync thread:

md\_check\_recovery  
if (!md\_is\_rdwr(mddev) &&  
    !test\_bit(MD\_RECOVERY\_NEEDED, &mddev->recovery))  
    return;  
-> -> MD\_RECOVERY\_RUNNING can't be cleared, hence step 4 hang;

The root cause is that dm-raid manipulate 'mddev->ro' by itself, however, dm-raid really should stop sync thread before setting the array read-only. Unfortunately, I need to read more code before I can refactor the handler of 'mddev->ro' in dm-raid, hence let's fix the problem the easy way for now to prevent dm-raid regression.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26757>

### **[CVE-2024-26758] kernel: md: Don't ignore suspended array in md\_check\_recovery() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

md: Don't ignore suspended array in md\_check\_recovery()

mddev\_suspend() never stop sync\_thread, hence it doesn't make sense to ignore suspended array in md\_check\_recovery(), which might cause sync\_thread can't be unregistered.

After commit f52f5c71f3d4 ("md: fix stopping sync thread"), following

hang can be triggered by test shell/integrity-caching.sh:

1) suspend the array:

```
raid_postsuspend
mddev_suspend
```

2) stop the array:

```
raid_dtr
md_stop
__md_stop_writes
stop_sync_thread
set_bit(MD_RECOVERY_INTR, &mddev->recovery);
md_wakeup_thread_directly(mddev->sync_thread);
wait_event(..., !test_bit(MD_RECOVERY_RUNNING, &mddev->recovery))
```

3) sync thread done:

```
md_do_sync
set_bit(MD_RECOVERY_DONE, &mddev->recovery);
md_wakeup_thread(mddev->thread);
```

4) daemon thread can't unregister sync thread:

```
md_check_recovery
if (mddev->suspended)
    return; -> return directly
md_read_sync_thread
clear_bit(MD_RECOVERY_RUNNING, &mddev->recovery);
-> MD_RECOVERY_RUNNING can't be cleared, hence step 2 hang;
```

This problem is not just related to dm-raid, fix it by ignoring suspended array in md\_check\_recovery(). And follow up patches will improve dm-raid better to frozen sync thread during suspend.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26758>

### **[CVE-2024-26767] kernel: drm/amd/display: fixed integer types and null check locations (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: fixed integer types and null check locations

[why]:

issues fixed:

- comparison with wider integer type in loop condition which can cause infinite loops
- pointer dereference before null check

More Info: <https://avd.aquasec.com/nvd/cve-2024-26767>

**[CVE-2024-26768] kernel: LoongArch: Change acpi\_core\_pic[NR\_CPUS] to acpi\_core\_pic[MAX\_CORE\_PIC] (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

LoongArch: Change acpi\_core\_pic[NR\_CPUS] to acpi\_core\_pic[MAX\_CORE\_PIC]

With default config, the value of NR\_CPUS is 64. When HW platform has more than 64 cpus, system will crash on these platforms. MAX\_CORE\_PIC is the maximum cpu number in MADT table (max physical number) which can exceed the supported maximum cpu number (NR\_CPUS, max logical number), but kernel should not crash. Kernel should boot cpus with NR\_CPUS, let the remainder cpus stay in BIOS.

The potential crash reason is that the array acpi\_core\_pic[NR\_CPUS] can be overflowed when parsing MADT table, and it is obvious that CORE\_PIC should be corresponding to physical core rather than logical core, so it is better to define the array as acpi\_core\_pic[MAX\_CORE\_PIC].

With the patch, system can boot up 64 vcpus with qemu parameter -smp 128, otherwise system will crash with the following message.

```
[ 0.000000] CPU 0 Unable to handle kernel paging request at virtual address 0000420000004259, era ==
90000000037a5f0c, ra == 90000000037a46ec
[ 0.000000] Oops[#1]:
[ 0.000000] CPU: 0 PID: 0 Comm: swapper Not tainted 6.8.0-rc2+ #192
[ 0.000000] Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 2/2/2022
[ 0.000000] pc 90000000037a5f0c ra 90000000037a46ec tp 9000000003c90000 sp 9000000003c93d60
[ 0.000000] a0 0000000000000019 a1 9000000003d93bc0 a2 0000000000000000 a3 9000000003c93bd8
[ 0.000000] a4 9000000003c93a74 a5 90000000083c93a67 a6 9000000003c938f0 a7 0000000000000005
[ 0.000000] t0 0000420000004201 t1 0000000000000000 t2 0000000000000001 t3 0000000000000001
[ 0.000000] t4 0000000000000003 t5 0000000000000000 t6 0000000000000030 t7 0000000000000063
[ 0.000000] t8 0000000000000014 u0 ffffffff s9 0000000000000000 s0 9000000003caee98
[ 0.000000] s1 90000000041b0480 s2 9000000003c93da0 s3 9000000003c93d98 s4 9000000003c93d90
[ 0.000000] s5 9000000003caa000 s6 000000000a7fd000 s7 000000000f556b60 s8 000000000e0a4330
[ 0.000000] ra: 90000000037a46ec platform_init+0x214/0x250
[ 0.000000] ERA: 90000000037a5f0c efi_runtime_init+0x30/0x94
[ 0.000000] CRMD: 000000b0 (PLV0 -IE -DA +PG DACF=CC DACM=CC -WE)
[ 0.000000] PRMD: 00000000 (PPLV0 -PIE -PWE)
[ 0.000000] EUEN: 00000000 (-FPE -SXE -ASXE -BTE)
[ 0.000000] ECFG: 00070800 (LIE=11 VS=7)
[ 0.000000] ESTAT: 00010000 [PIL] (IS= ECode=1 EsubCode=0)
[ 0.000000] BADV: 0000420000004259
[ 0.000000] PRID: 0014c010 (Loongson-64bit, Loongson-3A5000)
[ 0.000000] Modules linked in:
[ 0.000000] Process swapper (pid: 0, threadinfo=(____ptrval____), task=(____ptrval____))
[ 0.000000] Stack : 9000000003c93a14 9000000003800898 90000000041844f8 90000000037a46ec
[ 0.000000] 000000000a7fd000 0000000008290000 0000000000000000 0000000000000000
[ 0.000000] 0000000000000000 0000000000000000 00000000019d8000 000000000f556b60
```

```
[ 0.000000] 000000000a7fd000 000000000f556b08 9000000003ca7700 9000000003800000
[ 0.000000] 9000000003c93e50 9000000003800898 9000000003800108 90000000037a484c
[ 0.000000] 000000000e0a4330 000000000f556b60 000000000a7fd000 000000000f556b08
[ 0.000000] 9000000003ca7700 9000000004184000 0000000002000000 000000000e02b018
[ 0.000000] 000000000a7fd000 90000000037a0790 9000000003800108 0000000000000000
[ 0.000000] 0000000000000000 000000000e0a4330 000000000f556b60 000000000a7fd000
[ 0.000000] 000000000f556b08 000000000eaae298 000000000eaa5040 0000000002000000
[ 0.000000] ...
[ 0.000000] Call Trace:
[ 0.000000] [<90000000037a5f0c>] efi_runtime_init+0x30/0x94
[ 0.000000] [<90000000037a46ec>] platform_init+0x214/0x250
[ 0.000000] [<90000000037a484c>] setup_arch+0x124/0x45c
[ 0.000000] [<90000000037a0790>] start_kernel+0x90/0x670
[ 0.000000] [<900000000378b0d8>] kernel_entry+0xd8/0xdc
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26768>

## **[CVE-2024-26783] kernel: mm/vmscan: fix a bug calling wakeup\_kswapd() with a wrong zone index (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/vmscan: fix a bug calling wakeup\_kswapd() with a wrong zone index

With numa balancing on, when a numa system is running where a numa node doesn't have its local memory so it has no managed zones, the following oops has been observed. It's because wakeup\_kswapd() is called with a wrong zone index, -1. Fixed it by checking the index before calling wakeup\_kswapd().

```
> BUG: unable to handle page fault for address: 00000000000033f3
> #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: 2 PID: 895 Comm: masim Not tainted 6.6.0-dirty #255
> Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
> rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014
> RIP: 0010:wakeup_kswapd (./linux/mm/vmscan.c:7812)
> Code: (omitted)
> RSP: 0000:ffffc90004257d58 EFLAGS: 00010286
> RAX: ffffffff8883fff0480 RBX: ffff88883fff0480 RCX: 0000000000000003
> RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff88883fff0480
> RBP: ffffffff8883fff0480 R08: ff0003ffffffff R09: ffffffff8883fff0480
> R10: ffff888106c95540 R11: 0000000055555554 R12: 0000000000000003
> R13: 0000000000000000 R14: 0000000000000000 R15: ffff88883fff0940
> FS: 00007fc4b8124740(0000) GS:ffff888827c00000(0000) knlGS:0000000000000000
> CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
> CR2: 00000000000033f3 CR3: 000000026cc08004 CR4: 0000000000770ee0
```



```
> DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
> DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
> PKRU: 55555554
> Call Trace:
> <TASK>
> ? __die
> ? page_fault_oops
> ? __pte_offset_map_lock
> ? exc_page_fault
> ? asm_exc_page_fault
> ? wakeup_kswapd
> migrate_misplaced_page
> __handle_mm_fault
> handle_mm_fault
> do_user_addr_fault
> exc_page_fault
> asm_exc_page_fault
> RIP: 0033:0x55b897ba0808
> Code: (omitted)
> RSP: 002b:00007ffeefa821a0 EFLAGS: 00010287
> RAX: 000055b89983acd0 RBX: 00007ffeefa823f8 RCX: 000055b89983acd0
> RDX: 00007fc2f8122010 RSI: 0000000000020000 RDI: 000055b89983acd0
> RBP: 00007ffeefa821a0 R08: 0000000000000037 R09: 0000000000000075
> R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000000
> R13: 00007ffeefa82410 R14: 000055b897ba5dd8 R15: 00007fc4b8340000
> </TASK>
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26783>

### **[CVE-2024-26799] kernel: ASoC: qcom: Fix uninitialized pointer dmactl (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ASoC: qcom: Fix uninitialized pointer dmactl

In the case where `__lpass_get_dmactl_handle` is called and the driver id `dai_id` is invalid the pointer `dmactl` is not being assigned a value, and `dmactl` contains a garbage value since it has not been initialized and so the null check may not work. Fix this to initialize `dmactl` to `NULL`. One could argue that modern compilers will set this to zero, but it is useful to keep this initialized as per the same way in functions `__lpass_platform_codec_intf_init` and `lpass_cdc_dma_daiops_hw_params`.

Cleans up clang scan build warning:

sound/soc/qcom/lpass-cdc-dma.c:275:7: warning: Branch condition evaluates to a garbage value [core.uninitialized.Branch]

More Info: <https://avd.aquasec.com/nvd/cve-2024-26799>

### **[CVE-2024-26807] kernel: spi: cadence-qspi: fix pointer reference in runtime PM hooks (Severity:**

## **MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Both cadence-quadspi ->runtime\_suspend() and ->runtime\_resume() implementations start with:

```
struct cqspi_st *cqspi = dev_get_drvdata(dev);  
struct spi_controller *host = dev_get_drvdata(dev);
```

This obviously cannot be correct, unless "struct cqspi\_st" is the first member of "struct spi\_controller", or the other way around, but it is not the case. "struct spi\_controller" is allocated by devm\_spi\_alloc\_host(), which allocates an extra amount of memory for private data, used to store "struct cqspi\_st".

The ->probe() function of the cadence-quadspi driver then sets the device drvdata to store the address of the "struct cqspi\_st" structure. Therefore:

```
struct cqspi_st *cqspi = dev_get_drvdata(dev);
```

is correct, but:

```
struct spi_controller *host = dev_get_drvdata(dev);
```

is not, as it makes "host" point not to a "struct spi\_controller" but to the same "struct cqspi\_st" structure as above.

This obviously leads to bad things (memory corruption, kernel crashes) directly during ->probe(), as ->probe() enables the device using PM runtime, leading the ->runtime\_resume() hook being called, which in turns calls spi\_controller\_resume() with the wrong pointer.

This has at least been reported [0] to cause a kernel crash, but the exact behavior will depend on the memory contents.

[0] <https://lore.kernel.org/all/20240226121803.5a7r5wkpbbowcxgx@dhruva/>

This issue potentially affects all platforms that are currently using the cadence-quadspi driver.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26807>

## **[CVE-2024-26822] kernel: smb: client: set correct id, uid and cruid for multiuser automounts (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: set correct id, uid and cruid for multiuser automounts

When uid, gid and cruid are not specified, we need to dynamically set them into the filesystem context used for automounting otherwise they'll end up reusing the values from the parent mount.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26822>

## **[CVE-2024-26841] kernel: LoongArch: Update cpu\_sibling\_map when disabling nonboot CPUs (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

LoongArch: Update cpu\_sibling\_map when disabling nonboot CPUs

Update cpu\_sibling\_map when disabling nonboot CPUs by defining & calling clear\_cpu\_sibling\_map(), otherwise we get such errors on SMT systems:

jump label: negative count!

WARNING: CPU: 6 PID: 45 at kernel/jump\_label.c:263 \_\_static\_key\_slow\_dec\_cpuslocked+0xec/0x100

CPU: 6 PID: 45 Comm: cpuhp/6 Not tainted 6.8.0-rc5+ #1340

pc 90000000004c302c ra 90000000004c302c tp 90000001005bc000 sp 90000001005bfd20  
a0 000000000000001b a1 900000000224c278 a2 90000001005bfb58 a3 900000000224c280  
a4 900000000224c278 a5 90000001005bfb50 a6 0000000000000001 a7 0000000000000001  
t0 ce87a4763eb5234a t1 ce87a4763eb5234a t2 0000000000000000 t3 0000000000000000  
t4 0000000000000006 t5 0000000000000000 t6 0000000000000064 t7 0000000000001964  
t8 000000000009ebf6 u0 9000000001f2a068 s9 0000000000000000 s0 900000000246a2d8  
s1 ffffffff s2 ffffffff s3 90000000021518c0 s4 0000000000000040  
s5 9000000002151058 s6 9000000009828e40 s7 00000000000000b4 s8 0000000000000006

ra: 90000000004c302c \_\_static\_key\_slow\_dec\_cpuslocked+0xec/0x100

ERA: 90000000004c302c \_\_static\_key\_slow\_dec\_cpuslocked+0xec/0x100

CRMD: 000000b0 (PLV0 -IE -DA +PG DACF=CC DACM=CC -WE)

PRMD: 00000004 (PPLV0 +PIE -PWE)

EUEN: 00000000 (-FPE -SXE -ASXE -BTE)

ECFG: 00071c1c (LIE=2-4,10-12 VS=7)

ESTAT: 000c0000 [BRK] (IS= ECode=12 EsubCode=0)

PRID: 0014d000 (Loongson-64bit, Loongson-3A6000-HV)

CPU: 6 PID: 45 Comm: cpuhp/6 Not tainted 6.8.0-rc5+ #1340

Stack : 0000000000000000 900000000203f258 900000000179afc8 90000001005bc000  
90000001005bf980 0000000000000000 90000001005bf988 9000000001fe0be0  
900000000224c280 900000000224c278 90000001005bf8c0 0000000000000001  
0000000000000001 ce87a4763eb5234a 0000000007f38000 90000001003f8cc0  
0000000000000000 0000000000000006 0000000000000000 4c206e6f73676e6f  
6f4c203a656d616e 000000000009ec99 0000000007f38000 0000000000000000  
900000000214b000 9000000001fe0be0 0000000000000004 0000000000000000  
0000000000000107 0000000000000009 ffffffffafdbabe 00000000000000b4

0000000000000006 90000000004c302c 9000000000224528 00005555939a0c7c  
00000000000000b0 0000000000000004 0000000000000000 0000000000071c1c

...

Call Trace:

[<9000000000224528>] show\_stack+0x48/0x1a0  
[<9000000000179afc8>] dump\_stack\_lvl+0x78/0xa0  
[<9000000000263ed0>] \_\_warn+0x90/0x1a0  
[<900000000017419b8>] report\_bug+0x1b8/0x280  
[<9000000000179c564>] do\_bp+0x264/0x420  
[<90000000004c302c>] \_\_static\_key\_slow\_dec\_cpuslocked+0xec/0x100  
[<90000000002b4d7c>] sched\_cpu\_deactivate+0x2fc/0x300  
[<9000000000266498>] cpuhp\_invoke\_callback+0x178/0x8a0  
[<9000000000267f70>] cpuhp\_thread\_fun+0xf0/0x240  
[<90000000002a117c>] smpboot\_thread\_fn+0x1dc/0x2e0  
[<900000000029a720>] kthread+0x140/0x160  
[<9000000000222288>] ret\_from\_kernel\_thread+0xc/0xa4

More Info: <https://avd.aquasec.com/nvd/cve-2024-26841>

### [CVE-2024-26842] kernel: scsi: ufs: core: Fix shift issue in ufshcd\_clear\_cmd() (Severity: MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: ufs: core: Fix shift issue in ufshcd\_clear\_cmd()

When task\_tag >= 32 (in MCQ mode) and sizeof(unsigned int) == 4, 1U << task\_tag will out of bounds for a u32 mask. Fix this up to prevent SHIFT\_ISSUE (bitwise shifts that are out of bounds for their data type).

[name:debug\_monitors&]Unexpected kernel BRK exception at EL1  
[name:traps&]Internal error: BRK handler: 00000000f2005514 [#1] PREEMPT SMP  
[name:mediatek\_cpufreq\_hw&]cpufreq stop DVFS log done  
[name:mrddump&]Kernel Offset: 0x1ba5800000 from 0xfffffc0080000000  
[name:mrddump&]PHYS\_OFFSET: 0x80000000  
[name:mrddump&]pstate: 22400005 (nzCv daif +PAN -UAO)  
[name:mrddump&]pc : [0xfffffdba52bb2c] ufshcd\_clear\_cmd+0x280/0x288  
[name:mrddump&]lr : [0xfffffdba52a774] ufshcd\_wait\_for\_dev\_cmd+0x3e4/0x82c  
[name:mrddump&]sp : fffffc0081471b0  
<snip>

Workqueue: ufs\_eh\_wq\_0 ufshcd\_err\_handler

Call trace:

dump\_backtrace+0xf8/0x144  
show\_stack+0x18/0x24  
dump\_stack\_lvl+0x78/0x9c  
dump\_stack+0x18/0x44  
mrddump\_common\_die+0x254/0x480 [mrddump]  
ipanic\_die+0x20/0x30 [mrddump]  
notify\_die+0x15c/0x204  
die+0x10c/0x5f8  
arm64\_notify\_die+0x74/0x13c

do\_debug\_exception+0x164/0x26c  
el1\_dbg+0x64/0x80  
el1h\_64\_sync\_handler+0x3c/0x90  
el1h\_64\_sync+0x68/0x6c  
ufshcd\_clear\_cmd+0x280/0x288  
ufshcd\_wait\_for\_dev\_cmd+0x3e4/0x82c  
ufshcd\_exec\_dev\_cmd+0x5bc/0x9ac  
ufshcd\_verify\_dev\_init+0x84/0x1c8  
ufshcd\_probe\_hba+0x724/0x1ce0  
ufshcd\_host\_reset\_and\_restore+0x260/0x574  
ufshcd\_reset\_and\_restore+0x138/0xbd0  
ufshcd\_err\_handler+0x1218/0x2f28  
process\_one\_work+0x5fc/0x1140  
worker\_thread+0x7d8/0xe20  
kthread+0x25c/0x468  
ret\_from\_fork+0x10/0x20

More Info: <https://avd.aquasec.com/nvd/cve-2024-26842>

### **[CVE-2024-26866] kernel: spi: lpspi: Avoid potential use-after-free in probe() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

spi: lpspi: Avoid potential use-after-free in probe()

fsl\_lpspi\_probe() is allocating/disposing memory manually with spi\_alloc\_host()/spi\_alloc\_target(), but uses devm\_spi\_register\_controller(). In case of error after the latter call the memory will be explicitly freed in the probe function by spi\_controller\_put() call, but used afterwards by "devm" management outside probe() (spi\_unregister\_controller() <- devm\_spi\_unregister() below).

Unable to handle kernel NULL pointer dereference at virtual address 0000000000000070

...

Call trace:

kernfs\_find\_ns  
kernfs\_find\_and\_get\_ns  
sysfs\_remove\_group  
sysfs\_remove\_groups  
device\_remove\_attrs  
device\_del  
spi\_unregister\_controller  
devm\_spi\_unregister  
release\_nodes  
devres\_release\_all  
really\_probe  
driver\_probe\_device  
\_\_device\_attach\_driver  
bus\_for\_each\_drv  
\_\_device\_attach

device\_initial\_probe  
bus\_probe\_device  
deferred\_probe\_work\_func  
process\_one\_work  
worker\_thread  
kthread  
ret\_from\_fork

More Info: <https://avd.aquasec.com/nvd/cve-2024-26866>

### **[CVE-2024-26869] kernel: f2fs: fix to truncate meta inode pages forcely (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to truncate meta inode pages forcely

Below race case can cause data corruption:

Thread A   GC thread

- gc\_data\_segment
- ra\_data\_block
- locked meta\_inode page

- f2fs\_inplace\_write\_data

- invalidate\_mapping\_pages

: fail to invalidate meta\_inode page  
due to lock failure or dirty|writeback  
status

- f2fs\_submit\_page\_bio

: write last dirty data to old blkaddr

- move\_data\_block
- load old data from meta\_inode page
- f2fs\_submit\_page\_write

: write old data to new blkaddr

Because invalidate\_mapping\_pages() will skip invalidating page which has unclear status including locked, dirty, writeback and so on, so we need to use truncate\_inode\_pages\_range() instead of invalidate\_mapping\_pages() to make sure meta\_inode page will be dropped.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26869>

### **[CVE-2024-26876] kernel: drm/bridge: adv7511: fix crash on irq during probe (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/bridge: adv7511: fix crash on irq during probe

Moved IRQ registration down to end of adv7511\_probe().

If an IRQ already is pending during adv7511\_probe  
(before adv7511\_cec\_init) then cec\_received\_msg\_ts  
could crash using uninitialized data:

Unable to handle kernel read from unreadable memory at virtual address 00000000000003d5

Internal error: Oops: 96000004 [#1] PREEMPT\_RT SMP

Call trace:

```
cec_received_msg_ts+0x48/0x990 [cec]
adv7511_cec_irq_process+0x1cc/0x308 [adv7511]
adv7511_irq_process+0xd8/0x120 [adv7511]
adv7511_irq_handler+0x1c/0x30 [adv7511]
irq_thread_fn+0x30/0xa0
irq_thread+0x14c/0x238
kthread+0x190/0x1a8
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26876>

### **[CVE-2024-26902] kernel: perf: RISCv: Fix panic on pmu overflow handler (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

perf: RISCv: Fix panic on pmu overflow handler

(1 << idx) of int is not desired when setting bits in unsigned long  
overflowed\_ctrs, use BIT() instead. This panic happens when running  
'perf record -e branches' on sophgo sg2042.

```
[ 273.311852] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000098
[ 273.320851] Oops [#1]
[ 273.323179] Modules linked in:
[ 273.326303] CPU: 0 PID: 1475 Comm: perf Not tainted 6.6.0-rc3+ #9
[ 273.332521] Hardware name: Sophgo Mango (DT)
[ 273.336878] epc : riscv_pmu_ctr_get_width_mask+0x8/0x62
[ 273.342291] ra : pmu_sbi_ovf_handler+0x2e0/0x34e
[ 273.347091] epc : ffffffff80aecd98 ra : ffffffff80aee056 sp : ffffffff6e36928b0
[ 273.354454] gp : ffffffff821f82d0 tp : ffffffff90c353200 t0 : 0000002ade4f9978
[ 273.361815] t1 : 0000000000504d55 t2 : ffffffff8016cd8c s0 : ffffffff6e3692a70
[ 273.369180] s1 : 0000000000000020 a0 : 0000000000000000 a1 : 00001a8e81800000
[ 273.376540] a2 : 0000003c00070198 a3 : 0000003c00db75a4 a4 : 0000000000000015
[ 273.383901] a5 : ffffffff7ff8804b0 a6 : 0000000000000015 a7 : 000000000000002a
[ 273.391327] s2 : 000000000000ffff s3 : 0000000000000000 s4 : ffffffff7ff8803b0
[ 273.398773] s5 : 0000000000504d55 s6 : ffffffff905069800 s7 : ffffffff821fe210
[ 273.406139] s8 : 000000007fffffff s9 : ffffffff7ff8803b0 s10: ffffffff903f29098
[ 273.413660] s11: 0000000080000000 t3 : 0000000000000003 t4 : ffffffff8017a0ca
[ 273.421022] t5 : ffffffff8023cfc2 t6 : ffffffff9040780e8
[ 273.426437] status: 0000000200000100 badaddr: 0000000000000098 cause: 000000000000000d
[ 273.434512] [<fffffff80aecd98>] riscv_pmu_ctr_get_width_mask+0x8/0x62
```

```
[ 273.441169] [<ffffff80076bd8>] handle_percpu_devid_irq+0x98/0x1ee
[ 273.447562] [<ffffff80071158>] generic_handle_domain_irq+0x28/0x36
[ 273.454151] [<ffffff8047a99a>] riscv_intc_irq+0x36/0x4e
[ 273.459659] [<ffffff80c944de>] handle_riscv_irq+0x4a/0x74
[ 273.465442] [<ffffff80c94c48>] do_irq+0x62/0x92
[ 273.470360] Code: 0420 60a2 6402 5529 0141 8082 0013 0000 0013 0000 (6d5c) b783
[ 273.477921] ---[ end trace 0000000000000000 ]---
[ 273.482630] Kernel panic - not syncing: Fatal exception in interrupt
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-26902>

### **[CVE-2024-26914] kernel: drm/amd/display: fix incorrect mpc\_combine array size (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: fix incorrect mpc\_combine array size

[why]

MAX\_SURFACES is per stream, while MAX\_PLANES is per asic. The mpc\_combine is an array that records all the planes per asic. Therefore MAX\_PLANES should be used as the array size. Using MAX\_SURFACES causes array overflow when there are more than 3 planes.

[how]

Use the MAX\_PLANES for the mpc\_combine array size.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26914>

### **[CVE-2024-26947] kernel: ARM: 9359/1: flush: check if the folio is reserved for no-mapping addresses (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ARM: 9359/1: flush: check if the folio is reserved for no-mapping addresses

Since commit a4d5613c4dc6 ("arm: extend pfn\_valid to take into account freed memory map alignment") changes the semantics of pfn\_valid() to check presence of the memory map for a PFN. A valid page for an address which is reserved but not mapped by the kernel[1], the system crashed during some uio test with the following memory layout:

```
node 0: [mem 0x00000000c0a00000-0x00000000cc8fffff]
node 0: [mem 0x00000000d0000000-0x00000000da1fffff]
the uio layout is i%4$0xc0900000, 0x100000
```

the crash backtrace like:



Unable to handle kernel paging request at virtual address bff00000  
[...]  
CPU: 1 PID: 465 Comm: startapp.bin Tainted: G O 5.10.0 #1  
Hardware name: Generic DT based system  
PC is at b15\_flush\_kern\_dcache\_area+0x24/0x3c  
LR is at \_\_sync\_icache\_dcache+0x6c/0x98  
[...]  
(b15\_flush\_kern\_dcache\_area) from (\_\_sync\_icache\_dcache+0x6c/0x98)  
(\_\_sync\_icache\_dcache) from (set\_pte\_at+0x28/0x54)  
(set\_pte\_at) from (remap\_pfn\_range+0x1a0/0x274)  
(remap\_pfn\_range) from (uio\_mmap+0x184/0x1b8 [uio])  
(uio\_mmap [uio]) from (\_\_mmap\_region+0x264/0x5f4)  
(\_\_mmap\_region) from (\_\_do\_mmap\_mm+0x3ec/0x440)  
(\_\_do\_mmap\_mm) from (do\_mmap+0x50/0x58)  
(do\_mmap) from (vm\_mmap\_pgoff+0xfc/0x188)  
(vm\_mmap\_pgoff) from (ksys\_mmap\_pgoff+0xac/0xc4)  
(ksys\_mmap\_pgoff) from (ret\_fast\_syscall+0x0/0x5c)  
Code: e0801001 e2423001 e1c00003 f57ff04f (ee070f3e)  
---[ end trace 09cf0734c3805d52 ]---  
Kernel panic - not syncing: Fatal exception

So check if PG\_reserved was set to solve this issue.

[1]: <https://lore.kernel.org/lkml/Zbtdue57RO0QScJM@linux.ibm.com/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-26947>

### **[CVE-2024-26948] kernel: drm/amd/display: Add a dc\_state NULL check in dc\_state\_release (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add a dc\_state NULL check in dc\_state\_release

[How]

Check wheather state is NULL before releasing it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26948>

### **[CVE-2024-26953] kernel: net: esp: fix bad handling of pages from page\_pool (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: esp: fix bad handling of pages from page\_pool

When the skb is reorganized during esp\_output (!esp->inline), the pages coming from the original skb fragments are supposed to be released back to the system through put\_page. But if the skb fragment pages are originating from a page\_pool, calling put\_page on them will trigger a page\_pool leak which will eventually result in a crash.

This leak can be easily observed when using CONFIG\_DEBUG\_VM and doing ipsec + gre (non offloaded) forwarding:

```
BUG: Bad page state in process ksoftirqd/16 pfn:1451b6
page:00000000de2b8d32 refcount:0 mapcount:0 mapping:0000000000000000 index:0x1451b6000 pfn:0x1451b6
flags: 0x2000000000000000(node=0|zone=2)
page_type: 0xffffffff()
raw: 020000000000000000 dead00000000000040 ffff88810d23c000 0000000000000000
raw: 000000001451b6000 000000000000000001 000000000ffffff 0000000000000000
page dumped because: page_pool leak
Modules linked in: ip_gre gre mlx5_ib mlx5_core xt_contrack xt_MASQUERADE nf_contrack_netlink nfnetlink
iptable_nat nf_nat xt_addrtype br_netfilter rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm
ib_ipoib iw_cm ib_cm ib_uverbs ib_core overlay zram zsmalloc fuse [last unloaded: mlx5_core]
CPU: 16 PID: 96 Comm: ksoftirqd/16 Not tainted 6.8.0-rc4+ #22
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org
04/01/2014
Call Trace:
<TASK>
dump_stack_lvl+0x36/0x50
bad_page+0x70/0xf0
free_unref_page_prepare+0x27a/0x460
free_unref_page+0x38/0x120
esp_ssg_unref.isra.0+0x15f/0x200
esp_output_tail+0x66d/0x780
esp_xmit+0x2c5/0x360
validate_xmit_xfrm+0x313/0x370
? validate_xmit_skb+0x1d/0x330
validate_xmit_skb_list+0x4c/0x70
sch_direct_xmit+0x23e/0x350
__dev_queue_xmit+0x337/0xba0
? nf_hook_slow+0x3f/0xd0
ip_finish_output2+0x25e/0x580
iptunnel_xmit+0x19b/0x240
ip_tunnel_xmit+0x5fb/0xb60
ipgre_xmit+0x14d/0x280 [ip_gre]
dev_hard_start_xmit+0xc3/0x1c0
__dev_queue_xmit+0x208/0xba0
? nf_hook_slow+0x3f/0xd0
ip_finish_output2+0x1ca/0x580
ip_sublist_rcv_finish+0x32/0x40
ip_sublist_rcv+0x1b2/0x1f0
? ip_rcv_finish_core.constprop.0+0x460/0x460
ip_list_rcv+0x103/0x130
__netif_receive_skb_list_core+0x181/0x1e0
netif_receive_skb_list_internal+0x1b3/0x2c0
napi_gro_receive+0xc8/0x200
gro_cell_poll+0x52/0x90
```

```
__napi_poll+0x25/0x1a0
net_rx_action+0x28e/0x300
__do_softirq+0xc3/0x276
? sort_range+0x20/0x20
run_ksoftirqd+0x1e/0x30
smpboot_thread_fn+0xa6/0x130
kthread+0xcd/0x100
? kthread_complete_and_exit+0x20/0x20
ret_from_fork+0x31/0x50
? kthread_complete_and_exit+0x20/0x20
ret_from_fork_asm+0x11/0x20
</TASK>
```

The suggested fix is to introduce a new wrapper (skb\_page\_unref) that covers page refcounting for page\_pool pages as well.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26953>

### **[CVE-2024-26962] kernel: dm-raid456, md/raid456: fix a deadlock for dm-raid456 while io concurrent with reshape (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

dm-raid456, md/raid456: fix a deadlock for dm-raid456 while io concurrent with reshape

For raid456, if reshape is still in progress, then IO across reshape position will wait for reshape to make progress. However, for dm-raid, in following cases reshape will never make progress hence IO will hang:

- 1) the array is read-only;
- 2) MD\_RECOVERY\_WAIT is set;
- 3) MD\_RECOVERY\_FROZEN is set;

After commit c467e97f079f ("md/raid6: use valid sector values to determine if an I/O should wait on the reshape") fix the problem that IO across reshape position doesn't wait for reshape, the dm-raid test shell/lvconvert-raid-reshape.sh start to hang:

```
[root@fedora ~]# cat /proc/979/stack
[<0>] wait_woken+0x7d/0x90
[<0>] raid5_make_request+0x929/0x1d70 [raid456]
[<0>] md_handle_request+0xc2/0x3b0 [md_mod]
[<0>] raid_map+0x2c/0x50 [dm_raid]
[<0>] __map_bio+0x251/0x380 [dm_mod]
[<0>] dm_submit_bio+0x1f0/0x760 [dm_mod]
[<0>] __submit_bio+0xc2/0x1c0
[<0>] submit_bio_noacct_nocheck+0x17f/0x450
[<0>] submit_bio_noacct+0x2bc/0x780
[<0>] submit_bio+0x70/0xc0
```

```
[<0>] mpage_readahead+0x169/0x1f0
[<0>] blkdev_readahead+0x18/0x30
[<0>] read_pages+0x7c/0x3b0
[<0>] page_cache_ra_unbounded+0x1ab/0x280
[<0>] force_page_cache_ra+0x9e/0x130
[<0>] page_cache_sync_ra+0x3b/0x110
[<0>] filemap_get_pages+0x143/0xa30
[<0>] filemap_read+0xdc/0x4b0
[<0>] blkdev_read_iter+0x75/0x200
[<0>] vfs_read+0x272/0x460
[<0>] ksys_read+0x7a/0x170
[<0>] __x64_sys_read+0x1c/0x30
[<0>] do_syscall_64+0xc6/0x230
[<0>] entry_SYSCALL_64_after_hwframe+0x6c/0x74
```

This is because reshape can't make progress.

For md/raid, the problem doesn't exist because register new sync\_thread doesn't rely on the IO to be done any more:

- 1) If array is read-only, it can switch to read-write by ioctl/sysfs;
- 2) md/raid never set MD\_RECOVERY\_WAIT;
- 3) If MD\_RECOVERY\_FROZEN is set, mddev\_suspend() doesn't hold 'reconfig\_mutex', hence it can be cleared and reshape can continue by sysfs api 'sync\_action'.

However, I'm not sure yet how to avoid the problem in dm-raid yet. This patch on the one hand make sure raid\_message() can't change sync\_thread() through raid\_message() after presuspend(), on the other hand detect the above 3 cases before wait for IO do be done in dm\_suspend(), and let dm-raid requeue those IO.

More Info: <https://avd.aquasec.com/nvd/cve-2024-26962>

### **[CVE-2024-27005] kernel: interconnect: Don't access req\_list while it's being manipulated (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

interconnect: Don't access req\_list while it's being manipulated

The icc\_lock mutex was split into separate icc\_lock and icc\_bw\_lock mutexes in [1] to avoid lockdep splats. However, this didn't adequately protect access to icc\_node::req\_list.

The icc\_set\_bw() function will eventually iterate over req\_list while only holding icc\_bw\_lock, but req\_list can be modified while only holding icc\_lock. This causes races between icc\_set\_bw(), of\_icc\_get(), and icc\_put().

Example A:

```
CPU0          CPU1
----          ----
icc_set_bw(path_a)
mutex_lock(&icc_bw_lock);
                                icc_put(path_b)
                                mutex_lock(&icc_lock);
aggregate_requests()
hlist_for_each_entry(r, ...
                                hlist_del(...)
                                <r = invalid pointer>
```

Example B:

```
CPU0          CPU1
----          ----
icc_set_bw(path_a)
mutex_lock(&icc_bw_lock);
                                path_b = of_icc_get()
                                of_icc_get_by_index()
                                mutex_lock(&icc_lock);
                                path_find()
                                path_init()
aggregate_requests()
hlist_for_each_entry(r, ...
                                hlist_add_head(...)
                                <r = invalid pointer>
```

Fix this by ensuring `icc_bw_lock` is always held before manipulating `icc_node::req_list`. The additional places `icc_bw_lock` is held don't perform any memory allocations, so we should still be safe from the original lockdep splats that motivated the separate locks.

[1] commit af42269c3523 ("interconnect: Fix locking for runpm vs reclaim")

More Info: <https://avd.aquasec.com/nvd/cve-2024-27005>

### **[CVE-2024-27010] kernel: net/sched: Fix mirrored deadlock on device recursion (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/sched: Fix mirrored deadlock on device recursion

When the mirrored action is used on a classful egress qdisc and a packet is mirrored or redirected to self we hit a qdisc lock deadlock.

See trace below.

[..... other info removed for brevity.....]

```
[ 82.890906]
[ 82.890906] =====
[ 82.890906] WARNING: possible recursive locking detected
[ 82.890906] 6.8.0-05205-g77fadd89fe2d-dirty #213 Tainted: G      W
[ 82.890906] -----
[ 82.890906] ping/418 is trying to acquire lock:
[ 82.890906] ffff888006994110 (&sch->q.lock){+.-}-{3:3}, at:
__dev_queue_xmit+0x1778/0x3550
[ 82.890906]
[ 82.890906] but task is already holding lock:
[ 82.890906] ffff888006994110 (&sch->q.lock){+.-}-{3:3}, at:
__dev_queue_xmit+0x1778/0x3550
[ 82.890906]
[ 82.890906] other info that might help us debug this:
[ 82.890906] Possible unsafe locking scenario:
[ 82.890906]
[ 82.890906]      CPU0
[ 82.890906]      ----
[ 82.890906]  lock(&sch->q.lock);
[ 82.890906]  lock(&sch->q.lock);
[ 82.890906]
[ 82.890906] *** DEADLOCK ***
[ 82.890906]
[..... other info removed for brevity....]
```

Example setup (eth0->eth0) to recreate

```
tc qdisc add dev eth0 root handle 1: htb default 30
tc filter add dev eth0 handle 1: protocol ip prio 2 matchall \
    action mirred egress redirect dev eth0
```

Another example(eth0->eth1->eth0) to recreate

```
tc qdisc add dev eth0 root handle 1: htb default 30
tc filter add dev eth0 handle 1: protocol ip prio 2 matchall \
    action mirred egress redirect dev eth1
```

```
tc qdisc add dev eth1 root handle 1: htb default 30
tc filter add dev eth1 handle 1: protocol ip prio 2 matchall \
    action mirred egress redirect dev eth0
```

We fix this by adding an owner field (CPU id) to struct Qdisc set after root qdisc is entered. When the softirq enters it a second time, if the qdisc owner is the same CPU, the packet is dropped to break the loop.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27010>

### **[CVE-2024-27011] kernel: netfilter: nf\_tables: fix memleak in map from abort path (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

netfilter: nf\_tables: fix memleak in map from abort path

The delete set command does not rely on the transaction object for element removal, therefore, a combination of delete element + delete set from the abort path could result in restoring twice the refcount of the mapping.

Check for inactive element in the next generation for the delete element command in the abort path, skip restoring state if next generation bit has been already cleared. This is similar to the activate logic using the set walk iterator.

```
[ 6170.286929] -----[ cut here ]-----
[ 6170.286939] WARNING: CPU: 6 PID: 790302 at net/netfilter/nf_tables_api.c:2086
nf_tables_chain_destroy+0x1f7/0x220 [nf_tables]
[ 6170.287071] Modules linked in: [...]
[ 6170.287633] CPU: 6 PID: 790302 Comm: kworker/6:2 Not tainted 6.9.0-rc3+ #365
[ 6170.287768] RIP: 0010:nf_tables_chain_destroy+0x1f7/0x220 [nf_tables]
[ 6170.287886] Code: df 48 8d 7d 58 e8 69 2e 3b df 48 8b 7d 58 e8 80 1b 37 df 48 8d 7d 68 e8 57 2e 3b df 48 8b 7d 68
e8 6e 1b 37 df 48 89 ef eb c4 <0f> 0b 48 83 c4 08 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc 0f
[ 6170.287895] RSP: 0018:ffff888134b8fd08 EFLAGS: 00010202
[ 6170.287904] RAX: 0000000000000001 RBX: ffff888125bffb28 RCX: dffffc0000000000
[ 6170.287912] RDX: 0000000000000003 RSI: ffffffff8a20298ab RDI: ffff88811ebe4750
[ 6170.287919] RBP: ffff88811ebe4700 R08: ffff88838e812650 R09: fffffbfff0623a55
[ 6170.287926] R10: ffffffff8311d2af R11: 0000000000000001 R12: ffff888125bffb10
[ 6170.287933] R13: ffff888125bffb10 R14: dead000000000122 R15: dead000000000100
[ 6170.287940] FS: 0000000000000000(0000) GS:ffff888390b00000(0000) knlGS:0000000000000000
[ 6170.287948] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 6170.287955] CR2: 00007fd31fc00710 CR3: 0000000133f60004 CR4: 00000000001706f0
[ 6170.287962] Call Trace:
[ 6170.287967] <TASK>
[ 6170.287973] ? __warn+0x9f/0x1a0
[ 6170.287986] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables]
[ 6170.288092] ? report_bug+0x1b1/0x1e0
[ 6170.287986] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables]
[ 6170.288092] ? report_bug+0x1b1/0x1e0
[ 6170.288104] ? handle_bug+0x3c/0x70
[ 6170.288112] ? exc_invalid_op+0x17/0x40
[ 6170.288120] ? asm_exc_invalid_op+0x1a/0x20
[ 6170.288132] ? nf_tables_chain_destroy+0x2b/0x220 [nf_tables]
[ 6170.288243] ? nf_tables_chain_destroy+0x1f7/0x220 [nf_tables]
[ 6170.288366] ? nf_tables_chain_destroy+0x2b/0x220 [nf_tables]
[ 6170.288483] nf_tables_trans_destroy_work+0x588/0x590 [nf_tables]
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-27011>

### **[CVE-2024-27012] kernel: netfilter: nf\_tables: restore set elements when delete set fails (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

netfilter: nf\_tables: restore set elements when delete set fails

From abort path, nft\_mapelem\_activate() needs to restore refcounters to the original state. Currently, it uses the set->ops->walk() to iterate over these set elements. The existing set iterator skips inactive elements in the next generation, this does not work from the abort path to restore the original state since it has to skip active elements instead (not inactive ones).

This patch moves the check for inactive elements to the set iterator callback, then it reverses the logic for the .activate case which needs to skip active elements.

Toggle next generation bit for elements when delete set command is invoked and call nft\_clear() from .activate (abort) path to restore the next generation bit.

The splat below shows an object in mappings memleak:

```
[43929.457523] -----[ cut here ]-----
[43929.457532]   WARNING:   CPU:    0    PID:    1139    at    include/net/netfilter/nf_tables.h:1237
nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables]
[...]
[43929.458014] RIP: 0010:nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables]
[43929.458076] Code: 83 f8 01 77 ab 49 8d 7c 24 08 e8 37 5e d0 de 49 8b 6c 24 08 48 8d 7d 50 e8 e9 5c d0 de 8b 45
50 8d 50 ff 89 55 50 85 c0 75 86 <0f> 0b eb 82 0f 0b eb b3 0f 1f 40 00 90 90 90 90 90 90 90 90 90
[43929.458081] RSP: 0018:ffff888140f9f4b0 EFLAGS: 00010246
[43929.458086] RAX: 0000000000000000 RBX: ffff8881434f5288 RCX: dffffc0000000000
[43929.458090] RDX: 00000000ffffffff RSI: ffffffff26d28a7 RDI: ffff88810ecc9550
[43929.458093] RBP: ffff88810ecc9500 R08: 0000000000000001 R09: ffffed10281f3e8f
[43929.458096] R10: 0000000000000003 R11: ffff0000ffff0000 R12: ffff8881434f52a0
[43929.458100] R13: ffff888140f9f5f4 R14: ffff888151c7a800 R15: 0000000000000002
[43929.458103] FS: 00007f0c687c4740(0000) GS:ffff888390800000(0000) knlGS:0000000000000000
[43929.458107] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[43929.458111] CR2: 00007f58db5b008 CR3: 0000000123602005 CR4: 00000000001706f0
[43929.458114] Call Trace:
[43929.458118] <TASK>
[43929.458121] ? __warn+0x9f/0x1a0
[43929.458127] ? nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables]
[43929.458188] ? report_bug+0x1b1/0x1e0
[43929.458196] ? handle_bug+0x3c/0x70
[43929.458200] ? exc_invalid_op+0x17/0x40
[43929.458211] ? nft_setelem_data_deactivate+0xd7/0xf0 [nf_tables]
[43929.458271] ? nft_setelem_data_deactivate+0xe4/0xf0 [nf_tables]
[43929.458332] nft_mapelem_deactivate+0x24/0x30 [nf_tables]
[43929.458392] nft_rhash_walk+0xdd/0x180 [nf_tables]
[43929.458453] ? __pfx_nft_rhash_walk+0x10/0x10 [nf_tables]
[43929.458512] ? rb_insert_color+0x2e/0x280
[43929.458520] nft_map_deactivate+0xdc/0x1e0 [nf_tables]
[43929.458582] ? __pfx_nft_map_deactivate+0x10/0x10 [nf_tables]
[43929.458642] ? __pfx_nft_mapelem_deactivate+0x10/0x10 [nf_tables]
[43929.458701] ? __rcu_read_unlock+0x46/0x70
```



[43929.458709] nft\_delset+0xff/0x110 [nf\_tables]  
[43929.458769] nft\_flush\_table+0x16f/0x460 [nf\_tables]  
[43929.458830] nf\_tables\_deltab+0x501/0x580 [nf\_tables]

More Info: <https://avd.aquasec.com/nvd/cve-2024-27012>

**[CVE-2024-27041] kernel: drm/amd/display: fix NULL checks for adev->dm.dc in amdgpu\_dm\_fini() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: fix NULL checks for adev->dm.dc in amdgpu\_dm\_fini()

Since 'adev->dm.dc' in amdgpu\_dm\_fini() might turn out to be NULL before the call to dc\_enable\_dmub\_notifications(), check beforehand to ensure there will not be a possible NULL-ptr-deref there.

Also, since commit 1e88eb1b2c25 ("drm/amd/display: Drop CONFIG\_DRM\_AMD\_DC\_HDCP") there are two separate checks for NULL in 'adev->dm.dc' before dc\_deinit\_callbacks() and dc\_dmub\_srv\_destroy(). Clean up by combining them all under one 'if'.

Found by Linux Verification Center (linuxtesting.org) with static analysis tool SVACE.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27041>

**[CVE-2024-27056] kernel: wifi: iwlwifi: mvm: ensure offloading TID queue exists (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

wifi: iwlwifi: mvm: ensure offloading TID queue exists

The resume code path assumes that the TX queue for the offloading TID has been configured. At resume time it then tries to sync the write pointer as it may have been updated by the firmware.

In the unusual event that no packets have been send on TID 0, the queue will not have been allocated and this causes a crash. Fix this by ensuring the queue exist at suspend time.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27056>

**[CVE-2024-27057] kernel: ASoC: SOF: ipc4-pcm: Workaround for crashed firmware on system suspend (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ASoC: SOF: ipc4-pcm: Workaround for crashed firmware on system suspend

When the system is suspended while audio is active, the `sof_ipc4_pcm_hw_free()` is invoked to reset the pipelines since during suspend the DSP is turned off, streams will be re-started after resume.

If the firmware crashes during while audio is running (or when we reset the stream before suspend) then the `sof_ipc4_set_multi_pipeline_state()` will fail with IPC error and the state change is interrupted. This will cause misalignment between the kernel and firmware state on next DSP boot resulting errors returned by firmware for IPC messages, eventually failing the audio resume. On stream close the errors are ignored so the kernel state will be corrected on the next DSP boot, so the second boot after the DSP panic.

If `sof_ipc4_trigger_pipelines()` is called from `sof_ipc4_pcm_hw_free()` then state parameter is `SOF_IPC4_PIPE_RESET` and only in this case.

Treat a forced pipeline reset similarly to how we treat a `pcm_free` by ignoring error on state sending to allow the kernel's state to be consistent with the state the firmware will have after the next boot.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27057>

### **[CVE-2024-27062] kernel: nouveau: lock the client object tree. (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nouveau: lock the client object tree.

It appears the client object tree has no locking unless I've missed something else. Fix races around adding/removing client objects, mostly vram bar mappings.

4562.099306] general protection fault, probably for non-canonical address 0x6677ed422bceb80c: 0000 [#1] PREEMPT SMP PTI  
[ 4562.099314] CPU: 2 PID: 23171 Comm: deqp-vk Not tainted 6.8.0-rc6+ #27  
[ 4562.099324] Hardware name: Gigabyte Technology Co., Ltd. Z390 I AORUS PRO WIFI/Z390 I AORUS PRO WIFI-CF, BIOS F8 11/05/2021  
[ 4562.099330] RIP: 0010:nvkm\_object\_search+0x1d/0x70 [nouveau]  
[ 4562.099503] Code: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 66 0f 1f 00 0f 1f 44 00 00 48 89 f8 48 85 f6 74 39 48 8b 87 a0 00 00 00 48 85 c0 74 12 <48> 8b 48 f8 48 39 ce 73 15 48 8b 40 10 48 85 c0 75 ee 48 c7 c0 fe  
[ 4562.099506] RSP: 0000:ffffa94cc420bbf8 EFLAGS: 00010206  
[ 4562.099512] RAX: 6677ed422bceb814 RBX: ffff98108791f400 RCX: ffff9810f26b8f58

```
[ 4562.099517] RDX: 0000000000000000 RSI: ffff9810f26b9158 RDI: ffff98108791f400
[ 4562.099519] RBP: ffff9810f26b9158 R08: 0000000000000000 R09: 0000000000000000
[ 4562.099521] R10: ffffa94cc420bc48 R11: 0000000000000001 R12: ffff9810f02a7cc0
[ 4562.099526] R13: 0000000000000000 R14: 00000000000000ff R15: 0000000000000007
[ 4562.099528] FS: 00007f629c5017c0(0000) GS:ffff98142c700000(0000) knlGS:0000000000000000
[ 4562.099534] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 4562.099536] CR2: 00007f629a882000 CR3: 000000017019e004 CR4: 00000000003706f0
[ 4562.099541] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[ 4562.099542] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400
[ 4562.099544] Call Trace:
[ 4562.099555] <TASK>
[ 4562.099573] ? die_addr+0x36/0x90
[ 4562.099583] ? exc_general_protection+0x246/0x4a0
[ 4562.099593] ? asm_exc_general_protection+0x26/0x30
[ 4562.099600] ? nvkm_object_search+0x1d/0x70 [nouveau]
[ 4562.099730] nvkm_ioctl+0xa1/0x250 [nouveau]
[ 4562.099861] nvif_object_map_handle+0xc8/0x180 [nouveau]
[ 4562.099986] nouveau_ttm_io_mem_reserve+0x122/0x270 [nouveau]
[ 4562.100156] ? dma_resv_test_signaled+0x26/0xb0
[ 4562.100163] ttm_bo_vm_fault_reserved+0x97/0x3c0 [ttm]
[ 4562.100182] ? __mutex_unlock_slowpath+0x2a/0x270
[ 4562.100189] nouveau_ttm_fault+0x69/0xb0 [nouveau]
[ 4562.100356] __do_fault+0x32/0x150
[ 4562.100362] do_fault+0x7c/0x560
[ 4562.100369] __handle_mm_fault+0x800/0xc10
[ 4562.100382] handle_mm_fault+0x17c/0x3e0
[ 4562.100388] do_user_addr_fault+0x208/0x860
[ 4562.100395] exc_page_fault+0x7f/0x200
[ 4562.100402] asm_exc_page_fault+0x26/0x30
[ 4562.100412] RIP: 0033:0x9b9870
[ 4562.100419] Code: 85 a8 f7 ff ff 8b 8d 80 f7 ff ff 89 08 e9 18 f2 ff ff 0f 1f 84 00 00 00 00 00 44 89 32 e9 90 fa ff ff 0f 1f
84 00 00 00 00 00 <44> 89 32 e9 f8 f1 ff ff 0f 1f 84 00 00 00 00 00 66 44 89 32 e9 e7
[ 4562.100422] RSP: 002b:00007fff9ba2dc70 EFLAGS: 00010246
[ 4562.100426] RAX: 0000000000000004 RBX: 00000000dd65e10 RCX: 000000ffff000000
[ 4562.100428] RDX: 00007f629a882000 RSI: 00007f629a882000 RDI: 0000000000000066
[ 4562.100432] RBP: 00007fff9ba2e570 R08: 0000000000000000 R09: 0000000123ddf000
[ 4562.100434] R10: 0000000000000001 R11: 00000000000000246 R12: 000000007ffffff
[ 4562.100436] R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
[ 4562.100446] </TASK>
[ 4562.100448] Modules linked in: nf_contrack_netbios_ns nf_contrack_broadcast nft_fib_inet nft_fib_ipv4 nft_fib_ipv6
nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_contrack nf_defrag_ipv6
nf_defrag_ipv4 ip_set nf_tables libcrc32c nfnetlink cmac bnep sunrpc iwlmvm intel_rapl_msr intel_rapl_common
snd_sof_pci_intel_cnl x86_pkg_temp_thermal intel_powerclamp snd_sof_intel_hda_common mac80211 coretemp
snd_soc_acpi_intel_match kvm_intel snd_soc_acpi snd_soc_hdac_hda snd_sof_pci snd_sof_xtensa_dsp
snd_sof_intel_hda_mlink
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-27062>

### **[CVE-2024-27079] kernel: iommu/vt-d: Fix NULL domain on device release (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

iommu/vt-d: Fix NULL domain on device release

In the kdump kernel, the IOMMU operates in deferred\_attach mode. In this mode, info->domain may not yet be assigned by the time the release\_device function is called. It leads to the following crash in the crash kernel:

BUG: kernel NULL pointer dereference, address: 000000000000003c

...

RIP: 0010:do\_raw\_spin\_lock+0xa/0xa0

...

\_raw\_spin\_lock\_irqsave+0x1b/0x30

intel\_iommu\_release\_device+0x96/0x170

iommu\_deinit\_device+0x39/0xf0

\_\_iommu\_group\_remove\_device+0xa0/0xd0

iommu\_bus\_notifier+0x55/0xb0

notifier\_call\_chain+0x5a/0xd0

blocking\_notifier\_call\_chain+0x41/0x60

bus\_notify+0x34/0x50

device\_del+0x269/0x3d0

pci\_remove\_bus\_device+0x77/0x100

p2sb\_bar+0xae/0x1d0

...

i801\_probe+0x423/0x740

Use the release\_domain mechanism to fix it. The scalable mode context entry which is not part of release domain should be cleared in release\_device().

More Info: <https://avd.aquasec.com/nvd/cve-2024-27079>

### **[CVE-2024-27408] kernel: dmaengine: dw-edma: eDMA: Add sync read before starting the DMA transfer in remote setup (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

dmaengine: dw-edma: eDMA: Add sync read before starting the DMA transfer in remote setup

The Linked list element and pointer are not stored in the same memory as the eDMA controller register. If the doorbell register is toggled before the full write of the linked list a race condition error will occur.

In remote setup we can only use a readl to the memory to assure the full write has occurred.

More Info: <https://avd.aquasec.com/nvd/cve-2024-27408>

### **[CVE-2024-35784] kernel: btrfs: fix deadlock with fiemap and extent locking (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: fix deadlock with fiemap and extent locking

While working on the patchset to remove extent locking I got a lockdep splat with fiemap and pagefaulting with my new extent lock replacement lock.

This deadlock exists with our normal code, we just don't have lockdep annotations with the extent locking so we've never noticed it.

Since we're copying the fiemap extent to user space on every iteration we have the chance of pagefaulting. Because we hold the extent lock for the entire range we could mkwrite into a range in the file that we have mmap'ed. This would deadlock with the following stack trace

```
[<0>] lock_extent+0x28d/0x2f0
[<0>] btrfs_page_mkwrite+0x273/0x8a0
[<0>] do_page_mkwrite+0x50/0xb0
[<0>] do_fault+0xc1/0x7b0
[<0>] __handle_mm_fault+0x2fa/0x460
[<0>] handle_mm_fault+0xa4/0x330
[<0>] do_user_addr_fault+0x1f4/0x800
[<0>] exc_page_fault+0x7c/0x1e0
[<0>] asm_exc_page_fault+0x26/0x30
[<0>] rep_movs_alternative+0x33/0x70
[<0>] _copy_to_user+0x49/0x70
[<0>] fiemap_fill_next_extent+0xc8/0x120
[<0>] emit_fiemap_extent+0x4d/0xa0
[<0>] extent_fiemap+0x7f8/0xad0
[<0>] btrfs_fiemap+0x49/0x80
[<0>] __x64_sys_ioctl+0x3e1/0xb50
[<0>] do_syscall_64+0x94/0x1a0
[<0>] entry_SYSCALL_64_after_hwframe+0x6e/0x76
```

I wrote an fstest to reproduce this deadlock without my replacement lock and verified that the deadlock exists with our existing locking.

To fix this simply don't take the extent lock for the entire duration of the fiemap. This is safe in general because we keep track of where we are when we're searching the tree, so if an ordered extent updates in the middle of our fiemap call we'll still emit the correct extents because we know what offset we were on before.

The only place we maintain the lock is searching delalloc. Since the delalloc stuff can change during writeback we want to lock the extent range so we have a consistent view of delalloc at the time we're checking to see if we need to set the delalloc flag.

With this patch applied we no longer deadlock with my testcase.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35784>

**[CVE-2024-35790] kernel: usb: typec: altmodes/displayport: create sysfs nodes as driver's default device attribute group (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

usb: typec: altmodes/displayport: create sysfs nodes as driver's default device attribute group

The DisplayPort driver's sysfs nodes may be present to the userspace before typec\_altmode\_set\_drvdata() completes in dp\_altmode\_probe. This means that a sysfs read can trigger a NULL pointer error by deferencing dp->hpd in hpd\_show or dp->lock in pin\_assignment\_show, as dev\_get\_drvdata() returns NULL in those cases.

Remove manual sysfs node creation in favor of adding attribute group as default for devices bound to the driver. The ATTRIBUTE\_GROUPS() macro is not used here otherwise the path to the sysfs nodes is no longer compliant with the ABI.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35790>

**[CVE-2024-35794] kernel: dm-raid: really frozen sync\_thread during suspend (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

dm-raid: really frozen sync\_thread during suspend

- 1) commit f52f5c71f3d4 ("md: fix stopping sync thread") remove MD\_RECOVERY\_FROZEN from \_\_md\_stop\_writes() and doesn't realize that dm-raid relies on \_\_md\_stop\_writes() to frozen sync\_thread indirectly. Fix this problem by adding MD\_RECOVERY\_FROZEN in md\_stop\_writes(), and since stop\_sync\_thread() is only used for dm-raid in this case, also move stop\_sync\_thread() to md\_stop\_writes().
- 2) The flag MD\_RECOVERY\_FROZEN doesn't mean that sync thread is frozen, it only prevent new sync\_thread to start, and it can't stop the running sync thread; In order to frozen sync\_thread, after seting the flag, stop\_sync\_thread() should be used.
- 3) The flag MD\_RECOVERY\_FROZEN doesn't mean that writes are stopped, use it as condition for md\_stop\_writes() in raid\_postsuspend() doesn't look correct. Consider that reentrant stop\_sync\_thread() do nothing, always call md\_stop\_writes() in raid\_postsuspend().
- 4) raid\_message can set/clear the flag MD\_RECOVERY\_FROZEN at anytime,

and if MD\_RECOVERY\_FROZEN is cleared while the array is suspended, new sync\_thread can start unexpected. Fix this by disallow raid\_message() to change sync\_thread status during suspend.

Note that after commit f52f5c71f3d4 ("md: fix stopping sync thread"), the test shell/lvconvert-raid-reshape.sh start to hang in stop\_sync\_thread(), and with previous fixes, the test won't hang there anymore, however, the test will still fail and complain that ext4 is corrupted. And with this patch, the test won't hang due to stop\_sync\_thread() or fail due to ext4 is corrupted anymore. However, there is still a deadlock related to dm-raid456 that will be fixed in following patches.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35794>

### **[CVE-2024-35799] kernel: drm/amd/display: Prevent crash when disable stream (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Prevent crash when disable stream

[Why]

Disabling stream encoder invokes a function that no longer exists.

[How]

Check if the function declaration is NULL in disable stream encoder.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35799>

### **[CVE-2024-35808] kernel: md/dm-raid: don't call md\_reap\_sync\_thread() directly (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

md/dm-raid: don't call md\_reap\_sync\_thread() directly

Currently md\_reap\_sync\_thread() is called from raid\_message() directly without holding 'reconfig\_mutex', this is definitely unsafe because md\_reap\_sync\_thread() can change many fields that is protected by 'reconfig\_mutex'.

However, hold 'reconfig\_mutex' here is still problematic because this will cause deadlock, for example, commit 130443d60b1b ("md: refactor idle/frozen\_sync\_thread() to fix deadlock").

Fix this problem by using stop\_sync\_thread() to unregister sync\_thread, like md/raid did.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35808>

**[CVE-2024-35843] kernel: iommu/vt-d: Use device rbtree in iopf reporting path (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

iommu/vt-d: Use device rbtree in iopf reporting path

The existing I/O page fault handler currently locates the PCI device by calling `pci_get_domain_bus_and_slot()`. This function searches the list of all PCI devices until the desired device is found. To improve lookup efficiency, replace it with `device_rbtree_find()` to search the device within the probed device rbtree.

The I/O page fault is initiated by the device, which does not have any synchronization mechanism with the software to ensure that the device stays in the probed device tree. Theoretically, a device could be released by the IOMMU subsystem after `device_rbtree_find()` and before `iopf_get_dev_fault_param()`, which would cause a use-after-free problem.

Add a mutex to synchronize the I/O page fault reporting path and the IOMMU release device path. This lock doesn't introduce any performance overhead, as the conflict between I/O page fault reporting and device releasing is very rare.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35843>

**[CVE-2024-35860] kernel: bpf: support deferring bpf\_link dealloc to after RCU grace period (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: support deferring bpf\_link dealloc to after RCU grace period

BPF link for some program types is passed as a "context" which can be used by those BPF programs to look up additional information. E.g., for multi-kprobes and multi-uprobes, link is used to fetch BPF cookie values.

Because of this runtime dependency, when `bpf_link refcnt` drops to zero there could still be active BPF programs running accessing link data.

This patch adds generic support to defer `bpf_link dealloc` callback to after RCU GP, if requested. This is done by exposing two different deallocation callbacks, one synchronous and one deferred. If deferred one is provided, `bpf_link_free()` will schedule `dealloc_deferred()`



callback to happen after RCU GP.

BPF is using two flavors of RCU: "classic" non-sleepable one and RCU tasks trace one. The latter is used when sleepable BPF programs are used. `bpflink_free()` accommodates that by checking underlying BPF program's sleepable flag, and goes either through normal RCU GP only for non-sleepable, or through RCU tasks trace GP \*and\* then normal RCU GP (taking into account `rcu_trace_implies_rcu_gp()` optimization), if BPF program is sleepable.

We use this for multi-kprobe and multi-uprobe links, which dereference link during program run. We also preventively switch `raw_tp` link to use deferred dealloc callback, as upcoming changes in bpf-next tree expose `raw_tp` link data (specifically, cookie value) to BPF program at runtime as well.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35860>

**[CVE-2024-35869] kernel: smb: client: guarantee refcounted children from parent session (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: guarantee refcounted children from parent session

Avoid potential use-after-free bugs when walking DFS referrals, mounting and performing DFS failover by ensuring that all children from parent `@tcon->ses` are also refcounted. They're all needed across the entire DFS mount. Get rid of `@tcon->dfs_ses_list` while we're at it, too.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35869>

**[CVE-2024-35878] kernel: of: module: prevent NULL pointer dereference in vsnprintf() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

of: module: prevent NULL pointer dereference in vsnprintf()

In `of_modalias()`, we can get passed the `str` and `len` parameters which would cause a kernel oops in `vsnprintf()` since it only allows passing a NULL ptr when the length is also 0. Also, we need to filter out the negative values of the `len` parameter as these will result in a really huge buffer since `snprintf()` takes `size_t` parameter while ours is `ssize_t`...

Found by Linux Verification Center (linuxtesting.org) with the Svace static analysis tool.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35878>

**[CVE-2024-35904] kernel: selinux: avoid dereference of garbage after mount failure (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

selinux: avoid dereference of garbage after mount failure

In case kern\_mount() fails and returns an error pointer return in the error branch instead of continuing and dereferencing the error pointer.

While on it drop the never read static variable selinuxfs\_mount.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35904>

**[CVE-2024-35924] kernel: usb: typec: ucsi: Limit read size on v1.2 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

usb: typec: ucsi: Limit read size on v1.2

Between UCSI 1.2 and UCSI 2.0, the size of the MESSAGE\_IN region was increased from 16 to 256. In order to avoid overflowing reads for older systems, add a mechanism to use the read UCSI version to truncate read sizes on UCSI v1.2.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35924>

**[CVE-2024-35931] kernel: drm/amdgpu: Skip do PCI error slot reset during RAS recovery (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Skip do PCI error slot reset during RAS recovery

Why:

The PCI error slot reset maybe triggered after inject ue to UMC multi times, this caused system hang.

[ 557.371857] amdgpu 0000:af:00.0: amdgpu: GPU reset succeeded, trying to resume

```

[ 557.373718] [drm] PCIE GART of 512M enabled.
[ 557.373722] [drm] PTB located at 0x0000031FED700000
[ 557.373788] [drm] VRAM is lost due to GPU reset!
[ 557.373789] [drm] PSP is resuming...
[ 557.547012] mlx5_core 0000:55:00.0: mlx5_pci_err_detected Device state = 1 pci_status: 0. Exit, result = 3, need
reset
[ 557.547067] [drm] PCI error: detected callback, state(1)!!
[ 557.547069] [drm] No support for XGMI hive yet...
[ 557.548125] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 0. Enter
[ 557.607763] mlx5_core 0000:55:00.0: wait vital counter value 0x16b5b after 1 iterations
[ 557.607777] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 1. Exit, err = 0, result = 5,
recovered
[ 557.610492] [drm] PCI error: slot reset callback!!
...
[ 560.689382] amdgpu 0000:3f:00.0: amdgpu: GPU reset(2) succeeded!
[ 560.689546] amdgpu 0000:5a:00.0: amdgpu: GPU reset(2) succeeded!
[ 560.689562] general protection fault, probably for non-canonical address 0x5f080b54534f611f: 0000 [#1] SMP
NOPTI
[ 560.701008] CPU: 16 PID: 2361 Comm: kworker/u448:9 Tainted: G      OE   5.15.0-91-generic #101-Ubuntu
[ 560.712057] Hardware name: Microsoft C278A/C278A, BIOS C2789.5.BS.1C11.AG.1 11/08/2023
[ 560.720959] Workqueue: amdgpu-reset-hive amdgpu_ras_do_recovery [amdgpu]
[ 560.728887] RIP: 0010:amdgpu_device_gpu_recover.cold+0xbf1/0xcf5 [amdgpu]
[ 560.736891] Code: ff 41 89 c6 e9 1b ff ff ff 44 0f b6 45 b0 e9 4f ff ff ff be 01 00 00 00 4c 89 e7 e8 76 c9 8b ff 44 0f
b6 45 b0 e9 3c fd ff ff <48> 83 ba 18 02 00 00 00 0f 84 6a f8 ff ff 48 8d 7a 78 be 01 00 00
[ 560.757967] RSP: 0018:ffa0000032e53d80 EFLAGS: 00010202
[ 560.763848] RAX: ffa00000001dfd10 RBX: ffa0000000197090 RCX: ffa0000032e53db0
[ 560.771856] RDX: 5f080b54534f5f07 RSI: 0000000000000000 RDI: ff11000128100010
[ 560.779867] RBP: ffa0000032e53df0 R08: 0000000000000000 R09: ffffffff77f08
[ 560.787879] R10: 0000000000ffff0a R11: 0000000000000001 R12: 0000000000000000
[ 560.795889] R13: ffa0000032e53e00 R14: 0000000000000000 R15: 0000000000000000
[ 560.803889] FS: 0000000000000000(0000) GS:ff11007e7e800000(0000) knlGS:0000000000000000
[ 560.812973] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 560.819422] CR2: 000055a04c118e68 CR3: 0000000007410005 CR4: 0000000000771ee0
[ 560.827433] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[ 560.835433] DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7: 0000000000000400
[ 560.843444] PKRU: 55555554
[ 560.846480] Call Trace:
[ 560.849225] <TASK>
[ 560.851580] ? show_trace_log_lvl+0x1d6/0x2ea
[ 560.856488] ? show_trace_log_lvl+0x1d6/0x2ea
[ 560.861379] ? amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu]
[ 560.867778] ? show_regs.part.0+0x23/0x29
[ 560.872293] ? __die_body.cold+0x8/0xd
[ 560.876502] ? die_addr+0x3e/0x60
[ 560.880238] ? exc_general_protection+0x1c5/0x410
[ 560.885532] ? asm_exc_general_protection+0x27/0x30
[ 560.891025] ? amdgpu_device_gpu_recover.cold+0xbf1/0xcf5 [amdgpu]
[ 560.898323] amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu]
[ 560.904520] process_one_work+0x228/0x3d0

```

How:

In RAS recovery, mode-1 reset is issued from RAS fatal error handling and expected all the nodes in a hive to be reset. no need to issue another mode-1 during this procedure.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35931>

**[CVE-2024-35942] kernel: pmdomain: imx8mp-blk-ctrl: imx8mp\_blk: Add fdcc clock to hdmimix domain (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

pmdomain: imx8mp-blk-ctrl: imx8mp\_blk: Add fdcc clock to hdmimix domain

According to i.MX8MP RM and HDMI ADD, the fdcc clock is part of hdmi rx verification IP that should not enable for HDMI TX.

But actually if the clock is disabled before HDMI/LCDIF probe, LCDIF will not get pixel clock from HDMI PHY and print the error logs:

[CRTC:39:crtc-2] vblank wait timed out

WARNING: CPU: 2 PID: 9 at drivers/gpu/drm/drm\_atomic\_helper.c:1634  
drm\_atomic\_helper\_wait\_for\_vblanks.part.0+0x23c/0x260

Add fdcc clock to LCDIF and HDMI TX power domains to fix the issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35942>

**[CVE-2024-35945] kernel: net: phy: phy\_device: Prevent nullptr exceptions on ISR (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: phy: phy\_device: Prevent nullptr exceptions on ISR

If phydev->irq is set unconditionally, check for valid interrupt handler or fall back to polling mode to prevent nullptr exceptions in interrupt service routine.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35945>

**[CVE-2024-35946] kernel: wifi: rtw89: fix null pointer access when abort scan (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: rtw89: fix null pointer access when abort scan

During cancel scan we might use vif that weren't scanning.  
Fix this by using the actual scanning vif.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35946>

**[CVE-2024-35949] kernel: btrfs: make sure that WRITTEN is set on all metadata blocks (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: make sure that WRITTEN is set on all metadata blocks

We previously would call `btrfs_check_leaf()` if we had the check integrity code enabled, which meant that we could only run the extended leaf checks if we had WRITTEN set on the header flags.

This leaves a gap in our checking, because we could end up with corruption on disk where WRITTEN isn't set on the leaf, and then the extended leaf checks don't get run which we rely on to validate all of the item pointers to make sure we don't access memory outside of the extent buffer.

However, since 732fab95abe2 ("btrfs: check-integrity: remove CONFIG\_BTRFS\_FS\_CHECK\_INTEGRITY option") we no longer call `btrfs_check_leaf()` from `btrfs_mark_buffer_dirty()`, which means we only ever call it on blocks that are being written out, and thus have WRITTEN set, or that are being read in, which should have WRITTEN set.

Add checks to make sure we have WRITTEN set appropriately, and then make sure `__btrfs_check_leaf()` always does the item checking. This will protect us from file systems that have been corrupted and no longer have WRITTEN set on some of the blocks.

This was hit on a crafted image tweaking the WRITTEN bit and reported by KASAN as out-of-bound access in the eb accessors. The example is a dir item at the end of an eb.

```
[2.042] BTRFS warning (device loop1): bad eb member start: ptr 0x3fff start 30572544 member offset 16410 size 2
[2.040] general protection fault, probably for non-canonical address 0xe0009d1000000003: 0000 [#1] PREEMPT SMP
KASAN NOPTI
[2.537] KASAN: maybe wild-memory-access in range [0x0005088000000018-0x000508800000001f]
[2.729] CPU: 0 PID: 2587 Comm: mount Not tainted 6.8.2 #1
[2.729] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014
[2.621] RIP: 0010:btrfs_get_16+0x34b/0x6d0
[2.621] RSP: 0018:ffff88810871fab8 EFLAGS: 00000206
[2.621] RAX: 0000a11000000003 RBX: ffff888104ff8720 RCX: ffff88811b2288c0
[2.621] RDX: dffffc0000000000 RSI: ffffffff81dd8aca RDI: ffff88810871f748
[2.621] RBP: 000000000000401a R08: 0000000000000001 R09: ffffed10210e3ee9
[2.621] R10: ffff88810871f74f R11: 205d323430333737 R12: 000000000000001a
[2.621] R13: 000508800000001a R14: 1ffff110210e3f5d R15: ffffffff850011e8
[2.621] FS: 00007f56ea275840(0000) GS:ffff88811b200000(0000) knlGS:0000000000000000
```

[2.621] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
[2.621] CR2: 00007febd13b75c0 CR3: 000000010bb50000 CR4: 000000000000006f0  
[2.621] Call Trace:  
[2.621] <TASK>  
[2.621] ? show\_regs+0x74/0x80  
[2.621] ? die\_addr+0x46/0xc0  
[2.621] ? exc\_general\_protection+0x161/0x2a0  
[2.621] ? asm\_exc\_general\_protection+0x26/0x30  
[2.621] ? btrfs\_get\_16+0x33a/0x6d0  
[2.621] ? btrfs\_get\_16+0x34b/0x6d0  
[2.621] ? btrfs\_get\_16+0x33a/0x6d0  
[2.621] ? \_\_pfx\_btrfs\_get\_16+0x10/0x10  
[2.621] ? \_\_pfx\_mutex\_unlock+0x10/0x10  
[2.621] btrfs\_match\_dir\_item\_name+0x101/0x1a0  
[2.621] btrfs\_lookup\_dir\_item+0x1f3/0x280  
[2.621] ? \_\_pfx\_btrfs\_lookup\_dir\_item+0x10/0x10  
[2.621] btrfs\_get\_tree+0xd25/0x1910

[ copy more details from report ]

More Info: <https://avd.aquasec.com/nvd/cve-2024-35949>

**[CVE-2024-35951] kernel: drm/panfrost: Fix the error path in panfrost\_mmu\_map\_fault\_addr()  
(Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/panfrost: Fix the error path in panfrost\_mmu\_map\_fault\_addr()

Subject: [PATCH] drm/panfrost: Fix the error path in  
panfrost\_mmu\_map\_fault\_addr()

If some the pages or sgt allocation failed, we shouldn't release the  
pages ref we got earlier, otherwise we will end up with unbalanced  
get/put\_pages() calls. We should instead leave everything in place  
and let the BO release function deal with extra cleanup when the object  
is destroyed, or let the fault handler try again next time it's called.

More Info: <https://avd.aquasec.com/nvd/cve-2024-35951>

**[CVE-2024-35961] kernel: net/mlx5: Register devlink first under devlink lock (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5: Register devlink first under devlink lock

In case device is having a non fatal FW error during probe, the driver will report the error to user via devlink. This will trigger a WARN\_ON, since mlx5 is calling devlink\_register() last. In order to avoid the WARN\_ON[1], change mlx5 to invoke devl\_register() first under devlink lock.

[1]

WARNING: CPU: 5 PID: 227 at net/devlink/health.c:483 devlink\_recover\_notify.constprop.0+0xb8/0xc0  
CPU: 5 PID: 227 Comm: kworker/u16:3 Not tainted 6.4.0-rc5\_for\_upstream\_min\_debug\_2023\_06\_12\_12\_38 #1  
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org  
04/01/2014

Workqueue: mlx5\_health0000:08:00.0 mlx5\_fw\_reporter\_err\_work [mlx5\_core]

RIP: 0010:devlink\_recover\_notify.constprop.0+0xb8/0xc0

Call Trace:

<TASK>

? \_\_warn+0x79/0x120

? devlink\_recover\_notify.constprop.0+0xb8/0xc0

? report\_bug+0x17c/0x190

? handle\_bug+0x3c/0x60

? exc\_invalid\_op+0x14/0x70

? asm\_exc\_invalid\_op+0x16/0x20

? devlink\_recover\_notify.constprop.0+0xb8/0xc0

devlink\_health\_report+0x4a/0x1c0

mlx5\_fw\_reporter\_err\_work+0xa4/0xd0 [mlx5\_core]

process\_one\_work+0x1bb/0x3c0

? process\_one\_work+0x3c0/0x3c0

worker\_thread+0x4d/0x3c0

? process\_one\_work+0x3c0/0x3c0

kthread+0xc6/0xf0

? kthread\_complete\_and\_exit+0x20/0x20

ret\_from\_fork+0x1f/0x30

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2024-35961>

### **[CVE-2024-35974] kernel: block: fix q->blkcg\_list corruption during disk rebind (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

block: fix q->blkcg\_list corruption during disk rebind

Multiple gendisk instances can allocated/added for single request queue in case of disk rebind. blkcg may still stay in q->blkcg\_list when calling blkcg\_init\_disk() for rebind, then q->blkcg\_list becomes corrupted.

Fix the list corruption issue by:

- add blkcg\_init\_queue() to initialize q->blkcg\_list & q->blkcg\_mutex only
- move calling blkcg\_init\_queue() into blk\_alloc\_queue()

The list corruption should be started since commit f1c006f1c685 ("blk-cgroup: synchronize pd\_free\_fn() from blkcg\_free\_workfn() and blkcg\_deactivate\_policy()") which delays removing blkcg from q->blkcg\_list into blkcg\_free\_workfn().

More Info: <https://avd.aquasec.com/nvd/cve-2024-35974>

### **[CVE-2024-36022] kernel: drm/amdgpu: Init zone device and drm client after mode-1 reset on reload (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Init zone device and drm client after mode-1 reset on reload

In passthrough environment, when amdgpu is reloaded after unload, mode-1 is triggered after initializing the necessary IPs, That init does not include KFD, and KFD init waits until the reset is completed. KFD init is called in the reset handler, but in this case, the zone device and drm client is not initialized, causing app to create kernel panic.

v2: Removing the init KFD condition from amdgpu\_amdkfd\_drm\_client\_create. As the previous version has the potential of creating DRM client twice.

v3: v2 patch results in SDMA engine hung as DRM open causes VM clear to SDMA before SDMA init. Adding the condition to in drm client creation, on top of v1, to guard against drm client creation call multiple times.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36022>

### **[CVE-2024-36024] kernel: drm/amd/display: Disable idle reallow as part of command/gpint execution (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Disable idle reallow as part of command/gpint execution

[Why]

Workaroud for a race condition where DMCUB is in the process of committing to IPS1 during the handshake causing us to miss the transition into IPS2 and touch the INBOX1 RPTR causing a HW hang.

[How]

Disable the reallow to ensure that we have enough of a gap between entry and exit and we're not seeing back-to-back wake\_and\_executes.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36024>



**[CVE-2024-36881] kernel: mm/userfaultfd: reset ptes when close() for wr-protected ones (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/userfaultfd: reset ptes when close() for wr-protected ones

Userfaultfd unregister includes a step to remove wr-protect bits from all the relevant pgtable entries, but that only covered an explicit UFFDIO\_UNREGISTER ioctl, not a close() on the userfaultfd itself. Cover that too. This fixes a WARN trace.

The only user visible side effect is the user can observe leftover wr-protect bits even if the user close()ed on an userfaultfd when releasing the last reference of it. However hopefully that should be harmless, and nothing bad should happen even if so.

This change is now more important after the recent page-table-check patch we merged in mm-unstable (446dd9ad37d0 ("mm/page\_table\_check: support userfault wr-protect entries")), as we'll do sanity check on uffd-wp bits without vma context. So it's better if we can 100% guarantee no uffd-wp bit leftovers, to make sure each report will be valid.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36881>

**[CVE-2024-36903] kernel: ipv6: Fix potential uninit-value access in \_\_ip6\_make\_skb() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ipv6: Fix potential uninit-value access in \_\_ip6\_make\_skb()

As it was done in commit fc1092f51567 ("ipv4: Fix uninit-value access in \_\_ip\_make\_skb()") for IPv4, check FLOWI\_FLAG\_KNOWN\_NH on fl6->flowi6\_flags instead of testing HDRINCL on the socket to avoid a race condition which causes uninit-value access.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36903>

**[CVE-2024-36907] kernel: SUNRPC: add a missing rpc\_stat for TCP TLS (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

SUNRPC: add a missing rpc\_stat for TCP TLS

Commit 1548036ef120 ("nfs: make the rpc\_stat per net namespace") added functionality to specify rpc\_stats function but missed adding it to the TCP TLS functionality. As the result, mounting with xprtsec=tls lead to the following kernel oops.

```
[ 128.984192] Unable to handle kernel NULL pointer dereference at
virtual address 000000000000001c
[ 128.985058] Mem abort info:
[ 128.985372] ESR = 0x0000000096000004
[ 128.985709] EC = 0x25: DABT (current EL), IL = 32 bits
[ 128.986176] SET = 0, FnV = 0
[ 128.986521] EA = 0, S1PTW = 0
[ 128.986804] FSC = 0x04: level 0 translation fault
[ 128.987229] Data abort info:
[ 128.987597] ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000
[ 128.988169] CM = 0, WnR = 0, TnD = 0, TagAccess = 0
[ 128.988811] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0
[ 128.989302] user pgtbl: 4k pages, 48-bit VAs, pgdp=0000000106c84000
[ 128.990048] [000000000000001c] pgd=0000000000000000, p4d=0000000000000000
[ 128.990736] Internal error: Oops: 0000000096000004 [#1] SMP
[ 128.991168] Modules linked in: nfs_layout_nfsv41_files
rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace netfs
uinput dm_mod nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib
nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct
nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 rkill
ip_set nf_tables nfnetlink qrtr vsock_loopback
vmw_vsock_virtio_transport_common vmw_vsock_vmci_transport vsock
sunrpc vfat fat uvcvideo videobuf2_vmalloc videobuf2_memops uvc
videobuf2_v4l2 videodev videobuf2_common mc vmw_vmci xfs libcrc32c
e1000e crct10dif_ce ghash_ce sha2_ce vmwgfx nvme sha256_arm64
nvme_core sr_mod cdrom sha1_ce drm_ttm_helper ttm drm_kms_helper drm
sg fuse
[ 128.996466] CPU: 0 PID: 179 Comm: kworker/u4:26 Kdump: loaded Not
tainted 6.8.0-rc6+ #12
[ 128.997226] Hardware name: VMware, Inc. VMware20,1/VBSA, BIOS
VMW201.00V.21805430.BA64.2305221830 05/22/2023
[ 128.998084] Workqueue: xprtiod xs_tcp_tls_setup_socket [sunrpc]
[ 128.998701] pstate: 81400005 (Nzcv daif +PAN -UAO -TCO +DIT -SSBS BTYPE=--)
[ 128.999384] pc : call_start+0x74/0x138 [sunrpc]
[ 128.999809] lr : __rpc_execute+0xb8/0x3e0 [sunrpc]
[ 129.000244] sp : ffff8000832b3a00
[ 129.000508] x29: ffff8000832b3a00 x28: ffff800081ac79c0 x27: ffff800081ac7000
[ 129.001111] x26: 0000000004248060 x25: 0000000000000000 x24: ffff800081596008
[ 129.001757] x23: ffff80007b087240 x22: ffff00009a509d30 x21: 0000000000000000
[ 129.002345] x20: ffff000090075600 x19: ffff00009a509d00 x18: ffffffff
[ 129.002912] x17: 733d4d4554535953 x16: 42555300312d746e x15: ffff8000832b3a88
[ 129.003464] x14: ffffffff x13: ffff8000832b3a7d x12: 0000000000000008
[ 129.004021] x11: 0101010101010101 x10: ffff8000150cb560 x9 : ffff80007b087c00
[ 129.004577] x8 : ffff00009a509de0 x7 : 0000000000000000 x6 : 00000000be8c4ee3
[ 129.005026] x5 : 0000000000000000 x4 : 0000000000000000 x3 : ffff000094d56680
```

```
[ 129.005425] x2 : ffff80007b0637f8 x1 : ffff000090075600 x0 : ffff00009a509d00
[ 129.005824] Call trace:
[ 129.005967] call_start+0x74/0x138 [sunrpc]
[ 129.006233] __rpc_execute+0xb8/0x3e0 [sunrpc]
[ 129.006506] rpc_execute+0x160/0x1d8 [sunrpc]
[ 129.006778] rpc_run_task+0x148/0x1f8 [sunrpc]
[ 129.007204] tls_probe+0x80/0xd0 [sunrpc]
[ 129.007460] rpc_ping+0x28/0x80 [sunrpc]
[ 129.007715] rpc_create_xprt+0x134/0x1a0 [sunrpc]
[ 129.007999] rpc_create+0x128/0x2a0 [sunrpc]
[ 129.008264] xs_tcp_tls_setup_socket+0xdc/0x508 [sunrpc]
[ 129.008583] process_one_work+0x174/0x3c8
[ 129.008813] worker_thread+0x2c8/0x3e0
[ 129.009033] kthread+0x100/0x110
[ 129.009225] ret_from_fork+0x10/0x20
[ 129.009432] Code: f0fffc2 911fe042 aa1403e1 aa1303e0 (b9401c83)
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-36907>

### **[CVE-2024-36908] kernel: blk-iocost: do not WARN if iocg was already offlined (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

blk-iocost: do not WARN if iocg was already offlined

In `iocg_pay_debt()`, `warn` is triggered if `'active_list'` is empty, which is intended to confirm `iocg` is active when it has debt. However, `warn` can be triggered during a `blkcg` or disk removal, if `iocg_waitq_timer_fn()` is run at that time:

WARNING: CPU: 0 PID: 2344971 at block/blk-iocost.c:1402 `iocg_pay_debt+0x14c/0x190`

Call trace:

`iocg_pay_debt+0x14c/0x190`

`iocg_kick_waitq+0x438/0x4c0`

`iocg_waitq_timer_fn+0xd8/0x130`

`__run_hrtimer+0x144/0x45c`

`__hrtimer_run_queues+0x16c/0x244`

`hrtimer_interrupt+0x2cc/0x7b0`

The `warn` in this situation is meaningless. Since this `iocg` is being removed, the state of the `'active_list'` is irrelevant, and `'waitq_timer'` is canceled after removing `'active_list'` in `ioc_pd_free()`, which ensures `iocg` is freed after `iocg_waitq_timer_fn()` returns.

Therefore, add the check if `iocg` was already offlined to avoid `warn` when removing a `blkcg` or disk.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36908>

### **[CVE-2024-36911] kernel: hv\_netvsc: Don't free decrypted memory (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

hv\_netvsc: Don't free decrypted memory

In CoCo VMs it is possible for the untrusted host to cause `set_memory_encrypted()` or `set_memory_decrypted()` to fail such that an error is returned and the resulting memory is shared. Callers need to take care to handle these errors to avoid returning decrypted (shared) memory to the page allocator, which could lead to functional or security issues.

The netvsc driver could free decrypted/shared pages if `set_memory_decrypted()` fails. Check the decrypted field in the `gpadi` to decide whether to free the memory.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36911>

**[CVE-2024-36913] kernel: Drivers: hv: vmbus: Leak pages if set\_memory\_encrypted() fails (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Drivers: hv: vmbus: Leak pages if `set_memory_encrypted()` fails

In CoCo VMs it is possible for the untrusted host to cause `set_memory_encrypted()` or `set_memory_decrypted()` to fail such that an error is returned and the resulting memory is shared. Callers need to take care to handle these errors to avoid returning decrypted (shared) memory to the page allocator, which could lead to functional or security issues.

VMbus code could free decrypted pages if `set_memory_encrypted()/decrypted()` fails. Leak the pages if this happens.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36913>

**[CVE-2024-36922] kernel: wifi: iwlwifi: read txq->read\_ptr under lock (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: iwlwifi: read `txq->read_ptr` under lock

If we read `txq->read_ptr` without lock, we can read the same value twice, then obtain the lock, and reclaim from there to two different places, but crucially reclaim the same entry twice, resulting in the `WARN_ONCE()` a little later. Fix that by reading `txq->read_ptr` under lock.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36922>

## **[CVE-2024-36927] kernel: ipv4: Fix uninit-value access in `__ip_make_skb()` (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: `%ls(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

ipv4: Fix uninit-value access in `__ip_make_skb()`

KMSAN reported uninit-value access in `__ip_make_skb()` [1]. `__ip_make_skb()` tests `HDRINCL` to know if the `skb` has `icmphdr`. However, `HDRINCL` can cause a race condition. If calling `setsockopt(2)` with `IP_HDRINCL` changes `HDRINCL` while `__ip_make_skb()` is running, the function will access `icmphdr` in the `skb` even if it is not included. This causes the issue reported by KMSAN.

Check `FLOWI_FLAG_KNOWN_NH` on `fl4->flowi4_flags` instead of testing `HDRINCL` on the socket.

Also, `fl4->fl4_icmp_type` and `fl4->fl4_icmp_code` are not initialized. These are union in struct `flowi4` and are implicitly initialized by `flowi4_init_output()`, but we should not rely on specific union layout.

Initialize these explicitly in `raw_sendmsg()`.

[1]

BUG: KMSAN: uninit-value in `__ip_make_skb+0x2b74/0x2d20` net/ipv4/ip\_output.c:1481  
`__ip_make_skb+0x2b74/0x2d20` net/ipv4/ip\_output.c:1481  
`ip_finish_skb` include/net/ip.h:243 [inline]  
`ip_push_pending_frames+0x4c/0x5c0` net/ipv4/ip\_output.c:1508  
`raw_sendmsg+0x2381/0x2690` net/ipv4/raw.c:654  
`inet_sendmsg+0x27b/0x2a0` net/ipv4/af\_inet.c:851  
`sock_sendmsg_nosec` net/socket.c:730 [inline]  
`__sock_sendmsg+0x274/0x3c0` net/socket.c:745  
`__sys_sendto+0x62c/0x7b0` net/socket.c:2191  
`__do_sys_sendto` net/socket.c:2203 [inline]  
`__se_sys_sendto` net/socket.c:2199 [inline]  
`__x64_sys_sendto+0x130/0x200` net/socket.c:2199  
`do_syscall_64+0xd8/0x1f0` arch/x86/entry/common.c:83  
`entry_SYSCALL_64_after_hwframe+0x6d/0x75`

Uninit was created at:

`slab_post_alloc_hook` mm/slub.c:3804 [inline]  
`slab_alloc_node` mm/slub.c:3845 [inline]  
`kmem_cache_alloc_node+0x5f6/0xc50` mm/slub.c:3888  
`kmalloc_reserve+0x13c/0x4a0` net/core/skbuff.c:577

\_\_alloc\_skb+0x35a/0x7c0 net/core/skbuff.c:668  
alloc\_skb include/linux/skbuff.h:1318 [inline]  
\_\_ip\_append\_data+0x49ab/0x68c0 net/ipv4/ip\_output.c:1128  
ip\_append\_data+0x1e7/0x260 net/ipv4/ip\_output.c:1365  
raw\_sendmsg+0x22b1/0x2690 net/ipv4/raw.c:648  
inet\_sendmsg+0x27b/0x2a0 net/ipv4/af\_inet.c:851  
sock\_sendmsg\_nosec net/socket.c:730 [inline]  
\_\_sock\_sendmsg+0x274/0x3c0 net/socket.c:745  
\_\_sys\_sendto+0x62c/0x7b0 net/socket.c:2191  
\_\_do\_sys\_sendto net/socket.c:2203 [inline]  
\_\_se\_sys\_sendto net/socket.c:2199 [inline]  
\_\_x64\_sys\_sendto+0x130/0x200 net/socket.c:2199  
do\_syscall\_64+0xd8/0x1f0 arch/x86/entry/common.c:83  
entry\_SYSCALL\_64\_after\_hwframe+0x6d/0x75

CPU: 1 PID: 15709 Comm: syz-executor.7 Not tainted 6.8.0-11567-gb3603fcb79b1 #25  
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-1.fc39 04/01/2014

More Info: <https://avd.aquasec.com/nvd/cve-2024-36927>

**[CVE-2024-36949] kernel: amd/amdkfd: sync all devices to wait all processes being evicted (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

amd/amdkfd: sync all devices to wait all processes being evicted

If there are more than one device doing reset in parallel, the first device will call `kfd_suspend_all_processes()` to evict all processes on all devices, this call takes time to finish. other device will start reset and recover without waiting. if the process has not been evicted before doing recover, it will be restored, then caused page fault.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36949>

**[CVE-2024-36951] kernel: drm/amdkfd: range check cp bad op exception interrupts (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdkfd: range check cp bad op exception interrupts

Due to a CP interrupt bug, bad packet garbage exception codes are raised. Do a range check so that the debugger and runtime do not receive garbage codes.

Update the user api to guard exception code type checking as well.

More Info: <https://avd.aquasec.com/nvd/cve-2024-36951>

**[CVE-2024-36968] kernel: Bluetooth: L2CAP: Fix div-by-zero in l2cap\_le\_flowctl\_init() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: L2CAP: Fix div-by-zero in l2cap\_le\_flowctl\_init()

l2cap\_le\_flowctl\_init() can cause both div-by-zero and an integer overflow since hdev->le\_mtu may not fall in the valid range.

Move MTU from hci\_dev to hci\_conn to validate MTU and stop the connection process earlier if MTU is invalid.

Also, add a missing validation in read\_buffer\_size() and make it return an error value if the validation fails.

Now hci\_conn\_add() returns ERR\_PTR() as it can fail due to the both a kzalloc failure and invalid MTU value.

divide error: 0000 [#1] PREEMPT SMP KASAN NOPTI

CPU: 0 PID: 67 Comm: kworker/u5:0 Tainted: G W 6.9.0-rc5+ #20

Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014

Workqueue: hci0 hci\_rx\_work

RIP: 0010:l2cap\_le\_flowctl\_init+0x19e/0x3f0 net/bluetooth/l2cap\_core.c:547

Code: e8 17 17 0c 00 66 41 89 9f 84 00 00 00 bf 01 00 00 00 41 b8 02 00 00 00 4c

89 fe 4c 89 e2 89 d9 e8 27 17 0c 00 44 89 f0 31 d2 <66> f7 f3 89 c3 ff c3 4d 8d

b7 88 00 00 00 4c 89 f0 48 c1 e8 03 42

RSP: 0018:ffff88810bc0f858 EFLAGS: 00010246

RAX: 000000000000002a RBX: 0000000000000000 RCX: dffffc0000000000

RDX: 0000000000000000 RSI: ffff88810bc0f7c0 RDI: ffff90002dcb66f

RBP: ffff88810bc0f880 R08: aa69db2dda70ff01 R09: 0000faaaaaaaaaa

R10: 0084000000faaaaa R11: 0000000000000000 R12: ffff88810d65a084

R13: dffffc0000000000 R14: 000000000000002a R15: ffff88810d65a000

FS: 0000000000000000(0000) GS:ffff88811ac00000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 0000000020000100 CR3: 0000000103268003 CR4: 000000000770ef0

PKRU: 55555554

Call Trace:

<TASK>

l2cap\_le\_connect\_req net/bluetooth/l2cap\_core.c:4902 [inline]

l2cap\_le\_sig\_cmd net/bluetooth/l2cap\_core.c:5420 [inline]

l2cap\_le\_sig\_channel net/bluetooth/l2cap\_core.c:5486 [inline]

l2cap\_rcv\_frame+0xe59d/0x11710 net/bluetooth/l2cap\_core.c:6809

l2cap\_rcv\_acldata+0x544/0x10a0 net/bluetooth/l2cap\_core.c:7506

hci\_acldata\_packet net/bluetooth/hci\_core.c:3939 [inline]

hci\_rx\_work+0x5e5/0xb20 net/bluetooth/hci\_core.c:4176

process\_one\_work kernel/workqueue.c:3254 [inline]

```
process_scheduled_works+0x90f/0x1530 kernel/workqueue.c:3335
worker_thread+0x926/0xe70 kernel/workqueue.c:3416
kthread+0x2e3/0x380 kernel/kthread.c:388
ret_from_fork+0x5c/0x90 arch/x86/kernel/process.c:147
ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244
</TASK>
```

Modules linked in:

---[ end trace 0000000000000000 ]---

More Info: <https://avd.aquasec.com/nvd/cve-2024-36968>

### **[CVE-2024-38541] kernel: of: module: add buffer overflow check in of\_modalias() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

of: module: add buffer overflow check in of\_modalias()

In of\_modalias(), if the buffer happens to be too small even for the 1st snprintf() call, the len parameter will become negative and str parameter (if not NULL initially) will point beyond the buffer's end. Add the buffer overflow check after the 1st snprintf() call and fix such check after the strlen() call (accounting for the terminating NUL char).

More Info: <https://avd.aquasec.com/nvd/cve-2024-38541>

### **[CVE-2024-38557] kernel: net/mlx5: Reload only IB representors upon lag disable/enable (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5: Reload only IB representors upon lag disable/enable

On lag disable, the bond IB device along with all of its representors are destroyed, and then the slaves' representors get reloaded.

In case the slave IB representor load fails, the eswitch error flow unloads all representors, including ethernet representors, where the netdevs get detached and removed from lag bond. Such flow is inaccurate as the lag driver is not responsible for loading/unloading ethernet representors. Furthermore, the flow described above begins by holding lag lock to prevent bond changes during disable flow. However, when reaching the ethernet representors detachment from lag, the lag lock is required again, triggering the following deadlock:

Call trace:

\_\_switch\_to+0xf4/0x148



\_\_schedule+0x2c8/0x7d0  
schedule+0x50/0xe0  
schedule\_preempt\_disabled+0x18/0x28  
\_\_mutex\_lock.isra.13+0x2b8/0x570  
\_\_mutex\_lock\_slowpath+0x1c/0x28  
mutex\_lock+0x4c/0x68  
mlx5\_lag\_remove\_netdev+0x3c/0x1a0 [mlx5\_core]  
mlx5e\_uplink\_rep\_disable+0x70/0xa0 [mlx5\_core]  
mlx5e\_detach\_netdev+0x6c/0xb0 [mlx5\_core]  
mlx5e\_netdev\_change\_profile+0x44/0x138 [mlx5\_core]  
mlx5e\_netdev\_attach\_nic\_profile+0x28/0x38 [mlx5\_core]  
mlx5e\_vport\_rep\_unload+0x184/0x1b8 [mlx5\_core]  
mlx5\_esw\_offloads\_rep\_load+0xd8/0xe0 [mlx5\_core]  
mlx5\_eswitch\_reload\_reps+0x74/0xd0 [mlx5\_core]  
mlx5\_disable\_lag+0x130/0x138 [mlx5\_core]  
mlx5\_lag\_disable\_change+0x6c/0x70 [mlx5\_core] // hold ldev->lock  
mlx5\_devlink\_eswitch\_mode\_set+0xc0/0x410 [mlx5\_core]  
devlink\_nl\_cmd\_eswitch\_set\_doit+0xdc/0x180  
genl\_family\_rcv\_msg\_doit.isra.17+0xe8/0x138  
genl\_rcv\_msg+0xe4/0x220  
netlink\_rcv\_skb+0x44/0x108  
genl\_rcv+0x40/0x58  
netlink\_unicast+0x198/0x268  
netlink\_sendmsg+0x1d4/0x418  
sock\_sendmsg+0x54/0x60  
\_\_sys\_sendto+0xf4/0x120  
\_\_arm64\_sys\_sendto+0x30/0x40  
el0\_svc\_common+0x8c/0x120  
do\_el0\_svc+0x30/0xa0  
el0\_svc+0x20/0x30  
el0\_sync\_handler+0x90/0xb8  
el0\_sync+0x160/0x180

Thus, upon lag enable/disable, load and unload only the IB representors of the slaves preventing the deadlock mentioned above.

While at it, refactor the `mlx5_esw_offloads_rep_load()` function to have a static helper method for its internal logic, in symmetry with the representor unload design.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38557>

### **[CVE-2024-38564] kernel: bpf: Add BPF\_PROG\_TYPE\_CGROUP\_SKB attach type enforcement in BPF\_LINK\_CREATE (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Add BPF\_PROG\_TYPE\_CGROUP\_SKB attach type enforcement in BPF\_LINK\_CREATE

bpf\_prog\_attach uses attach\_type\_to\_prog\_type to enforce proper attach type for BPF\_PROG\_TYPE\_CGROUP\_SKB. link\_create uses bpf\_prog\_get and relies on bpf\_prog\_attach\_check\_attach\_type to properly verify prog\_type <> attach\_type association.

Add missing attach\_type enforcement for the link\_create case. Otherwise, it's currently possible to attach cgroup\_skb prog types to other cgroup hooks.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38564>

## **[CVE-2024-38594] kernel: net: stmmac: move the EST lock to struct stmmac\_priv (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: stmmac: move the EST lock to struct stmmac\_priv

Reinitialize the whole EST structure would also reset the mutex lock which is embedded in the EST structure, and then trigger the following warning. To address this, move the lock to struct stmmac\_priv. We also need to reacquire the mutex lock when doing this initialization.

DEBUG\_LOCKS\_WARN\_ON(lock->magic != lock)

WARNING: CPU: 3 PID: 505 at kernel/locking/mutex.c:587 \_\_mutex\_lock+0xd84/0x1068

Modules linked in:

CPU: 3 PID: 505 Comm: tc Not tainted 6.9.0-rc6-00053-g0106679839f7-dirty #29

Hardware name: NXP i.MX8MPlus EVK board (DT)

pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--)

pc : \_\_mutex\_lock+0xd84/0x1068

lr : \_\_mutex\_lock+0xd84/0x1068

sp : fffffc0864e3570

x29: fffffc0864e3570 x28: fffffc0817bdc78 x27: 0000000000000003

x26: fffff80c54f1808 x25: fffff80c9164080 x24: fffffc080d723ac

x23: 0000000000000000 x22: 0000000000000002 x21: 0000000000000000

x20: 0000000000000000 x19: fffffc083bc3000 x18: ffffffff

x17: fffffc0817b080 x16: 0000000000000002 x15: fffff80d2d40000

x14: 000000000000002da x13: fffff80d2d404b8 x12: fffffc082b5a5c8

x11: fffffc082bca680 x10: fffffc082bb2640 x9 : fffffc082bb2698

x8 : 0000000000017fe8 x7 : c0000000ffff x6 : 0000000000000001

x5 : fffff8178fe0d48 x4 : 0000000000000000 x3 : 0000000000000027

x2 : fffff8178fe0d50 x1 : 0000000000000000 x0 : 0000000000000000

Call trace:

\_\_mutex\_lock+0xd84/0x1068

mutex\_lock\_nested+0x28/0x34

tc\_setup\_taprio+0x118/0x68c

stmmac\_setup\_tc+0x50/0xf0

taprio\_change+0x868/0xc9c

More Info: <https://avd.aquasec.com/nvd/cve-2024-38594>

## [CVE-2024-38608] kernel: net/mlx5e: Fix netif state handling (Severity: MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5e: Fix netif state handling

mlx5e\_suspend cleans resources only if netif\_device\_present() returns true. However, mlx5e\_resume changes the state of netif, via mlx5e\_nic\_enable, only if reg\_state == NETREG\_REGISTERED. In the below case, the above leads to NULL-ptr Oops[1] and memory leaks:

```
mlx5e_probe
  _mlx5e_resume
    mlx5e_attach_netdev
      mlx5e_nic_enable <-- netdev not reg, not calling netif_device_attach()
      register_netdev <-- failed for some reason.
ERROR_FLOW:
  _mlx5e_suspend <-- netif_device_present return false, resources aren't freed :(
```

Hence, clean resources in this case as well.

[1]

BUG: kernel NULL pointer dereference, address: 0000000000000000

PGD 0 P4D 0

Oops: 0010 [#1] SMP

CPU: 2 PID: 9345 Comm: test-ovs-ct-gen Not tainted 6.5.0\_for\_upstream\_min\_debug\_2023\_09\_05\_16\_01 #1

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org  
04/01/2014

RIP: 0010:0x0

Code: Unable to access opcode bytes at 0xffffffffffffd6.

RSP: 0018:ffff888178aaf758 EFLAGS: 00010246

Call Trace:

<TASK>

? \_\_die+0x20/0x60

? page\_fault\_oops+0x14c/0x3c0

? exc\_page\_fault+0x75/0x140

? asm\_exc\_page\_fault+0x22/0x30

notifier\_call\_chain+0x35/0xb0

blocking\_notifier\_call\_chain+0x3d/0x60

mlx5\_blocking\_notifier\_call\_chain+0x22/0x30 [mlx5\_core]

mlx5\_core\_uplink\_netdev\_event\_replay+0x3e/0x60 [mlx5\_core]

mlx5\_mdev\_netdev\_track+0x53/0x60 [mlx5\_ib]

mlx5\_ib\_roce\_init+0xc3/0x340 [mlx5\_ib]

\_\_mlx5\_ib\_add+0x34/0xd0 [mlx5\_ib]

mlx5r\_probe+0xe1/0x210 [mlx5\_ib]

? auxiliary\_match\_id+0x6a/0x90

auxiliary\_bus\_probe+0x38/0x80

? driver\_sysfs\_add+0x51/0x80

```

really_probe+0xc9/0x3e0
? driver_probe_device+0x90/0x90
__driver_probe_device+0x80/0x160
driver_probe_device+0x1e/0x90
__device_attach_driver+0x7d/0x100
bus_for_each_drv+0x80/0xd0
__device_attach+0xbc/0x1f0
bus_probe_device+0x86/0xa0
device_add+0x637/0x840
__auxiliary_device_add+0x3b/0xa0
add_adev+0xc9/0x140 [mlx5_core]
mlx5_rescan_drivers_locked+0x22a/0x310 [mlx5_core]
mlx5_register_device+0x53/0xa0 [mlx5_core]
mlx5_init_one_devl_locked+0x5c4/0x9c0 [mlx5_core]
mlx5_init_one+0x3b/0x60 [mlx5_core]
probe_one+0x44c/0x730 [mlx5_core]
local_pci_probe+0x3e/0x90
pci_device_probe+0xbf/0x210
? kernfs_create_link+0x5d/0xa0
? sysfs_do_create_link_sd+0x60/0xc0
really_probe+0xc9/0x3e0
? driver_probe_device+0x90/0x90
__driver_probe_device+0x80/0x160
driver_probe_device+0x1e/0x90
__device_attach_driver+0x7d/0x100
bus_for_each_drv+0x80/0xd0
__device_attach+0xbc/0x1f0
pci_bus_add_device+0x54/0x80
pci_iov_add_virtfn+0x2e6/0x320
sriov_enable+0x208/0x420
mlx5_core_sriov_configure+0x9e/0x200 [mlx5_core]
sriov_numvfs_store+0xae/0x1a0
kernfs_fop_write_iter+0x10c/0x1a0
vfs_write+0x291/0x3c0
ksys_write+0x5f/0xe0
do_syscall_64+0x3d/0x90
entry_SYSCALL_64_after_hwframe+0x46/0xb0
CR2: 0000000000000000
---[ end trace 0000000000000000 ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-38608>

**[CVE-2024-38611] kernel: media: i2c: et8ek8: Don't strip remove function when driver is builtin (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: i2c: et8ek8: Don't strip remove function when driver is builtin

Using `__exit` for the remove function results in the remove callback being discarded with `CONFIG_VIDEO_ET8EK8=y`. When such a device gets unbound (e.g. using `sysfs` or `hotplug`), the driver is just removed without the cleanup being performed. This results in resource leaks. Fix it by compiling in the remove callback unconditionally.

This also fixes a W=1 modpost warning:

WARNING: modpost: drivers/media/i2c/et8ek8/et8ek8: section mismatch in reference: et8ek8\_i2c\_driver+0x10 (section: .data) -> et8ek8\_remove (section: .exit.text)

More Info: <https://avd.aquasec.com/nvd/cve-2024-38611>

### **[CVE-2024-38620] kernel: Bluetooth: HCI: Remove HCI\_AMP support (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: HCI: Remove HCI\_AMP support

Since BT\_HS has been remove HCI\_AMP controllers no longer has any use so remove it along with the capability of creating AMP controllers.

Since we no longer need to differentiate between AMP and Primary controllers, as only HCI\_PRIMARY is left, this also remove `hdev->dev_type` altogether.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38620>

### **[CVE-2024-38622] kernel: drm/msm/dpu: Add callback function pointer check before its call (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/msm/dpu: Add callback function pointer check before its call

In `dpu_core_irq_callback_handler()` callback function pointer is compared to `NULL`, but then callback function is unconditionally called by this pointer. Fix this bug by adding conditional return.

Found by Linux Verification Center ([linuxtesting.org](https://linuxtesting.org)) with SVACE.

Patchwork: <https://patchwork.freedesktop.org/patch/588237/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-38622>

### **[CVE-2024-38625] kernel: fs/ntfs3: Check `folio` pointer for NULL (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fs/ntfs3: Check 'folio' pointer for NULL

It can be NULL if bmap is called.

More Info: <https://avd.aquasec.com/nvd/cve-2024-38625>

### **[CVE-2024-40965] kernel: i2c: lpi2c: Avoid calling clk\_get\_rate during transfer (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

i2c: lpi2c: Avoid calling clk\_get\_rate during transfer

Instead of repeatedly calling clk\_get\_rate for each transfer, lock the clock rate and cache the value.

A deadlock has been observed while adding tlv320aic32x4 audio codec to the system. When this clock provider adds its clock, the clk mutex is locked already, it needs to access i2c, which in return needs the mutex for clk\_get\_rate as well.

More Info: <https://avd.aquasec.com/nvd/cve-2024-40965>

### **[CVE-2024-40969] kernel: f2fs: don't set RO when shutting down f2fs (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: don't set RO when shutting down f2fs

Shutdown does not check the error of thaw\_super due to readonly, which causes a deadlock like below.

```
f2fs_ioc_shutdown(F2FS_GOING_DOWN_FULLSYNC)    issue_discard_thread
- bdev_freeze
- freeze_super
- f2fs_stop_checkpoint()
- f2fs_handle_critical_error                    - sb_start_write
  - set RO                                    - waiting
- bdev_thaw
- thaw_super_locked
  - return -EINVAL, if sb_rdonly()
- f2fs_stop_discard_thread
```

-> wait for kthread\_stop(discard\_thread);

More Info: <https://avd.aquasec.com/nvd/cve-2024-40969>

**[CVE-2024-40973] kernel: media: mtk-vcodec: potential null pointer deference in SCP (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: mtk-vcodec: potential null pointer deference in SCP

The return value of devm\_kzalloc() needs to be checked to avoid NULL pointer deference. This is similar to CVE-2022-3113.

More Info: <https://avd.aquasec.com/nvd/cve-2024-40973>

**[CVE-2024-40975] kernel: platform/x86: x86-android-tablets: Unregister devices in reverse order (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

platform/x86: x86-android-tablets: Unregister devices in reverse order

Not all subsystems support a device getting removed while there are still consumers of the device with a reference to the device.

One example of this is the regulator subsystem. If a regulator gets unregistered while there are still drivers holding a reference a WARN() at drivers/regulator/core.c:5829 triggers, e.g.:

WARNING: CPU: 1 PID: 1587 at drivers/regulator/core.c:5829 regulator\_unregister  
Hardware name: Intel Corp. VALLEYVIEW C0 PLATFORM/BYT-T FFD8, BIOS BLADE\_21.X64.0005.R00.1504101516  
FFD8\_X64\_R\_2015\_04\_10\_1516 04/10/2015  
RIP: 0010:regulator\_unregister  
Call Trace:  
<TASK>  
regulator\_unregister  
devres\_release\_group  
i2c\_device\_remove  
device\_release\_driver\_internal  
bus\_remove\_device  
device\_del  
device\_unregister  
x86\_android\_tablet\_remove

On the Lenovo Yoga Tablet 2 series the bq24190 charger chip also provides

a 5V boost converter output for powering USB devices connected to the micro USB port, the bq24190-charger driver exports this as a Vbus regulator.

On the 830 (8") and 1050 ("10") models this regulator is controlled by a platform\_device and x86\_android\_tablet\_remove() removes platform\_device-s before i2c\_clients so the consumer gets removed first.

But on the 1380 (13") model there is a lc824206xa micro-USB switch connected over I2C and the extcon driver for that controls the regulator. The bq24190 i2c-client *\*must\** be registered first, because that creates the regulator with the lc824206xa listed as its consumer. If the regulator has not been registered yet the lc824206xa driver will end up getting a dummy regulator.

Since in this case both the regulator provider and consumer are I2C devices, the only way to ensure that the consumer is unregistered first is to unregister the I2C devices in reverse order of in which they were created.

For consistency and to avoid similar problems in the future change x86\_android\_tablet\_remove() to unregister all device types in reverse order.

More Info: <https://avd.aquasec.com/nvd/cve-2024-40975>

**[CVE-2024-40998] kernel: ext4: fix uninitialized ratelimit\_state->lock access in \_\_ext4\_fill\_super() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ext4: fix uninitialized ratelimit\_state->lock access in \_\_ext4\_fill\_super()

In the following concurrency we will access the uninitialized rs->lock:

```
ext4_fill_super
ext4_register_sysfs
    // sysfs registered msg_ratelimit_interval_ms
        // Other processes modify rs->interval to
        // non-zero via msg_ratelimit_interval_ms
ext4_orphan_cleanup
ext4_msg(sb, KERN_INFO, "Errors on filesystem, "
__ext4_msg
    __ratelimit(&(EXT4_SB(sb)->s_msg_ratelimit_state)
    if (!rs->interval) // do nothing if interval is 0
        return 1;
    raw_spin_trylock_irqsave(&rs->lock, flags)
    raw_spin_trylock(lock)
    __raw_spin_trylock
    __raw_spin_trylock
```



```

spin_acquire(&lock->dep_map, 0, 1, _RET_IP_)
lock_acquire
__lock_acquire
register_lock_class
assign_lock_key
dump_stack();
ratelimit_state_init(&sbi->s_msg_ratelimit_state, 5 * HZ, 10);
raw_spin_lock_init(&rs->lock);
// init rs->lock here

```

and get the following dump\_stack:

```

=====
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: 12 PID: 753 Comm: mount Tainted: G E 6.7.0-rc6-next-20231222 #504
[...]
Call Trace:
dump_stack_lvl+0xc5/0x170
dump_stack+0x18/0x30
register_lock_class+0x740/0x7c0
__lock_acquire+0x69/0x13a0
lock_acquire+0x120/0x450
_raw_spin_trylock+0x98/0xd0
__ratelimit+0xf6/0x220
__ext4_msg+0x7f/0x160 [ext4]
ext4_orphan_cleanup+0x665/0x740 [ext4]
__ext4_fill_super+0x21ea/0x2b10 [ext4]
ext4_fill_super+0x14d/0x360 [ext4]
[...]
=====

```

Normally interval is 0 until s\_msg\_ratelimit\_state is initialized, so \_\_ratelimit() does nothing. But registering sysfs precedes initializing rs->lock, so it is possible to change rs->interval to a non-zero value via the msg\_ratelimit\_interval\_ms interface of sysfs while rs->lock is uninitialized, and then a call to ext4\_msg triggers the problem by accessing an uninitialized rs->lock. Therefore register sysfs after all initializations are complete to avoid such problems.

More Info: <https://avd.aquasec.com/nvd/cve-2024-40998>

**[CVE-2024-40999] kernel: net: ena: Add validation for completion descriptors consistency (Severity: MEDIUM)**

Package: linux-libc-dev  
 Installed: 6.1.129-1  
 Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: ena: Add validation for completion descriptors consistency

Validate that `first` flag is set only for the first descriptor in multi-buffer packets.

In case of an invalid descriptor, a reset will occur.

A new reset reason for RX data corruption has been added.

More Info: <https://avd.aquasec.com/nvd/cve-2024-40999>

### **[CVE-2024-41008] kernel: drm/amdgpu: change vm->task\_info handling (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: change vm->task\_info handling

This patch changes the handling and lifecycle of vm->task\_info object.

The major changes are:

- vm->task\_info is a dynamically allocated ptr now, and its usage is reference counted.
- introducing two new helper funcs for task\_info lifecycle management
  - amdgpu\_vm\_get\_task\_info: reference counts up task\_info before returning this info
  - amdgpu\_vm\_put\_task\_info: reference counts down task\_info
- last put to task\_info() frees task\_info from the vm.

This patch also does logistical changes required for existing usage of vm->task\_info.

V2: Do not block all the prints when task\_info not found (Felix)

V3: Fixed review comments from Felix

- Fix wrong indentation
- No debug message for -ENOMEM
- Add NULL check for task\_info
- Do not duplicate the debug messages (ti vs no ti)
- Get first reference of task\_info in vm\_init(), put last in vm\_fini()

V4: Fixed review comments from Felix

- fix double reference increment in create\_task\_info
- change amdgpu\_vm\_get\_task\_info\_pasid
- additional changes in amdgpu\_gem.c while porting

More Info: <https://avd.aquasec.com/nvd/cve-2024-41008>

### **[CVE-2024-41023] kernel: sched/deadline: Fix task\_struct reference leak (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

sched/deadline: Fix task\_struct reference leak

During the execution of the following stress test with linux-rt:

```
stress-ng --cyclic 30 --timeout 30 --minimize --quiet
```

kmemleak frequently reported a memory leak concerning the task\_struct:

unreferenced object 0xffff8881305b8000 (size 16136):

comm "stress-ng", pid 614, jiffies 4294883961 (age 286.412s)

object hex dump (first 32 bytes):

```
02 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .@.....
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 
```

debug hex dump (first 16 bytes):

```
53 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 S.....
```

backtrace:

```
[<00000000046b6790>] dup_task_struct+0x30/0x540
```

```
[<00000000c5ca0f0b>] copy_process+0x3d9/0x50e0
```

```
[<00000000ced59777>] kernel_clone+0xb0/0x770
```

```
[<00000000a50befdc>] __do_sys_clone+0xb6/0xf0
```

```
[<000000001dbf2008>] do_syscall_64+0x5d/0xf0
```

```
[<00000000552900ff>] entry_SYSCALL_64_after_hwframe+0x6e/0x76
```

The issue occurs in start\_dl\_timer(), which increments the task\_struct reference count and sets a timer. The timer callback, dl\_task\_timer, is supposed to decrement the reference count upon expiration. However, if enqueue\_task\_dl() is called before the timer expires and cancels it, the reference count is not decremented, leading to the leak.

This patch fixes the reference leak by ensuring the task\_struct reference count is properly decremented when the timer is canceled.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41023>

### **[CVE-2024-41031] kernel: mm/filemap: skip to create PMD-sized page cache if needed (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/filemap: skip to create PMD-sized page cache if needed

On ARM64, HPAGE\_PMD\_ORDER is 13 when the base page size is 64KB. The PMD-sized page cache can't be supported by xarray as the following error messages indicate.

-----[ cut here ]-----

WARNING: CPU: 35 PID: 7484 at lib/xarray.c:1025 xas\_split\_alloc+0xf8/0x128  
Modules linked in: nft\_fib\_inet nft\_fib\_ipv4 nft\_fib\_ipv6 nft\_fib \nft\_reject\_inet nf\_reject\_ipv4 nf\_reject\_ipv6 nft\_reject nft\_ct \nft\_chain\_nat nf\_nat nf\_conntrack nf\_defrag\_ipv6 nf\_defrag\_ipv4 \nip\_set rkill nf\_tables nfnetlink vfat fat virtio\_balloon drm \nfuse xfs libcrc32c crct10dif\_ce ghash\_ce sha2\_ce sha256\_arm64 \sha1\_ce virtio\_net net\_failover virtio\_console virtio\_blk failover \dimlib virtio\_mmio  
CPU: 35 PID: 7484 Comm: test Kdump: loaded Tainted: G W 6.10.0-rc5-gavin+ #9  
Hardware name: QEMU KVM Virtual Machine, BIOS edk2-20240524-1.el9 05/24/2024  
pstate: 83400005 (Nzcv daif +PAN -UAO +TCO +DIT -SSBS BTYPE=--)  
pc : xas\_split\_alloc+0xf8/0x128  
lr : split\_huge\_page\_to\_list\_to\_order+0x1c4/0x720  
sp : ffff800087a4f6c0  
x29: ffff800087a4f6c0 x28: ffff800087a4f720 x27: 000000001ffffff  
x26: 0000000000000c40 x25: 000000000000000d x24: ffff00010625b858  
x23: ffff800087a4f720 x22: fffffdfc0780000 x21: 0000000000000000  
x20: 0000000000000000 x19: fffffdfc0780000 x18: 000000001ff40000  
x17: 00000000ffffff x16: 0000018000000000 x15: 51ec004000000000  
x14: 0000e00000000000 x13: 0000000000002000 x12: 0000000000000020  
x11: 51ec000000000000 x10: 51ece1c0fff8000 x9: ffffb961a44d28  
x8: 0000000000000003 x7: fffffdfc0456420 x6: ffff0000e1aa6eb8  
x5: 20bf08b4fe778fca x4: fffffdfc0456420 x3: 0000000000000c40  
x2: 000000000000000d x1: 000000000000000c x0: 0000000000000000  
Call trace:  
xas\_split\_alloc+0xf8/0x128  
split\_huge\_page\_to\_list\_to\_order+0x1c4/0x720  
truncate\_inode\_partial\_folio+0xdc/0x160  
truncate\_inode\_pages\_range+0x1b4/0x4a8  
truncate\_pagecache\_range+0x84/0xa0  
xfs\_flush\_unmap\_range+0x70/0x90 [xfs]  
xfs\_file\_fallocate+0xfc/0x4d8 [xfs]  
vfs\_fallocate+0x124/0x2e8  
ksys\_fallocate+0x4c/0xa0  
\_\_arm64\_sys\_fallocate+0x24/0x38  
invoke\_syscall.constprop.0+0x7c/0xd8  
do\_el0\_svc+0xb4/0xd0  
el0\_svc+0x44/0x1d8  
el0t\_64\_sync\_handler+0x134/0x150  
el0t\_64\_sync+0x17c/0x180

Fix it by skipping to allocate PMD-sized page cache when its size is larger than MAX\_PAGECACHE\_ORDER. For this specific case, we will fall to regular path where the readahead window is determined by BDI's sysfs file (read\_ahead\_kb).

More Info: <https://avd.aquasec.com/nvd/cve-2024-41031>

### **[CVE-2024-41045] kernel: bpf: Defer work in bpf\_timer\_cancel\_and\_free (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Defer work in bpf\_timer\_cancel\_and\_free

Currently, the same case as previous patch (two timer callbacks trying to cancel each other) can be invoked through bpf\_map\_update\_elem as well, or more precisely, freeing map elements containing timers. Since this relies on hrtimer\_cancel as well, it is prone to the same deadlock situation as the previous patch.

It would be sufficient to use hrtimer\_try\_to\_cancel to fix this problem, as the timer cannot be enqueued after async\_cancel\_and\_free. Once async\_cancel\_and\_free has been done, the timer must be reinitialized before it can be armed again. The callback running in parallel trying to arm the timer will fail, and freeing bpf\_hrtimer without waiting is sufficient (given kfree\_rcu), and bpf\_timer\_cb will return HRTIMER\_NORESTART, preventing the timer from being rearmed again.

However, there exists a UAF scenario where the callback arms the timer before entering this function, such that if cancellation fails (due to timer callback invoking this routine, or the target timer callback running concurrently). In such a case, if the timer expiration is significantly far in the future, the RCU grace period expiration happening before it will free the bpf\_hrtimer state and along with it the struct hrtimer, that is enqueued.

Hence, it is clear cancellation needs to occur after async\_cancel\_and\_free, and yet it cannot be done inline due to deadlock issues. We thus modify bpf\_timer\_cancel\_and\_free to defer work to the global workqueue, adding a work\_struct alongside rcu\_head (both used at \_different\_ points of time, so can share space).

Update existing code comments to reflect the new state of affairs.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41045>

### **[CVE-2024-41067] kernel: btrfs: scrub: handle RST lookup error correctly (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: scrub: handle RST lookup error correctly

[BUG]

When running btrfs/060 with forced RST feature, it would crash the following ASSERT() inside scrub\_read\_endio():

```
ASSERT(sector_nr < stripe->nr_sectors);
```

Before that, we would have tree dump from btrfs\_get\_raid\_extent\_offset(), as we failed to find the RST entry for

the range.

#### [CAUSE]

Inside `scrub_submit_extent_sector_read()` every time we allocated a new `bbio` we immediately called `btrfs_map_block()` to make sure there was some RST range covering the scrub target.

But if `btrfs_map_block()` fails, we immediately call `endio` for the `bbio`, while the `bbio` is newly allocated, it's completely empty.

Then inside `scrub_read_endio()`, we go through the `bvecs` to find the sector number (as `bi_sector` is no longer reliable if the `bio` is submitted to lower layers).

And since the `bio` is empty, such `bvecs` iteration would not find any sector matching the sector, and return `sector_nr == stripe->nr_sectors`, triggering the `ASSERT()`.

#### [FIX]

Instead of calling `btrfs_map_block()` after allocating a new `bbio`, call `btrfs_map_block()` first.

Since our only objective of calling `btrfs_map_block()` is only to update `stripe_len`, there is really no need to do that after `btrfs_alloc_bio()`.

This new timing would avoid the problem of handling empty `bbio` completely, and in fact fixes a possible race window for the old code, where if the submission thread is the only owner of the `pending_io`, the scrub would never finish (since we didn't decrease the `pending_io` counter).

Although the root cause of RST lookup failure still needs to be addressed.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41067>

### **[CVE-2024-41082] kernel: nvme-fabrics: use reserved tag for reg read/write command (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

`nvme-fabrics: use reserved tag for reg read/write command`

In some scenarios, if too many commands are issued by `nvme` command in the same time by user tasks, this may exhaust all tags of `admin_q`. If a reset (`nvme` reset or IO timeout) occurs before these commands finish, reconnect routine may fail to update `nvme` regs due to insufficient tags, which will cause kernel hang forever. In order to workaround this issue, maybe we can let `reg_read32()/reg_read64()/reg_write32()` use reserved

tags. This maybe safe for nvme:

1. For the disable ctrl path, we will not issue connect command
2. For the enable ctrl / fw activate path, since connect and reg\_xx() are called serially.

So the reserved tags may still be enough while reg\_xx() use reserved tags.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41082>

### **[CVE-2024-41935] kernel: f2fs: fix to shrink read extent node in batches (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to shrink read extent node in batches

We use rwlock to protect core structure data of extent tree during its shrink, however, if there is a huge number of extent nodes in extent tree, during shrink of extent tree, it may hold rwlock for a very long time, which may trigger kernel hang issue.

This patch fixes to shrink read extent node in batches, so that, critical region of the rwlock can be shrunk to avoid its extreme long time hold.

More Info: <https://avd.aquasec.com/nvd/cve-2024-41935>

### **[CVE-2024-42067] kernel: bpf: Take return from set\_memory\_rox() into account with bpf\_jit\_binary\_lock\_ro() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Take return from set\_memory\_rox() into account with bpf\_jit\_binary\_lock\_ro()

set\_memory\_rox() can fail, leaving memory unprotected.

Check return and bail out when bpf\_jit\_binary\_lock\_ro() returns an error.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42067>

### **[CVE-2024-42079] kernel: gfs2: Fix NULL pointer dereference in gfs2\_log\_flush (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

gfs2: Fix NULL pointer dereference in gfs2\_log\_flush

In gfs2\_jindex\_free(), set sdp->sd\_jdesc to NULL under the log flush lock to provide exclusion against gfs2\_log\_flush().

In gfs2\_log\_flush(), check if sdp->sd\_jdesc is non-NULL before dereferencing it. Otherwise, we could run into a NULL pointer dereference when outstanding glock work races with an unmount (glock\_work\_func -> run\_queue -> do\_xmote -> inode\_go\_sync -> gfs2\_log\_flush).

More Info: <https://avd.aquasec.com/nvd/cve-2024-42079>

### **[CVE-2024-42107] kernel: ice: Don't process extts if PTP is disabled (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ice: Don't process extts if PTP is disabled

The ice\_ptp\_extts\_event() function can race with ice\_ptp\_release() and result in a NULL pointer dereference which leads to a kernel panic.

Panic occurs because the ice\_ptp\_extts\_event() function calls ptp\_clock\_event() with a NULL pointer. The ice driver has already released the PTP clock by the time the interrupt for the next external timestamp event occurs.

To fix this, modify the ice\_ptp\_extts\_event() function to check the PTP state and bail early if PTP is not ready.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42107>

### **[CVE-2024-42118] kernel: drm/amd/display: Do not return negative stream id for array (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Do not return negative stream id for array

[WHY]

resource\_stream\_to\_stream\_idx returns an array index and it return -1 when not found; however, -1 is not a valid array index number.

[HOW]



When this happens, call `ASSERT()`, and return a zero instead.

This fixes an `OVERRUN` and an `NEGATIVE_RETURNS` issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42118>

**[CVE-2024-42123] kernel: drm/amdgpu: fix double free err\_addr pointer warnings (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: fix double free err\_addr pointer warnings

In `amdgpu_umc_bad_page_polling_timeout`, the `amdgpu_umc_handle_bad_pages` will be run many times so that double free `err_addr` in some special case. So set the `err_addr` to `NULL` to avoid the warnings.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42123>

**[CVE-2024-42125] kernel: wifi: rtw89: fw: scan offload prohibit all 6 GHz channel if no 6 GHz sband (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: rtw89: fw: scan offload prohibit all 6 GHz channel if no 6 GHz sband

We have some policy via BIOS to block uses of 6 GHz. In this case, 6 GHz sband will be `NULL` even if it is WiFi 7 chip. So, add `NULL` handling here to avoid crash.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42125>

**[CVE-2024-42128] kernel: leds: an30259a: Use devm\_mutex\_init() for mutex initialization (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

leds: an30259a: Use `devm_mutex_init()` for mutex initialization

In this driver LEDs are registered using `devm_led_classdev_register()` so they are automatically unregistered after module's `remove()` is done. `led_classdev_unregister()` calls module's `led_set_brightness()` to turn off the LEDs and that callback uses mutex which was destroyed already in module's `remove()` so use `devm` API instead.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42128>

**[CVE-2024-42129] kernel: leds: mlxreg: Use devm\_mutex\_init() for mutex initialization (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

leds: mlxreg: Use devm\_mutex\_init() for mutex initialization

In this driver LEDs are registered using devm\_led\_classdev\_register() so they are automatically unregistered after module's remove() is done. led\_classdev\_unregister() calls module's led\_set\_brightness() to turn off the LEDs and that callback uses mutex which was destroyed already in module's remove() so use devm API instead.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42129>

**[CVE-2024-42135] kernel: vhost\_task: Handle SIGKILL by flushing work and exiting (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

vhost\_task: Handle SIGKILL by flushing work and exiting

Instead of lingering until the device is closed, this has us handle SIGKILL by:

1. marking the worker as killed so we no longer try to use it with new virtqueues and new flush operations.
2. setting the virtqueue to worker mapping so no new works are queued.
3. running all the exiting works.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42135>

**[CVE-2024-42139] kernel: ice: Fix improper extts handling (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ice: Fix improper extts handling

Extts events are disabled and enabled by the application ts2phc. However, in case where the driver is removed when the application is

running, a specific extts event remains enabled and can cause a kernel crash.

As a side effect, when the driver is reloaded and application is started again, remaining extts event for the channel from a previous run will keep firing and the message "extts on unexpected channel" might be printed to the user.

To avoid that, extts events shall be disabled when PTP is released.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42139>

### **[CVE-2024-42156] kernel: s390/pkey: Wipe copies of clear-key structures on failure (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

s390/pkey: Wipe copies of clear-key structures on failure

Wipe all sensitive data from stack for all IOCTLs, which convert a clear-key into a protected- or secure-key.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42156>

### **[CVE-2024-42158] kernel: s390/pkey: Use kfree\_sensitive() to fix Coccinelle warnings (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

s390/pkey: Use kfree\_sensitive() to fix Coccinelle warnings

Replace memzero\_explicit() and kfree() with kfree\_sensitive() to fix warnings reported by Coccinelle:

WARNING opportunity for kfree\_sensitive/kvfree\_sensitive (line 1506)

WARNING opportunity for kfree\_sensitive/kvfree\_sensitive (line 1643)

WARNING opportunity for kfree\_sensitive/kvfree\_sensitive (line 1770)

More Info: <https://avd.aquasec.com/nvd/cve-2024-42158>

### **[CVE-2024-42239] kernel: bpf: Fail bpf\_timer\_cancel when callback is being cancelled (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Fail bpf\_timer\_cancel when callback is being cancelled

Given a schedule:

timer1 cb timer2 cb

```
bpf_timer_cancel(timer2); bpf_timer_cancel(timer1);
```

Both bpf\_timer\_cancel calls would wait for the other callback to finish executing, introducing a lockup.

Add an atomic\_t count named 'cancelling' in bpf\_hrtimer. This keeps track of all in-flight cancellation requests for a given BPF timer. Whenever cancelling a BPF timer, we must check if we have outstanding cancellation requests, and if so, we must fail the operation with an error (-EDEADLK) since cancellation is synchronous and waits for the callback to finish executing. This implies that we can enter a deadlock situation involving two or more timer callbacks executing in parallel and attempting to cancel one another.

Note that we avoid incrementing the cancelling counter for the target timer (the one being cancelled) if bpf\_timer\_cancel is not invoked from a callback, to avoid spurious errors. The whole point of detecting cur->cancelling and returning -EDEADLK is to not enter a busy wait loop (which may or may not lead to a lockup). This does not apply in case the caller is in a non-callback context, the other side can continue to cancel as it sees fit without running into errors.

Background on prior attempts:

Earlier versions of this patch used a bool 'cancelling' bit and used the following pattern under timer->lock to publish cancellation status.

```
lock(t->lock);
t->cancelling = true;
mb();
if (cur->cancelling)
    return -EDEADLK;
unlock(t->lock);
hrtimer_cancel(t->timer);
t->cancelling = false;
```

The store outside the critical section could overwrite a parallel requests t->cancelling assignment to true, to ensure the parallelly executing callback observes its cancellation status.

It would be necessary to clear this cancelling bit once hrtimer\_cancel is done, but lack of serialization introduced races. Another option was explored where bpf\_timer\_start would clear the bit when (re)starting the timer under timer->lock. This would ensure serialized access to the cancelling bit, but may allow it to be cleared before in-flight hrtimer\_cancel has finished executing, such that lockups can occur

again.

Thus, we choose an atomic counter to keep track of all outstanding cancellation requests and use it to prevent lockups in case callbacks attempt to cancel each other while executing in parallel.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42239>

### **[CVE-2024-42241] kernel: mm/shmem: disable PMD-sized page cache if needed (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/shmem: disable PMD-sized page cache if needed

For shmem files, it's possible that PMD-sized page cache can't be supported by xarray. For example, 512MB page cache on ARM64 when the base page size is 64KB can't be supported by xarray. It leads to errors as the following messages indicate when this sort of xarray entry is split.

```
WARNING: CPU: 34 PID: 7578 at lib/xarray.c:1025 xas_split_alloc+0xf8/0x128
Modules linked in: binfmt_misc nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 \
nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject \
nft_ct nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 \
ip_set rkill nf_tables nfnetlink vfat fat virtio_balloon drm fuse xfs \
libcrc32c crct10dif_ce ghash_ce sha2_ce sha256_arm64 sha1_ce virtio_net \
net_failover virtio_console virtio_blk failover dimlib virtio_mmio
CPU: 34 PID: 7578 Comm: test Kdump: loaded Tainted: G W 6.10.0-rc5-gavin+ #9
Hardware name: QEMU KVM Virtual Machine, BIOS edk2-20240524-1.el9 05/24/2024
pstate: 83400005 (Nzcv daif +PAN -UAO +TCO +DIT -SSBS BTYPE=--)
pc : xas_split_alloc+0xf8/0x128
lr : split_huge_page_to_list_to_order+0x1c4/0x720
sp : ffff8000882af5f0
x29: ffff8000882af5f0 x28: ffff8000882af650 x27: ffff8000882af768
x26: 00000000000000cc0 x25: 000000000000000d x24: ffff00010625b858
x23: ffff8000882af650 x22: fffffdfc09000000 x21: 0000000000000000
x20: 0000000000000000 x19: fffffdfc09000000 x18: 0000000000000000
x17: 0000000000000000 x16: 0000018000000000 x15: 52f8004000000000
x14: 0000e00000000000 x13: 0000000000002000 x12: 0000000000000020
x11: 52f8000000000000 x10: 52f8e1c0fff6000 x9 : ffffbef9619a681c
x8 : 0000000000000003 x7 : 0000000000000000 x6 : ffff00010b02ddb0
x5 : ffffbef96395e378 x4 : 0000000000000000 x3 : 00000000000000cc0
x2 : 000000000000000d x1 : 000000000000000c x0 : 0000000000000000
Call trace:
xas_split_alloc+0xf8/0x128
split_huge_page_to_list_to_order+0x1c4/0x720
truncate_inode_partial_folio+0xdc/0x160
shmem_undo_range+0x2bc/0x6a8
shmem_fallocate+0x134/0x430
vfs_fallocate+0x124/0x2e8
ksys_fallocate+0x4c/0xa0
```

```
__arm64_sys_fallocate+0x24/0x38
invoke_syscall.constprop.0+0x7c/0xd8
do_el0_svc+0xb4/0xd0
el0_svc+0x44/0x1d8
el0t_64_sync_handler+0x134/0x150
el0t_64_sync+0x17c/0x180
```

Fix it by disabling PMD-sized page cache when HPAGE\_PMD\_ORDER is larger than MAX\_PAGECACHE\_ORDER. As Matthew Wilcox pointed, the page cache in a shmem file isn't represented by a multi-index entry and doesn't have this limitation when the xarray entry is split until commit 6b24ca4a1a8d ("mm: Use multi-index entries in the page cache").

More Info: <https://avd.aquasec.com/nvd/cve-2024-42241>

**[CVE-2024-42243] kernel: mm/filemap: make MAX\_PAGECACHE\_ORDER acceptable to xarray (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/filemap: make MAX\_PAGECACHE\_ORDER acceptable to xarray

Patch series "mm/filemap: Limit page cache size to that supported by xarray", v2.

Currently, xarray can't support arbitrary page cache size. More details can be found from the WARN\_ON() statement in xas\_split\_alloc(). In our test whose code is attached below, we hit the WARN\_ON() on ARM64 system where the base page size is 64KB and huge page size is 512MB. The issue was reported long time ago and some discussions on it can be found here [1].

[1] <https://www.spinics.net/lists/linux-xfs/msg75404.html>

In order to fix the issue, we need to adjust MAX\_PAGECACHE\_ORDER to one supported by xarray and avoid PMD-sized page cache if needed. The code changes are suggested by David Hildenbrand.

PATCH[1] adjusts MAX\_PAGECACHE\_ORDER to that supported by xarray  
PATCH[2-3] avoids PMD-sized page cache in the synchronous readahead path  
PATCH[4] avoids PMD-sized page cache for shmem files if needed

Test program

=====

```
# cat test.c
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

```

#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/syscall.h>
#include <sys/mman.h>

#define TEST_XFS_FILENAME "/tmp/data"
#define TEST_SHMEM_FILENAME "/dev/shm/data"
#define TEST_MEM_SIZE 0x20000000

int main(int argc, char **argv)
{
    const char *filename;
    int fd = 0;
    void *buf = (void *)-1, *p;
    int pgsz = getpagesize();
    int ret;

    if (pgsz != 0x10000) {
        fprintf(stderr, "64KB base page size is required\n");
        return -EPERM;
    }

    system("echo force > /sys/kernel/mm/transparent_hugepage/shmem_enabled");
    system("rm -fr /tmp/data");
    system("rm -fr /dev/shm/data");
    system("echo 1 > /proc/sys/vm/drop_caches");

    /* Open xfs or shmem file */
    filename = TEST_XFS_FILENAME;
    if (argc > 1 && !strcmp(argv[1], "shmem"))
        filename = TEST_SHMEM_FILENAME;

    fd = open(filename, O_CREAT | O_RDWR | O_TRUNC);
    if (fd < 0) {
        fprintf(stderr, "Unable to open <%s>\n", filename);
        return -EIO;
    }

    /* Extend file size */
    ret = ftruncate(fd, TEST_MEM_SIZE);
    if (ret) {
        fprintf(stderr, "Error %d to ftruncate()\n", ret);
        goto cleanup;
    }

    /* Create VMA */
    buf = mmap(NULL, TEST_MEM_SIZE,
        PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
    if (buf == (void *)-1) {
        fprintf(stderr, "Unable to mmap <%s>\n", filename);
        goto cleanup;
    }
}

```

```

fprintf(stdout, "mapped buffer at 0x%p\n", buf);
ret = madvise(buf, TEST_MEM_SIZE, MADV_HUGEPAGE);
    if (ret) {
        fprintf(stderr, "Unable to madvise(MADV_HUGEPAGE)\n");
        goto cleanup;
    }

/* Populate VMA */
ret = madvise(buf, TEST_MEM_SIZE, MADV_POPULATE_WRITE);
if (ret) {
    fprintf(stderr, "Error %d to madvise(MADV_POPULATE_WRITE)\n", ret);
    goto cleanup;
}

/* Punch the file to enforce xarray split */
ret = fallocate(fd, FALLOC_FL_KEEP_SIZE | FALLOC_FL_PUNCH_HOLE,
    TEST_MEM_SIZE - pgsz, pgsz);
if (ret)
    fprintf(stderr, "Error %d to fallocate()\n", ret);

cleanup:
if (buf != (void *)-1)
    munmap(buf, TEST_MEM_SIZE);
if (fd > 0)
    close(fd);

return 0;
}

# gcc test.c -o test
# cat /proc/1/smmaps | grep KernelPageSize | head -n 1
KernelPageSize:      64 kB
# ./test shmem
:
-----[ cut here ]-----
WARNING: CPU: 17 PID: 5253 at lib/xarray.c:1025 xas_split_alloc+0xf8/0x128
Modules linked in: nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib \
nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct \
nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 \
ip_set nf_tables rfkill nfnetlink vfat fat virtio_balloon \
drm fuse xfs libcrc32c crct10dif_ce ghash_ce sha2_ce sha256_arm64 \
virtio_net sha1_ce net_failover failover virtio_console virtio_blk \
dimlib virtio_mmio
CPU: 17 PID: 5253 Comm: test Kdump: loaded Tainted: G W 6.10.0-rc5-gavin+ #12
Hardware name: QEMU KVM Virtual Machine, BIOS edk2-20240524-1.el9 05/24/2024
pstate: 83400005 (Nzcv daif +PAN -UAO +TC
---truncated---

More Info: https://avd.aquasec.com/nvd/cve-2024-42243

```

**[CVE-2024-42279] kernel: spi: microchip-core: ensure TX and RX FIFOs are empty at start of a transfer (Severity: MEDIUM)**



Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

spi: microchip-core: ensure TX and RX FIFOs are empty at start of a transfer

While transmitting with rx\_len == 0, the RX FIFO is not going to be emptied in the interrupt handler. A subsequent transfer could then read crap from the previous transfer out of the RX FIFO into the start RX buffer. The core provides a register that will empty the RX and TX FIFOs, so do that before each transfer.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42279>

### **[CVE-2024-42317] kernel: mm/huge\_memory: avoid PMD-size page cache if needed (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/huge\_memory: avoid PMD-size page cache if needed

xarray can't support arbitrary page cache size. the largest and supported page cache size is defined as MAX\_PAGECACHE\_ORDER by commit 099d90642a71 ("mm/filemap: make MAX\_PAGECACHE\_ORDER acceptable to xarray"). However, it's possible to have 512MB page cache in the huge memory's collapsing path on ARM64 system whose base page size is 64KB. 512MB page cache is breaking the limitation and a warning is raised when the xarray entry is split as shown in the following example.

```
[root@dhcp-10-26-1-207 ~]# cat /proc/1/smaps | grep KernelPageSize
```

```
KernelPageSize:      64 kB
```

```
[root@dhcp-10-26-1-207 ~]# cat /tmp/test.c
```

```
:
int main(int argc, char **argv)
{
    const char *filename = TEST_XFS_FILENAME;
    int fd = 0;
    void *buf = (void *)-1, *p;
    int pgsz = getpagesize();
    int ret = 0;

    if (pgsz != 0x10000) {
        fprintf(stdout, "System with 64KB base page size is required!\n");
        return -EPERM;
    }
}
```

```
system("echo 0 > /sys/devices/virtual/bdi/253:0/read_ahead_kb");
```

```

system("echo 1 > /proc/sys/vm/drop_caches");

/* Open the xfs file */
fd = open(filename, O_RDONLY);
assert(fd > 0);

/* Create VMA */
buf = mmap(NULL, TEST_MEM_SIZE, PROT_READ, MAP_SHARED, fd, 0);
assert(buf != (void *)-1);
fprintf(stdout, "mapped buffer at 0x%p\n", buf);

/* Populate VMA */
ret = madvise(buf, TEST_MEM_SIZE, MADV_NOHUGEPAGE);
assert(ret == 0);
ret = madvise(buf, TEST_MEM_SIZE, MADV_POPULATE_READ);
assert(ret == 0);

/* Collapse VMA */
ret = madvise(buf, TEST_MEM_SIZE, MADV_HUGEPAGE);
assert(ret == 0);
ret = madvise(buf, TEST_MEM_SIZE, MADV_COLLAPSE);
if (ret) {
    fprintf(stdout, "Error %d to madvise(MADV_COLLAPSE)\n", errno);
    goto out;
}

/* Split xarray entry. Write permission is needed */
munmap(buf, TEST_MEM_SIZE);
buf = (void *)-1;
close(fd);
fd = open(filename, O_RDWR);
assert(fd > 0);
fallocate(fd, FALLOC_FL_KEEP_SIZE | FALLOC_FL_PUNCH_HOLE,
    TEST_MEM_SIZE - pgsz, pgsz);
out:
if (buf != (void *)-1)
    munmap(buf, TEST_MEM_SIZE);
if (fd > 0)
    close(fd);

return ret;
}

```

```
[root@dhcp-10-26-1-207 ~]# gcc /tmp/test.c -o /tmp/test
```

```
[root@dhcp-10-26-1-207 ~]# ./tmp/test
```

```
-----[ cut here ]-----
```

```
WARNING: CPU: 25 PID: 7560 at lib/xarray.c:1025 xas_split_alloc+0xf8/0x128
```

```

Modules linked in: nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib \
nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct \
nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 \
ip_set rfkill nf_tables nfnetlink vfat fat virtio_balloon drm fuse \
xfs libcrc32c crct10dif_ce ghash_ce sha2_ce sha256_arm64 virtio_net \
sha1_ce net_failover virtio_blk virtio_console failover dimlib virtio_mmio

```

CPU: 25 PID: 7560 Comm: test Kdump: loaded Not tainted 6.10.0-rc7-gavin+ #9  
Hardware name: QEMU KVM Virtual Machine, BIOS edk2-20240524-1.el9 05/24/2024  
pstate: 83400005 (Nzcv daif +PAN -UAO +TCO +DIT -SSBS BTYPE=--)  
pc : xas\_split\_alloc+0xf8/0x128  
lr : split\_huge\_page\_to\_list\_to\_order+0x1c4/0x780  
sp : ffff8000ac32f660  
x29: ffff8000ac32f660 x28: ffff0000e0969eb0 x27: ffff8000ac32f6c0  
x26: 0000000000000c40 x25: ffff0000e0969eb0 x24: 000000000000000d  
x23: ffff8000ac32f6c0 x22: fffffdfc0700000 x21: 0000000000000000  
x20: 0000000000000000 x19: fffffdfc0700000 x18: 0000000000000000  
x17: 0000000000000000 x16: ffffd5f3708ffc70 x15: 0000000000000000  
x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000  
x11: ffffffffcc0 x10: 0000000000000040 x9 : ffffd5f3708e692c  
x8 : 0000000000000003 x7 : 0000000000000000 x6 : ffff0000e0969eb8  
x5 : ffffd5f37289e378 x4 : 0000000000000000 x3 : 0000000000000c40  
x2 : 000000000000000d x1 : 000000000000000c x0 : 0000000000000000  
Call trace:  
xas\_split\_alloc+0xf8/0x128  
split\_huge\_page\_to\_list\_to\_order+0x1c4/0x780  
truncate\_inode\_partial\_folio+0xdc/0x160  
truncate\_inode\_pages\_range+0x1b4/0x4a8  
truncate\_pagecache\_range+0x84/0xa  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-42317>

**[CVE-2024-43819] kernel: kvm: s390: Reject memory region operations for ucontrol VMs (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

kvm: s390: Reject memory region operations for ucontrol VMs

This change rejects the KVM\_SET\_USER\_MEMORY\_REGION and KVM\_SET\_USER\_MEMORY\_REGION2 ioctls when called on a ucontrol VM. This is necessary since ucontrol VMs have kvm->arch.gmap set to 0 and would thus result in a null pointer dereference further in. Memory management needs to be performed in userspace and using the ioctls KVM\_S390\_UCAS\_MAP and KVM\_S390\_UCAS\_UNMAP.

Also improve s390 specific documentation for KVM\_SET\_USER\_MEMORY\_REGION and KVM\_SET\_USER\_MEMORY\_REGION2.

[frankja@linux.ibm.com: commit message spelling fix, subject prefix fix]

More Info: <https://avd.aquasec.com/nvd/cve-2024-43819>

**[CVE-2024-43824] kernel: PCI: endpoint: pci-epf-test: Make use of cached '&#39;epc\_features&#39; in pci\_epf\_test\_core\_init() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

PCI: endpoint: pci-epf-test: Make use of cached 'epc\_features' in pci\_epf\_test\_core\_init()

Instead of getting the epc\_features from pci\_epc\_get\_features() API, use the cached pci\_epf\_test::epc\_features value to avoid the NULL check. Since the NULL check is already performed in pci\_epf\_test\_bind(), having one more check in pci\_epf\_test\_core\_init() is redundant and it is not possible to hit the NULL pointer dereference.

Also with commit a01e7214bef9 ("PCI: endpoint: Remove "core\_init\_notifier" flag"), 'epc\_features' got dereferenced without the NULL check, leading to the following false positive Smatch warning:

drivers/pci/endpoint/functions/pci-epf-test.c:784 pci\_epf\_test\_core\_init() error: we previously assumed 'epc\_features' could be null (see line 747)

Thus, remove the redundant NULL check and also use the epc\_features::{msix\_capable/msi\_capable} flags directly to avoid local variables.

[kwilczynski: commit log]

More Info: <https://avd.aquasec.com/nvd/cve-2024-43824>

### **[CVE-2024-43831] kernel: media: mediatek: vcodec: Handle invalid decoder vsi (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: mediatek: vcodec: Handle invalid decoder vsi

Handle an invalid decoder vsi in vpu\_dec\_init to ensure the decoder vsi is valid for future use.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43831>

### **[CVE-2024-43840] kernel: bpf, arm64: Fix trampoline for BPF\_TRAMP\_F\_CALL\_ORIG (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf, arm64: Fix trampoline for BPF\_TRAMP\_F\_CALL\_ORIG

When BPF\_TRAMP\_F\_CALL\_ORIG is set, the trampoline calls `__bpf_tramp_enter()` and `__bpf_tramp_exit()` functions, passing them the struct `bpf_tramp_image *im` pointer as an argument in R0.

The trampoline generation code uses `emit_addr_mov_i64()` to emit instructions for moving the `bpf_tramp_image` address into R0, but `emit_addr_mov_i64()` assumes the address to be in the `vmalloc()` space and uses only 48 bits. Because `bpf_tramp_image` is allocated using `kzalloc()`, its address can use more than 48-bits, in this case the trampoline will pass an invalid address to `__bpf_tramp_enter/exit()` causing a kernel crash.

Fix this by using `emit_a64_mov_i64()` in place of `emit_addr_mov_i64()` as it can work with addresses that are greater than 48-bits.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43840>

### **[CVE-2024-43850] kernel: soc: qcom: icc-bwmon: Fix refcount imbalance seen during bwmon\_remove (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

soc: qcom: icc-bwmon: Fix refcount imbalance seen during bwmon\_remove

The following warning is seen during `bwmon_remove` due to refcount imbalance, fix this by releasing the OPPs after use.

Logs:

WARNING: at drivers/opp/core.c:1640 \_opp\_table\_kref\_release+0x150/0x158

Hardware name: Qualcomm Technologies, Inc. X1E80100 CRD (DT)

...

Call trace:

\_opp\_table\_kref\_release+0x150/0x158

dev\_pm\_opp\_remove\_table+0x100/0x1b4

devm\_pm\_opp\_of\_table\_release+0x10/0x1c

devm\_action\_release+0x14/0x20

devres\_release\_all+0xa4/0x104

device\_unbind\_cleanup+0x18/0x60

device\_release\_driver\_internal+0x1ec/0x228

driver\_detach+0x50/0x98

bus\_remove\_driver+0x6c/0xbc

driver\_unregister+0x30/0x60

platform\_driver\_unregister+0x14/0x20

bwmon\_driver\_exit+0x18/0x524 [icc\_bwmon]

\_\_arm64\_sys\_delete\_module+0x184/0x264

invoke\_syscall+0x48/0x118

el0\_svc\_common.constprop.0+0xc8/0xe8

do\_el0\_svc+0x20/0x2c

el0\_svc+0x34/0xdc

```
el0t_64_sync_handler+0x13c/0x158
el0t_64_sync+0x190/0x194
--[ end trace 0000000000000000 ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-43850>

**[CVE-2024-43872] kernel: RDMA/hns: Fix soft lockup under heavy CEQE load (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/hns: Fix soft lockup under heavy CEQE load

CEQEs are handled in interrupt handler currently. This may cause the CPU core staying in interrupt context too long and lead to soft lockup under heavy load.

Handle CEQEs in BH workqueue and set an upper limit for the number of CEQE handled by a single call of work handler.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43872>

**[CVE-2024-43886] kernel: drm/amd/display: Add null check in resource\_log\_pipe\_topology\_update (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add null check in resource\_log\_pipe\_topology\_update

[WHY]

When switching from "Extend" to "Second Display Only" we sometimes call resource\_get\_otg\_master\_for\_stream on a stream for the eDP, which is disconnected. This leads to a null pointer dereference.

[HOW]

Added a null check in dc\_resource.c/resource\_log\_pipe\_topology\_update.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43886>

**[CVE-2024-43899] kernel: drm/amd/display: Fix null pointer deref in dcn20\_resource.c (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix null pointer deref in dcn20\_resource.c

Fixes a hang thats triggered when MPV is run on a DCN401 dGPU:

```
mpv --hwdec=vaapi --vo=gpu --hwdec-codecs=all
```

and then enabling fullscreen playback (double click on the video)

The following calltrace will be seen:

```
[ 181.843989] BUG: kernel NULL pointer dereference, address: 0000000000000000
[ 181.843997] #PF: supervisor instruction fetch in kernel mode
[ 181.844003] #PF: error_code(0x0010) - not-present page
[ 181.844009] PGD 0 P4D 0
[ 181.844020] Oops: 0010 [#1] PREEMPT SMP NOPTI
[ 181.844028] CPU: 6 PID: 1892 Comm: gnome-shell Tainted: G      W OE      6.5.0-41-generic #41~22.04.2-Ubuntu
[ 181.844038] Hardware name: System manufacturer System Product Name/CROSSHAIR VI HERO, BIOS 6302
10/23/2018
[ 181.844044] RIP: 0010:0x0
[ 181.844079] Code: Unable to access opcode bytes at 0xffffffffffffd6.
[ 181.844084] RSP: 0018:ffffb593c2b8f7b0 EFLAGS: 00010246
[ 181.844093] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000004
[ 181.844099] RDX: fffffb593c2b8f804 RSI: fffffb593c2b8f7e0 RDI: ffff9e3c8e758400
[ 181.844105] RBP: fffffb593c2b8f7b8 R08: fffffb593c2b8f9c8 R09: fffffb593c2b8f96c
[ 181.844110] R10: 0000000000000000 R11: 0000000000000000 R12: fffffb593c2b8f9c8
[ 181.844115] R13: 0000000000000000 R14: ffff9e3c88000000 R15: 0000000000000005
[ 181.844121] FS: 00007c6e323bb5c0(0000) GS:ffff9e3f85f80000(0000) knlGS:0000000000000000
[ 181.844128] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 181.844134] CR2: fffffb593c2b8f7b8 CR3: 0000000140f80000 CR4: 00000000003506e0
[ 181.844141] Call Trace:
[ 181.844146] <TASK>
[ 181.844153] ? show_regs+0x6d/0x80
[ 181.844167] ? __die+0x24/0x80
[ 181.844179] ? page_fault_oops+0x99/0x1b0
[ 181.844192] ? do_user_addr_fault+0x31d/0x6b0
[ 181.844204] ? exc_page_fault+0x83/0x1b0
[ 181.844216] ? asm_exc_page_fault+0x27/0x30
[ 181.844237] dcn20_get_dcc_compression_cap+0x23/0x30 [amdgpu]
[ 181.845115] amdgpu_dm_plane_validate_dcc.constprop.0+0xe5/0x180 [amdgpu]
[ 181.845985] amdgpu_dm_plane_fill_plane_buffer_attributes+0x300/0x580 [amdgpu]
[ 181.846848] fill_dc_plane_info_and_addr+0x258/0x350 [amdgpu]
[ 181.847734] fill_dc_plane_attributes+0x162/0x350 [amdgpu]
[ 181.848748] dm_update_plane_state.constprop.0+0x4e3/0x6b0 [amdgpu]
[ 181.849791] ? dm_update_plane_state.constprop.0+0x4e3/0x6b0 [amdgpu]
[ 181.850840] amdgpu_dm_atomic_check+0xdfc/0x1760 [amdgpu]
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-43899>

**[CVE-2024-43901] kernel: drm/amd/display: Fix NULL pointer dereference for DTN log in DCN401 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix NULL pointer dereference for DTN log in DCN401

When users run the command:

```
cat /sys/kernel/debug/dri/0/amdgpu_dm_dtn_log
```

The following NULL pointer dereference happens:

```
[ +0.000003] BUG: kernel NULL pointer dereference, address: NULL
[ +0.000005] #PF: supervisor instruction fetch in kernel mode
[ +0.000002] #PF: error_code(0x0010) - not-present page
[ +0.000002] PGD 0 P4D 0
[ +0.000004] Oops: 0010 [#1] PREEMPT SMP NOPTI
[ +0.000003] RIP: 0010:0x0
[ +0.000008] Code: Unable to access opcode bytes at 0xffffffffffffd6.
[...]
```

```
[ +0.000002] PKRU: 55555554
[ +0.000002] Call Trace:
[ +0.000002] <TASK>
[ +0.000003] ? show_regs+0x65/0x70
[ +0.000006] ? __die+0x24/0x70
[ +0.000004] ? page_fault_oops+0x160/0x470
[ +0.000006] ? do_user_addr_fault+0x2b5/0x690
[ +0.000003] ? prb_read_valid+0x1c/0x30
[ +0.000005] ? exc_page_fault+0x8c/0x1a0
[ +0.000005] ? asm_exc_page_fault+0x27/0x30
[ +0.000012] dcn10_log_color_state+0xf9/0x510 [amdgpu]
[ +0.000306] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000003] ? vsnprintf+0x2fb/0x600
[ +0.000009] dcn10_log_hw_state+0xfd0/0xfe0 [amdgpu]
[ +0.000218] ? __mod_memcg_lruvec_state+0xe8/0x170
[ +0.000008] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000002] ? debug_smp_processor_id+0x17/0x20
[ +0.000003] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000002] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000002] ? set_ptes.isra.0+0x2b/0x90
[ +0.000004] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000002] ? _raw_spin_unlock+0x19/0x40
[ +0.000004] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000002] ? do_anonymous_page+0x337/0x700
[ +0.000004] dtn_log_read+0x82/0x120 [amdgpu]
[ +0.000207] full_proxy_read+0x66/0x90
[ +0.000007] vfs_read+0xb0/0x340
[ +0.000005] ? __count_memcg_events+0x79/0xe0
[ +0.000002] ? srso_alias_return_thunk+0x5/0xfbef5
[ +0.000003] ? count_memcg_events.constprop.0+0x1e/0x40
[ +0.000003] ? handle_mm_fault+0xb2/0x370
[ +0.000003] ksys_read+0x6b/0xf0
[ +0.000004] __x64_sys_read+0x19/0x20
```



```
[ +0.000003] do_syscall_64+0x60/0x130
[ +0.000004] entry_SYSCALL_64_after_hwframe+0x6e/0x76
[ +0.000003] RIP: 0033:0x7fd32f147e2
[...]
```

This error happens when the color log tries to read the gamut remap information from DCN401 which is not initialized in the dcn401\_dpp\_funcs which leads to a null pointer dereference. This commit addresses this issue by adding a proper guard to access the gamut\_remap callback in case the specific ASIC did not implement this function.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43901>

### **[CVE-2024-43913] kernel: nvme: apple: fix device reference counting (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nvme: apple: fix device reference counting

Drivers must call nvme\_uninit\_ctrl after a successful nvme\_init\_ctrl. Split the allocation side out to make the error handling boundary easier to navigate. The apple driver had been doing this wrong, leaking the controller device memory on a tagset failure.

More Info: <https://avd.aquasec.com/nvd/cve-2024-43913>

### **[CVE-2024-44955] kernel: drm/amd/display: Don't refer to dc\_sink in is\_dsc\_need\_re\_compute (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Don't refer to dc\_sink in is\_dsc\_need\_re\_compute

[Why]

When unplug one of monitors connected after mst hub, encounter null pointer dereference.

It's due to dc\_sink get released immediately in early\_unregister() or detect\_ctx(). When commit new state which directly referring to info stored in dc\_sink will cause null pointer dereference.

[how]

Remove redundant checking condition. Relevant condition should already be covered by checking if dsc\_aux is null or not. Also reset dsc\_aux to NULL when the connector is disconnected.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44955>

## **[CVE-2024-44957] kernel: xen: privcmd: Switch from mutex to spinlock for irqfds (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

xen: privcmd: Switch from mutex to spinlock for irqfds

irqfd\_wakeup() gets EPOLLHUP, when it is called by eventfd\_release() by way of wake\_up\_poll(&ctx->wqh, EPOLLHUP), which gets called under spin\_lock\_irqsave(). We can't use a mutex here as it will lead to a deadlock.

Fix it by switching over to a spin lock.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44957>

## **[CVE-2024-44961] kernel: drm/amdgpu: Forward soft recovery errors to userspace (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Forward soft recovery errors to userspace

As we discussed before[1], soft recovery should be forwarded to userspace, or we can get into a really bad state where apps will keep submitting hanging command buffers cascading us to a hard reset.

1: <https://lore.kernel.org/all/bf23d5ed-9a6b-43e7-84ee-8cbfd0d60f18@froggi.es/>  
(cherry picked from commit 434967aadbbbe3ad9103cc29e9a327de20fdb01)

More Info: <https://avd.aquasec.com/nvd/cve-2024-44961>

## **[CVE-2024-44963] kernel: btrfs: do not BUG\_ON() when freeing tree block after error (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: do not BUG\_ON() when freeing tree block after error

When freeing a tree block, at btrfs\_free\_tree\_block(), if we fail to create a delayed reference we don't deal with the error and just do a BUG\_ON(). The error most likely to happen is -ENOMEM, and we have a comment mentioning that only -ENOMEM can happen, but that is not true,

because in case qgroups are enabled any error returned from `btrfs_qgroup_trace_extent_post()` (can be `-EUCLEAN` or anything returned from `btrfs_search_slot()` for example) can be propagated back to `btrfs_free_tree_block()`.

So stop doing a `BUG_ON()` and return the error to the callers and make them abort the transaction to prevent leaking space. Syzbot was triggering this, likely due to memory allocation failure injection.

More Info: <https://avd.aquasec.com/nvd/cve-2024-44963>

## **[CVE-2024-44972] kernel: btrfs: do not clear page dirty inside extent\_write\_locked\_range() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: do not clear page dirty inside `extent_write_locked_range()`

### **[BUG]**

For subpage + zoned case, the following workload can lead to rsv data leak at unmount time:

```
# mkfs.btrfs -f -s 4k $dev
# mount $dev $mnt
# fsstress -w -n 8 -d $mnt -s 1709539240
0/0: fiemap - no filename
0/1: copyrange read - no filename
0/2: write - no filename
0/3: rename - no source filename
0/4: creat f0 x:0 0 0
0/4: creat add id=0,parent=-1
0/5: writev f0[259 1 0 0 0 0] [778052,113,965] 0
0/6: ioctl(FIEMAP) f0[259 1 0 0 224 887097] [1294220,2291618343991484791,0x10000] -1
0/7: dwrite - xfsctl(XFS_IOC_DIOINFO) f0[259 1 0 0 224 887097] return 25, fallback to stat()
0/7: dwrite f0[259 1 0 0 224 887097] [696320,102400] 0
# umount $mnt
```

The dmesg includes the following rsv leak detection warning (all call trace skipped):

```
-----[ cut here ]-----
WARNING: CPU: 2 PID: 4528 at fs/btrfs/inode.c:8653 btrfs_destroy_inode+0x1e0/0x200 [btrfs]
---[ end trace 0000000000000000 ]---
-----[ cut here ]-----
WARNING: CPU: 2 PID: 4528 at fs/btrfs/inode.c:8654 btrfs_destroy_inode+0x1a8/0x200 [btrfs]
---[ end trace 0000000000000000 ]---
-----[ cut here ]-----
WARNING: CPU: 2 PID: 4528 at fs/btrfs/inode.c:8660 btrfs_destroy_inode+0x1a0/0x200 [btrfs]
---[ end trace 0000000000000000 ]---
```

```

BTRFS info (device sda): last unmount of filesystem 1b4abba9-de34-4f07-9e7f-157cf12a18d6
-----[ cut here ]-----
WARNING: CPU: 3 PID: 4528 at fs/btrfs/block-group.c:4434 btrfs_free_block_groups+0x338/0x500 [btrfs]
---[ end trace 0000000000000000 ]---
BTRFS info (device sda): space_info DATA has 268218368 free, is not full
  BTRFS info (device sda): space_info total=268435456, used=204800, pinned=0, reserved=0, may_use=12288,
readonly=0 zone_unusable=0
  BTRFS info (device sda): global_block_rsv: size 0 reserved 0
  BTRFS info (device sda): trans_block_rsv: size 0 reserved 0
  BTRFS info (device sda): chunk_block_rsv: size 0 reserved 0
  BTRFS info (device sda): delayed_block_rsv: size 0 reserved 0
  BTRFS info (device sda): delayed_refs_rsv: size 0 reserved 0
-----[ cut here ]-----
WARNING: CPU: 3 PID: 4528 at fs/btrfs/block-group.c:4434 btrfs_free_block_groups+0x338/0x500 [btrfs]
---[ end trace 0000000000000000 ]---
BTRFS info (device sda): space_info METADATA has 267796480 free, is not full
  BTRFS info (device sda): space_info total=268435456, used=131072, pinned=0, reserved=0, may_use=262144,
readonly=0 zone_unusable=245760
  BTRFS info (device sda): global_block_rsv: size 0 reserved 0
  BTRFS info (device sda): trans_block_rsv: size 0 reserved 0
  BTRFS info (device sda): chunk_block_rsv: size 0 reserved 0
  BTRFS info (device sda): delayed_block_rsv: size 0 reserved 0
  BTRFS info (device sda): delayed_refs_rsv: size 0 reserved 0

```

More Info: <https://avd.aquasec.com/nvd/cve-2024-44972>

**[CVE-2024-45015] kernel: drm/msm/dpu: move dpu\_encoder's connector assignment to atomic\_enable() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/msm/dpu: move dpu\_encoder's connector assignment to atomic\_enable()

For cases where the crtc's connectors\_changed was set without enable/active getting toggled , there is an atomic\_enable() call followed by an atomic\_disable() but without an atomic\_mode\_set().

This results in a NULL ptr access for the dpu\_encoder\_get\_drm\_fmt() call in the atomic\_enable() as the dpu\_encoder's connector was cleared in the atomic\_disable() but not re-assigned as there was no atomic\_mode\_set() call.

Fix the NULL ptr access by moving the assignment for atomic\_enable() and also use drm\_atomic\_get\_new\_connector\_for\_encoder() to get the connector from the atomic\_state.

Patchwork: <https://patchwork.freedesktop.org/patch/606729/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-45015>

**[CVE-2024-46678] kernel: bonding: change ipsec\_lock from spin lock to mutex (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bonding: change ipsec\_lock from spin lock to mutex

In the cited commit, bond->ipsec\_lock is added to protect ipsec\_list, hence xdo\_dev\_state\_add and xdo\_dev\_state\_delete are called inside this lock. As ipsec\_lock is a spin lock and such xfrmdev ops may sleep, "scheduling while atomic" will be triggered when changing bond's active slave.

```
[ 101.055189] BUG: scheduling while atomic: bash/902/0x00000200
[ 101.055726] Modules linked in:
[ 101.058211] CPU: 3 PID: 902 Comm: bash Not tainted 6.9.0-rc4+ #1
[ 101.058760] Hardware name:
[ 101.059434] Call Trace:
[ 101.059436]  <TASK>
[ 101.060873]  dump_stack_lvl+0x51/0x60
[ 101.061275]  __schedule_bug+0x4e/0x60
[ 101.061682]  __schedule+0x612/0x7c0
```

```
[ 101.062078] ? __mod_timer+0x25c/0x370
[ 101.062486] schedule+0x25/0xd0
[ 101.062845] schedule_timeout+0x77/0xf0
[ 101.063265] ? asm_common_interrupt+0x22/0x40
[ 101.063724] ? __bpf_trace_itimer_state+0x10/0x10
[ 101.064215] __wait_for_common+0x87/0x190
[ 101.064648] ? usleep_range_state+0x90/0x90
[ 101.065091] cmd_exec+0x437/0xb20 [mlx5_core]
[ 101.065569] mlx5_cmd_do+0x1e/0x40 [mlx5_core]
[ 101.066051] mlx5_cmd_exec+0x18/0x30 [mlx5_core]
[ 101.066552] mlx5_crypto_create_dek_key+0xea/0x120 [mlx5_core]
[ 101.067163] ? bonding_sysfs_store_option+0x4d/0x80 [bonding]
[ 101.067738] ? kmalloc_trace+0x4d/0x350
[ 101.068156] mlx5_ipsec_create_sa_ctx+0x33/0x100 [mlx5_core]
[ 101.068747] mlx5e_xfrm_add_state+0x47b/0xaa0 [mlx5_core]
[ 101.069312] bond_change_active_slave+0x392/0x900 [bonding]
[ 101.069868] bond_option_active_slave_set+0x1c2/0x240 [bonding]
[ 101.070454] __bond_opt_set+0xa6/0x430 [bonding]
[ 101.070935] __bond_opt_set_notify+0x2f/0x90 [bonding]
[ 101.071453] bond_opt_tryset_rtnl+0x72/0xb0 [bonding]
[ 101.071965] bonding_sysfs_store_option+0x4d/0x80 [bonding]
[ 101.072567] kernfs_fop_write_iter+0x10c/0x1a0
[ 101.073033] vfs_write+0x2d8/0x400
[ 101.073416] ? alloc_fd+0x48/0x180
[ 101.073798] ksys_write+0x5f/0xe0
[ 101.074175] do_syscall_64+0x52/0x110
[ 101.074576] entry_SYSCALL_64_after_hwframe+0x4b/0x53
```

As bond\_ipsec\_add\_sa\_all and bond\_ipsec\_del\_sa\_all are only called from bond\_change\_active\_slave, which requires holding the RTNL lock. And bond\_ipsec\_add\_sa and bond\_ipsec\_del\_sa are xfrm state xdo\_dev\_state\_add and xdo\_dev\_state\_delete APIs, which are in user context. So ipsec\_lock doesn't have to be spin lock, change it to mutex, and thus the above issue can be resolved.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46678>

### **[CVE-2024-46681] kernel: pktgen: use cpus\_read\_lock() in pg\_net\_init() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

pktgen: use cpus\_read\_lock() in pg\_net\_init()

I have seen the WARN\_ON(smp\_processor\_id() != cpu) firing in pktgen\_thread\_worker() during tests.

We must use cpus\_read\_lock()/cpus\_read\_unlock() around the for\_each\_online\_cpu(cpu) loop.

While we are at it use WARN\_ON\_ONCE() to avoid a possible syslog flood.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46681>

**[CVE-2024-46698] kernel: video/aperture: optionally match the device in sysfb\_disable() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

video/aperture: optionally match the device in sysfb\_disable()

In aperture\_remove\_conflicting\_pci\_devices(), we currently only call sysfb\_disable() on vga class devices. This leads to the following problem when the primary device is not VGA compatible:

1. A PCI device with a non-VGA class is the boot display
2. That device is probed first and it is not a VGA device so sysfb\_disable() is not called, but the device resources are freed by aperture\_detach\_platform\_device()
3. Non-primary GPU has a VGA class and it ends up calling sysfb\_disable()
4. NULL pointer dereference via sysfb\_disable() since the resources have already been freed by aperture\_detach\_platform\_device() when it was called by the other device.

Fix this by passing a device pointer to sysfb\_disable() and checking the device to determine if we should execute it or not.

v2: Fix build when CONFIG\_SCREEN\_INFO is not set

v3: Move device check into the mutex

Drop primary variable in aperture\_remove\_conflicting\_pci\_devices()

Drop \_\_init on pci\_sysfb\_pci\_dev\_is\_enabled()

More Info: <https://avd.aquasec.com/nvd/cve-2024-46698>

**[CVE-2024-46727] kernel: drm/amd/display: Add otg\_master NULL check within resource\_log\_pipe\_topology\_update (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add otg\_master NULL check within resource\_log\_pipe\_topology\_update

[Why]

Coverity reports NULL\_RETURN warning.

[How]

Add otg\_master NULL check.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46727>

**[CVE-2024-46728] kernel: drm/amd/display: Check index for aux\_rd\_interval before using (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Check index for aux\_rd\_interval before using

aux\_rd\_interval has size of 7 and should be checked.

This fixes 3 OVERRUN and 1 INTEGER\_OVERFLOW issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46728>

**[CVE-2024-46729] kernel: drm/amd/display: Fix incorrect size calculation for loop (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix incorrect size calculation for loop

[WHY]

fe\_clk\_en has size of 5 but sizeof(fe\_clk\_en) has byte size 20 which is larger than the array size.

[HOW]

Divide byte size 20 by its element size.

This fixes 2 OVERRUN issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46729>

**[CVE-2024-46730] kernel: drm/amd/display: Ensure array index tg\_inst won't be -1 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Ensure array index tg\_inst won't be -1

[WHY & HOW]

tg\_inst will be a negative if timing\_generator\_count equals 0, which should be checked before used.



This fixes 2 OVERRUN issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46730>

**[CVE-2024-46733] kernel: btrfs: fix qgroup reserve leaks in cow\_file\_range (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: fix qgroup reserve leaks in cow\_file\_range

In the buffered write path, the dirty page owns the qgroup reserve until it creates an ordered\_extent.

Therefore, any errors that occur before the ordered\_extent is created must free that reservation, or else the space is leaked. The fstest generic/475 exercises various IO error paths, and is able to trigger errors in cow\_file\_range where we fail to get to allocating the ordered extent. Note that because we \*do\* clear delalloc, we are likely to remove the inode from the delalloc list, so the inodes/pages to not have invalidate/laundry called on them in the commit abort path.

This results in failures at the unmount stage of the test that look like:

```
BTRFS: error (device dm-8 state EA) in cleanup_transaction:2018: errno=-5 IO failure
BTRFS: error (device dm-8 state EA) in btrfs_replace_file_extents:2416: errno=-5 IO failure
BTRFS warning (device dm-8 state EA): qgroup 0/5 has unreleased space, type 0 rsv 28672
-----[ cut here ]-----
WARNING: CPU: 3 PID: 22588 at fs/btrfs/disk-io.c:4333 close_ctree+0x222/0x4d0 [btrfs]
Modules linked in: btrfs blake2b_generic libcrc32c xor zstd_compress raid6_pq
CPU: 3 PID: 22588 Comm: umount Kdump: loaded Tainted: G W      6.10.0-rc7-gab56fde445b8 #21
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Arch Linux 1.16.3-1-1 04/01/2014
RIP: 0010:close_ctree+0x222/0x4d0 [btrfs]
RSP: 0018:ffffb4465283be00 EFLAGS: 00010202
RAX: 0000000000000001 RBX: ffffa1a1818e1000 RCX: 0000000000000001
RDX: 0000000000000000 RSI: ffffb4465283bbe0 RDI: ffffa1a19374fcb8
RBP: ffffa1a1818e13c0 R08: 0000000100028b16 R09: 0000000000000000
R10: 0000000000000003 R11: 0000000000000003 R12: ffffa1a18ad7972c
R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
FS: 00007f9168312b80(0000) GS:ffffa1a4afcc0000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007f91683c9140 CR3: 000000010acaa000 CR4: 000000000000006f0
Call Trace:
<TASK>
? close_ctree+0x222/0x4d0 [btrfs]
? __warn.cold+0x8e/0xea
? close_ctree+0x222/0x4d0 [btrfs]
? report_bug+0xff/0x140
? handle_bug+0x3b/0x70
? exc_invalid_op+0x17/0x70
? asm_exc_invalid_op+0x1a/0x20
```

```
? close_ctree+0x222/0x4d0 [btrfs]
generic_shutdown_super+0x70/0x160
kill_anon_super+0x11/0x40
btrfs_kill_super+0x11/0x20 [btrfs]
deactivate_locked_super+0x2e/0xa0
cleanup_mnt+0xb5/0x150
task_work_run+0x57/0x80
syscall_exit_to_user_mode+0x121/0x130
do_syscall_64+0xab/0x1a0
entry_SYSCALL_64_after_hwframe+0x77/0x7f
RIP: 0033:0x7f916847a887
---[ end trace 0000000000000000 ]---
BTRFS error (device dm-8 state EA): qgroup reserved space leaked
```

Cases 2 and 3 in the out\_reserve path both pertain to this type of leak and must free the reserved qgroup data. Because it is already an error path, I opted not to handle the possible errors in btrfs\_free\_qgroup\_data.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46733>

**[CVE-2024-46742] kernel: smb/server: fix potential null-ptr-deref of lease\_ctx\_info in smb2\_open() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb/server: fix potential null-ptr-deref of lease\_ctx\_info in smb2\_open()

null-ptr-deref will occur when (req\_op\_level == SMB2\_OPLOCK\_LEVEL\_LEASE) and parse\_lease\_state() return NULL.

Fix this by check if 'lease\_ctx\_info' is NULL.

Additionally, remove the redundant parentheses in parse\_durable\_handle\_context().

More Info: <https://avd.aquasec.com/nvd/cve-2024-46742>

**[CVE-2024-46748] kernel: cachefiles: Set the max subreq size for cache writes to MAX\_RW\_COUNT (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

cachefiles: Set the max subreq size for cache writes to MAX\_RW\_COUNT

Set the maximum size of a subrequest that writes to cachefiles to be

MAX\_RW\_COUNT so that we don't overrun the maximum write we can make to the backing filesystem.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46748>

**[CVE-2024-46751] kernel: btrfs: don't BUG\_ON() when 0 reference count at btrfs\_lookup\_extent\_info() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: don't BUG\_ON() when 0 reference count at btrfs\_lookup\_extent\_info()

Instead of doing a BUG\_ON() handle the error by returning -EUCLEAN, aborting the transaction and logging an error message.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46751>

**[CVE-2024-46753] kernel: btrfs: handle errors from btrfs\_dec\_ref() properly (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: handle errors from btrfs\_dec\_ref() properly

In walk\_up\_proc() we BUG\_ON(ret) from btrfs\_dec\_ref(). This is incorrect, we have proper error handling here, return the error.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46753>

**[CVE-2024-46754] kernel: bpf: Remove tst\_run from lwt\_seg6local\_prog\_ops. (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Remove tst\_run from lwt\_seg6local\_prog\_ops.

The syzbot reported that the lwt\_seg6 related BPF ops can be invoked via bpf\_test\_run() without without entering input\_action\_end\_bpf() first.

Martin KaFai Lau said that self test for BPF\_PROG\_TYPE\_LWT\_SEG6LOCAL probably didn't work since it was introduced in commit 04d4b274e2a ("ipv6: sr: Add seg6local action End.BPF"). The reason is that the per-CPU variable seg6\_bpf\_srh\_states::srh is never assigned in the self test case but each BPF function expects it.

Remove test\_run for BPF\_PROG\_TYPE\_LWT\_SEG6LOCAL.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46754>

**[CVE-2024-46760] kernel: wifi: rtw88: usb: schedule rx work after everything is set up (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: rtw88: usb: schedule rx work after everything is set up

Right now it's possible to hit NULL pointer dereference in rtw\_rx\_fill\_rx\_status on hw object and/or its fields because initialization routine can start getting USB replies before rtw\_dev is fully setup.

The stack trace looks like this:

```
rtw_rx_fill_rx_status
rtw8821c_query_rx_desc
rtw_usb_rx_handler
...
queue_work
rtw_usb_read_port_complete
...
usb_submit_urb
rtw_usb_rx_resubmit
rtw_usb_init_rx
rtw_usb_probe
```

So while we do the async stuff rtw\_usb\_probe continues and calls rtw\_register\_hw, which does all kinds of initialization (e.g. via ieee80211\_register\_hw) that rtw\_rx\_fill\_rx\_status relies on.

Fix this by moving the first usb\_submit\_urb after everything is set up.

For me, this bug manifested as:

```
[ 8.893177] rtw_8821cu 1-1:1.2: band wrong, packet dropped
[ 8.910904] rtw_8821cu 1-1:1.2: hw->conf.chandef.chan NULL in rtw_rx_fill_rx_status
because I'm using Larry's backport of rtw88 driver with the NULL
checks in rtw_rx_fill_rx_status.
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-46760>

**[CVE-2024-46762] kernel: xen: privcmd: Fix possible access to a freed kirqfd instance (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

xen: privcmd: Fix possible access to a freed irqfd instance

Nothing prevents simultaneous ioctl calls to privcmd\_irqfd\_assign() and privcmd\_irqfd\_deassign(). If that happens, it is possible that a irqfd created and added to the irqfds\_list by privcmd\_irqfd\_assign() may get removed by another thread executing privcmd\_irqfd\_deassign(), while the former is still using it after dropping the locks.

This can lead to a situation where an already freed irqfd instance may be accessed and cause kernel oops.

Use SRCU locking to prevent the same, as is done for the KVM implementation for irqfds.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46762>

### **[CVE-2024-46765] kernel: ice: protect XDP configuration with a mutex (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ice: protect XDP configuration with a mutex

The main threat to data consistency in ice\_xdp() is a possible asynchronous PF reset. It can be triggered by a user or by TX timeout handler.

XDP setup and PF reset code access the same resources in the following sections:

- \* ice\_vsi\_close() in ice\_prepare\_for\_reset() - already rtnl-locked
- \* ice\_vsi\_rebuild() for the PF VSI - not protected
- \* ice\_vsi\_open() - already rtnl-locked

With an unfortunate timing, such accesses can result in a crash such as the one below:

```
[ +1.999878] ice 0000:b1:00.0: Registered XDP mem model MEM_TYPE_XSK_BUFF_POOL on Rx ring 14
[ +2.002992] ice 0000:b1:00.0: Registered XDP mem model MEM_TYPE_XSK_BUFF_POOL on Rx ring 18
[Mar15 18:17] ice 0000:b1:00.0 ens801f0np0: NETDEV WATCHDOG: CPU: 38: transmit queue 14 timed out 80692736
ms
[ +0.000093] ice 0000:b1:00.0 ens801f0np0: tx_timeout: VSI_num: 6, Q 14, NTC: 0x0, HW_HEAD: 0x0, NTU: 0x0, INT:
0x4000001
[ +0.000012] ice 0000:b1:00.0 ens801f0np0: tx_timeout recovery level 1, txqueue 14
[ +0.394718] ice 0000:b1:00.0: PTP reset successful
[ +0.006184] BUG: kernel NULL pointer dereference, address: 0000000000000098
[ +0.000045] #PF: supervisor read access in kernel mode
[ +0.000023] #PF: error_code(0x0000) - not-present page
```

```

[ +0.000023] PGD 0 P4D 0
[ +0.000018] Oops: 0000 [#1] PREEMPT SMP NOPTI
[ +0.000023] CPU: 38 PID: 7540 Comm: kworker/38:1 Not tainted 6.8.0-rc7 #1
[      +0.000031]      Hardware      name:      Intel      Corporation      S2600WFT/S2600WFT,      BIOS
SE5C620.86B.02.01.0014.082620210524 08/26/2021
[ +0.000036] Workqueue: ice ice_service_task [ice]
[ +0.000183] RIP: 0010:ice_clean_tx_ring+0xa/0xd0 [ice]
[...]
[ +0.000013] Call Trace:
[ +0.000016] <TASK>
[ +0.000014] ? __die+0x1f/0x70
[ +0.000029] ? page_fault_oops+0x171/0x4f0
[ +0.000029] ? schedule+0x3b/0xd0
[ +0.000027] ? exc_page_fault+0x7b/0x180
[ +0.000022] ? asm_exc_page_fault+0x22/0x30
[ +0.000031] ? ice_clean_tx_ring+0xa/0xd0 [ice]
[ +0.000194] ice_free_tx_ring+0xe/0x60 [ice]
[ +0.000186] ice_destroy_xdp_rings+0x157/0x310 [ice]
[ +0.000151] ice_vsi_decfg+0x53/0xe0 [ice]
[ +0.000180] ice_vsi_rebuild+0x239/0x540 [ice]
[ +0.000186] ice_vsi_rebuild_by_type+0x76/0x180 [ice]
[ +0.000145] ice_rebuild+0x18c/0x840 [ice]
[ +0.000145] ? delay_tsc+0x4a/0xc0
[ +0.000022] ? delay_tsc+0x92/0xc0
[ +0.000020] ice_do_reset+0x140/0x180 [ice]
[ +0.000886] ice_service_task+0x404/0x1030 [ice]
[ +0.000824] process_one_work+0x171/0x340
[ +0.000685] worker_thread+0x277/0x3a0
[ +0.000675] ? preempt_count_add+0x6a/0xa0
[ +0.000677] ? _raw_spin_lock_irqsave+0x23/0x50
[ +0.000679] ? __pfx_worker_thread+0x10/0x10
[ +0.000653] kthread+0xf0/0x120
[ +0.000635] ? __pfx_kthread+0x10/0x10
[ +0.000616] ret_from_fork+0x2d/0x50
[ +0.000612] ? __pfx_kthread+0x10/0x10
[ +0.000604] ret_from_fork_asm+0x1b/0x30
[ +0.000604] </TASK>

```

The previous way of handling this through returning -EBUSY is not viable, particularly when destroying AF\_XDP socket, because the kernel proceeds with removal anyway.

There is plenty of code between those calls and there is no need to create a large critical section that covers all of them, same as there is no need to protect ice\_vsi\_rebuild() with rtnl\_lock().

Add xdp\_state\_lock mutex to protect ice\_vsi\_rebuild() and ice\_xdp().

Leaving unprotected sections in between would result in two states that have to be considered:

1. when the VSI is closed, but not yet rebuild
2. when VSI is already rebuild, but not yet open

The latter case is actually already handled through !netif\_running() case, we just need to adjust flag checking a little. The former one is not as trivial, because between ice\_vsi\_close() and ice\_vsi\_rebuild(), a lot of hardware interaction happens, this can make adding/deleting rings exit with an error. Luckily, VSI rebuild is pending and can apply new configuration for us in a managed fashion.

Therefore, add an additional VSI state flag ICE\_VSI\_REBUILD\_PENDING to indicate that ice\_x  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-46765>

**[CVE-2024-46772] kernel: drm/amd/display: Check denominator crb\_pipes before used (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Check denominator crb\_pipes before used

[WHAT & HOW]

A denominator cannot be 0, and is checked before used.

This fixes 2 DIVIDE\_BY\_ZERO issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46772>

**[CVE-2024-46775] kernel: drm/amd/display: Validate function returns (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Validate function returns

[WHAT & HOW]

Function return values must be checked before data can be used in subsequent functions.

This fixes 4 CHECKED\_RETURN issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46775>

**[CVE-2024-46776] kernel: drm/amd/display: Run DC\_LOG\_DC after checking link->link\_enc (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Run DC\_LOG\_DC after checking link->link\_enc

[WHAT]

The DC\_LOG\_DC should be run after link->link\_enc is checked, not before.

This fixes 1 REVERSE\_NULL issue reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46776>

### **[CVE-2024-46787] kernel: userfaultfd: fix checks for huge PMDs (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

userfaultfd: fix checks for huge PMDs

Patch series "userfaultfd: fix races around pmd\_trans\_huge() check", v2.

The pmd\_trans\_huge() code in mfill\_atomic() is wrong in three different ways depending on kernel version:

1. The pmd\_trans\_huge() check is racy and can lead to a BUG\_ON() (if you hit the right two race windows) - I've tested this in a kernel build with some extra mdelay() calls. See the commit message for a description of the race scenario.  
On older kernels (before 6.5), I think the same bug can even theoretically lead to accessing transhuge page contents as a page table if you hit the right 5 narrow race windows (I haven't tested this case).
2. As pointed out by Qi Zheng, pmd\_trans\_huge() is not sufficient for detecting PMDs that don't point to page tables.  
On older kernels (before 6.5), you'd just have to win a single fairly wide race to hit this.  
I've tested this on 6.1 stable by racing migration (with a mdelay() patched into try\_to\_migrate()) against UFFDIO\_ZEROPAGE - on my x86 VM, that causes a kernel oops in ptlock\_ptr().
3. On newer kernels (>=6.5), for shmem mappings, khugepaged is allowed to yank page tables out from under us (though I haven't tested that), so I think the BUG\_ON() checks in mfill\_atomic() are just wrong.

I decided to write two separate fixes for these (one fix for bugs 1+2, one fix for bug 3), so that the first fix can be backported to kernels affected by bugs 1+2.

This patch (of 2):

This fixes two issues.



I discovered that the following race can occur:

```
mfill_atomic          other thread
=====              =====
                        <zap PMD>
pmdp_get_lockless() [reads none pmd]
<bail if trans_huge>
<if none:>
    <pagefault creates transhuge zeropage>
    __pte_alloc [no-op]
    <zap PMD>
    <bail if pmd_trans_huge(*dst_pmd)>
    BUG_ON(pmd_none(*dst_pmd))
```

I have experimentally verified this in a kernel with extra mdelay() calls; the BUG\_ON(pmd\_none(\*dst\_pmd)) triggers.

On kernels newer than commit 0d940a9b270b ("mm/pgtable: allow pte\_offset\_map[\_lock]() to fail"), this can't lead to anything worse than a BUG\_ON(), since the page table access helpers are actually designed to deal with page tables concurrently disappearing; but on older kernels (<=6.4), I think we could probably theoretically race past the two BUG\_ON() checks and end up treating a hugepage as a page table.

The second issue is that, as Qi Zheng pointed out, there are other types of huge PMDs that pmd\_trans\_huge() can't catch: devmap PMDs and swap PMDs (in particular, migration PMDs).

On <=6.4, this is worse than the first issue: If mfill\_atomic() runs on a PMD that contains a migration entry (which just requires winning a single, fairly wide race), it will pass the PMD to pte\_offset\_map\_lock(), which assumes that the PMD points to a page table.

Breakage follows: First, the kernel tries to take the PTE lock (which will crash or maybe worse if there is no "struct page" for the address bits in the migration entry PMD - I think at least on X86 there usually is no corresponding "struct page" thanks to the PTE inversion mitigation, amd64 looks different).

If that didn't crash, the kernel would next try to write a PTE into what it wrongly thinks is a page table.

As part of fixing these issues, get rid of the check for pmd\_trans\_huge() before \_\_pte\_alloc() - that's redundant, we're going to have to check for that after the \_\_pte\_alloc() anyway.

Backport note: pmdp\_get\_lockless() is pmd\_read\_atomic() in older kernels.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46787>

**[CVE-2024-46803] kernel: drm/amdkfd: Check debug trap enable before write dbg\_ev\_file (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdkfd: Check debug trap enable before write dbg\_ev\_file

In interrupt context, write dbg\_ev\_file will be run by work queue. It will cause write dbg\_ev\_file execution after debug\_trap\_disable, which will cause NULL pointer access.  
v2: cancel work "debug\_event\_workarea" before set dbg\_ev\_file as NULL.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46803>

**[CVE-2024-46806] kernel: drm/amdgpu: Fix the warning division or modulo by zero (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amdgpu: Fix the warning division or modulo by zero

Checks the partition mode and returns an error for an invalid mode.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46806>

**[CVE-2024-46808] kernel: drm/amd/display: Add missing NULL pointer check within dpcd\_extend\_address\_range (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add missing NULL pointer check within dpcd\_extend\_address\_range

[Why & How]  
ASSERT if return NULL from kcalloc.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46808>

**[CVE-2024-46816] kernel: drm/amd/display: Stop amdgpu\_dm initialize when link nums greater than max\_links (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Stop amdgpu\_dm initialize when link nums greater than max\_links

[Why]

Coverity report OVERRUN warning. There are only max\_links elements within dc->links. link count could up to AMDGPU\_DM\_MAX\_DISPLAY\_INDEX 31.

[How]

Make sure link count less than max\_links.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46816>

**[CVE-2024-46823] kernel: kunit/overflow: Fix UB in overflow\_allocation\_test (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

kunit/overflow: Fix UB in overflow\_allocation\_test

The 'device\_name' array doesn't exist out of the 'overflow\_allocation\_test' function scope. However, it is being used as a driver name when calling 'kunit\_driver\_create' from 'kunit\_device\_register'. It produces the kernel panic with KASAN enabled.

Since this variable is used in one place only, remove it and pass the device name into kunit\_device\_register directly as an ascii string.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46823>

**[CVE-2024-46825] kernel: wifi: iwlwifi: mvm: use IWL\_FW\_CHECK for link ID check (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: iwlwifi: mvm: use IWL\_FW\_CHECK for link ID check

The lookup function iwl\_mvm\_rcu\_fw\_link\_id\_to\_link\_conf() is normally called with input from the firmware, so it should use IWL\_FW\_CHECK() instead of WARN\_ON().

More Info: <https://avd.aquasec.com/nvd/cve-2024-46825>

**[CVE-2024-46834] kernel: ethtool: fail closed if we can't get max channel used in indirection tables (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ethtool: fail closed if we can't get max channel used in indirection tables

Commit 0d1b7d6c9274 ("bnxt: fix crashes when reducing ring count with active RSS contexts") proves that allowing indirection table to contain channels with out of bounds IDs may lead to crashes. Currently the max channel check in the core gets skipped if driver can't fetch the indirection table or when we can't allocate memory.

Both of those conditions should be extremely rare but if they do happen we should try to be safe and fail the channel change.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46834>

**[CVE-2024-46842] kernel: scsi: lpfc: Handle mailbox timeouts in lpfc\_get\_sfp\_info (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: lpfc: Handle mailbox timeouts in lpfc\_get\_sfp\_info

The MBX\_TIMEOUT return code is not handled in lpfc\_get\_sfp\_info and the routine unconditionally frees submitted mailbox commands regardless of return status. The issue is that for MBX\_TIMEOUT cases, when firmware returns SFP information at a later time, that same mailbox memory region references previously freed memory in its cmpl routine.

Fix by adding checks for the MBX\_TIMEOUT return code. During mailbox resource cleanup, check the mbox flag to make sure that the wait did not timeout. If the MBOX\_WAKE flag is not set, then do not free the resources because it will be freed when firmware completes the mailbox at a later time in its cmpl routine.

Also, increase the timeout from 30 to 60 seconds to accommodate boot scripts requiring longer timeouts.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46842>

**[CVE-2024-46843] kernel: scsi: ufs: core: Remove SCSI host only if added (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: ufs: core: Remove SCSI host only if added

If host tries to remove ufshcd driver from a UFS device it would cause a kernel panic if ufshcd\_async\_scan fails during ufshcd\_probe\_hba before adding a SCSI host with scsi\_add\_host and MCQ is enabled since SCSI host has been deferred after MCQ configuration introduced by commit 0cab4023ec7b ("scsi: ufs: core: Defer adding host to SCSI if MCQ is supported").

To guarantee that SCSI host is removed only if it has been added, set the scsi\_host\_added flag to true after adding a SCSI host and check whether it is set or not before removing it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46843>

**[CVE-2024-46860] kernel: wifi: mt76: mt7921: fix NULL pointer access in mt7921\_ipv6\_addr\_change (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: mt76: mt7921: fix NULL pointer access in mt7921\_ipv6\_addr\_change

When disabling wifi mt7921\_ipv6\_addr\_change() is called as a notifier.

At this point mvif->phy is already NULL so we cannot use it here.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46860>

**[CVE-2024-46870] kernel: drm/amd/display: Disable DMCUB timeout for DCN35 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Disable DMCUB timeout for DCN35

[Why]

DMCUB can intermittently take longer than expected to process commands.

Old ASIC policy was to continue while logging a diagnostic error - which works fine for ASIC without IPS, but with IPS this could lead to a race condition where we attempt to access DCN state while it's inaccessible, leading to a system hang when the NIU port is not disabled or register accesses that timeout and the display configuration in an undefined state.

[How]

We need to investigate why these accesses take longer than expected, but for now we should disable the timeout on DCN35 to avoid this race condition. Since the waits happen only at lower interrupt levels the risk of taking too long at higher IRQ and causing a system watchdog timeout are minimal.

More Info: <https://avd.aquasec.com/nvd/cve-2024-46870>

**[CVE-2024-47141] kernel: pinmux: Use sequential access to access desc->pinmux data (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

pinmux: Use sequential access to access desc->pinmux data

When two client of the same gpio call pinctrl\_select\_state() for the same functionality, we are seeing NULL pointer issue while accessing desc->mux\_owner.

Let's say two processes A, B executing in pin\_request() for the same pin and process A updates the desc->mux\_usecount but not yet updated the desc->mux\_owner while process B see the desc->mux\_usecount which got updated by A path and further executes strcmp and while accessing desc->mux\_owner it crashes with NULL pointer.

Serialize the access to mux related setting with a mutex lock.

cpu0 (process A)    cpu1(process B)

```
pinctrl_select_state() { pinctrl_select_state() {
pin_request() { pin_request() {
...
....
} else {
desc->mux_usecount++;
desc->mux_usecount && strcmp(desc->mux_owner, owner)) {

if (desc->mux_usecount > 1)
return 0;
desc->mux_owner = owner;

} }
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-47141>

**[CVE-2024-47658] kernel: crypto: stm32/cryp - call finalize with bh disabled (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

crypto: stm32/cryp - call finalize with bh disabled

The finalize operation in interrupt mode produce a produces a spinlock recursion warning. The reason is the fact that BH must be disabled during this process.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47658>

**[CVE-2024-47661] kernel: drm/amd/display: Avoid overflow from uint32\_t to uint8\_t (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Avoid overflow from uint32\_t to uint8\_t

[WHAT & HOW]

dmub\_rb\_cmd's ramping\_boundary has size of uint8\_t and it is assigned 0xFFFF. Fix it by changing it to uint8\_t with value of 0xFF.

This fixes 2 INTEGER\_OVERFLOW issues reported by Coverity.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47661>

**[CVE-2024-47662] kernel: drm/amd/display: Remove register from DCN35 DMCUB diagnostic collection (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Remove register from DCN35 DMCUB diagnostic collection

[Why]

These registers should not be read from driver and triggering the security violation when DMCUB work times out and diagnostics are collected blocks Z8 entry.

[How]

Remove the register read from DCN35.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47662>

**[CVE-2024-47664] kernel: spi: hisi-kunpeng: Add verification for the max\_frequency provided by the firmware (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

spi: hisi-kunpeng: Add verification for the max\_frequency provided by the firmware

If the value of max\_speed\_hz is 0, it may cause a division by zero error in hisi\_calc\_effective\_speed().

The value of max\_speed\_hz is provided by firmware.

Firmware is generally considered as a trusted domain. However, as division by zero errors can cause system failure, for defense measure, the value of max\_speed is validated here. So 0 is regarded as invalid and an error code is returned.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47664>

**[CVE-2024-47666] kernel: scsi: pm80xx: Set phy->enable\_completion only when we wait for it (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: pm80xx: Set phy->enable\_completion only when we wait for it

pm8001\_phy\_control() populates the enable\_completion pointer with a stack address, sends a PHY\_LINK\_RESET / PHY\_HARD\_RESET, waits 300 ms, and returns. The problem arises when a phy control response comes late. After 300 ms the pm8001\_phy\_control() function returns and the passed enable\_completion stack address is no longer valid. Late phy control response invokes complete() on a dangling enable\_completion pointer which leads to a kernel crash.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47666>

**[CVE-2024-47703] kernel: bpf, lsm: Add check for BPF LSM return value (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf, lsm: Add check for BPF LSM return value

A bpf prog returning a positive number attached to file\_alloc\_security hook makes kernel panic.

This happens because file system can not filter out the positive number returned by the LSM prog using IS\_ERR, and misinterprets this positive number as a file pointer.

Given that hook file\_alloc\_security never returned positive number before the introduction of BPF LSM, and other BPF LSM hooks may encounter similar issues, this patch adds LSM return value check in verifier, to ensure no unexpected value is returned.



More Info: <https://avd.aquasec.com/nvd/cve-2024-47703>

**[CVE-2024-47704] kernel: drm/amd/display: Check link\_res->hpo\_dp\_link\_enc before using it (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47704>

**[CVE-2024-47736] kernel: erofs: handle overlapped pclusters out of crafted images properly (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

erofs: handle overlapped pclusters out of crafted images properly

syzbot reported a task hang issue due to a deadlock case where it is waiting for the folio lock of a cached folio that will be used for cache I/Os.

After looking into the crafted fuzzed image, I found it's formed with several overlapped big pclusters as below:

Ext:	logical offset	length	physical offset	length
0:	0.. 16384	16384 :	151552.. 167936	16384
1:	16384.. 32768	16384 :	155648.. 172032	16384
2:	32768.. 49152	16384 :	537223168.. 537239552	16384

...

Here, extent 0/1 are physically overlapped although it's entirely \_impossible\_ for normal filesystem images generated by mkfs.

First, managed folios containing compressed data will be marked as up-to-date and then unlocked immediately (unlike in-place folios) when compressed I/Os are complete. If physical blocks are not submitted in the incremental order, there should be separate BIOs to avoid dependency

issues. However, the current code mis-arranges `z_erofs_fill_bio_vec()` and BIO submission which causes unexpected BIO waits.

Second, managed folios will be connected to their own pclusters for efficient inter-queries. However, this is somewhat hard to implement easily if overlapped big pclusters exist. Again, these only appear in fuzzed images so let's simply fall back to temporary short-lived pages for correctness.

Additionally, it justifies that referenced managed folios cannot be truncated for now and reverts part of commit 2080ca1ed3e4 ("erofs: tidy up `struct z_erofs_bvec``") for simplicity although it shouldn't be any difference.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47736>

**[CVE-2024-47752] kernel: media: mediatek: vcodec: Fix H264 stateless decoder smatch warning (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47752>

**[CVE-2024-47753] kernel: media: mediatek: vcodec: Fix VP8 stateless decoder smatch warning (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47753>

**[CVE-2024-47754] kernel: media: mediatek: vcodec: Fix H264 multi stateless decoder smatch warning (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: mediatek: vcodec: Fix H264 multi stateless 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`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47754>

### **[CVE-2024-47794] kernel: bpf: Prevent tailcall infinite loop caused by freplace (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: `6.1.129-1`

Fixed: `%ls(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

bpf: Prevent tailcall infinite loop caused by freplace

There is a potential infinite loop issue that can occur when using a combination of tail calls and freplace.

In an upcoming selftest, the attach target for `entry_freplace` of `tailcall_freplace.c` is `subprog_tc` of `tc_bpf2bpf.c`, while the tail call in `entry_freplace` leads to `entry_tc`. This results in an infinite loop:

```
entry_tc -> subprog_tc -> entry_freplace --tailcall-> entry_tc.
```

The problem arises because the `tail_call_cnt` in `entry_freplace` resets to zero each time `entry_freplace` is executed, causing the tail call mechanism to never terminate, eventually leading to a kernel panic.

To fix this issue, the solution is twofold:

1. Prevent updating a program extended by an freplace program to a `prog_array` map.
2. Prevent extending a program that is already part of a `prog_array` map with an freplace program.

This ensures that:

- \* If a program or its subprogram has been extended by an freplace program, it can no longer be updated to a `prog_array` map.
- \* If a program has been added to a `prog_array` map, neither it nor its subprograms can be extended by an freplace program.

Moreover, an extension program should not be tailcalled. As such, return `-EINVAL` if the program has a type of `BPF_PROG_TYPE_EXT` when adding it to a `prog_array` map.

Additionally, fix a minor code style issue by replacing eight spaces with a tab for proper formatting.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47794>

## **[CVE-2024-47809] kernel: dlm: fix possible lkb\_resource null dereference (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

dlm: fix possible lkb\_resource null dereference

This patch fixes a possible null pointer dereference when this function is called from request\_lock() as lkb->lkb\_resource is not assigned yet, only after validate\_lock\_args() by calling attach\_lkb(). Another issue is that a resource name could be a non printable bytearray and we cannot assume to be ASCII coded.

The log functionality is probably never being hit when DLM is used in normal way and no debug logging is enabled. The null pointer dereference can only occur on a new created lkb that does not have the resource assigned yet, it probably never hits the null pointer dereference but we should be sure that other changes might not change this behaviour and we actually can hit the mentioned null pointer dereference.

In this patch we just drop the printout of the resource name, the lkb id is enough to make a possible connection to a resource name if this exists.

More Info: <https://avd.aquasec.com/nvd/cve-2024-47809>

## **[CVE-2024-48875] kernel: btrfs: don't take dev\_replace rwsem on task already holding it (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: don't take dev\_replace rwsem on task already holding it

Running fstests btrfs/011 with MKFS\_OPTIONS="-O rst" to force the usage of the RAID stripe-tree, we get the following splat from lockdep:

BTRFS info (device sdd): dev\_replace from /dev/sdd (devid 1) to /dev/sdb started

=====

WARNING: possible recursive locking detected

6.11.0-rc3-btrfs-for-next #599 Not tainted

-----

btrfs/2326 is trying to acquire lock:

ffff88810f215c98 (&fs\_info->dev\_replace.rwsem){++++}-{3:3}, at: btrfs\_map\_block+0x39f/0x2250

but task is already holding lock:

ffff88810f215c98 (&fs\_info->dev\_replace.rwsem){++++}-{3:3}, at: btrfs\_map\_block+0x39f/0x2250

other info that might help us debug this:

Possible unsafe locking scenario:

CPU0

----

lock(&fs\_info->dev\_replace.rwsem);

lock(&fs\_info->dev\_replace.rwsem);

\*\*\* DEADLOCK \*\*\*

May be due to missing lock nesting notation

1 lock held by btrfs/2326:

#0: ffff88810f215c98 (&fs\_info->dev\_replace.rwsem){++++}-{3:3}, at: btrfs\_map\_block+0x39f/0x2250

stack backtrace:

CPU: 1 UID: 0 PID: 2326 Comm: btrfs Not tainted 6.11.0-rc3-btrfs-for-next #599

Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011

Call Trace:

<TASK>

dump\_stack\_lvl+0x5b/0x80

\_\_lock\_acquire+0x2798/0x69d0

? \_\_pfx\_\_lock\_acquire+0x10/0x10

? \_\_pfx\_\_lock\_acquire+0x10/0x10

lock\_acquire+0x19d/0x4a0

? btrfs\_map\_block+0x39f/0x2250

? \_\_pfx\_lock\_acquire+0x10/0x10

? find\_held\_lock+0x2d/0x110

? lock\_is\_held\_type+0x8f/0x100

down\_read+0x8e/0x440

? btrfs\_map\_block+0x39f/0x2250

? \_\_pfx\_down\_read+0x10/0x10

? do\_raw\_read\_unlock+0x44/0x70

? \_raw\_read\_unlock+0x23/0x40

btrfs\_map\_block+0x39f/0x2250

? btrfs\_dev\_replace\_by\_ioctl+0xd69/0x1d00

? btrfs\_bio\_counter\_inc\_blocked+0xd9/0x2e0

? \_\_kasan\_slab\_alloc+0x6e/0x70

? \_\_pfx\_btrfs\_map\_block+0x10/0x10

? \_\_pfx\_btrfs\_bio\_counter\_inc\_blocked+0x10/0x10

? kmem\_cache\_alloc\_noprof+0x1f2/0x300

? mempool\_alloc\_noprof+0xed/0x2b0

btrfs\_submit\_chunk+0x28d/0x17e0

? \_\_pfx\_btrfs\_submit\_chunk+0x10/0x10

? bvec\_alloc+0xd7/0x1b0

? bio\_add\_folio+0x171/0x270

? \_\_pfx\_bio\_add\_folio+0x10/0x10

? \_\_kasan\_check\_read+0x20/0x20

btrfs\_submit\_bio+0x37/0x80

read\_extent\_buffer\_pages+0x3df/0x6c0

btrfs\_read\_extent\_buffer+0x13e/0x5f0

read\_tree\_block+0x81/0xe0

read\_block\_for\_search+0x4bd/0x7a0  
? \_\_pfx\_read\_block\_for\_search+0x10/0x10  
btrfs\_search\_slot+0x78d/0x2720  
? \_\_pfx\_btrfs\_search\_slot+0x10/0x10  
? lock\_is\_held\_type+0x8f/0x100  
? kasan\_save\_track+0x14/0x30  
? \_\_kasan\_slab\_alloc+0x6e/0x70  
? kmem\_cache\_alloc\_noprof+0x1f2/0x300  
btrfs\_get RAID\_extent\_offset+0x181/0x820  
? \_\_pfx\_lock\_acquire+0x10/0x10  
? \_\_pfx\_btrfs\_get RAID\_extent\_offset+0x10/0x10  
? down\_read+0x194/0x440  
? \_\_pfx\_down\_read+0x10/0x10  
? do\_raw\_read\_unlock+0x44/0x70  
? \_raw\_read\_unlock+0x23/0x40  
btrfs\_map\_block+0x5b5/0x2250  
? \_\_pfx\_btrfs\_map\_block+0x10/0x10  
scrub\_submit\_initial\_read+0x8fe/0x11b0  
? \_\_pfx\_scrub\_submit\_initial\_read+0x10/0x10  
submit\_initial\_group\_read+0x161/0x3a0  
? lock\_release+0x20e/0x710  
? \_\_pfx\_submit\_initial\_group\_read+0x10/0x10  
? \_\_pfx\_lock\_release+0x10/0x10  
scrub\_simple\_mirror.isra.0+0x3eb/0x580  
scrub\_stripe+0xe4d/0x1440  
? lock\_release+0x20e/0x710  
? \_\_pfx\_scrub\_stripe+0x10/0x10  
? \_\_pfx\_lock\_release+0x10/0x10  
? do\_raw\_read\_unlock+0x44/0x70  
? \_raw\_read\_unlock+0x23/0x40  
scrub\_chunk+0x257/0x4a0  
scrub\_enumerate\_chunks+0x64c/0xf70  
? \_\_mutex\_unlock\_slowpath+0x147/0x5f0  
? \_\_pfx\_scrub\_enumerate\_chunks+0x10/0x10  
? bit\_wait\_timeout+0xb0/0x170  
? \_\_up\_read+0x189/0x700  
? scrub\_workers\_get+0x231/0x300  
? up\_write+0x490/0x4f0  
btrfs\_scrub\_dev+0x52e/0xcd0  
? create\_pending\_snapshots+0x230/0x250  
? \_\_pfx\_btrfs\_scrub\_dev+0x10/0x10  
btrfs\_dev\_replace\_by\_ioctl+0xd69/0x1d00  
? lock\_acquire+0x19d/0x4a0  
? \_\_pfx\_btrfs\_dev\_replace\_by\_ioctl+0x10/0x10  
?  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-48875>

**[CVE-2024-49568] kernel: net/smc: check v2\_ext\_offset/eid\_cnt/ism\_gid\_cnt when receiving proposal msg (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/smc: check v2\_ext\_offset/eid\_cnt/ism\_gid\_cnt when receiving proposal msg

When receiving proposal msg in server, the fields v2\_ext\_offset/eid\_cnt/ism\_gid\_cnt in proposal msg are from the remote client and can not be fully trusted. Especially the field v2\_ext\_offset, once exceed the max value, there has the chance to access wrong address, and crash may happen.

This patch checks the fields v2\_ext\_offset/eid\_cnt/ism\_gid\_cnt before using them.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49568>

### **[CVE-2024-49569] kernel: nvme-rdma: unquiesce admin\_q before destroy it (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nvme-rdma: unquiesce admin\_q before destroy it

Kernel will hang on destroy admin\_q while we create ctrl failed, such as following calltrace:

```
PID: 23644 TASK: ff2d52b40f439fc0 CPU: 2 COMMAND: "nvme"
#0 [ff61d23de260fb78] __schedule at ffffffff8323bc15
#1 [ff61d23de260fc08] schedule at ffffffff8323c014
#2 [ff61d23de260fc28] blk_mq_freeze_queue_wait at ffffffff82a3dba1
#3 [ff61d23de260fc78] blk_freeze_queue at ffffffff82a4113a
#4 [ff61d23de260fc90] blk_cleanup_queue at ffffffff82a33006
#5 [ff61d23de260fcb0] nvme_rdma_destroy_admin_queue at ffffffff812686ce
#6 [ff61d23de260fcc8] nvme_rdma_setup_ctrl at ffffffff81268ced
#7 [ff61d23de260fd28] nvme_rdma_create_ctrl at ffffffff8126919b
#8 [ff61d23de260fd68] nvme_dev_write at ffffffff8024f362
#9 [ff61d23de260fe38] vfs_write at ffffffff827d5f25
RIP: 00007fda7891d574 RSP: 00007ffe2ef06958 RFLAGS: 00000202
RAX: ffffffff812686ce RBX: 000055e8122a4d90 RCX: 00007fda7891d574
RDX: 0000000000000012b RSI: 000055e8122a4d90 RDI: 0000000000000004
RBP: 00007ffe2ef079c0 R8: 0000000000000012b R9: 000055e8122a4d90
R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000004
R13: 000055e8122923c0 R14: 0000000000000012b R15: 00007fda78a54500
ORIG_RAX: 0000000000000001 CS: 0033 SS: 002b
```

This due to we have quiesced admin\_q before cancel requests, but forgot to unquiesce before destroy it, as a result we fail to drain the pending requests, and hang on blk\_mq\_freeze\_queue\_wait() forever. Here try to reuse nvme\_rdma\_teardown\_admin\_queue() to fix this issue and

simplify the code.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49569>

**[CVE-2024-49893] kernel: drm/amd/display: Check stream\_status before it is used (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49893>

**[CVE-2024-49901] kernel: drm/msm/adreno: Assign msm\_gpu->pdev earlier to avoid nullptrs (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-49901>

**[CVE-2024-49906] kernel: drm/amd/display: Check null pointer before try to access it (Severity:**



## MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49906>

### **[CVE-2024-49908] kernel: drm/amd/display: Add null check for 'afb' in amdgpu\_dm\_update\_cursor (v2) (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Add null check for 'afb' in amdgpu\_dm\_update\_cursor (v2)

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)

More Info: <https://avd.aquasec.com/nvd/cve-2024-49908>

### **[CVE-2024-49910] kernel: drm/amd/display: Add NULL check for function pointer in dcn401\_set\_output\_transfer\_func (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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 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`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49910>

### **[CVE-2024-49914] kernel: drm/amd/display: Add null check for `pipe_ctx->plane_state` in `dcn20_program_pipe` (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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)

More Info: <https://avd.aquasec.com/nvd/cve-2024-49914>

### **[CVE-2024-49916] kernel: drm/amd/display: Add NULL check for `clk_mgr` and `clk_mgr->funcs` in `dcn401_init_hw` (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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)

More Info: <https://avd.aquasec.com/nvd/cve-2024-49916>

**[CVE-2024-49918] kernel: drm/amd/display: Add null check for head\_pipe in dcn32\_acquire\_idle\_pipe\_for\_head\_pipe\_in\_layer (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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)

More Info: <https://avd.aquasec.com/nvd/cve-2024-49918>

**[CVE-2024-49919] kernel: drm/amd/display: Add null check for head\_pipe in dcn201\_acquire\_free\_pipe\_for\_layer (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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 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)

More Info: <https://avd.aquasec.com/nvd/cve-2024-49919>

**[CVE-2024-49920] kernel: drm/amd/display: Check null pointers before multiple uses (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Check null pointers before multiple uses

**[WHAT & HOW]**

Pointers, 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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49920>

**[CVE-2024-49921] kernel: drm/amd/display: Check null pointers before used (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Check null pointers before used

**[WHAT & HOW]**

Pointers, 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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49921>

**[CVE-2024-49922] kernel: drm/amd/display: Check null pointers before using them (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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 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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49922>

**[CVE-2024-49923] kernel: drm/amd/display: Pass non-null to dcn20\_validate\_apply\_pipe\_split\_flags (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49923>

**[CVE-2024-49926] kernel: rcu-tasks: Fix access non-existent percpu rtpcp variable in rcu\_tasks\_need\_gpccb() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

rcu-tasks: Fix access non-existent percpu rtpcp variable in rcu\_tasks\_need\_gpccb()

For kernels built with CONFIG\_FORCE\_NR\_CPUS=y, the nr\_cpu\_ids is defined as NR\_CPUS instead of the number of possible cpus, this will cause the following system panic:

smpboot: Allowing 4 CPUs, 0 hotplug CPUs

...

setup\_percpu: NR\_CPUS:512 nr\_cpumask\_bits:512 nr\_cpu\_ids:512 nr\_node\_ids:1

...

BUG: unable to handle page fault for address: ffffffff9911c8c8

Oops: 0000 [#1] PREEMPT SMP PTI

CPU: 0 PID: 15 Comm: rcu\_tasks\_trace Tainted: G W

6.6.21 #1 5dc7acf91a5e8e9ac9dcfc35bee0245691283ea6

RIP: 0010:rcu\_tasks\_need\_gpccb+0x25d/0x2c0

RSP: 0018:ffffa371c00a3e60 EFLAGS: 00010082

CR2: ffffffff9911c8c8 CR3: 000000040fa20005 CR4: 00000000001706f0

Call Trace:

<TASK>

? \_\_die+0x23/0x80

```
? page_fault_oops+0xa4/0x180
? exc_page_fault+0x152/0x180
? asm_exc_page_fault+0x26/0x40
? rcu_tasks_need_gpcb+0x25d/0x2c0
? __pfx_rcu_tasks_kthread+0x40/0x40
rcu_tasks_one_gp+0x69/0x180
rcu_tasks_kthread+0x94/0xc0
kthread+0xe8/0x140
? __pfx_kthread+0x40/0x40
ret_from_fork+0x34/0x80
? __pfx_kthread+0x40/0x40
ret_from_fork_asm+0x1b/0x80
</TASK>
```

Considering that there may be holes in the CPU numbers, use the maximum possible cpu number, instead of nr\_cpu\_ids, for configuring enqueue and dequeue limits.

[ neeraj.upadhyay: Fix htmldocs build error reported by Stephen Rothwell ]

More Info: <https://avd.aquasec.com/nvd/cve-2024-49926>

### **[CVE-2024-49932] kernel: btrfs: don't readahead the relocation inode on RST (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: don't readahead the relocation inode on RST

On relocation we're doing readahead on the relocation inode, but if the filesystem is backed by a RAID stripe tree we can get ENOENT (e.g. due to preallocated extents not being mapped in the RST) from the lookup.

But readahead doesn't handle the error and submits invalid reads to the device, causing an assertion in the scatter-gather list code:

```
BTRFS info (device nvme1n1): balance: start -d -m -s
BTRFS info (device nvme1n1): relocating block group 6480920576 flags data|raid0
BTRFS error (device nvme1n1): cannot find raid-stripe for logical [6481928192, 6481969152] devid 2, profile raid0
-----[ cut here ]-----
kernel BUG at include/linux/scatterlist.h:115!
Oops: invalid opcode: 0000 [#1] PREEMPT SMP PTI
CPU: 0 PID: 1012 Comm: btrfs Not tainted 6.10.0-rc7+ #567
RIP: 0010: __blk_rq_map_sg+0x339/0x4a0
RSP: 0018:ffffc90001a43820 EFLAGS: 00010202
RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffea00045d4802
RDX: 0000000011752000 RSI: 0000000000000000 RDI: ffff8881027d1000
RBP: 0000000000003000 R08: ffffea00045d4902 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000001000 R12: ffff8881003d10b8
R13: ffffc90001a438f0 R14: 0000000000000000 R15: 0000000000003000
FS: 00007fcc048a6900(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000
```

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 000000002cd11000 CR3: 00000001109ea001 CR4: 0000000000370eb0

Call Trace:

<TASK>

? \_\_die\_body.cold+0x14/0x25  
? die+0x2e/0x50  
? do\_trap+0xca/0x110  
? do\_error\_trap+0x65/0x80  
? \_\_blk\_rq\_map\_sg+0x339/0x4a0  
? exc\_invalid\_op+0x50/0x70  
? \_\_blk\_rq\_map\_sg+0x339/0x4a0  
? asm\_exc\_invalid\_op+0x1a/0x20  
? \_\_blk\_rq\_map\_sg+0x339/0x4a0  
nvme\_prep\_rq.part.0+0x9d/0x770  
nvme\_queue\_rq+0x7d/0x1e0  
\_\_blk\_mq\_issue\_directly+0x2a/0x90  
? blk\_mq\_get\_budget\_and\_tag+0x61/0x90  
blk\_mq\_try\_issue\_list\_directly+0x56/0xf0  
blk\_mq\_flush\_plug\_list.part.0+0x52b/0x5d0  
\_\_blk\_flush\_plug+0xc6/0x110  
blk\_finish\_plug+0x28/0x40  
read\_pages+0x160/0x1c0  
page\_cache\_ra\_unbounded+0x109/0x180  
relocate\_file\_extent\_cluster+0x611/0x6a0  
? btrfs\_search\_slot+0xba4/0xd20  
? balance\_dirty\_pages\_ratelimited\_flags+0x26/0xb00  
relocate\_data\_extent.constprop.0+0x134/0x160  
relocate\_block\_group+0x3f2/0x500  
btrfs\_relocate\_block\_group+0x250/0x430  
btrfs\_relocate\_chunk+0x3f/0x130  
btrfs\_balance+0x71b/0xef0  
? kmalloc\_trace\_noprof+0x13b/0x280  
btrfs\_ioctl+0x2c2e/0x3030  
? kvfree\_call\_rcu+0x1e6/0x340  
? list\_lru\_add\_obj+0x66/0x80  
? mntput\_no\_expire+0x3a/0x220  
\_\_x64\_sys\_ioctl+0x96/0xc0  
do\_syscall\_64+0x54/0x110  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

RIP: 0033:0x7fcc04514f9b

Code: Unable to access opcode bytes at 0x7fcc04514f71.

RSP: 002b:00007ffe923370 EFLAGS: 00000246 ORIG\_RAX: 0000000000000010

RAX: ffffffffda RBX: 0000000000000003 RCX: 00007fcc04514f9b

RDX: 00007ffe923460 RSI: 00000000c4009420 RDI: 0000000000000003

RBP: 0000000000000000 R08: 0000000000000013 R09: 0000000000000001

R10: 00007fcc043fba8 R11: 0000000000000246 R12: 00007ffe924fc5

R13: 00007ffe923460 R14: 0000000000000002 R15: 00000000004d4bb0

</TASK>

Modules linked in:

---[ end trace 0000000000000000 ]---

RIP: 0010:\_\_blk\_rq\_map\_sg+0x339/0x4a0

RSP: 0018:ffff90001a43820 EFLAGS: 00010202

RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffea00045d4802

RDX: 0000000117520000 RSI: 0000000000000000 RDI: ffff8881027d1000  
RBP: 0000000000003000 R08: ffffea00045d4902 R09: 0000000000000000  
R10: 0000000000000000 R11: 0000000000001000 R12: ffff8881003d10b8  
R13: ffffc90001a438f0 R14: 0000000000000000 R15: 0000000000003000  
FS: 00007fcc048a6900(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 00007fcc04514f71 CR3: 00000001109ea001 CR4: 0000000000370eb0  
Kernel p  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-49932>

### **[CVE-2024-49940] kernel: l2tp: prevent possible tunnel refcount underflow (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

l2tp: prevent possible tunnel refcount underflow

When a session is created, it sets a backpointer to its tunnel. When the session refcount drops to 0, l2tp\_session\_free drops the tunnel refcount if session->tunnel is non-NULL. However, session->tunnel is set in l2tp\_session\_create, before the tunnel refcount is incremented by l2tp\_session\_register, which leaves a small window where session->tunnel is non-NULL when the tunnel refcount hasn't been bumped.

Moving the assignment to l2tp\_session\_register is trivial but l2tp\_session\_create calls l2tp\_session\_set\_header\_len which uses session->tunnel to get the tunnel's encap. Add an encap arg to l2tp\_session\_set\_header\_len to avoid using session->tunnel.

If l2tpv3 sessions have colliding IDs, it is possible for l2tp\_v3\_session\_get to race with l2tp\_session\_register and fetch a session which doesn't yet have session->tunnel set. Add a check for this case.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49940>

### **[CVE-2024-49945] kernel: net/ncsi: Disable the ncsi work before freeing the associated structure (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49945>

**[CVE-2024-49968] kernel: ext4: filesystems without casefold feature cannot be mounted with siphash (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ext4: filesystems without casefold feature cannot be mounted with siphash

When mounting the ext4 filesystem, if the default hash version is set to DX\_HASH\_SIPHASH but the casefold feature is not set, exit the mounting.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49968>

**[CVE-2024-49970] kernel: drm/amd/display: Implement bounds check for stream encoder creation in DCN401 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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 dcn401\_stream\_encoder\_create()  
error: buffer overflow 'stream\_enc\_regs' 4 <= 5

More Info: <https://avd.aquasec.com/nvd/cve-2024-49970>

**[CVE-2024-49972] kernel: drm/amd/display: Deallocate DML memory if allocation fails (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49972>

### **[CVE-2024-49987] kernel: bpftool: Fix undefined behavior in qsort(NULL, 0, ...) (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49987>

### **[CVE-2024-49988] kernel: ksmbd: add refcnt to ksmbd\_conn struct (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-49988>

### **[CVE-2024-49998] kernel: net: dsa: improve shutdown sequence (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: dsa: improve shutdown sequence

Alexander Sverdlin presents 2 problems during shutdown with the lan9303 driver. One is specific to lan9303 and the other just happens to reproduce there.

The first problem is that lan9303 is unique among DSA drivers in that it calls dev\_get\_drvdata() at "arbitrary runtime" (not probe, not shutdown, not remove):

```
phy_state_machine()
-> ...
-> dsa_user_phy_read()
  -> ds->ops->phy_read()
    -> lan9303_phy_read()
      -> chip->ops->phy_read()
        -> lan9303_mdio_phy_read()
          -> dev_get_drvdata()
```

But we never stop the phy\_state\_machine(), so it may continue to run after dsa\_switch\_shutdown(). Our common pattern in all DSA drivers is to set drvdata to NULL to suppress the remove() method that may come afterwards. But in this case it will result in an NPD.

The second problem is that the way in which we set dp->conduit->dsa\_ptr = NULL; is concurrent with receive packet processing. dsa\_switch\_rcv() checks once whether dev->dsa\_ptr is NULL, but afterwards, rather than continuing to use that non-NULL value, dev->dsa\_ptr is dereferenced again and again without NULL checks: dsa\_conduit\_find\_user() and many other places. In between dereferences, there is no locking to ensure that what was valid once continues to be

valid.

Both problems have the common aspect that closing the conduit interface solves them.

In the first case, `dev_close(conduit)` triggers the `NETDEV_GOING_DOWN` event in `dsa_user_netdevice_event()` which closes user ports as well. `dsa_port_disable_rt()` calls `phylink_stop()`, which synchronously stops the phylink state machine, and `ds->ops->phy_read()` will thus no longer call into the driver after this point.

In the second case, `dev_close(conduit)` should do this, as per Documentation/networking/driver.rst:

```
| Quiescence
| -----
|
| After the ndo_stop routine has been called, the hardware must
| not receive or transmit any data. All in flight packets must
| be aborted. If necessary, poll or wait for completion of
| any reset commands.
```

So it should be sufficient to ensure that later, when we zeroize `conduit->dsa_ptr`, there will be no concurrent `dsa_switch_rcv()` call on this conduit.

The addition of the `netif_device_detach()` function is to ensure that `ioctl`s, `rtnetlinks` and `ethtool` requests on the user ports no longer propagate down to the driver - we're no longer prepared to handle them.

The race condition actually did not exist when commit 0650bf52b31f ("net: dsa: be compatible with masters which unregister on shutdown") first introduced `dsa_switch_shutdown()`. It was created later, when we stopped unregistering the user interfaces from a bad spot, and we just replaced that sequence with a racy zeroization of `conduit->dsa_ptr` (one which doesn't ensure that the interfaces aren't up).

More Info: <https://avd.aquasec.com/nvd/cve-2024-49998>

**[CVE-2024-50009] kernel: cpufreq: amd-pstate: add check for cpufreq\_cpu\_get's return value (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50009>

**[CVE-2024-50016] kernel: drm/amd/display: Avoid overflow assignment in link\_dp\_cts (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50016>

**[CVE-2024-50017] kernel: x86/mm/ident\_map: Use gbpages only where full GB page should be mapped. (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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 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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50017>

**[CVE-2024-50028] kernel: thermal: core: Reference count the zone in thermal\_zone\_get\_by\_id() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50028>

**[CVE-2024-50032] kernel: rcu/nocb: Fix rcuog wake-up from offline softirq (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

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.

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().

More Info: <https://avd.aquasec.com/nvd/cve-2024-50032>

### **[CVE-2024-50056] kernel: usb: gadget: uvc: Fix ERR\_PTR dereference in uvc\_v4l2.c (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usb: gadget: uvc: Fix ERR\_PTR dereference in uvc\_v4l2.c

Fix potential dereferencing of ERR\_PTR() in find\_format\_by\_pix() and uvc\_v4l2\_enum\_format().

Fix the following smatch errors:

drivers/usb/gadget/function/uvc\_v4l2.c:124 find\_format\_by\_pix()  
error: 'fmtdesc' dereferencing possible ERR\_PTR()

drivers/usb/gadget/function/uvc\_v4l2.c:392 uvc\_v4l2\_enum\_format()  
error: 'fmtdesc' dereferencing possible ERR\_PTR()

Also, fix similar issue in uvc\_v4l2\_try\_format() for potential dereferencing of ERR\_PTR().

More Info: <https://avd.aquasec.com/nvd/cve-2024-50056>

### **[CVE-2024-50111] kernel: LoongArch: Enable IRQ if do\_ale() triggered in irq-enabled context (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

LoongArch: Enable IRQ if do\_ale() triggered in irq-enabled context

Unaligned access exception can be triggered in irq-enabled context such

as user mode, in this case do\_ale() may call get\_user() which may cause sleep. Then we will get:

BUG: sleeping function called from invalid context at arch/loongarch/kernel/access-helper.h:7  
in\_atomic(): 0, irqs\_disabled(): 1, non\_block: 0, pid: 129, name: modprobe  
preempt\_count: 0, expected: 0  
RCU nest depth: 0, expected: 0  
CPU: 0 UID: 0 PID: 129 Comm: modprobe Tainted: G W 6.12.0-rc1+ #1723  
Tainted: [W]=WARN

Stack : 9000000105e0bd48 0000000000000000 9000000003803944 9000000105e08000  
9000000105e0bc70 9000000105e0bc78 0000000000000000 0000000000000000  
9000000105e0bc78 0000000000000001 9000000185e0ba07 9000000105e0b890  
ffffffffffffff 9000000105e0bc78 73924b81763be05b 9000000100194500  
000000000000020c 000000000000000a 0000000000000000 0000000000000003  
000000000000023f0 00000000000e1401 00000000072f8000 0000007ffbb0e260  
0000000000000000 0000000000000000 9000000005437650 90000000055d5000  
0000000000000000 0000000000000003 0000007ffbb0e1f0 0000000000000000  
0000005567b00490 0000000000000000 9000000003803964 0000007ffbb0dfec  
00000000000000b0 0000000000000007 0000000000000003 0000000000071c1d

...

Call Trace:

[<9000000003803964>] show\_stack+0x64/0x1a0  
[<9000000004c57464>] dump\_stack\_lvl+0x74/0xb0  
[<9000000003861ab4>] \_\_might\_resched+0x154/0x1a0  
[<900000000380c96c>] emulate\_load\_store\_insn+0x6c/0xf60  
[<9000000004c58118>] do\_ale+0x78/0x180  
[<9000000003801bc8>] handle\_ale+0x128/0x1e0

So enable IRQ if unaligned access exception is triggered in irq-enabled context to fix it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50111>

## **[CVE-2024-50135] kernel: nvme-pci: fix race condition between reset and nvme\_dev\_disable() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

nvme-pci: fix race condition between reset and nvme\_dev\_disable()

nvme\_dev\_disable() modifies the dev->online\_queues field, therefore nvme\_pci\_update\_nr\_queues() should avoid racing against it, otherwise we could end up passing invalid values to blk\_mq\_update\_nr\_hw\_queues().

WARNING: CPU: 39 PID: 61303 at drivers/pci/msi/api.c:347  
pci\_irq\_get\_affinity+0x187/0x210  
Workqueue: nvme-reset-wq nvme\_reset\_work [nvme]  
RIP: 0010:pci\_irq\_get\_affinity+0x187/0x210  
Call Trace:



<TASK>

? blk\_mq\_pci\_map\_queues+0x87/0x3c0

? pci\_irq\_get\_affinity+0x187/0x210

blk\_mq\_pci\_map\_queues+0x87/0x3c0

nvme\_pci\_map\_queues+0x189/0x460 [nvme]

blk\_mq\_update\_nr\_hw\_queues+0x2a/0x40

nvme\_reset\_work+0x1be/0x2a0 [nvme]

Fix the bug by locking the shutdown\_lock mutex before using dev->online\_queues. Give up if nvme\_dev\_disable() is running or if it has been executed already.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50135>

### **[CVE-2024-50166] kernel: fsl/fman: Fix refcount handling of fman-related devices (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fsl/fman: Fix refcount handling of fman-related devices

In mac\_probe() there are multiple calls to of\_find\_device\_by\_node(), fman\_bind() and fman\_port\_bind() which takes references to of\_dev->dev. Not all references taken by these calls are released later on error path in mac\_probe() and in mac\_remove() which lead to reference leaks.

Add references release.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50166>

### **[CVE-2024-50277] kernel: dm: fix a crash if blk\_alloc\_disk fails (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

dm: fix a crash if blk\_alloc\_disk fails

If blk\_alloc\_disk fails, the variable md->disk is set to an error value. cleanup\_mapped\_device will see that md->disk is non-NULL and it will attempt to access it, causing a crash on this statement "md->disk->private\_data = NULL;".

More Info: <https://avd.aquasec.com/nvd/cve-2024-50277>

### **[CVE-2024-50285] kernel: ksmbd: check outstanding simultaneous SMB operations (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: check outstanding simultaneous SMB operations

If Client send simultaneous SMB operations to ksmbd, It exhausts too much memory through the "ksmbd\_work\_cache". It will cause OOM issue.

ksmbd has a credit mechanism but it can't handle this problem. This patch add the check if it exceeds max credits to prevent this problem by assuming that one smb request consumes at least one credit.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50285>

### **[CVE-2024-50289] kernel: media: av7110: fix a spectre vulnerability (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

media: av7110: fix a spectre vulnerability

As warned by smatch:

drivers/staging/media/av7110/av7110\_ca.c:270 dvb\_ca\_ioctl() warn: potential spectre issue 'av7110->ci\_slot' [w] (local cap)

There is a spectre-related vulnerability at the code. Fix it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50289>

### **[CVE-2024-50298] kernel: net: enetc: allocate vf\_state during PF probes (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: enetc: allocate vf\_state during PF probes

In the previous implementation, vf\_state is allocated memory only when VF is enabled. However, net\_device\_ops::ndo\_set\_vf\_mac() may be called before VF is enabled to configure the MAC address of VF. If this is the case, enetc\_pf\_set\_vf\_mac() will access vf\_state, resulting in access to a null pointer. The simplified error log is as follows.

```
root@ls1028ardb:~# ip link set eno0 vf 1 mac 00:0c:e7:66:77:89
```

```
[ 173.543315] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000004
```

```
[ 173.637254] pc : enetc_pf_set_vf_mac+0x3c/0x80 Message from sy
```

```
[ 173.641973] lr : do_setlink+0x4a8/0xec8
```

```
[ 173.732292] Call trace:
```

```
[ 173.734740] enetc_pf_set_vf_mac+0x3c/0x80
```

[ 173.738847] \_\_rtnl\_newlink+0x530/0x89c  
[ 173.742692] rtnl\_newlink+0x50/0x7c  
[ 173.746189] rtnetlink\_rcv\_msg+0x128/0x390  
[ 173.750298] netlink\_rcv\_skb+0x60/0x130  
[ 173.754145] rtnetlink\_rcv+0x18/0x24  
[ 173.757731] netlink\_unicast+0x318/0x380  
[ 173.761665] netlink\_sendmsg+0x17c/0x3c8

More Info: <https://avd.aquasec.com/nvd/cve-2024-50298>

**[CVE-2024-52559] kernel: drm/msm/gem: prevent integer overflow in msm\_ioctl\_gem\_submit() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/msm/gem: prevent integer overflow in msm\_ioctl\_gem\_submit()

The "submit->cmd[i].size" and "submit->cmd[i].offset" variables are u32 values that come from the user via the submit\_lookup\_cmds() function. This addition could lead to an integer wrapping bug so use size\_add() to prevent that.

Patchwork: <https://patchwork.freedesktop.org/patch/624696/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-52559>

**[CVE-2024-52560] kernel: fs/ntfs3: Mark inode as bad as soon as error detected in mi\_enum\_attr() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fs/ntfs3: Mark inode as bad as soon as error detected in mi\_enum\_attr()

Extended the `mi\_enum\_attr()` function interface with an additional parameter, `struct ntfs\_inode \*ni`, to allow marking the inode as bad as soon as an error is detected.

More Info: <https://avd.aquasec.com/nvd/cve-2024-52560>

**[CVE-2024-53050] kernel: drm/i915/hdcp: Add encoder check in hdcp2\_get\_capability (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/i915/hdcp: Add encoder check in hdcp2\_get\_capability

Add encoder check in intel\_hdcp2\_get\_capability to avoid null pointer error.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53050>

**[CVE-2024-53051] kernel: drm/i915/hdcp: Add encoder check in intel\_hdcp\_get\_capability (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/i915/hdcp: Add encoder check in intel\_hdcp\_get\_capability

Sometimes during hotplug scenario or suspend/resume scenario encoder is not always initialized when intel\_hdcp\_get\_capability add a check to avoid kernel null pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53051>

**[CVE-2024-53056] kernel: drm/mediatek: Fix potential NULL dereference in mtk\_crtc\_destroy() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/mediatek: Fix potential NULL dereference in mtk\_crtc\_destroy()

In mtk\_crtc\_create(), if the call to mbox\_request\_channel() fails then we set the "mtk\_crtc->cmdq\_client.chan" pointer to NULL. In that situation, we do not call cmdq\_pkt\_create().

During the cleanup, we need to check if the "mtk\_crtc->cmdq\_client.chan" is NULL first before calling cmdq\_pkt\_destroy(). Calling cmdq\_pkt\_destroy() is unnecessary if we didn't call cmdq\_pkt\_create() and it will result in a NULL pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53056>

**[CVE-2024-53079] kernel: mm/thp: fix deferred split unqueue naming and locking (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/thp: fix deferred split unqueue naming and locking

Recent changes are putting more pressure on THP deferred split queues: under load revealing long-standing races, causing list\_del corruptions, "Bad page state"s and worse (I keep BUGs in both of those, so usually don't get to see how badly they end up without). The relevant recent changes being 6.8's mTHP, 6.10's mTHP swapout, and 6.12's mTHP swapin, improved swap allocation, and underused THP splitting.

Before fixing locking: rename misleading folio\_undo\_large\_rmappable(), which does not undo large\_rmappable, to folio\_unqueue\_deferred\_split(), which is what it does. But that and its out-of-line \_\_callee are mm internals of very limited usability: add comment and WARN\_ON\_ONCEs to check usage; and return a bool to say if a deferred split was unqueued, which can then be used in WARN\_ON\_ONCEs around safety checks (sparing callers the arcane conditionals in \_\_folio\_unqueue\_deferred\_split()).

Just omit the folio\_unqueue\_deferred\_split() from free\_unref\_folios(), all of whose callers now call it beforehand (and if any forget then bad\_page() will tell) - except for its caller put\_pages\_list(), which itself no longer has any callers (and will be deleted separately).

Swapout: mem\_cgroup\_swapout() has been resetting folio->memcg\_data 0 without checking and unqueueing a THP folio from deferred split list; which is unfortunate, since the split\_queue\_lock depends on the memcg (when memcg is enabled); so swapout has been unqueueing such THPs later, when freeing the folio, using the pgdat's lock instead: potentially corrupting the memcg's list. \_\_remove\_mapping() has frozen refcount to 0 here, so no problem with calling folio\_unqueue\_deferred\_split() before resetting memcg\_data.

That goes back to 5.4 commit 87eaceb3faa5 ("mm: thp: make deferred split shrinker memcg aware"): which included a check on swapcache before adding to deferred queue, but no check on deferred queue before adding THP to swapcache. That worked fine with the usual sequence of events in reclaim (though there were a couple of rare ways in which a THP on deferred queue could have been swapped out), but 6.12 commit daff3f4c850 ("mm: split underused THPs") avoids splitting underused THPs in reclaim, which makes swapcache THPs on deferred queue commonplace.

Keep the check on swapcache before adding to deferred queue? Yes: it is no longer essential, but preserves the existing behaviour, and is likely to be a worthwhile optimization (vmstat showed much more traffic on the queue under swapping load if the check was removed); update its comment.

Memcg-v1 move (deprecated): mem\_cgroup\_move\_account() has been changing folio->memcg\_data without checking and unqueueing a THP folio from the deferred list, sometimes corrupting "from" memcg's list, like swapout. Refcount is non-zero here, so folio\_unqueue\_deferred\_split() can only be used in a WARN\_ON\_ONCE to validate the fix, which must be done earlier: mem\_cgroup\_move\_charge\_pte\_range() first try to split the THP (splitting of course unqueues), or skip it if that fails. Not ideal, but moving charge has been requested, and khugepaged should repair the THP later:

nobody wants new custom unqueueing code just for this deprecated case.

The 87eaceb3faa5 commit did have the code to move from one deferred list to another (but was not conscious of its unsafety while refcount non-0); but that was removed by 5.6 commit fac0516b5534 ("mm: thp: don't need care deferred split queue in memcg charge move path"), which argued that the existence of a PMD mapping guarantees that the THP cannot be on a deferred list. As above, false in rare cases, and now commonly false.

Backport to 6.11 should be straightforward. Earlier backports must take care that other `_deferred_list` fixes and dependencies are included. There is not a strong case for backports, but they can fix cornercases.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53079>

### **[CVE-2024-53085] kernel: tpm: Lock TPM chip in tpm\_pm\_suspend() first (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

tpm: Lock TPM chip in tpm\_pm\_suspend() first

Setting `TPM_CHIP_FLAG_SUSPENDED` in the end of `tpm_pm_suspend()` can be racy according, as this leaves window for `tpm_hwrng_read()` to be called while the operation is in progress. The recent bug report gives also evidence of this behaviour.

Address this by locking the TPM chip before checking any `chip->flags` both in `tpm_pm_suspend()` and `tpm_hwrng_read()`. Move `TPM_CHIP_FLAG_SUSPENDED` check inside `tpm_get_random()` so that it will be always checked only when the lock is reserved.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53085>

### **[CVE-2024-53089] kernel: LoongArch: KVM: Mark hrtimer to expire in hard interrupt context (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

LoongArch: KVM: Mark hrtimer to expire in hard interrupt context

Like commit 2c0d278f3293f ("KVM: LAPIC: Mark hrtimer to expire in hard interrupt context") and commit 9090825fa9974 ("KVM: arm/arm64: Let the timer expire in hardirq context on RT"), On `PREEMPT_RT` enabled kernels unmarked hrtimers are moved into soft interrupt expiry mode by default. Then the timers are canceled from an preempt-notifier which is invoked with disabled preemption which is not allowed on `PREEMPT_RT`.

The timer callback is short so it could be invoked in hard-IRQ context.  
So let the timer expire on hard-IRQ context even on -RT.

This fixes a "scheduling while atomic" bug for PREEMPT\_RT enabled kernels:

```
BUG: scheduling while atomic: qemu-system-loo/1011/0x00000002
Modules linked in: amdgpu rkill nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6
nft_reject nft_ct nft_chain_nat ns
CPU: 1 UID: 0 PID: 1011 Comm: qemu-system-loo Tainted: G      W      6.12.0-rc2+ #1774
Tainted: [W]=WARN
Hardware name: Loongson Loongson-3A5000-7A1000-1w-CRB/Loongson-LS3A5000-7A1000-1w-CRB, BIOS
vUDK2018-LoongArch-V2.0.0-prebeta9 10/21/2022
Stack : ffffffff 0000000000000000 9000000004e3ea38 9000000116744000
        90000001167475a0 0000000000000000 90000001167475a8 9000000005644830
        90000000058dc000 90000000058dbff8 9000000116747420 0000000000000001
        0000000000000001 6a613fc938313980 000000000790c000 90000001001c1140
        00000000000003fe 0000000000000001 000000000000000d 0000000000000003
        0000000000000030 00000000000003f3 000000000790c000 9000000116747830
        90000000057ef000 0000000000000000 9000000005644830 0000000000000004
        0000000000000000 90000000057f4b58 0000000000000001 9000000116747868
        900000000451b600 9000000005644830 9000000003a13998 0000000010000020
        00000000000000b0 0000000000000004 0000000000000000 0000000000071c1d
        ...
```

Call Trace:

```
[<9000000003a13998>] show_stack+0x38/0x180
[<9000000004e3ea34>] dump_stack_lvl+0x84/0xc0
[<9000000003a71708>] __schedule_bug+0x48/0x60
[<9000000004e45734>] __schedule+0x1114/0x1660
[<9000000004e46040>] schedule_rtlock+0x20/0x60
[<9000000004e4e330>] rtlock_slowlock_locked+0x3f0/0x10a0
[<9000000004e4f038>] rt_spin_lock+0x58/0x80
[<9000000003b02d68>] hrtimer_cancel_wait_running+0x68/0xc0
[<9000000003b02e30>] hrtimer_cancel+0x70/0x80
[<ffff80000235eb70>] kvm_restore_timer+0x50/0x1a0 [kvm]
[<ffff8000023616c8>] kvm_arch_vcpu_load+0x68/0x2a0 [kvm]
[<ffff80000234c2d4>] kvm_sched_in+0x34/0x60 [kvm]
[<9000000003a749a0>] finish_task_switch.isra.0+0x140/0x2e0
[<9000000004e44a70>] __schedule+0x450/0x1660
[<9000000004e45cb0>] schedule+0x30/0x180
[<ffff800002354c70>] kvm_vcpu_block+0x70/0x120 [kvm]
[<ffff800002354d80>] kvm_vcpu_halt+0x60/0x3e0 [kvm]
[<ffff80000235b194>] kvm_handle_gspr+0x3f4/0x4e0 [kvm]
[<ffff80000235f548>] kvm_handle_exit+0x1c8/0x260 [kvm]
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-53089>

### [CVE-2024-53090] kernel: afs: Fix lock recursion (Severity: MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

afs: Fix lock recursion

afs\_wake\_up\_async\_call() can incur lock recursion. The problem is that it is called from AF\_RXRPC whilst holding the ->notify\_lock, but it tries to take a ref on the afs\_call struct in order to pass it to a work queue - but if the afs\_call is already queued, we then have an extraneous ref that must be put... calling afs\_put\_call() may call back down into AF\_RXRPC through rxrpc\_kernel\_shutdown\_call(), however, which might try taking the ->notify\_lock again.

This case isn't very common, however, so defer it to a workqueue. The oops looks something like:

BUG: spinlock recursion on CPU#0, krxrpcio/7001/1646

lock: 0xffff888141399b30, .magic: dead4ead, .owner: krxrpcio/7001/1646, .owner\_cpu: 0

CPU: 0 UID: 0 PID: 1646 Comm: krxrpcio/7001 Not tainted 6.12.0-rc2-build3+ #4351

Hardware name: ASUS All Series/H97-PLUS, BIOS 2306 10/09/2014

Call Trace:

<TASK>

dump\_stack\_lvl+0x47/0x70

do\_raw\_spin\_lock+0x3c/0x90

rxrpc\_kernel\_shutdown\_call+0x83/0xb0

afs\_put\_call+0xd7/0x180

rxrpc\_notify\_socket+0xa0/0x190

rxrpc\_input\_split\_jumbo+0x198/0x1d0

rxrpc\_input\_data+0x14b/0x1e0

? rxrpc\_input\_call\_packet+0xc2/0x1f0

rxrpc\_input\_call\_event+0xad/0x6b0

rxrpc\_input\_packet\_on\_conn+0x1e1/0x210

rxrpc\_input\_packet+0x3f2/0x4d0

rxrpc\_io\_thread+0x243/0x410

? \_\_pfx\_rxrpc\_io\_thread+0x10/0x10

kthread+0xcf/0xe0

? \_\_pfx\_kthread+0x10/0x10

ret\_from\_fork+0x24/0x40

? \_\_pfx\_kthread+0x10/0x10

ret\_from\_fork\_asm+0x1a/0x30

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2024-53090>

**[CVE-2024-53091] kernel: bpf: Add sk\_is\_inet and IS\_ICSK check in tls\_sw\_has\_ctx\_tx/rx (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Add sk\_is\_inet and IS\_ICSK check in tls\_sw\_has\_ctx\_tx/rx



As the introduction of the support for vsock and unix sockets in sockmap, `tls_sw_has_ctx_tx/rx` cannot presume the socket passed in must be `IS_ICSK`. `vsock` and `af_unix` sockets have `vsock_sock` and `unix_sock` instead of `inet_connection_sock`. For these sockets, `tls_get_ctx` may return an invalid pointer and cause page fault in function `tls_sw_ctx_rx`.

BUG: unable to handle page fault for address: 0000000000040030

Workqueue: vsock-loopback vsock\_loopback\_work

RIP: 0010:sk\_psock\_strp\_data\_ready+0x23/0x60

Call Trace:

? \_\_die+0x81/0xc3  
? no\_context+0x194/0x350  
? do\_page\_fault+0x30/0x110  
? async\_page\_fault+0x3e/0x50  
? sk\_psock\_strp\_data\_ready+0x23/0x60  
virtio\_transport\_recv\_pkt+0x750/0x800  
? update\_load\_avg+0x7e/0x620  
vsock\_loopback\_work+0xd0/0x100  
process\_one\_work+0x1a7/0x360  
worker\_thread+0x30/0x390  
? create\_worker+0x1a0/0x1a0  
kthread+0x112/0x130  
? \_\_kthread\_cancel\_work+0x40/0x40  
ret\_from\_fork+0x1f/0x40

v2:

- Add `IS_ICSK` check

v3:

- Update the commits in Fixes

More Info: <https://avd.aquasec.com/nvd/cve-2024-53091>

### **[CVE-2024-53094] kernel: RDMA/siw: Add `sendpage_ok()` check to disable `MSG_SPLICE_PAGES` (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/siw: Add `sendpage_ok()` check to disable `MSG_SPLICE_PAGES`

While running ISER over SIW, the initiator machine encounters a warning from `skb_splice_from_iter()` indicating that a slab page is being used in `send_page`. To address this, it is better to add a `sendpage_ok()` check within the driver itself, and if it returns 0, then `MSG_SPLICE_PAGES` flag should be disabled before entering the network stack.

A similar issue has been discussed for NVMe in this thread:

<https://lore.kernel.org/all/20240530142417.146696-1-ofir.gal@volumez.com/>

WARNING: CPU: 0 PID: 5342 at net/core/skbuff.c:7140 `skb_splice_from_iter`+0x173/0x320

Call Trace:

```
tcp_sendmsg_locked+0x368/0xe40
siw_tx_hdt+0x695/0xa40 [siw]
siw_qp_sq_process+0x102/0xb00 [siw]
siw_sq_resume+0x39/0x110 [siw]
siw_run_sq+0x74/0x160 [siw]
kthread+0xd2/0x100
ret_from_fork+0x34/0x40
ret_from_fork_asm+0x1a/0x30
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-53094>

**[CVE-2024-53095] kernel: smb: client: Fix use-after-free of network namespace. (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: Fix use-after-free of network namespace.

Recently, we got a customer report that CIFS triggers oops while reconnecting to a server. [0]

The workload runs on Kubernetes, and some pods mount CIFS servers in non-root network namespaces. The problem rarely happened, but it was always while the pod was dying.

The root cause is wrong reference counting for network namespace.

CIFS uses kernel sockets, which do not hold refcnt of the netns that the socket belongs to. That means CIFS must ensure the socket is always freed before its netns; otherwise, use-after-free happens.

The repro steps are roughly:

1. mount CIFS in a non-root netns
2. drop packets from the netns
3. destroy the netns
4. unmount CIFS

We can reproduce the issue quickly with the script [1] below and see the splat [2] if CONFIG\_NET\_NS\_REFCNT\_TRACKER is enabled.

When the socket is TCP, it is hard to guarantee the netns lifetime without holding refcnt due to async timers.

Let's hold netns refcnt for each socket as done for SMC in commit 9744d2bf1976 ("smc: Fix use-after-free in tcp\_write\_timer\_handler().").

Note that we need to move put\_net() from cifs\_put\_tcp\_session() to clean\_demultiplex\_info(); otherwise, \_\_sock\_create() still could touch a freed netns while cifsd tries to reconnect from cifs\_demultiplex\_thread().

Also, maybe\_get\_net() cannot be put just before \_\_sock\_create() because the code is not under RCU and there is a small chance that the same address happened to be reallocated to another netns.

[0]:

CIFS: VFS: \\XXXXXXXXXXXX has not responded in 15 seconds. Reconnecting...

CIFS: Serverclose failed 4 times, giving up

Unable to handle kernel paging request at virtual address 14de99e461f84a07

Mem abort info:

ESR = 0x0000000096000004

EC = 0x25: DABT (current EL), IL = 32 bits

SET = 0, FnV = 0

EA = 0, S1PTW = 0

FSC = 0x04: level 0 translation fault

Data abort info:

ISV = 0, ISS = 0x00000004

CM = 0, WnR = 0

[14de99e461f84a07] address between user and kernel address ranges

Internal error: Oops: 0000000096000004 [#1] SMP

Modules linked in: cls\_bpf sch\_ingress nls\_utf8 cifs cifs\_arc4 cifs\_md4 dns\_resolver tcp\_diag inet\_diag veth xt\_state xt\_connmark nf\_conntrack\_netlink xt\_nat xt\_statistic xt\_MASQUERADE xt\_mark xt\_addrtype ipt\_REJECT nf\_reject\_ipv4 nft\_chain\_nat nf\_nat xt\_conntrack nf\_conntrack nf\_defrag\_ipv6 nf\_defrag\_ipv4 xt\_comment nft\_compat nf\_tables nfnetlink overlay nls\_ascii nls\_cp437 sunrpc vfat fat aes\_ce\_blk aes\_ce\_cipher ghash\_ce sm4\_ce\_cipher sm4 sm3\_ce sm3 sha3\_ce sha512\_ce sha512\_arm64 sha1\_ce ena button sch\_fq\_codel loop fuse configfs dmi\_sysfs sha2\_ce sha256\_arm64 dm\_mirror dm\_region\_hash dm\_log dm\_mod dax efivarfs

CPU: 5 PID: 2690970 Comm: cifs Not tainted 6.1.103-109.184.amzn2023.aarch64 #1

Hardware name: Amazon EC2 r7g.4xlarge/, BIOS 1.0 11/1/2018

pstate: 00400005 (nzcw daif +PAN -UAO -TCO -DIT -SSBS BTYP=--)

pc : fib\_rules\_lookup+0x44/0x238

lr : \_\_fib\_lookup+0x64/0xbc

sp : ffff8000265db790

x29: ffff8000265db790 x28: 0000000000000000 x27: 000000000000bd01

x26: 0000000000000000 x25: ffff000b4baf8000 x24: ffff00047b5e4580

x23: ffff8000265db7e0 x22: 0000000000000000 x21: ffff00047b5e4500

x20: ffff0010e3f694f8 x19: 14de99e461f849f7 x18: 0000000000000000

x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000

x14: 0000000000000000 x13: 0000000000000000 x12: 3f92800abd010002

x11: 0000000000000001 x10: ffff0010e3f69420 x9 : ffff800008a6f294

x8 : 0000000000000000 x7 : 0000000000000006 x6 : 0000000000000000

x5 : 0000000000000001 x4 : ffff001924354280 x3 : ffff8000265db7e0

x2 : 0000000000000000 x1 : ffff0010e3f694f8 x0 : ffff00047b5e4500

Call trace:

fib\_rules\_lookup+0x44/0x238

\_\_fib\_lookup+0x64/0xbc

ip\_route\_output\_key\_hash\_rcu+0x2c4/0x398

ip\_route\_output\_key\_hash+0x60/0x8c

tcp\_v4\_connect+0x290/0x488

\_\_inet\_stream\_connect+0x108/0x3d0

inet\_stream\_connect+0x50/0x78

kernel\_connect+0x6c/0xac

generic\_ip\_conne

---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-53095>

**[CVE-2024-53114] kernel: x86/CPU/AMD: Clear virtualized VMLOAD/VMSAVE on Zen4 client (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

x86/CPU/AMD: Clear virtualized VMLOAD/VMSAVE on Zen4 client

A number of Zen4 client SoCs advertise the ability to use virtualized VMLOAD/VMSAVE, but using these instructions is reported to be a cause of a random host reboot.

These instructions aren't intended to be advertised on Zen4 client so clear the capability.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53114>

**[CVE-2024-53134] kernel: pmdomain: imx93-blk-ctrl: correct remove path (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

pmdomain: imx93-blk-ctrl: correct remove path

The check condition should be 'i < bc->onecell\_data.num\_domains', not 'bc->onecell\_data.num\_domains' which will make the look never finish and cause kernel panic.

Also disable runtime to address  
"imx93-blk-ctrl 4ac10000.system-controller: Unbalanced pm\_runtime\_enable!"

More Info: <https://avd.aquasec.com/nvd/cve-2024-53134>

**[CVE-2024-53147] kernel: exfat: fix out-of-bounds access of directory entries (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

exfat: fix out-of-bounds access of directory entries

In the case of the directory size is greater than or equal to the cluster size, if start\_clu becomes an EOF cluster(an invalid cluster) due to file system corruption, then the directory entry

where `ei->hint_femp.eidx` hint is outside the directory, resulting in an out-of-bounds access, which may cause further file system corruption.

This commit adds a check for `start_clu`, if it is an invalid cluster, the file or directory will be treated as empty.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53147>

## **[CVE-2024-53176] kernel: smb: During unmount, ensure all cached dir instances drop their dentry (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: During unmount, ensure all cached dir instances drop their dentry

The unmount process (`cifs_kill_sb()` calling `close_all_cached_dirs()`) can race with various cached directory operations, which ultimately results in dentries not being dropped and these kernel BUGs:

BUG: Dentry ffff88814f37e358{i=10000000000080,n=/} still in use (2) [unmount of cifs cifs]

VFS: Busy inodes after unmount of cifs (cifs)

-----[ cut here ]-----

kernel BUG at fs/super.c:661!

This happens when a `cfid` is in the process of being cleaned up when, and has been removed from the `cfids->entries` list, including:

- Receiving a lease break from the server
- Server reconnection triggers `invalidate_all_cached_dirs()`, which removes all the `cfids` from the list
- The `laundromat` thread decides to expire an old `cfid`.

To solve these problems, dropping the dentry is done in queued work done in a newly-added `cfid_put_wq` workqueue, and `close_all_cached_dirs()` flushes that workqueue after it drops all the dentries of which it's aware. This is a global workqueue (rather than scoped to a mount), but the queued work is minimal.

The final cleanup work for cleaning up a `cfid` is performed via work queued in the `serverclose_wq` workqueue; this is done separate from dropping the dentries so that `close_all_cached_dirs()` doesn't block on any server operations.

Both of these queued works expect to be invoked with a `cfid` reference and a `tcon` reference to avoid those objects from being freed while the work is ongoing.

While we're here, add proper locking to `close_all_cached_dirs()`, and

locking around the freeing of cfid->dentry.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53176>

**[CVE-2024-53177] kernel: smb: prevent use-after-free due to open\_cached\_dir error paths (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: prevent use-after-free due to open\_cached\_dir error paths

If open\_cached\_dir() encounters an error parsing the lease from the server, the error handling may race with receiving a lease break, resulting in open\_cached\_dir() freeing the cfid while the queued work is pending.

Update open\_cached\_dir() to drop refs rather than directly freeing the cfid.

Have cached\_dir\_lease\_break(), cfids\_laundromat\_worker(), and invalidate\_all\_cached\_dirs() clear has\_lease immediately while still holding cfids->cfid\_list\_lock, and then use this to also simplify the reference counting in cfids\_laundromat\_worker() and invalidate\_all\_cached\_dirs().

Fixes this KASAN splat (which manually injects an error and lease break in open\_cached\_dir()):

=====

BUG: KASAN: slab-use-after-free in smb2\_cached\_lease\_break+0x27/0xb0

Read of size 8 at addr ffff88811cc24c10 by task kworker/3:1/65

CPU: 3 UID: 0 PID: 65 Comm: kworker/3:1 Not tainted 6.12.0-rc6-g255cf264e6e5-dirty #87

Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020

Workqueue: cifsiod smb2\_cached\_lease\_break

Call Trace:

<TASK>

dump\_stack\_lvl+0x77/0xb0

print\_report+0xce/0x660

kasan\_report+0xd3/0x110

smb2\_cached\_lease\_break+0x27/0xb0

process\_one\_work+0x50a/0xc50

worker\_thread+0x2ba/0x530

kthread+0x17c/0x1c0

ret\_from\_fork+0x34/0x60

ret\_from\_fork\_asm+0x1a/0x30

</TASK>

Allocated by task 2464:

kasan\_save\_stack+0x33/0x60  
kasan\_save\_track+0x14/0x30  
\_\_kasan\_kmalloc+0xaa/0xb0  
open\_cached\_dir+0xa7d/0x1fb0  
smb2\_query\_path\_info+0x43c/0x6e0  
cifs\_get\_fattr+0x346/0xf10  
cifs\_get\_inode\_info+0x157/0x210  
cifs\_revalidate\_dentry\_attr+0x2d1/0x460  
cifs\_getattr+0x173/0x470  
vfs\_statx\_path+0x10f/0x160  
vfs\_statx+0xe9/0x150  
vfs\_fstatat+0x5e/0xc0  
\_\_do\_sys\_newfstatat+0x91/0xf0  
do\_syscall\_64+0x95/0x1a0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Freed by task 2464:

kasan\_save\_stack+0x33/0x60  
kasan\_save\_track+0x14/0x30  
kasan\_save\_free\_info+0x3b/0x60  
\_\_kasan\_slab\_free+0x51/0x70  
kfree+0x174/0x520  
open\_cached\_dir+0x97f/0x1fb0  
smb2\_query\_path\_info+0x43c/0x6e0  
cifs\_get\_fattr+0x346/0xf10  
cifs\_get\_inode\_info+0x157/0x210  
cifs\_revalidate\_dentry\_attr+0x2d1/0x460  
cifs\_getattr+0x173/0x470  
vfs\_statx\_path+0x10f/0x160  
vfs\_statx+0xe9/0x150  
vfs\_fstatat+0x5e/0xc0  
\_\_do\_sys\_newfstatat+0x91/0xf0  
do\_syscall\_64+0x95/0x1a0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Last potentially related work creation:

kasan\_save\_stack+0x33/0x60  
\_\_kasan\_record\_aux\_stack+0xad/0xc0  
insert\_work+0x32/0x100  
\_\_queue\_work+0x5c9/0x870  
queue\_work\_on+0x82/0x90  
open\_cached\_dir+0x1369/0x1fb0  
smb2\_query\_path\_info+0x43c/0x6e0  
cifs\_get\_fattr+0x346/0xf10  
cifs\_get\_inode\_info+0x157/0x210  
cifs\_revalidate\_dentry\_attr+0x2d1/0x460  
cifs\_getattr+0x173/0x470  
vfs\_statx\_path+0x10f/0x160  
vfs\_statx+0xe9/0x150  
vfs\_fstatat+0x5e/0xc0  
\_\_do\_sys\_newfstatat+0x91/0xf0  
do\_syscall\_64+0x95/0x1a0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

The buggy address belongs to the object at ffff88811cc24c00  
which belongs to the cache kmalloc-1k of size 1024  
The buggy address is located 16 bytes inside of  
freed 1024-byte region [ffff88811cc24c00, ffff88811cc25000)

More Info: <https://avd.aquasec.com/nvd/cve-2024-53177>

**[CVE-2024-53178] kernel: smb: Don't leak cfid when reconnect races with open\_cached\_dir (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: Don't leak cfid when reconnect races with open\_cached\_dir

open\_cached\_dir() may either race with the tcon reconnection even before compound\_send\_recv() or directly trigger a reconnection via SMB2\_open\_init() or SMB\_query\_info\_init().

The reconnection process invokes invalidate\_all\_cached\_dirs() via cifs\_mark\_open\_files\_invalid(), which removes all cfids from the cfids->entries list but doesn't drop a ref if has\_lease isn't true. This results in the currently-being-constructed cfid not being on the list, but still having a refcount of 2. It leaks if returned from open\_cached\_dir().

Fix this by setting cfid->has\_lease when the ref is actually taken; the cfid will not be used by other threads until it has a valid time.

Addresses these kmemleaks:

unreferenced object 0xffff8881090c4000 (size 1024):

comm "bash", pid 1860, jiffies 4295126592

hex dump (first 32 bytes):

```
00 01 00 00 00 00 ad de 22 01 00 00 00 00 ad de ..... " .....  
00 ca 45 22 81 88 ff ff f8 dc 4f 04 81 88 ff ff ..E".....O.....
```

backtrace (crc 6f58c20f):

```
[<ffffff8b895a1e>] __kmalloc_cache_noprof+0x2be/0x350  
[<ffffff8bda06e3>] open_cached_dir+0x993/0x1fb0  
[<ffffff8bdaa750>] cifs_readdir+0x15a0/0x1d50  
[<ffffff8b9a853f>] iterate_dir+0x28f/0x4b0  
[<ffffff8b9a9aed>] __x64_sys_getdents64+0xfd/0x200  
[<ffffff8cf6da05>] do_syscall_64+0x95/0x1a0  
[<ffffff8d00012f>] entry_SYSCALL_64_after_hwframe+0x76/0x7e
```

unreferenced object 0xffff8881044fdcf8 (size 8):

comm "bash", pid 1860, jiffies 4295126592

hex dump (first 8 bytes):

```
00 cc cc cc cc cc cc cc .....  
.....
```

backtrace (crc 10c106a9):



```
[<ffffff8b89a3d3>] __kmalloc_node_track_caller_noprof+0x363/0x480
[<ffffff8b7d7256>] kstrdup+0x36/0x60
[<ffffff8bda0700>] open_cached_dir+0x9b0/0x1fb0
[<ffffff8bdaa750>] cifs_readdir+0x15a0/0x1d50
[<ffffff8b9a853f>] iterate_dir+0x28f/0x4b0
[<ffffff8b9a9aed>] __x64_sys_getdents64+0xfd/0x200
[<ffffff8cf6da05>] do_syscall_64+0x95/0x1a0
[<ffffff8d00012f>] entry_SYSCALL_64_after_hwframe+0x76/0x7e
```

And addresses these BUG splats when unmounting the SMB filesystem:

```
BUG: Dentry ffff888140590ba0{i=10000000000080,n=/} still in use (2) [unmount of cifs cifs]
WARNING: CPU: 3 PID: 3433 at fs/dcache.c:1536 umount_check+0xd0/0x100
Modules linked in:
CPU: 3 UID: 0 PID: 3433 Comm: bash Not tainted 6.12.0-rc4-g850925a8133c-dirty #49
Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020
RIP: 0010:umount_check+0xd0/0x100
Code: 8d 7c 24 40 e8 31 5a f4 ff 49 8b 54 24 40 41 56 49 89 e9 45 89 e8 48 89 d9 41 57 48 89 de 48 c7 c7 80 e7 db ac
e8 f0 72 9a ff <0f> 0b 58 31 c0 5a 5b 5d 41 5c 41 5d 41 5e 41 5f e9 2b e5 5d 01 41
RSP: 0018:ffff88811cc27978 EFLAGS: 00010286
RAX: 0000000000000000 RBX: ffff888140590ba0 RCX: ffffffff00000000
RDX: dffffc0000000000 RSI: 0000000000000000 RDI: ffff8881f6fb6f40
RBP: ffff8881462ec000 R08: 0000000000000000 R09: ffffed1023984ee3
R10: ffff88811cc2771f R11: 00000000016cfcc0 R12: ffff888134383e08
R13: 0000000000000002 R14: ffff8881462ec668 R15: ffffffff00000000
FS: 00007f23bfa98740(0000) GS:ffff8881f6f80000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 0000556de4a6f808 CR3: 00000000123c8000 CR4: 0000000000350ef0
Call Trace:
<TASK>
d_walk+0x6a/0x530
shrink_dcache_for_umount+0x6a/0x200
generic_shutdown_super+0x52/0x2a0
kill_anon_super+0x22/0x40
cifs_kill_sb+0x159/0x1e0
deactivate_locked_super+0x66/0xe0
cleanup_mnt+0x140/0x210
task_work_run+0xfb/0x170
syscall_exit_to_user_mode+0x29f/0x2b0
do_syscall_64+0xa1/0x1a0
entry_SYSCALL_64_after_hwframe+0x76/0x7e
RIP: 0033:0x7f23bfb93ae7
Code: ff ff ff ff c3 66 0f 1f 44 00 00 48 8b 0d 11 93 0d 00 f7 d8 64 89 01 b8 ff ff ff eb bf 0f 1f 44 00 00 b8 50 00 00 00 0f
05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d e9 92 0d 00 f7 d8 64 89
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-53178>

### [CVE-2024-53187] kernel: io\_uring: check for overflows in io\_pin\_pages (Severity: MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

io\_uring: check for overflows in io\_pin\_pages

WARNING: CPU: 0 PID: 5834 at io\_uring/memmap.c:144 io\_pin\_pages+0x149/0x180 io\_uring/memmap.c:144  
CPU: 0 UID: 0 PID: 5834 Comm: syz-executor825 Not tainted 6.12.0-next-20241118-syzkaller #0

Call Trace:

<TASK>

\_\_io\_uaddr\_map+0xfb/0x2d0 io\_uring/memmap.c:183  
io\_rings\_map io\_uring/io\_uring.c:2611 [inline]  
io\_allocate\_scq\_urings+0x1c0/0x650 io\_uring/io\_uring.c:3470  
io\_uring\_create+0x5b5/0xc00 io\_uring/io\_uring.c:3692  
io\_uring\_setup io\_uring/io\_uring.c:3781 [inline]

...

</TASK>

io\_pin\_pages()'s uaddr parameter came directly from the user and can be garbage. Don't just add size to it as it can overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53187>

### **[CVE-2024-53195] kernel: KVM: arm64: Get rid of userspace\_irqchip\_in\_use (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

KVM: arm64: Get rid of userspace\_irqchip\_in\_use

Improper use of userspace\_irqchip\_in\_use led to syzbot hitting the following WARN\_ON() in kvm\_timer\_update\_irq():

WARNING: CPU: 0 PID: 3281 at arch/arm64/kvm/arch\_timer.c:459

kvm\_timer\_update\_irq+0x21c/0x394

Call trace:

kvm\_timer\_update\_irq+0x21c/0x394 arch/arm64/kvm/arch\_timer.c:459  
kvm\_timer\_vcpu\_reset+0x158/0x684 arch/arm64/kvm/arch\_timer.c:968  
kvm\_reset\_vcpu+0x3b4/0x560 arch/arm64/kvm/reset.c:264  
kvm\_vcpu\_set\_target arch/arm64/kvm/arm.c:1553 [inline]  
kvm\_arch\_vcpu\_ioctl\_vcpu\_init arch/arm64/kvm/arm.c:1573 [inline]  
kvm\_arch\_vcpu\_ioctl+0x112c/0x1b3c arch/arm64/kvm/arm.c:1695  
kvm\_vcpu\_ioctl+0x4ec/0xf74 virt/kvm/kvm\_main.c:4658  
vfs\_ioctl fs/ioctl.c:51 [inline]  
\_\_do\_sys\_ioctl fs/ioctl.c:907 [inline]  
\_\_se\_sys\_ioctl fs/ioctl.c:893 [inline]  
\_\_arm64\_sys\_ioctl+0x108/0x184 fs/ioctl.c:893  
\_\_invoke\_syscall arch/arm64/kernel/syscall.c:35 [inline]  
invoke\_syscall+0x78/0x1b8 arch/arm64/kernel/syscall.c:49  
el0\_svc\_common+0xe8/0x1b0 arch/arm64/kernel/syscall.c:132  
do\_el0\_svc+0x40/0x50 arch/arm64/kernel/syscall.c:151  
el0\_svc+0x54/0x14c arch/arm64/kernel/entry-common.c:712  
el0t\_64\_sync\_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:730

The following sequence led to the scenario:

- Userspace creates a VM and a vCPU.
- The vCPU is initialized with KVM\_ARM\_VCPU\_PMU\_V3 during KVM\_ARM\_VCPU\_INIT.
- Without any other setup, such as vGIC or vPMU, userspace issues KVM\_RUN on the vCPU. Since the vPMU is requested, but not setup, `kvm_arm_pmu_v3_enable()` fails in `kvm_arch_vcpu_run_pid_change()`. As a result, KVM\_RUN returns after enabling the timer, but before incrementing 'userspace\_irqchip\_in\_use':

```
kvm_arch_vcpu_run_pid_change()
    ret = kvm_arm_pmu_v3_enable()
    if (!vcpu->arch.pmu.created)
        return -EINVAL;
```

```
    if (ret)
        return ret;
```

```
    [...]
```

```
    if (!irqchip_in_kernel(kvm))
        static_branch_inc(&userspace_irqchip_in_use);
```

- Userspace ignores the error and issues KVM\_ARM\_VCPU\_INIT again. Since the timer is already enabled, control moves through the following flow, ultimately hitting the `WARN_ON()`:

```
kvm_timer_vcpu_reset()
    if (timer->enabled)
        kvm_timer_update_irq()
        if (!userspace_irqchip())
            ret = kvm_vgic_inject_irq()
            ret = vgic_lazy_init()
            if (unlikely(!vgic_initialized(kvm)))
                if (kvm->arch.vgic.vgic_model !=
                    KVM_DEV_TYPE_ARM_VGIC_V2)
                        return -EBUSY;
        WARN_ON(ret);
```

Theoretically, since `userspace_irqchip_in_use`'s functionality can be simply replaced by '`irqchip_in_kernel()`', get rid of the static key to avoid the mismanagement, which also helps with the syzbot issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53195>

## **[CVE-2024-53209] kernel: bnxt\_en: Fix receive ring space parameters when XDP is active (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

`bnxt_en`: Fix receive ring space parameters when XDP is active

The MTU setting at the time an XDP multi-buffer is attached

determines whether the aggregation ring will be used and the rx\_skb\_func handler. This is done in bnxt\_set\_rx\_skb\_mode().

If the MTU is later changed, the aggregation ring setting may need to be changed and it may become out-of-sync with the settings initially done in bnxt\_set\_rx\_skb\_mode(). This may result in random memory corruption and crashes as the HW may DMA data larger than the allocated buffer size, such as:

```
BUG: kernel NULL pointer dereference, address: 00000000000003c0
PGD 0 P4D 0
Oops: 0000 [#1] PREEMPT SMP NOPTI
CPU: 17 PID: 0 Comm: swapper/17 Kdump: loaded Tainted: G S      OE      6.1.0-226bf9805506 #1
Hardware name: Wiyynn Delta Lake PVT BZA.02601.0150/Delta Lake-Class1, BIOS F0E_3A12 08/26/2021
RIP: 0010:bnxt_rx_pkt+0xe97/0x1ae0 [bnxt_en]
Code: 8b 95 70 ff ff ff 4c 8b 9d 48 ff ff ff 66 41 89 87 b4 00 00 00 e9 0b f7 ff ff 0f b7 43 0a 49 8b 95 a8 04 00 00 25 ff 0f
00 00 <0f> b7 14 42 48 c1 e2 06 49 03 95 a0 04 00 00 0f b6 42 33f
RSP: 0018:ffffa19f40cc0d18 EFLAGS: 00010202
RAX: 000000000000001e0 RBX: ffff8e2c805c6100 RCX: 000000000000007ff
RDX: 0000000000000000 RSI: ffff8e2c271ab990 RDI: ffff8e2c84f12380
RBP: fffffa19f40cc0e48 R08: 0000000000001000d R09: 974ea2fcddfa4cbf
R10: 0000000000000000 R11: fffffa19f40cc0ff8 R12: ffff8e2c94b58980
R13: ffff8e2c952d6600 R14: 0000000000000016 R15: ffff8e2c271ab990
FS: 0000000000000000(0000) GS:ffff8e3b3f840000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 000000000000003c0 CR3: 00000000e8580a004 CR4: 00000000007706e0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400
PKRU: 55555554
Call Trace:
<IRQ>
__bnxt_poll_work+0x1c2/0x3e0 [bnxt_en]
```

To address the issue, we now call bnxt\_set\_rx\_skb\_mode() within bnxt\_change\_mtu() to properly set the AGG rings configuration and update rx\_skb\_func based on the new MTU value. Additionally, BNXT\_FLAG\_NO\_AGG\_RINGS is cleared at the beginning of bnxt\_set\_rx\_skb\_mode() to make sure it gets set or cleared based on the current MTU.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53209>

### **[CVE-2024-53218] kernel: f2fs: fix race in concurrent f2fs\_stop\_gc\_thread (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix race in concurrent f2fs\_stop\_gc\_thread

In my test case, concurrent calls to f2fs shutdown report the following stack trace:

Oops: general protection fault, probably for non-canonical address 0xc6cfff63bb5513fc: 0000 [#1] PREEMPT SMP PTI  
CPU: 0 UID: 0 PID: 678 Comm: f2fs\_rep\_shutdo Not tainted 6.12.0-rc5-next-20241029-g6fb2fa9805c5-dirty #85

Call Trace:

<TASK>

? show\_regs+0x8b/0xa0  
? \_\_die\_body+0x26/0xa0  
? die\_addr+0x54/0x90  
? exc\_general\_protection+0x24b/0x5c0  
? asm\_exc\_general\_protection+0x26/0x30  
? kthread\_stop+0x46/0x390  
f2fs\_stop\_gc\_thread+0x6c/0x110  
f2fs\_do\_shutdown+0x309/0x3a0  
f2fs\_ioc\_shutdown+0x150/0x1c0  
\_\_f2fs\_ioctl+0xffd/0x2ac0  
f2fs\_ioctl+0x76/0xe0  
vfs\_ioctl+0x23/0x60  
\_\_x64\_sys\_ioctl+0xce/0xf0  
x64\_sys\_call+0x2b1b/0x4540  
do\_syscall\_64+0xa7/0x240  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

The root cause is a race condition in f2fs\_stop\_gc\_thread() called from different f2fs shutdown paths:

[CPU0]	[CPU1]
f2fs_stop_gc_thread	f2fs_stop_gc_thread
	gc_th = sbi->gc_thread
gc_th = sbi->gc_thread	
kfree(gc_th)	
sbi->gc_thread = NULL	
	< gc_th != NULL >
	kthread_stop(gc_th->f2fs_gc_task) //UAF

The commit c7f114d864ac ("f2fs: fix to avoid use-after-free in f2fs\_stop\_gc\_thread()") attempted to fix this issue by using a read semaphore to prevent races between shutdown and remount threads, but it fails to prevent all race conditions.

Fix it by converting to write lock of s\_umount in f2fs\_do\_shutdown().

More Info: <https://avd.aquasec.com/nvd/cve-2024-53218>

### [CVE-2024-53219] kernel: virtiofs: use pages instead of pointer for kernel direct IO (Severity: MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

virtiofs: use pages instead of pointer for kernel direct IO

When trying to insert a 10MB kernel module kept in a virtio-fs with cache disabled, the following warning was reported:

```
-----[ cut here ]-----
WARNING: CPU: 1 PID: 404 at mm/page_alloc.c:4551 .....
Modules linked in:
CPU: 1 PID: 404 Comm: insmod Not tainted 6.9.0-rc5+ #123
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) .....
RIP: 0010:___alloc_pages+0x2bf/0x380
.....
Call Trace:
<TASK>
? __warn+0x8e/0x150
? __alloc_pages+0x2bf/0x380
__kmalloc_large_node+0x86/0x160
__kmalloc+0x33c/0x480
virtio_fs_enqueue_req+0x240/0x6d0
virtio_fs_wake_pending_and_unlock+0x7f/0x190
queue_request_and_unlock+0x55/0x60
fuse_simple_request+0x152/0x2b0
fuse_direct_io+0x5d2/0x8c0
fuse_file_read_iter+0x121/0x160
__kernel_read+0x151/0x2d0
kernel_read+0x45/0x50
kernel_read_file+0x1a9/0x2a0
init_module_from_file+0x6a/0xe0
idempotent_init_module+0x175/0x230
__x64_sys_finit_module+0x5d/0xb0
x64_sys_call+0x1c3/0x9e0
do_syscall_64+0x3d/0xc0
entry_SYSCALL_64_after_hwframe+0x4b/0x53
.....
</TASK>
---[ end trace 0000000000000000 ]---
```

The warning is triggered as follows:

1) syscall finit\_module() handles the module insertion and it invokes kernel\_read\_file() to read the content of the module first.

2) kernel\_read\_file() allocates a 10MB buffer by using vmalloc() and passes it to kernel\_read(). kernel\_read() constructs a kvec iter by using iov\_iter\_kvec() and passes it to fuse\_file\_read\_iter().

3) virtio-fs disables the cache, so fuse\_file\_read\_iter() invokes fuse\_direct\_io(). As for now, the maximal read size for kvec iter is only limited by fc->max\_read. For virtio-fs, max\_read is UINT\_MAX, so fuse\_direct\_io() doesn't split the 10MB buffer. It saves the address and the size of the 10MB-sized buffer in out\_args[0] of a fuse request and passes the fuse request to virtio\_fs\_wake\_pending\_and\_unlock().

4) virtio\_fs\_wake\_pending\_and\_unlock() uses virtio\_fs\_enqueue\_req() to queue the request. Because virtiofs need DMA-able address, so

virtio\_fs\_enqueue\_req() uses kmalloc() to allocate a bounce buffer for all fuse args, copies these args into the bounce buffer and passed the physical address of the bounce buffer to virtiofsd. The total length of these fuse args for the passed fuse request is about 10MB, so copy\_args\_to\_argbuf() invokes kmalloc() with a 10MB size parameter and it triggers the warning in \_\_alloc\_pages():

```
if (WARN_ON_ONCE_GFP(order > MAX_PAGE_ORDER, gfp))
    return NULL;
```

5) virtio\_fs\_enqueue\_req() will retry the memory allocation in a kworker, but it won't help, because kmalloc() will always return NULL due to the abnormal size and finit\_module() will hang forever.

A feasible solution is to limit the value of max\_read for virtio-fs, so the length passed to kmalloc() will be limited. However it will affect the maximal read size for normal read. And for virtio-fs write initiated from kernel, it has the similar problem but now there is no way to limit fc->max\_write in kernel.

So instead of limiting both the values of max\_read and max\_write in kernel, introducing use\_pages\_for\_kvec\_io in fuse\_conn and setting it as true in virtiofs. When use\_pages\_for\_kvec\_io is enabled, fuse will use pages instead of pointer to pass the KVEC\_IO data.

After switching to pages for KVEC\_IO data, these pages will be used for DMA through virtio-fs. If these pages are backed by vmalloc(), {flush|invalidate}\_kernel\_vmap\_range() are necessary to flush or invalidate the cache before the DMA operation. So add two new fields in fuse\_args\_pages to record the base address of vmalloc area and the condition indicating whether invalidation is needed. Perform the flush in fuse\_get\_user\_pages() for write operations and the invalidation in fuse\_release\_user\_pages() for read operations.

It may seem necessary to introduce another file  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-53219>

### **[CVE-2024-53221] kernel: f2fs: fix null-ptr-deref in f2fs\_submit\_page\_bio() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix null-ptr-deref in f2fs\_submit\_page\_bio()

There's issue as follows when concurrently installing the f2fs.ko module and mounting the f2fs file system:

KASAN: null-ptr-deref in range [0x0000000000000020-0x0000000000000027]

RIP: 0010:\_\_bio\_alloc+0x2fb/0x6c0 [f2fs]

Call Trace:

<TASK>

```
f2fs_submit_page_bio+0x126/0x8b0 [f2fs]
__get_meta_page+0x1d4/0x920 [f2fs]
get_checkpoint_version.constprop.0+0x2b/0x3c0 [f2fs]
validate_checkpoint+0xac/0x290 [f2fs]
f2fs_get_valid_checkpoint+0x207/0x950 [f2fs]
f2fs_fill_super+0x1007/0x39b0 [f2fs]
mount_bdev+0x183/0x250
legacy_get_tree+0xf4/0x1e0
vfs_get_tree+0x88/0x340
do_new_mount+0x283/0x5e0
path_mount+0x2b2/0x15b0
__x64_sys_mount+0x1fe/0x270
do_syscall_64+0x5f/0x170
entry_SYSCALL_64_after_hwframe+0x76/0x7e
```

Above issue happens as the biset of the f2fs file system is not initialized before register "f2fs\_fs\_type".

To address above issue just register "f2fs\_fs\_type" at the last in init\_f2fs\_fs(). Ensure that all f2fs file system resources are initialized.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53221>

### **[CVE-2024-53224] kernel: RDMA/mlx5: Move events notifier registration to be after device registration (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/mlx5: Move events notifier registration to be after device registration

Move pkey change work initialization and cleanup from device resources stage to notifier stage, since this is the stage which handles this work events.

Fix a race between the device deregistration and pkey change work by moving MLX5\_IB\_STAGE\_DEVICE\_NOTIFIER to be after MLX5\_IB\_STAGE\_IB\_REG in order to ensure that the notifier is deregistered before the device during cleanup. Which ensures there are no works that are being executed after the device has already unregistered which can cause the panic below.

BUG: kernel NULL pointer dereference, address: 0000000000000000

PGD 0 P4D 0

Oops: 0000 [#1] PREEMPT SMP PTI

CPU: 1 PID: 630071 Comm: kworker/1:2 Kdump: loaded Tainted: G W OE ----- --- 5.14.0-162.6.1.el9\_1.x86\_64 #1

Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS 090008 02/27/2023

Workqueue: events pkey\_change\_handler [mlx5\_ib]

RIP: 0010:setup\_qp+0x38/0x1f0 [mlx5\_ib]

Code: ee 41 54 45 31 e4 55 89 f5 53 48 89 fb 48 83 ec 20 8b 77 08 65 48 8b 04 25 28 00 00 00 48 89 44 24 18 48 8b



07 48 8d 4c 24 16 <4c> 8b 38 49 8b 87 80 0b 00 00 4c 89 ff 48 8b 80 08 05 00 00 8b 40

RSP: 0018:ffffbcc54068be20 EFLAGS: 00010282

RAX: 0000000000000000 RBX: ffff954054494128 RCX: fffffcc54068be36

RDY: ffff954004934000 RSI: 0000000000000001 RDI: ffff954054494128

RBP: 0000000000000023 R08: ffff954001be2c20 R09: 0000000000000001

R10: ffff954001be2c20 R11: ffff9540260133c0 R12: 0000000000000000

R13: 0000000000000023 R14: 0000000000000000 R15: ffff9540ffcb0905

FS: 0000000000000000(0000) GS:ffff9540ffc80000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 0000000000000000 CR3: 000000010625c001 CR4: 00000000003706e0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

Call Trace:

mlx5\_ib\_gsi\_pkey\_change+0x20/0x40 [mlx5\_ib]

process\_one\_work+0x1e8/0x3c0

worker\_thread+0x50/0x3b0

? rescuer\_thread+0x380/0x380

kthread+0x149/0x170

? set\_kthread\_struct+0x50/0x50

ret\_from\_fork+0x22/0x30

Modules linked in: rdma\_ucm(OE) rdma\_cm(OE) iw\_cm(OE) ib\_ipoib(OE) ib\_cm(OE) ib\_umad(OE) mlx5\_ib(OE) mlx5\_fwctl(OE) fwctl(OE) ib\_uverbs(OE) mlx5\_core(OE) mlxdevm(OE) ib\_core(OE) mlx\_compat(OE) psample mlxfw(OE) tls knem(OE) netconsole nfsv3 nfs\_acl nfs lockd grace fscache netfs qrtr rkill sunrpc intel\_rapl\_msr intel\_rapl\_common rapl hv\_balloon hv\_utils i2c\_piix4 pcspkr joydev fuse ext4 mbcache jbd2 sr\_mod sd\_mod cdrom t10\_pi sg ata\_generic pci\_hyperv pci\_hyperv\_intf hyperv\_drm drm\_shmem\_helper drm\_kms\_helper hv\_storvsc syscopyarea hv\_netvsc sysfillrect sysimgblt hid\_hyperv fb\_sys\_fops scsi\_transport\_fc hyperv\_keyboard drm ata\_piix crct10dif\_pclmul crc32\_pclmul crc32c\_intel libata ghash\_clmulni\_intel hv\_vmbus serio\_raw [last unloaded: ib\_core]

CR2: 0000000000000000

---[ end trace f6f8be4eae12f7bc ]---

More Info: <https://avd.aquasec.com/nvd/cve-2024-53224>

## **[CVE-2024-53687] kernel: riscv: Fix IPIs usage in kfence\_protect\_page() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

riscv: Fix IPIs usage in kfence\_protect\_page()

flush\_tlb\_kernel\_range() may use IPIs to flush the TLBs of all the cores, which triggers the following warning when the irqs are disabled:

```
[ 3.455330] WARNING: CPU: 1 PID: 0 at kernel/smp.c:815 smp_call_function_many_cond+0x452/0x520
[ 3.456647] Modules linked in:
[ 3.457218] CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Not tainted 6.12.0-rc7-00010-g91d3de7240b8 #1
[ 3.457416] Hardware name: QEMU QEMU Virtual Machine, BIOS
[ 3.457633] epc : smp_call_function_many_cond+0x452/0x520
[ 3.457736] ra : on_each_cpu_cond_mask+0x1e/0x30
[ 3.457786] epc : ffffffff800b669a ra : ffffffff800b67c2 sp : ff2000000000bb50
[ 3.457824] gp : ffffffff815212b8 tp : ff6000008014f080 t0 : 0000000000000003f
[ 3.457859] t1 : ffffffff815221e0 t2 : 000000000000000f s0 : ff2000000000bc10
```

```
[ 3.457920] s1 : 0000000000000040 a0 : ffffffff815221e0 a1 : 0000000000000001
[ 3.457953] a2 : 0000000000010000 a3 : 0000000000000003 a4 : 0000000000000000
[ 3.458006] a5 : 0000000000000000 a6 : ffffffff80000000 a7 : 0000000000000000
[ 3.458042] s2 : ffffffff815223be s3 : 00ffffffff0000 s4 : ff600001ffe38fc0
[ 3.458076] s5 : ff600001ff950d00 s6 : 0000000200000120 s7 : 0000000000000001
[ 3.458109] s8 : 0000000000000001 s9 : ff60000080841ef0 s10: 0000000000000001
[ 3.458141] s11: ffffffff81524812 t3 : 0000000000000001 t4 : ff60000080092bc0
[ 3.458172] t5 : 0000000000000000 t6 : ff200000000236d0
[ 3.458203] status: 0000000200000100 badaddr: ffffffff800b669a cause: 0000000000000003
[ 3.458373] [<ffffffff800b669a>] smp_call_function_many_cond+0x452/0x520
[ 3.458593] [<ffffffff800b67c2>] on_each_cpu_cond_mask+0x1e/0x30
[ 3.458625] [<ffffffff8000e4ca>] __flush_tlb_range+0x118/0x1ca
[ 3.458656] [<ffffffff8000e6b2>] flush_tlb_kernel_range+0x1e/0x26
[ 3.458683] [<ffffffff801ea56a>] kfence_protect+0xc0/0xce
[ 3.458717] [<ffffffff801e9456>] kfence_guarded_free+0xc6/0x1c0
[ 3.458742] [<ffffffff801e9d6c>] __kfence_free+0x62/0xc6
[ 3.458764] [<ffffffff801c57d8>] kfree+0x106/0x32c
[ 3.458786] [<ffffffff80588cf2>] detach_buf_split+0x188/0x1a8
[ 3.458816] [<ffffffff8058708c>] virtqueue_get_buf_ctx+0xb6/0x1f6
[ 3.458839] [<ffffffff805871da>] virtqueue_get_buf+0xe/0x16
[ 3.458880] [<ffffffff80613d6a>] virtblk_done+0x5c/0xe2
[ 3.458908] [<ffffffff8058766e>] vring_interrupt+0x6a/0x74
[ 3.458930] [<ffffffff800747d8>] __handle_irq_event_percpu+0x7c/0xe2
[ 3.458956] [<ffffffff800748f0>] handle_irq_event+0x3c/0x86
[ 3.458978] [<ffffffff800786cc>] handle_simple_irq+0x9e/0xbe
[ 3.459004] [<ffffffff80073934>] generic_handle_domain_irq+0x1c/0x2a
[ 3.459027] [<ffffffff804bf87c>] imsic_handle_irq+0xba/0x120
[ 3.459056] [<ffffffff80073934>] generic_handle_domain_irq+0x1c/0x2a
[ 3.459080] [<ffffffff804bdb76>] riscv_intc_aia_irq+0x24/0x34
[ 3.459103] [<ffffffff809d0452>] handle_riscv_irq+0x2e/0x4c
[ 3.459133] [<ffffffff809d923e>] call_on_irq_stack+0x32/0x40
```

So only flush the local TLB and let the lazy kfence page fault handling deal with the faults which could happen when a core has an old protected pte version cached in its TLB. That leads to potential inaccuracies which can be tolerated when using kfence.

More Info: <https://avd.aquasec.com/nvd/cve-2024-53687>

## **[CVE-2024-54456] kernel: NFS: Fix potential buffer overflow in nfs\_sysfs\_link\_rpc\_client() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

NFS: Fix potential buffer overflow in nfs\_sysfs\_link\_rpc\_client()

name is char[64] where the size of clnt->cl\_program->name remains unknown. Invoking strcat() directly will also lead to potential buffer overflow. Change them to strncpy() and strncat() to fix potential

issues.

More Info: <https://avd.aquasec.com/nvd/cve-2024-54456>

**[CVE-2024-54683] kernel: netfilter: IDLETIMER: Fix for possible ABBA deadlock (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

netfilter: IDLETIMER: Fix for possible ABBA deadlock

Deletion of the last rule referencing a given idletimer may happen at the same time as a read of its file in sysfs:

```
| =====  
| WARNING: possible circular locking dependency detected  
| 6.12.0-rc7-01692-g5e9a28f41134-dirty #594 Not tainted  
| -----  
| iptables/3303 is trying to acquire lock:  
| ffff8881057e04b8 (kn->active#48){++++}-{0:0}, at: __kernfs_remove+0x20  
|  
| but task is already holding lock:  
| ffffffff80249068 (list_mutex){+..+}-{3:3}, at: idletimer_tg_destroy_v]  
|  
| which lock already depends on the new lock.
```

A simple reproducer is:

```
| #!/bin/bash  
|  
| while true; do  
|     iptables -A INPUT -i foo -j IDLETIMER --timeout 10 --label "testme"  
|     iptables -D INPUT -i foo -j IDLETIMER --timeout 10 --label "testme"  
| done &  
| while true; do  
|     cat /sys/class/xt_idletimer/timers/testme >/dev/null  
| done
```

Avoid this by freeing list\_mutex right after deleting the element from the list, then continuing with the teardown.

More Info: <https://avd.aquasec.com/nvd/cve-2024-54683>

**[CVE-2024-56544] kernel: udmabuf: change folios array from kmalloc to kvmalloc (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

udmabuf: change folios array from kmalloc to kvmalloc

When PAGE\_SIZE 4096, MAX\_PAGE\_ORDER 10, 64bit machine,  
page\_alloc only support 4MB.  
If above this, trigger this warn and return NULL.

udmabuf can change size limit, if change it to 3072(3GB), and then alloc  
3GB udmabuf, will fail create.

```
[ 4080.876581] -----[ cut here ]-----  
[ 4080.876843] WARNING: CPU: 3 PID: 2015 at mm/page_alloc.c:4556 __alloc_pages+0x2c8/0x350  
[ 4080.878839] RIP: 0010:__alloc_pages+0x2c8/0x350  
[ 4080.879470] Call Trace:  
[ 4080.879473] <TASK>  
[ 4080.879473] ? __alloc_pages+0x2c8/0x350  
[ 4080.879475] ? __warn.cold+0x8e/0xe8  
[ 4080.880647] ? __alloc_pages+0x2c8/0x350  
[ 4080.880909] ? report_bug+0xff/0x140  
[ 4080.881175] ? handle_bug+0x3c/0x80  
[ 4080.881556] ? exc_invalid_op+0x17/0x70  
[ 4080.881559] ? asm_exc_invalid_op+0x1a/0x20  
[ 4080.882077] ? udmabuf_create+0x131/0x400
```

Because MAX\_PAGE\_ORDER, kmalloc can max alloc  $4096 * (1 \ll 10)$ , 4MB  
memory, each array entry is pointer(8byte), so can save 524288 pages(2GB).

Further more, costly order(order 3) may not be guaranteed that it can be  
applied for, due to fragmentation.

This patch change udmabuf array use kvmalloc\_array, this can fallback  
alloc into vmalloc, which can guarantee allocation for any size and does  
not affect the performance of kmalloc allocations.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56544>

### **[CVE-2024-56565] kernel: f2fs: fix to drop all discards after creating snapshot on lvm device (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to drop all discards after creating snapshot on lvm device

Piergiorgio reported a bug in bugzilla as below:

```
-----[ cut here ]-----  
WARNING: CPU: 2 PID: 969 at fs/f2fs/segment.c:1330  
RIP: 0010:__submit_discard_cmd+0x27d/0x400 [f2fs]  
Call Trace:  
__issue_discard_cmd+0x1ca/0x350 [f2fs]
```

```
issue_discard_thread+0x191/0x480 [f2fs]
kthread+0xcf/0x100
ret_from_fork+0x31/0x50
ret_from_fork_asm+0x1a/0x30
```

w/ below testcase, it can reproduce this bug quickly:

```
- pvcreate /dev/vdb
- vgcreate myvg1 /dev/vdb
- lvcreate -L 1024m -n mylv1 myvg1
- mount /dev/myvg1/mylv1 /mnt/f2fs
- dd if=/dev/zero of=/mnt/f2fs/file bs=1M count=20
- sync
- rm /mnt/f2fs/file
- sync
- lvcreate -L 1024m -s -n mylv1-snapshot /dev/myvg1/mylv1
- umount /mnt/f2fs
```

The root cause is: it will update `discard_max_bytes` of mounted lvm device to zero after creating snapshot on this lvm device, then, `__submit_discard_cmd()` will pass parameter `@nr_sects` w/ zero value to `__blkdev_issue_discard()`, it returns a NULL bio pointer, result in panic.

This patch changes as below for fixing:

1. Let's drop all remained discards in `f2fs_unfreeze()` if snapshot of lvm device is created.
2. Checking `discard_max_bytes` before submitting discard during `__submit_discard_cmd()`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56565>

### **[CVE-2024-56566] kernel: mm/slub: Avoid list corruption when removing a slab from the full list (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/slub: Avoid list corruption when removing a slab from the full list

Boot with `slub_debug=UFPZ`.

If allocated object failed in `alloc_consistency_checks`, all objects of the slab will be marked as used, and then the slab will be removed from the partial list.

When an object belonging to the slab got freed later, the `remove_full()` function is called. Because the slab is neither on the partial list nor on the full list, it eventually lead to a list corruption (actually a list poison being detected).

So we need to mark and isolate the slab page with metadata corruption,  
do not put it back in circulation.

Because the debug caches avoid all the fastpaths, reusing the frozen bit  
to mark slab page with metadata corruption seems to be fine.

```
[ 4277.385669] list_del corruption, ffffea00044b3e50->next is LIST_POISON1 (dead000000000100)
[ 4277.387023] -----[ cut here ]-----
[ 4277.387880] kernel BUG at lib/list_debug.c:56!
[ 4277.388680] invalid opcode: 0000 [#1] PREEMPT SMP PTI
[ 4277.389562] CPU: 5 PID: 90 Comm: kworker/5:1 Kdump: loaded Tainted: G      OE   6.6.1-1 #1
[ 4277.392113] Workqueue: xfs-inodegc/vda1 xfs_inodegc_worker [xfs]
[ 4277.393551] RIP: 0010:___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.394518] Code: 48 91 82 e8 37 f9 9a ff 0f 0b 48 89 fe 48 c7 c7 28 49 91 82 e8 26 f9 9a ff 0f 0b 48 89 fe 48 c7 c7
58 49 91
[ 4277.397292] RSP: 0018:ffffc90000333b38 EFLAGS: 00010082
[ 4277.398202] RAX: 0000000000000004 RBX: ffffea00044b3e50 RCX: 0000000000000000
[ 4277.399340] RDX: 0000000000000002 RSI: ffffffff828f8715 RDI: 00000000ffffffff
[ 4277.400545] RBP: ffffea00044b3e40 R08: 0000000000000000 R09: ffffc900003339f0
[ 4277.401710] R10: 0000000000000003 R11: ffffffff82d44088 R12: ffff888112cf9910
[ 4277.402887] R13: 0000000000000001 R14: 0000000000000001 R15: ffff8881000424c0
[ 4277.404049] FS: 0000000000000000(0000) GS:ffff88842fd40000(0000) knlGS:0000000000000000
[ 4277.405357] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 4277.406389] CR2: 00007f2ad0b24000 CR3: 0000000102a3a006 CR4: 00000000007706e0
[ 4277.407589] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[ 4277.408780] DR3: 0000000000000000 DR6: 000000000ffe0ff0 DR7: 0000000000000040
[ 4277.410000] PKRU: 55555554
[ 4277.410645] Call Trace:
[ 4277.411234] <TASK>
[ 4277.411777] ? die+0x32/0x80
[ 4277.412439] ? do_trap+0xd6/0x100
[ 4277.413150] ? ___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.414158] ? do_error_trap+0x6a/0x90
[ 4277.414948] ? ___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.415915] ? exc_invalid_op+0x4c/0x60
[ 4277.416710] ? ___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.417675] ? asm_exc_invalid_op+0x16/0x20
[ 4277.418482] ? ___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.419466] ? ___list_del_entry_valid_or_report+0x7b/0xc0
[ 4277.420410] free_to_partial_list+0x515/0x5e0
[ 4277.421242] ? xfs_iext_remove+0x41a/0xa10 [xfs]
[ 4277.422298] xfs_iext_remove+0x41a/0xa10 [xfs]
[ 4277.423316] ? xfs_inodegc_worker+0xb4/0x1a0 [xfs]
[ 4277.424383] xfs_bmap_del_extents_delay+0x4fe/0x7d0 [xfs]
[ 4277.425490] __xfs_bunmap+0x50d/0x840 [xfs]
[ 4277.426445] xfs_itruncate_extents_flags+0x13a/0x490 [xfs]
[ 4277.427553] xfs_inactive_truncate+0xa3/0x120 [xfs]
[ 4277.428567] xfs_inactive+0x22d/0x290 [xfs]
[ 4277.429500] xfs_inodegc_worker+0xb4/0x1a0 [xfs]
[ 4277.430479] process_one_work+0x171/0x340
[ 4277.431227] worker_thread+0x277/0x390
[ 4277.431962] ? __pfx_worker_thread+0x10/0x10
[ 4277.432752] kthread+0xf0/0x120
```

```
[ 4277.433382] ? __pfx_kthread+0x10/0x10
[ 4277.434134] ret_from_fork+0x2d/0x50
[ 4277.434837] ? __pfx_kthread+0x10/0x10
[ 4277.435566] ret_from_fork_asm+0x1b/0x30
[ 4277.436280] </TASK>
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-56566>

**[CVE-2024-56583] kernel: sched/deadline: Fix warning in migrate\_enable for boosted tasks (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

sched/deadline: Fix warning in migrate\_enable for boosted tasks

When running the following command:

```
while true; do
    stress-ng --cyclic 30 --timeout 30s --minimize --quiet
done
```

a warning is eventually triggered:

```
WARNING: CPU: 43 PID: 2848 at kernel/sched/deadline.c:794
setup_new_dl_entity+0x13e/0x180
```

...

Call Trace:

<TASK>

```
? show_trace_log_lvl+0x1c4/0x2df
? enqueue_dl_entity+0x631/0x6e0
? setup_new_dl_entity+0x13e/0x180
? __warn+0x7e/0xd0
? report_bug+0x11a/0x1a0
? handle_bug+0x3c/0x70
? exc_invalid_op+0x14/0x70
? asm_exc_invalid_op+0x16/0x20
enqueue_dl_entity+0x631/0x6e0
enqueue_task_dl+0x7d/0x120
__do_set_cpus_allowed+0xe3/0x280
__set_cpus_allowed_ptr_locked+0x140/0x1d0
__set_cpus_allowed_ptr+0x54/0xa0
migrate_enable+0x7e/0x150
rt_spin_unlock+0x1c/0x90
group_send_sig_info+0xf7/0x1a0
? kill_pid_info+0x1f/0x1d0
kill_pid_info+0x78/0x1d0
kill_proc_info+0x5b/0x110
__x64_sys_kill+0x93/0xc0
do_syscall_64+0x5c/0xf0
```

entry\_SYSCALL\_64\_after\_hwframe+0x6e/0x76  
RIP: 0033:0x7f0dab31f92b

This warning occurs because set\_cpus\_allowed dequeues and enqueues tasks with the ENQUEUE\_RESTORE flag set. If the task is boosted, the warning is triggered. A boosted task already had its parameters set by rt\_mutex\_setprio, and a new call to setup\_new\_dl\_entity is unnecessary, hence the WARN\_ON call.

Check if we are requeueing a boosted task and avoid calling setup\_new\_dl\_entity if that's the case.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56583>

### **[CVE-2024-56588] kernel: scsi: hisi\_sas: Create all dump files during debugfs initialization (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: hisi\_sas: Create all dump files during debugfs initialization

For the current debugfs of hisi\_sas, after user triggers dump, the driver allocate memory space to save the register information and create debugfs files to display the saved information. In this process, the debugfs files created after each dump.

Therefore, when the dump is triggered while the driver is unbind, the following hang occurs:

```
[67840.853907] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000a0
[67840.862947] Mem abort info:
[67840.865855]  ESR = 0x0000000009600004
[67840.869713]  EC = 0x25: DABT (current EL), IL = 32 bits
[67840.875125]  SET = 0, FnV = 0
[67840.878291]  EA = 0, S1PTW = 0
[67840.881545]  FSC = 0x04: level 0 translation fault
[67840.886528] Data abort info:
[67840.889524]  ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000
[67840.895117]  CM = 0, WnR = 0, TnD = 0, TagAccess = 0
[67840.900284]  GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0
[67840.905709] user pgtable: 4k pages, 48-bit VAs, pgdp=0000002803a1f000
[67840.912263] [00000000000000a0] pgd=0000000000000000, p4d=0000000000000000
[67840.919177] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP
[67840.996435] pstate: 80400009 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--)
[67841.003628] pc : down_write+0x30/0x98
[67841.007546] lr : start_creating.part.0+0x60/0x198
[67841.012495] sp : ffff8000b979ba20
[67841.016046] x29: ffff8000b979ba20 x28: 0000000000000010 x27: 00000000000024b40
[67841.023412] x26: 0000000000000012 x25: ffff20202b355ae8 x24: ffff20202b35a8c8
```



[67841.030779] x23: fffa36877928208 x22: fffa368b4972240 x21: ffff8000b979bb18  
[67841.038147] x20: ffff00281dc1e3c0 x19: ffffffffef x18: 0000000000000020  
[67841.045515] x17: 0000000000000000 x16: fffa368b128a530 x15: ffffffffef  
[67841.052888] x14: ffff8000b979bc18 x13: ffffffffef x12: ffff8000b979bb18  
[67841.060263] x11: 0000000000000000 x10: 0000000000000000 x9 : fffa368b1289b18  
[67841.067640] x8 : 0000000000000012 x7 : 0000000000000000 x6 : 00000000000003a9  
[67841.075014] x5 : 0000000000000000 x4 : ffff002818c5cb00 x3 : 0000000000000001  
[67841.082388] x2 : 0000000000000000 x1 : ffff002818c5cb00 x0 : 00000000000000a0  
[67841.089759] Call trace:  
[67841.092456] down\_write+0x30/0x98  
[67841.096017] start\_creating.part.0+0x60/0x198  
[67841.100613] debugfs\_create\_dir+0x48/0x1f8  
[67841.104950] debugfs\_create\_files\_v3\_hw+0x88/0x348 [hisi\_sas\_v3\_hw]  
[67841.111447] debugfs\_snapshot\_regs\_v3\_hw+0x708/0x798 [hisi\_sas\_v3\_hw]  
[67841.118111] debugfs\_trigger\_dump\_v3\_hw\_write+0x9c/0x120 [hisi\_sas\_v3\_hw]  
[67841.125115] full\_proxy\_write+0x68/0xc8  
[67841.129175] vfs\_write+0xd8/0x3f0  
[67841.132708] ksys\_write+0x70/0x108  
[67841.136317] \_\_arm64\_sys\_write+0x24/0x38  
[67841.140440] invoke\_syscall+0x50/0x128  
[67841.144385] el0\_svc\_common.constprop.0+0xc8/0xf0  
[67841.149273] do\_el0\_svc+0x24/0x38  
[67841.152773] el0\_svc+0x38/0xd8  
[67841.156009] el0t\_64\_sync\_handler+0xc0/0xc8  
[67841.160361] el0t\_64\_sync+0x1a4/0x1a8  
[67841.164189] Code: b9000882 d2800002 d2800023 f9800011 (c85ffc05)  
[67841.170443] ---[ end trace 0000000000000000 ]---

To fix this issue, create all directories and files during debugfs initialization. In this way, the driver only needs to allocate memory space to save information each time the user triggers dumping.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56588>

### **[CVE-2024-56591] kernel: Bluetooth: hci\_conn: Use disable\_delayed\_work\_sync (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: hci\_conn: Use disable\_delayed\_work\_sync

This makes use of disable\_delayed\_work\_sync instead cancel\_delayed\_work\_sync as it not only cancel the ongoing work but also disables new submit which is disarable since the object holding the work is about to be freed.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56591>

### **[CVE-2024-56592] kernel: bpf: Call free\_htab\_elem() after htab\_unlock\_bucket() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Call free\_htab\_elem() after htab\_unlock\_bucket()

For htab of maps, when the map is removed from the htab, it may hold the last reference of the map. bpf\_map\_fd\_put\_ptr() will invoke bpf\_map\_free\_id() to free the id of the removed map element. However, bpf\_map\_fd\_put\_ptr() is invoked while holding a bucket lock (raw\_spin\_lock\_t), and bpf\_map\_free\_id() attempts to acquire map\_idr\_lock (spinlock\_t), triggering the following lockdep warning:

```
=====
[ BUG: Invalid wait context ]
6.11.0-rc4+ #49 Not tainted
-----
test_maps/4881 is trying to lock:
fffffff84884578 (map_idr_lock){+...}-{3:3}, at: bpf_map_free_id.part.0+0x21/0x70
other info that might help us debug this:
context-{5:5}
2 locks held by test_maps/4881:
#0: fffffff846caf60 (rcu_read_lock){....}-{1:3}, at: bpf_fd_htab_map_update_elem+0xf9/0x270
#1: ffff888149ced148 (&htab->lockdep_key#2){....}-{2:2}, at: htab_map_update_elem+0x178/0xa80
stack backtrace:
CPU: 0 UID: 0 PID: 4881 Comm: test_maps Not tainted 6.11.0-rc4+ #49
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), ...
Call Trace:
<TASK>
dump_stack_lvl+0x6e/0xb0
dump_stack+0x10/0x20
__lock_acquire+0x73e/0x36c0
lock_acquire+0x182/0x450
_raw_spin_lock_irqsave+0x43/0x70
bpf_map_free_id.part.0+0x21/0x70
bpf_map_put+0xcf/0x110
bpf_map_fd_put_ptr+0x9a/0xb0
free_htab_elem+0x69/0xe0
htab_map_update_elem+0x50f/0xa80
bpf_fd_htab_map_update_elem+0x131/0x270
htab_map_update_elem+0x50f/0xa80
bpf_fd_htab_map_update_elem+0x131/0x270
bpf_map_update_value+0x266/0x380
__sys_bpf+0x21bb/0x36b0
__x64_sys_bpf+0x45/0x60
x64_sys_call+0x1b2a/0x20d0
do_syscall_64+0x5d/0x100
entry_SYSCALL_64_after_hwframe+0x76/0x7e
```

One way to fix the lockdep warning is using raw\_spinlock\_t for map\_idr\_lock as well. However, bpf\_map\_alloc\_id() invokes idr\_alloc\_cyclic() after acquiring map\_idr\_lock, it will trigger a

similar lockdep warning because the slab's lock (s->cpu\_slab->lock) is still a spinlock.

Instead of changing map\_idr\_lock's type, fix the issue by invoking htab\_put\_fd\_value() after htab\_unlock\_bucket(). However, only deferring the invocation of htab\_put\_fd\_value() is not enough, because the old map pointers in htab of maps can not be saved during batched deletion. Therefore, also defer the invocation of free\_htab\_elem(), so these to-be-freed elements could be linked together similar to lru map.

There are four callers for ->map\_fd\_put\_ptr:

(1) alloc\_htab\_elem() (through htab\_put\_fd\_value())

It invokes ->map\_fd\_put\_ptr() under a raw\_spinlock\_t. The invocation of htab\_put\_fd\_value() can not simply move after htab\_unlock\_bucket(), because the old element has already been stashed in htab->extra\_elems. It may be reused immediately after htab\_unlock\_bucket() and the invocation of htab\_put\_fd\_value() after htab\_unlock\_bucket() may release the newly-added element incorrectly. Therefore, saving the map pointer of the old element for htab of maps before unlocking the bucket and releasing the map\_ptr after unlock. Beside the map pointer in the old element, should do the same thing for the special fields in the old element as well.

(2) free\_htab\_elem() (through htab\_put\_fd\_value())

Its caller includes \_\_htab\_map\_lookup\_and\_delete\_elem(), htab\_map\_delete\_elem() and \_\_htab\_map\_lookup\_and\_delete\_batch().

For htab\_map\_delete\_elem(), simply invoke free\_htab\_elem() after htab\_unlock\_bucket(). For \_\_htab\_map\_lookup\_and\_delete\_batch(), just like lru map, linking the to-be-freed element into node\_to\_free list and invoking free\_htab\_elem() for these element after unlock. It is safe to reuse batch\_flink as the link for node\_to\_free, because these elements have been removed from the hash llist.

Because htab of maps doesn't support lookup\_and\_delete operation, \_\_htab\_map\_lookup\_and\_delete\_elem() doesn't have the problem, so kept it as  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-56592>

**[CVE-2024-56609] kernel: wifi: rtw88: use ieee80211\_purge\_tx\_queue() to purge TX skb (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: rtw88: use ieee80211\_purge\_tx\_queue() to purge TX skb

When removing kernel modules by:

```
rmmod rtw88_8723cs rtw88_8703b rtw88_8723x rtw88_sdio rtw88_core
```

Driver uses `skb_queue_purge()` to purge TX skb, but not report tx status causing "Have pending ack frames!" warning. Use `ieee80211_purge_tx_queue()` to correct this.

Since `ieee80211_purge_tx_queue()` doesn't take locks, to prevent racing between TX work and purge TX queue, flush and destroy TX work in advance.

```
wlan0: deauthenticating from aa:f5:fd:60:4c:a8 by local  
choice (Reason: 3=DEAUTH_LEAVING)
```

```
-----[ cut here ]-----
```

Have pending ack frames!

WARNING: CPU: 3 PID: 9232 at net/mac80211/main.c:1691

```
ieee80211_free_ack_frame+0x5c/0x90 [mac80211]
```

CPU: 3 PID: 9232 Comm: rmmod Tainted: G C

6.10.1-200.fc40.aarch64 #1

Hardware name: pine64 Pine64 PinePhone Braveheart

(1.1)/Pine64 PinePhone Braveheart (1.1), BIOS 2024.01 01/01/2024

pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--)

```
pc : ieee80211_free_ack_frame+0x5c/0x90 [mac80211]
```

```
lr : ieee80211_free_ack_frame+0x5c/0x90 [mac80211]
```

```
sp : ffff80008c1b37b0
```

```
x29: ffff80008c1b37b0 x28: ffff000003be8000 x27: 0000000000000000
```

```
x26: 0000000000000000 x25: ffff000003dc14b8 x24: ffff80008c1b37d0
```

```
x23: ffff000000ff9f80 x22: 0000000000000000 x21: 000000007ffffff
```

```
x20: ffff80007c7e93d8 x19: ffff00006e66f400 x18: 0000000000000000
```

```
x17: ffff7fffd2b3000 x16: ffff800083fc0000 x15: 0000000000000000
```

```
x14: 0000000000000000 x13: 2173656d61726620 x12: 6b636120676e6964
```

```
x11: 0000000000000000 x10: 000000000000005d x9 : ffff8000802af2b0
```

```
x8 : ffff80008c1b3430 x7 : 0000000000000001 x6 : 0000000000000001
```

```
x5 : 0000000000000000 x4 : 0000000000000000 x3 : 0000000000000000
```

```
x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff000003be8000
```

Call trace:

```
ieee80211_free_ack_frame+0x5c/0x90 [mac80211]
```

```
idr_for_each+0x74/0x110
```

```
ieee80211_free_hw+0x44/0xe8 [mac80211]
```

```
rtw_sdio_remove+0x9c/0xc0 [rtw88_sdio]
```

```
sdio_bus_remove+0x44/0x180
```

```
device_remove+0x54/0x90
```

```
device_release_driver_internal+0x1d4/0x238
```

```
driver_detach+0x54/0xc0
```

```
bus_remove_driver+0x78/0x108
```

```
driver_unregister+0x38/0x78
```

```
sdio_unregister_driver+0x2c/0x40
```

```
rtw_8723cs_driver_exit+0x18/0x1000 [rtw88_8723cs]
```

```
__do_sys_delete_module.isra.0+0x190/0x338
```

```
__arm64_sys_delete_module+0x1c/0x30
```

```
invoke_syscall+0x74/0x100
```

```
el0_svc_common.constprop.0+0x48/0xf0
```

```
do_el0_svc+0x24/0x38
```

```
el0_svc+0x3c/0x158
```

```
el0t_64_sync_handler+0x120/0x138
el0t_64_sync+0x194/0x198
---[ end trace 0000000000000000 ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-56609>

## **[CVE-2024-56611] kernel: mm/mempolicy: fix migrate\_to\_node() assuming there is at least one VMA in a MM (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/mempolicy: fix migrate\_to\_node() assuming there is at least one VMA in a MM

We currently assume that there is at least one VMA in a MM, which isn't true.

So we might end up having find\_vma() return NULL, to then de-reference NULL. So properly handle find\_vma() returning NULL.

This fixes the report:

Oops: general protection fault, probably for non-canonical address 0xdfffc00000000000: 0000 [#1] PREEMPT SMP KASAN PTI

KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007]

CPU: 1 UID: 0 PID: 6021 Comm: syz-executor284 Not tainted 6.12.0-rc7-syzkaller-00187-gf868cd251776 #0

Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/30/2024

RIP: 0010:migrate\_to\_node mm/mempolicy.c:1090 [inline]

RIP: 0010:do\_migrate\_pages+0x403/0x6f0 mm/mempolicy.c:1194

Code: ...

RSP: 0018:ffff9000375fd08 EFLAGS: 00010246

RAX: 0000000000000000 RBX: ffff9000375fd78 RCX: 0000000000000000

RDX: ffff88807e171300 RSI: dfffc00000000000 RDI: ffff88803390c044

RBP: ffff88807e171428 R08: 0000000000000014 R09: fffffbfff2039ef1

R10: fffffff901cf78f R11: 0000000000000000 R12: 0000000000000003

R13: ffff9000375fe90 R14: ffff9000375fe98 R15: ffff9000375fdf8

FS: 00005555919e1380(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 00005555919e1ca8 CR3: 000000007f12a000 CR4: 00000000003526f0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400

Call Trace:

<TASK>

kernel\_migrate\_pages+0x5b2/0x750 mm/mempolicy.c:1709

\_\_do\_sys\_migrate\_pages mm/mempolicy.c:1727 [inline]

\_\_se\_sys\_migrate\_pages mm/mempolicy.c:1723 [inline]

\_\_x64\_sys\_migrate\_pages+0x96/0x100 mm/mempolicy.c:1723

do\_syscall\_x64 arch/x86/entry/common.c:52 [inline]

do\_syscall\_64+0xcd/0x250 arch/x86/entry/common.c:83

entry\_SYSCALL\_64\_after\_hwframe+0x77/0x7f

[akpm@linux-foundation.org: add unlikely()]

More Info: <https://avd.aquasec.com/nvd/cve-2024-56611>

**[CVE-2024-56647] kernel: net: Fix icmp host relookup triggering ip\_rt\_bug (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: Fix icmp host relookup triggering ip\_rt\_bug

arp link failure may trigger ip\_rt\_bug while xfrm enabled, call trace is:

WARNING: CPU: 0 PID: 0 at net/ipv4/route.c:1241 ip\_rt\_bug+0x14/0x20

Modules linked in:

CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.0-rc6-00077-g2e1b3cc9d7f7

Hardware name: QEMU Standard PC (i440FX + PIIX, 1996),

BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014

RIP: 0010:ip\_rt\_bug+0x14/0x20

Call Trace:

<IRQ>

ip\_send\_skb+0x14/0x40

\_\_icmp\_send+0x42d/0x6a0

ipv4\_link\_failure+0xe2/0x1d0

arp\_error\_report+0x3c/0x50

neigh\_invalidate+0x8d/0x100

neigh\_timer\_handler+0x2e1/0x330

call\_timer\_fn+0x21/0x120

\_\_run\_timer\_base.part.0+0x1c9/0x270

run\_timer\_softirq+0x4c/0x80

handle\_softirqs+0xac/0x280

irq\_exit\_rcu+0x62/0x80

sysvec\_apic\_timer\_interrupt+0x77/0x90

The script below reproduces this scenario:

```
ip xfrm policy add src 0.0.0.0/0 dst 0.0.0.0/0 \
```

```
dir out priority 0 ptype main flag localok icmp
```

```
ip l a veth1 type veth
```

```
ip a a 192.168.141.111/24 dev veth0
```

```
ip l s veth0 up
```

```
ping 192.168.141.155 -c 1
```

icmp\_route\_lookup() create input routes for locally generated packets while xfrm relookup ICMP traffic. Then it will set input route (dst->out = ip\_rt\_bug) to skb for DESTUNREACH.

For ICMP err triggered by locally generated packets, dst->dev of output route is loopback. Generally, xfrm relookup verification is not required on loopback interfaces (net.ipv4.conf.lo.disable\_xfrm = 1).

Skip icmp relookup for locally generated packets to fix it.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56647>

**[CVE-2024-56657] kernel: ALSA: control: Avoid WARN() for symlink errors (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ALSA: control: Avoid WARN() for symlink errors

Using WARN() for showing the error of symlink creations don't give more information than telling that something goes wrong, since the usual code path is a lregister callback from each control element creation. More badly, the use of WARN() rather confuses fuzzer as if it were serious issues.

This patch downgrades the warning messages to use the normal dev\_err() instead of WARN(). For making it clearer, add the function name to the prefix, too.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56657>

**[CVE-2024-56692] kernel: f2fs: fix to do sanity check on node blkaddr in truncate\_node() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

f2fs: fix to do sanity check on node blkaddr in truncate\_node()

syzbot reports a f2fs bug as below:

-----[ cut here ]-----

kernel BUG at fs/f2fs/segment.c:2534!

RIP: 0010:f2fs\_invalidate\_blocks+0x35f/0x370 fs/f2fs/segment.c:2534

Call Trace:

truncate\_node+0x1ae/0x8c0 fs/f2fs/node.c:909

f2fs\_remove\_inode\_page+0x5c2/0x870 fs/f2fs/node.c:1288

f2fs\_evict\_inode+0x879/0x15c0 fs/f2fs/inode.c:856

evict+0x4e8/0x9b0 fs/inode.c:723

f2fs\_handle\_failed\_inode+0x271/0x2e0 fs/f2fs/inode.c:986

f2fs\_create+0x357/0x530 fs/f2fs/namei.c:394

lookup\_open fs/namei.c:3595 [inline]

open\_last\_lookups fs/namei.c:3694 [inline]

path\_openat+0x1c03/0x3590 fs/namei.c:3930

do\_filp\_open+0x235/0x490 fs/namei.c:3960

do\_sys\_openat2+0x13e/0x1d0 fs/open.c:1415

```
do_sys_open fs/open.c:1430 [inline]
__do_sys_openat fs/open.c:1446 [inline]
__se_sys_openat fs/open.c:1441 [inline]
__x64_sys_openat+0x247/0x2a0 fs/open.c:1441
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: 0010:f2fs_invalidate_blocks+0x35f/0x370 fs/f2fs/segment.c:2534
```

The root cause is: on a fuzzed image, blkaddr in nat entry may be corrupted, then it will cause system panic when using it in f2fs\_invalidate\_blocks(), to avoid this, let's add sanity check on nat blkaddr in truncate\_node().

More Info: <https://avd.aquasec.com/nvd/cve-2024-56692>

### **[CVE-2024-56712] kernel: udmabuf: fix memory leak on last export\_udmabuf() error path (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

udmabuf: fix memory leak on last export\_udmabuf() error path

In export\_udmabuf(), if dma\_buf\_fd() fails because the FD table is full, a dma\_buf owning the udmabuf has already been created; but the error handling in udmabuf\_create() will tear down the udmabuf without doing anything about the containing dma\_buf.

This leaves a dma\_buf in memory that contains a dangling pointer; though that doesn't seem to lead to anything bad except a memory leak.

Fix it by moving the dma\_buf\_fd() call out of export\_udmabuf() so that we can give it different error handling.

Note that the shape of this code changed a lot in commit 5e72b2b41a21 ("udmabuf: convert udmabuf driver to use folios"); but the memory leak seems to have existed since the introduction of udmabuf.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56712>

### **[CVE-2024-56719] kernel: net: stmmac: fix TSO DMA API usage causing oops (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: stmmac: fix TSO DMA API usage causing oops



Commit 66600fac7a98 ("net: stmmac: TSO: Fix unbalanced DMA map/unmap for non-paged SKB data") moved the assignment of tx\_skbuff\_dma[]'s members to be later in stmmac\_tso\_xmit().

The buf (dma cookie) and len stored in this structure are passed to dma\_unmap\_single() by stmmac\_tx\_clean(). The DMA API requires that the dma cookie passed to dma\_unmap\_single() is the same as the value returned from dma\_map\_single(). However, by moving the assignment later, this is not the case when priv->dma\_cap.addr64 > 32 as "des" is offset by proto\_hdr\_len.

This causes problems such as:

dwc-eth-dwmac 2490000.ethernet eth0: Tx DMA map failed

and with DMA\_API\_DEBUG enabled:

DMA-API: dwc-eth-dwmac 2490000.ethernet: device driver tries to +free DMA memory it has not allocated [device address=0x000000ffffcf65c0] [size=66 bytes]

Fix this by maintaining "des" as the original DMA cookie, and use tso\_des to pass the offset DMA cookie to stmmac\_tso\_allocator().

Full details of the crashes can be found at:

<https://lore.kernel.org/all/d8112193-0386-4e14-b516-37c2d838171a@nvidia.com/>

<https://lore.kernel.org/all/klkzp5yn5kq5efgtrow6wbvnc46bcqfxs65nz3qy77ujr5turc@bwwhelz2l4dw/>

More Info: <https://avd.aquasec.com/nvd/cve-2024-56719>

### **[CVE-2024-56729] kernel: smb: Initialize cfid->tcon before performing network ops (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: Initialize cfid->tcon before performing network ops

Avoid leaking a tcon ref when a lease break races with opening the cached directory. Processing the leak break might take a reference to the tcon in cached\_dir\_lease\_break() and then fail to release the ref in cached\_dir\_offload\_close, since cfid->tcon is still NULL.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56729>

### **[CVE-2024-56742] kernel: vfio/mlx5: Fix an unwind issue in mlx5vf\_add\_migration\_pages() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

vfio/mlx5: Fix an unwind issue in mlx5vf\_add\_migration\_pages()

Fix an unwind issue in mlx5vf\_add\_migration\_pages().

If a set of pages is allocated but fails to be added to the SG table, they need to be freed to prevent a memory leak.

Any pages successfully added to the SG table will be freed as part of mlx5vf\_free\_data\_buffer().

More Info: <https://avd.aquasec.com/nvd/cve-2024-56742>

**[CVE-2024-56757] kernel: Bluetooth: btusb: mediatek: add intf release flow when usb disconnect (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: btusb: mediatek: add intf release flow when usb disconnect

MediaTek claim an special usb intr interface for ISO data transmission. The interface need to be released before unregistering hci device when usb disconnect. Removing BT usb dongle without properly releasing the interface may cause Kernel panic while unregister hci device.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56757>

**[CVE-2024-56758] kernel: btrfs: check folio mapping after unlock in relocate\_one\_folio() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: check folio mapping after unlock in relocate\_one\_folio()

When we call btrfs\_read\_folio() to bring a folio uptodate, we unlock the folio. The result of that is that a different thread can modify the mapping (like remove it with invalidate) before we call folio\_lock(). This results in an invalid page and we need to try again.

In particular, if we are relocating concurrently with aborting a transaction, this can result in a crash like the following:

```
BUG: kernel NULL pointer dereference, address: 0000000000000000
PGD 0 P4D 0
Oops: 0000 [#1] SMP
```

CPU: 76 PID: 1411631 Comm: kworker/u322:5  
Workqueue: events\_unbound btrfs\_reclaim\_bgs\_work  
RIP: 0010:set\_page\_extent\_mapped+0x20/0xb0  
RSP: 0018:ffffc900516a7be8 EFLAGS: 00010246  
RAX: ffffea009e851d08 RBX: ffffea009e0b1880 RCX: 0000000000000000  
RDX: 0000000000000000 RSI: ffffc900516a7b90 RDI: ffffea009e0b1880  
RBP: 0000000003573000 R08: 0000000000000001 R09: ffff88c07fd2f3f0  
R10: 0000000000000000 R11: 0000194754b575be R12: 0000000003572000  
R13: 0000000003572fff R14: 0000000000100cca R15: 0000000005582fff  
FS: 0000000000000000(0000) GS:ffff88c07fd00000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 0000000000000000 CR3: 000000407d00f002 CR4: 00000000007706f0  
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000  
DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040  
PKRU: 55555554  
Call Trace:  
<TASK>  
? \_\_die+0x78/0xc0  
? page\_fault\_oops+0x2a8/0x3a0  
? \_\_switch\_to+0x133/0x530  
? wq\_worker\_running+0xa/0x40  
? exc\_page\_fault+0x63/0x130  
? asm\_exc\_page\_fault+0x22/0x30  
? set\_page\_extent\_mapped+0x20/0xb0  
relocate\_file\_extent\_cluster+0x1a7/0x940  
relocate\_data\_extent+0xaf/0x120  
relocate\_block\_group+0x20f/0x480  
btrfs\_relocate\_block\_group+0x152/0x320  
btrfs\_relocate\_chunk+0x3d/0x120  
btrfs\_reclaim\_bgs\_work+0x2ae/0x4e0  
process\_scheduled\_works+0x184/0x370  
worker\_thread+0xc6/0x3e0  
? blk\_add\_timer+0xb0/0xb0  
kthread+0xae/0xe0  
? flush\_tlb\_kernel\_range+0x90/0x90  
ret\_from\_fork+0x2f/0x40  
? flush\_tlb\_kernel\_range+0x90/0x90  
ret\_from\_fork\_asm+0x11/0x20  
</TASK>

This occurs because `cleanup_one_transaction()` calls `destroy_delalloc_inodes()` which calls `invalidate_inode_pages2()` which takes the `folio_lock` before setting mapping to NULL. We fail to check this, and subsequently call `set_extent_mapping()`, which assumes that `mapping != NULL` (in fact it asserts that in debug mode)

Note that the "fixes" patch here is not the one that introduced the race (the very first iteration of this code from 2009) but a more recent change that made this particular crash happen in practice.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56758>

**[CVE-2024-56782] kernel: ACPI: x86: Add adev NULL check to acpi\_quirk\_skip\_serdev\_enumeration()**

**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ACPI: x86: Add adev NULL check to acpi\_quirk\_skip\_serdev\_enumeration()

acpi\_dev\_hid\_match() does not check for adev == NULL, dereferencing it unconditional.

Add a check for adev being NULL before calling acpi\_dev\_hid\_match().

At the moment acpi\_quirk\_skip\_serdev\_enumeration() is never called with a controller\_parent without an ACPI companion, but better safe than sorry.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56782>

**[CVE-2024-56786] kernel: bpf: put bpf\_link's program when link is safe to be deallocated (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: put bpf\_link's program when link is safe to be deallocated

In general, BPF link's underlying BPF program should be considered to be reachable through attach hook -> link -> prog chain, and, pessimistically, we have to assume that as long as link's memory is not safe to free, attach hook's code might hold a pointer to BPF program and use it.

As such, it's not (generally) correct to put link's program early before waiting for RCU GPs to go through. More eager bpf\_prog\_put() that we currently do is mostly correct due to BPF program's release code doing similar RCU GP waiting, but as will be shown in the following patches, BPF program can be non-sleepable (and, thus, reliant on only "classic" RCU GP), while BPF link's attach hook can have sleepable semantics and needs to be protected by RCU Tasks Trace, and for such cases BPF link has to go through RCU Tasks Trace + "classic" RCU GPs before being deallocated. And so, if we put BPF program early, we might free BPF program before we free BPF link, leading to use-after-free situation.

So, this patch defers bpf\_prog\_put() until we are ready to perform bpf\_link's deallocation. At worst, this delays BPF program freeing by one extra RCU GP, but that seems completely acceptable. Alternatively, we'd need more elaborate ways to determine BPF hook, BPF link, and BPF program lifetimes, and how they relate to each other, which seems like an unnecessary complication.

Note, for most BPF links we still will perform eager bpf\_prog\_put() and link dealloc, so for those BPF links there are no observable changes whatsoever. Only BPF links that use deferred dealloc might notice slightly delayed freeing of BPF programs.

Also, to reduce code and logic duplication, extract program put + link dealloc logic into bpf\_link\_dealloc() helper.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56786>

### **[CVE-2024-57795] kernel: RDMA/rxe: Remove the direct link to net\_device (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/rxe: Remove the direct link to net\_device

The similar patch in siw is in the link:

<https://git.kernel.org/rdma/rdma/c/16b87037b48889>

This problem also occurred in RXE. The following analyze this problem.

In the following Call Traces:

"

BUG: KASAN: slab-use-after-free in dev\_get\_flags+0x188/0x1d0 net/core/dev.c:8782

Read of size 4 at addr ffff8880554640b0 by task kworker/1:4/5295

CPU: 1 UID: 0 PID: 5295 Comm: kworker/1:4 Not tainted

6.12.0-rc3-syzkaller-00399-g9197b73fd7bb #0

Hardware name: Google Compute Engine/Google Compute Engine,

BIOS Google 09/13/2024

Workqueue: infiniband ib\_cache\_event\_task

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

dev\_get\_flags+0x188/0x1d0 net/core/dev.c:8782

rxe\_query\_port+0x12d/0x260 drivers/infiniband/sw/rxe/rxe\_verbs.c:60

\_\_ib\_query\_port drivers/infiniband/core/device.c:2111 [inline]

ib\_query\_port+0x168/0x7d0 drivers/infiniband/core/device.c:2143

ib\_cache\_update+0x1a9/0xb80 drivers/infiniband/core/cache.c:1494

ib\_cache\_event\_task+0xf3/0x1e0 drivers/infiniband/core/cache.c:1568

process\_one\_work kernel/workqueue.c:3229 [inline]

process\_scheduled\_works+0xa65/0x1850 kernel/workqueue.c:3310

worker\_thread+0x870/0xd30 kernel/workqueue.c:3391

kthread+0x2f2/0x390 kernel/kthread.c:389

ret\_from\_fork+0x4d/0x80 arch/x86/kernel/process.c:147

ret\_from\_fork\_asm+0x1a/0x30 arch/x86/entry/entry\_64.S:244

</TASK>

"

1). In the link [1],

"

infiniband syz2: set down

"

This means that on 839.350575, the event `ib_cache_event_task` was sent and queued in `ib_wq`.

2). In the link [1],

"

team0 (unregistering): Port device `team_slave_0` removed

"

It indicates that before 843.251853, the net device should be freed.

3). In the link [1],

"

BUG: KASAN: slab-use-after-free in `dev_get_flags+0x188/0x1d0`

"

This means that on 850.559070, this slab-use-after-free problem occurred.

In all, on 839.350575, the event `ib_cache_event_task` was sent and queued in `ib_wq`,

before 843.251853, the net device `veth` was freed.

on 850.559070, this event was executed, and the mentioned freed net device was called. Thus, the above call trace occurred.

[1] <https://syzkaller.appspot.com/x/log.txt?x=12e7025f980000>

More Info: <https://avd.aquasec.com/nvd/cve-2024-57795>

### **[CVE-2024-57804] kernel: scsi: mpi3mr: Fix corrupt config pages PHY state is switched in sysfs (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: `%ls(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

scsi: mpi3mr: Fix corrupt config pages PHY state is switched in sysfs

The driver, through the SAS transport, exposes a `sysfs` interface to enable/disable PHYs in a controller/expander setup. When multiple PHYs are disabled and enabled in rapid succession, the persistent and current

config pages related to SAS IO unit/SAS Expander pages could get corrupted.

Use separate memory for each config request.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57804>

### **[CVE-2024-57809] kernel: PCI: imx6: Fix suspend/resume support on i.MX6QDL (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

PCI: imx6: Fix suspend/resume support on i.MX6QDL

The suspend/resume functionality is currently broken on the i.MX6QDL platform, as documented in the NXP errata (ERR005723):

<https://www.nxp.com/docs/en/errata/IMX6DQCE.pdf>

This patch addresses the issue by sharing most of the suspend/resume sequences used by other i.MX devices, while avoiding modifications to critical registers that disrupt the PCIe functionality. It targets the same problem as the following downstream commit:

<https://github.com/nxp-imx/linux-imx/commit/4e92355e1f79d225ea842511fcd42b343b32995>

Unlike the downstream commit, this patch also resets the connected PCIe device if possible. Without this reset, certain drivers, such as ath10k or iwlwifi, will crash on resume. The device reset is also done by the driver on other i.MX platforms, making this patch consistent with existing practices.

Upon resuming, the kernel will hang and display an error. Here's an example of the error encountered with the ath10k driver:

ath10k\_pci 0000:01:00.0: Unable to change power state from D3hot to D0, device inaccessible  
Unhandled fault: imprecise external abort (0x1406) at 0x0106f944

Without this patch, suspend/resume will fail on i.MX6QDL devices if a PCIe device is connected.

[kwilczynski: commit log, added tag for stable releases]

More Info: <https://avd.aquasec.com/nvd/cve-2024-57809>

### **[CVE-2024-57843] kernel: virtio-net: fix overflow inside virtnet\_rq\_alloc (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

virtio-net: fix overflow inside virtnet\_rq\_alloc

When the frag just got a page, then may lead to regression on VM. Specially if the sysctl net.core.high\_order\_alloc\_disable value is 1, then the frag always get a page when do refill.

Which could see reliable crashes or scp failure (scp a file 100M in size to VM).

The issue is that the virtnet\_rq\_dma takes up 16 bytes at the beginning of a new frag. When the frag size is larger than PAGE\_SIZE, everything is fine. However, if the frag is only one page and the total size of the buffer and virtnet\_rq\_dma is larger than one page, an overflow may occur.

The commit f9dac92ba908 ("virtio\_ring: enable premapped mode whatever use\_dma\_api") introduced this problem. And we reverted some commits to fix this in last linux version. Now we try to enable it and fix this bug directly.

Here, when the frag size is not enough, we reduce the buffer len to fix this problem.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57843>

**[CVE-2024-57852] kernel: firmware: qcom: scm: smc: Handle missing SCM device (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

firmware: qcom: scm: smc: Handle missing SCM device

Commit ca61d6836e6f ("firmware: qcom: scm: fix a NULL-pointer dereference") makes it explicit that qcom\_scm\_get\_tzmem\_pool() can return NULL, therefore its users should handle this.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57852>

**[CVE-2024-57857] kernel: RDMA/siw: Remove direct link to net\_device (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/siw: Remove direct link to net\_device

Do not manage a per device direct link to net\_device. Rely



on associated `ib_devices` `net_device` management, not doubling the effort locally. A badly managed local link to `net_device` was causing a 'KASAN: slab-use-after-free' exception during `siw_query_port()` call.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57857>

**[CVE-2024-57872] kernel: scsi: ufs: pltfrm: Dellocate HBA during ufshcd\_pltfrm\_remove() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: ufs: pltfrm: Dellocate HBA during ufshcd\_pltfrm\_remove()

This will ensure that the scsi host is cleaned up properly using `scsi_host_dev_release()`. Otherwise, it may lead to memory leaks.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57872>

**[CVE-2024-57875] kernel: block: RCU protect disk->conv\_zones\_bitmap (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

block: RCU protect disk->conv\_zones\_bitmap

Ensure that a disk revalidation changing the conventional zones bitmap of a disk does not cause invalid memory references when using the `disk_zone_is_conv()` helper by RCU protecting the `disk->conv_zones_bitmap` pointer.

`disk_zone_is_conv()` is modified to operate under the RCU read lock and the function `disk_set_conv_zones_bitmap()` is added to update a disk `conv_zones_bitmap` pointer using `rcu_replace_pointer()` with the disk `zone_wplugs_lock` spinlock held.

`disk_free_zone_resources()` is modified to call `disk_update_zone_resources()` with a NULL bitmap pointer to free the disk `conv_zones_bitmap`. `disk_set_conv_zones_bitmap()` is also used in `disk_update_zone_resources()` to set the new (revalidated) bitmap and free the old one.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57875>

**[CVE-2024-57883] kernel: mm: hugetlb: independent PMD page table shared count (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm: hugetlb: independent PMD page table shared count

The folio refcount may be increased unexpectedly through try\_get\_folio() by caller such as split\_huge\_pages. In huge\_pmd\_unshare(), we use refcount to check whether a pmd page table is shared. The check is incorrect if the refcount is increased by the above caller, and this can cause the page table leaked:

```
BUG: Bad page state in process sh pfn:109324
page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x66 pfn:0x109324
flags: 0x17ffff8000000000(node=0|zone=2|lastcpupid=0xfffff)
page_type: f2(table)
raw: 017ffff800000000 0000000000000000 0000000000000000 0000000000000000
raw: 0000000000000066 0000000000000000 00000000f2000000 0000000000000000
page dumped because: nonzero mapcount
...
CPU: 31 UID: 0 PID: 7515 Comm: sh Kdump: loaded Tainted: G B 6.13.0-rc2master+ #7
Tainted: [B]=BAD_PAGE
Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015
Call trace:
show_stack+0x20/0x38 (C)
dump_stack_lvl+0x80/0xf8
dump_stack+0x18/0x28
bad_page+0x8c/0x130
free_page_is_bad_report+0xa4/0xb0
free_unref_page+0x3cc/0x620
__folio_put+0xf4/0x158
split_huge_pages_all+0x1e0/0x3e8
split_huge_pages_write+0x25c/0x2d8
full_proxy_write+0x64/0xd8
vfs_write+0xcc/0x280
ksys_write+0x70/0x110
__arm64_sys_write+0x24/0x38
invoke_syscall+0x50/0x120
el0_svc_common.constprop.0+0xc8/0xf0
do_el0_svc+0x24/0x38
el0_svc+0x34/0x128
el0t_64_sync_handler+0xc8/0xd0
el0t_64_sync+0x190/0x198
```

The issue may be triggered by damon, offline\_page, page\_idle, etc, which will increase the refcount of page table.

1. The page table itself will be discarded after reporting the "nonzero mapcount".
2. The HugeTLB page mapped by the page table miss freeing since we treat the page table as shared and a shared page table will not be

unmapped.

Fix it by introducing independent PMD page table shared count. As described by comment, `pt_index/pt_mm/pt_frag_refcount` are used for s390 gmap, x86 pgds and powerpc, `pt_share_count` is used for x86/arm64/riscv pmds, so we can reuse the field as `pt_share_count`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57883>

**[CVE-2024-57888] kernel: workqueue: Do not warn when cancelling WQ\_MEM\_RECLAIM work from !WQ\_MEM\_RECLAIM worker (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

workqueue: Do not warn when cancelling WQ\_MEM\_RECLAIM work from !WQ\_MEM\_RECLAIM worker

After commit

746ae46c1113 ("drm/sched: Mark scheduler work queues with WQ\_MEM\_RECLAIM")

amdgpu started seeing the following warning:

```
[ ] workqueue: WQ_MEM_RECLAIM sdma0:drm_sched_run_job_work [gpu_sched] is flushing !WQ_MEM_RECLAIM
events:amdgpu_device_delay_enable_gfx_off [amdgpu]
```

...

```
[ ] Workqueue: sdma0 drm_sched_run_job_work [gpu_sched]
```

...

```
[ ] Call Trace:
```

```
[ ] <TASK>
```

...

```
[ ] ? check_flush_dependency+0xf5/0x110
```

...

```
[ ] cancel_delayed_work_sync+0x6e/0x80
```

```
[ ] amdgpu_gfx_off_ctrl+0xab/0x140 [amdgpu]
```

```
[ ] amdgpu_ring_alloc+0x40/0x50 [amdgpu]
```

```
[ ] amdgpu_ib_schedule+0xf4/0x810 [amdgpu]
```

```
[ ] ? drm_sched_run_job_work+0x22c/0x430 [gpu_sched]
```

```
[ ] amdgpu_job_run+0xaa/0x1f0 [amdgpu]
```

```
[ ] drm_sched_run_job_work+0x257/0x430 [gpu_sched]
```

```
[ ] process_one_work+0x217/0x720
```

...

```
[ ] </TASK>
```

The intent of the verification done in `check_flush_dependency` is to ensure forward progress during memory reclaim, by flagging cases when either a memory reclaim process, or a memory reclaim work item is flushed from a context not marked as memory reclaim safe.

This is correct when flushing, but when called from the `cancel(_delayed)_work_sync()` paths it is a false positive because work is either already running, or will not be running at all. Therefore

cancelling it is safe and we can relax the warning criteria by letting the helper know of the calling context.

References: 746ae46c1113 ("drm/sched: Mark scheduler work queues with WQ\_MEM\_RECLAIM")

More Info: <https://avd.aquasec.com/nvd/cve-2024-57888>

### **[CVE-2024-57895] kernel: ksmbd: set ATTR\_CTIME flags when setting mtime (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: set ATTR\_CTIME flags when setting mtime

David reported that the new warning from setattr\_copy\_mgtime is coming like the following.

```
[ 113.215316] -----[ cut here ]-----
[ 113.215974] WARNING: CPU: 1 PID: 31 at fs/attr.c:300 setattr_copy+0x1ee/0x200
[ 113.219192] CPU: 1 UID: 0 PID: 31 Comm: kworker/1:1 Not tainted 6.13.0-rc1+ #234
[ 113.220127] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
rel-1.16.2-3-gd478f380-rebuilt.opensuse.org 04/01/2014
[ 113.221530] Workqueue: ksmbd-io handle_ksmbd_work [ksmbd]
[ 113.222220] RIP: 0010:setattr_copy+0x1ee/0x200
[ 113.222833] Code: 24 28 49 8b 44 24 30 48 89 53 58 89 43 6c 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc 48 89 df
e8 77 d6 ff ff e9 cd fe ff ff <0f> 0b e9 be fe ff ff 66 0
[ 113.225110] RSP: 0018:ffffaf218010fb68 EFLAGS: 00010202
[ 113.225765] RAX: 00000000000000120 RBX: ffffa446815f8568 RCX: 0000000000000003
[ 113.226667] RDX: ffffaf218010fd38 RSI: ffffa446815f8568 RDI: ffffffff94eb03a0
[ 113.227531] RBP: ffffaf218010fb90 R08: 00000001a251e217d R09: 00000000675259fa
[ 113.228426] R10: 0000000002ba8a6d R11: ffffa4468196c7a8 R12: ffffaf218010fd38
[ 113.229304] R13: 00000000000000120 R14: ffffffff94eb03a0 R15: 0000000000000000
[ 113.230210] FS: 0000000000000000(0000) GS:ffffa44739d00000(0000) knlGS:0000000000000000
[ 113.231215] CS: 0010 DS: 0000 ES: 0000 CRO: 0000000080050033
[ 113.232055] CR2: 00007efe0053d27e CR3: 000000000331a000 CR4: 000000000000006b0
[ 113.232926] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[ 113.233812] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400
[ 113.234797] Call Trace:
[ 113.235116] <TASK>
[ 113.235393] ? __warn+0x73/0xd0
[ 113.235802] ? setattr_copy+0x1ee/0x200
[ 113.236299] ? report_bug+0xf3/0x1e0
[ 113.236757] ? handle_bug+0x4d/0x90
[ 113.237202] ? exc_invalid_op+0x13/0x60
[ 113.237689] ? asm_exc_invalid_op+0x16/0x20
[ 113.238185] ? setattr_copy+0x1ee/0x200
[ 113.238692] btrfs_setattr+0x80/0x820 [btrfs]
[ 113.239285] ? get_stack_info_noinstr+0x12/0xf0
[ 113.239857] ? __module_address+0x22/0xa0
[ 113.240368] ? handle_ksmbd_work+0x6e/0x460 [ksmbd]
[ 113.240993] ? __module_text_address+0x9/0x50
```

```
[ 113.241545] ? __module_address+0x22/0xa0
[ 113.242033] ? unwind_next_frame+0x10e/0x920
[ 113.242600] ? __pfx_stack_trace_consume_entry+0x10/0x10
[ 113.243268] notify_change+0x2c2/0x4e0
[ 113.243746] ? stack_depot_save_flags+0x27/0x730
[ 113.244339] ? set_file_basic_info+0x130/0x2b0 [ksmbd]
[ 113.244993] set_file_basic_info+0x130/0x2b0 [ksmbd]
[ 113.245613] ? process_scheduled_works+0xbe/0x310
[ 113.246181] ? worker_thread+0x100/0x240
[ 113.246696] ? kthread+0xc8/0x100
[ 113.247126] ? ret_from_fork+0x2b/0x40
[ 113.247606] ? ret_from_fork_asm+0x1a/0x30
[ 113.248132] smb2_set_info+0x63f/0xa70 [ksmbd]
```

ksmbd is trying to set the atime and mtime via notify\_change without also setting the ctime. so This patch add ATTR\_CTIME flags when setting mtime to avoid a warning.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57895>

### **[CVE-2024-57898] kernel: wifi: cfg80211: clear link ID from bitmap during link delete after clean up (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: cfg80211: clear link ID from bitmap during link delete after clean up

Currently, during link deletion, the link ID is first removed from the valid\_links bitmap before performing any clean-up operations. However, some functions require the link ID to remain in the valid\_links bitmap. One such example is cfg80211\_cac\_event(). The flow is -

```
nl80211_remove_link()
  cfg80211_remove_link()
    ieee80211_del_intf_link()
      ieee80211_vif_set_links()
        ieee80211_vif_update_links()
          ieee80211_link_stop()
            cfg80211_cac_event()
```

cfg80211\_cac\_event() requires link ID to be present but it is cleared already in cfg80211\_remove\_link(). Ultimately, WARN\_ON() is hit.

Therefore, clear the link ID from the bitmap only after completing the link clean-up.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57898>

### **[CVE-2024-57899] kernel: wifi: mac80211: fix mbss changed flags corruption on 32 bit systems**

**(Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: mac80211: fix mbss changed flags corruption on 32 bit systems

On 32-bit systems, the size of an unsigned long is 4 bytes, while a u64 is 8 bytes. Therefore, when using `or_each_set_bit(bit, &bits, sizeof(changed) * BITS_PER_BYTE)`, the code is incorrectly searching for a bit in a 32-bit variable that is expected to be 64 bits in size, leading to incorrect bit finding.

Solution: Ensure that the size of the bits variable is correctly adjusted for each architecture.

Call Trace:

```
? show_regs+0x54/0x58
? __warn+0x6b/0xd4
? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211]
? report_bug+0x113/0x150
? exc_overflow+0x30/0x30
? handle_bug+0x27/0x44
? exc_invalid_op+0x18/0x50
? handle_exception+0xf6/0xf6
? exc_overflow+0x30/0x30
? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211]
? exc_overflow+0x30/0x30
? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211]
? ieee80211_mesh_work+0xff/0x260 [mac80211]
? cfg80211_wiphy_work+0x72/0x98 [cfg80211]
? process_one_work+0xf1/0x1fc
? worker_thread+0x2c0/0x3b4
? kthread+0xc7/0xf0
? mod_delayed_work_on+0x4c/0x4c
? kthread_complete_and_exit+0x14/0x14
? ret_from_fork+0x24/0x38
? kthread_complete_and_exit+0x14/0x14
? ret_from_fork_asm+0xf/0x14
? entry_INT80_32+0xf0/0xf0
```

[restore no-op path for no changes]

More Info: <https://avd.aquasec.com/nvd/cve-2024-57899>

**[CVE-2024-57924] kernel: fs: relax assertions on failure to encode file handles (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fs: relax assertions on failure to encode file handles

Encoding file handles is usually performed by a filesystem `>encode_fh()` method that may fail for various reasons.

The legacy users of `exportfs_encode_fh()`, namely, `nfsd` and `name_to_handle_at(2)` syscall are ready to cope with the possibility of failure to encode a file handle.

There are a few other users of `exportfs_encode_{fh,fid}()` that currently have a `WARN_ON()` assertion when `->encode_fh()` fails. Relax those assertions because they are wrong.

The second linked bug report states commit `16aac5ad1fa9` ("ovl: support encoding non-decodable file handles") in v6.6 as the regressing commit, but this is not accurate.

The aforementioned commit only increases the chances of the assertion and allows triggering the assertion with the reproducer using overlaysfs, `inotify` and `drop_caches`.

Triggering this assertion was always possible with other filesystems and other reasons of `->encode_fh()` failures and more particularly, it was also possible with the exact same reproducer using overlaysfs that is mounted with options `index=on,nfs_export=on` also on kernels `< v6.6`. Therefore, I am not listing the aforementioned commit as a Fixes commit.

Backport hint: this patch will have a trivial conflict applying to v6.6.y, and other trivial conflicts applying to stable kernels `< v6.6`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57924>

## **[CVE-2024-57945] kernel: riscv: mm: Fix the out of bound issue of vmemmap address (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: `6.1.129-1`

Fixed: `%!s(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

riscv: mm: Fix the out of bound issue of vmemmap address

In sparse vmemmap model, the virtual address of vmemmap is calculated as:  
`((struct page *)VMEMMAP_START - (phys_ram_base >> PAGE_SHIFT)).`  
And the struct page's va can be calculated with an offset:  
`(vmemmap + (pfn)).`

However, when initializing struct pages, kernel actually starts from the first page from the same section that `phys_ram_base` belongs to. If the

first page's physical address is not (`phys_ram_base >> PAGE_SHIFT`), then we get an va below `VMEMMAP_START` when calculating va for it's struct page.

For example, if `phys_ram_base` starts from `0x82000000` with pfn `0x82000`, the first page in the same section is actually pfn `0x80000`. During `init_unavailable_range()`, we will initialize struct page for pfn `0x80000` with virtual address `((struct page *)VMEMMAP_START - 0x2000)`, which is below `VMEMMAP_START` as well as `PCI_IO_END`.

This commit fixes this bug by introducing a new variable `'vmemmap_start_pfn'` which is aligned with memory section size and using it to calculate vmemmap address instead of `phys_ram_base`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57945>

### **[CVE-2024-57950] kernel: drm/amd/display: Initialize denominator defaults to 1 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Initialize denominator defaults to 1

[WHAT & HOW]

Variables, used as denominators and maybe not assigned to other values, should be initialized to non-zero to avoid `DIVIDE_BY_ZERO`, as reported by Coverity.

(cherry picked from commit `e2c4c6c10542ccfe4a0830bb6c9fd5b177b7bbb7`)

More Info: <https://avd.aquasec.com/nvd/cve-2024-57950>

### **[CVE-2024-57952] kernel: Revert "libfs: fix infinite directory reads for offset dir" (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Revert "libfs: fix infinite directory reads for offset dir"

The current directory offset allocator (based on `mtree_alloc_cyclic`) stores the next offset value to return in `octx->next_offset`. This mechanism typically returns values that increase monotonically over time. Eventually, though, the newly allocated offset value wraps back to a low number (say, 2) which is smaller than other already-allocated offset values.

Yu Kuai <yukuai3@huawei.com> reports that, after commit `64a7ce76fb90` ("libfs: fix infinite directory reads for offset dir"), if a directory's offset allocator wraps, existing entries are no longer



visible via `readdir/getdents` because `offset_readdir()` stops listing entries once an entry's offset is larger than `octx->next_offset`. These entries vanish persistently -- they can be looked up, but will never again appear in `readdir(3)` output.

The reason for this is that the commit treats directory offsets as monotonically increasing integer values rather than opaque cookies, and introduces this comparison:

```
if (dentry2offset(dentry) >= last_index) {
```

On 64-bit platforms, the directory offset value upper bound is  $2^{63} - 1$ . Directory offsets will monotonically increase for millions of years without wrapping.

On 32-bit platforms, however, `LONG_MAX` is  $2^{31} - 1$ . The allocator can wrap after only a few weeks (at worst).

Revert commit `64a7ce76fb90` ("libfs: fix infinite directory reads for offset dir") to prepare for a fix that can work properly on 32-bit systems and might apply to recent LTS kernels where `shmem` employs the `simple_offset` mechanism.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57952>

## **[CVE-2024-57974] kernel: udp: Deal with race between UDP socket address change and rehash (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: `6.1.129-1`

Fixed: `%ls(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

udp: Deal with race between UDP socket address change and rehash

If a UDP socket changes its local address while it's receiving datagrams, as a result of `connect()`, there is a period during which a lookup operation might fail to find it, after the address is changed but before the secondary hash (port and address) and the four-tuple hash (local and remote ports and addresses) are updated.

Secondary hash chains were introduced by commit `30fff9231fad` ("udp: bind() optimisation") and, as a result, a rehash operation became needed to make a bound socket reachable again after a `connect()`.

This operation was introduced by commit `719f835853a9` ("udp: add rehash on connect()") which isn't however a complete fix: the socket will be found once the rehashing completes, but not while it's pending.

This is noticeable with a `socat(1)` server in `UDP4-LISTEN` mode, and a client sending datagrams to it. After the server receives the first

datagram (cf. `_xioopen_ipdgram_listen()`), it issues a `connect()` to the address of the sender, in order to set up a directed flow.

Now, if the client, running on a different CPU thread, happens to send a (subsequent) datagram while the server's socket changes its address, but is not rehashed yet, this will result in a failed lookup and a port unreachable error delivered to the client, as apparent from the following reproducer:

```
LEN=$((cat /proc/sys/net/core/wmem_default) / 4))
dd if=/dev/urandom bs=1 count=${LEN} of=tmp.in

while ;; do
  taskset -c 1 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,trunc &
  sleep 0.1 || sleep 1
  taskset -c 2 socat OPEN:tmp.in UDP4:localhost:1337,shut-null
  wait
done
```

where the client will eventually get `ECONNREFUSED` on a `write()` (typically the second or third one of a given iteration):

```
2024/11/13 21:28:23 socat[46901] E write(6, 0x556db2e3c000, 8192): Connection refused
```

This issue was first observed as a seldom failure in Podman's tests checking UDP functionality while using `pasta(1)` to connect the container's network namespace, which leads us to a reproducer with the lookup error resulting in an ICMP packet on a tap device:

```
LOCAL_ADDR="$(ip -j -4 addr show|jq -rM '[] | .addr_info[0] | select(.scope == "global").local')"
```

```
while ;; do
  ./pasta --config-net -p pasta.pcap -u 1337 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,trunc &
  sleep 0.2 || sleep 1
  socat OPEN:tmp.in UDP4:${LOCAL_ADDR}:1337,shut-null
  wait
  cmp tmp.in tmp.out
done
```

Once this fails:

```
tmp.in tmp.out differ: char 8193, line 29
```

we can finally have a look at what's going on:

```
$ tshark -r pasta.pcap
 1  0.000000      :: ? ff02::16   ICMPv6 110 Multicast Listener Report Message v2
 2  0.168690 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
 3  0.168767 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
 4  0.168806 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
 5  0.168827 c6:47:05:8d:dc:04 ? Broadcast  ARP 42 Who has 88.198.0.161? Tell 88.198.0.164
 6  0.168851 9a:55:9a:55:9a:55 ? c6:47:05:8d:dc:04 ARP 42 88.198.0.161 is at 9a:55:9a:55:9a:55
 7  0.168875 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
```

```
8 0.168896 88.198.0.164 ? 88.198.0.161 ICMP 590 Destination unreachable (Port unreachable)
9 0.168926 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
10 0.168959 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192
11 0.168989 88.198.0.161 ? 88.198.0.164 UDP 4138 60260 ? 1337 Len=4096
12 0.169010 88.198.0.161 ? 88.198.0.164 UDP 42 60260 ? 1337 Len=0
```

On the third datagram received, the network namespace of the container initiates an ARP lookup to deliver the ICMP message.

In another variant of this reproducer, starting the client with:

```
strace -f pasta --config-net -u 1337 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,tru
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-57974>

### **[CVE-2024-57975] kernel: btrfs: do proper folio cleanup when run\_delalloc\_nocow() failed (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: do proper folio cleanup when run\_delalloc\_nocow() failed

[BUG]

With CONFIG\_DEBUG\_VM set, test case generic/476 has some chance to crash with the following VM\_BUG\_ON\_FOLIO():

```
BTRFS error (device dm-3): cow_file_range failed, start 1146880 end 1253375 len 106496 ret -28
BTRFS error (device dm-3): run_delalloc_nocow failed, start 1146880 end 1253375 len 106496 ret -28
page: refcount:4 mapcount:0 mapping:00000000592787cc index:0x12 pfn:0x10664
aops:btrfs_aops [btrfs] ino:101 dentry name(?):"f1774"
flags: 0x2ffff80004028(uptodate|lru|private|node=0|zone=2|lastcpupid=0xffff)
page dumped because: VM_BUG_ON_FOLIO(!folio_test_locked(folio))
-----[ cut here ]-----
kernel BUG at mm/page-writeback.c:2992!
Internal error: Oops - BUG: 00000000f2000800 [#1] SMP
CPU: 2 UID: 0 PID: 3943513 Comm: kworker/u24:15 Tainted: G      OE      6.12.0-rc7-custom+ #87
Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE
Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022
Workqueue: events_unbound btrfs_async_reclaim_data_space [btrfs]
pc : folio_clear_dirty_for_io+0x128/0x258
lr : folio_clear_dirty_for_io+0x128/0x258
Call trace:
folio_clear_dirty_for_io+0x128/0x258
btrfs_folio_clamp_clear_dirty+0x80/0xd0 [btrfs]
__process_folios_contig+0x154/0x268 [btrfs]
extent_clear_unlock_delalloc+0x5c/0x80 [btrfs]
run_delalloc_nocow+0x5f8/0x760 [btrfs]
btrfs_run_delalloc_range+0xa8/0x220 [btrfs]
```

```

writepage_delalloc+0x230/0x4c8 [btrfs]
extent_writepage+0xb8/0x358 [btrfs]
extent_write_cache_pages+0x21c/0x4e8 [btrfs]
btrfs_writepages+0x94/0x150 [btrfs]
do_writepages+0x74/0x190
filemap_fdatawrite_wbc+0x88/0xc8
start_delalloc_inodes+0x178/0x3a8 [btrfs]
btrfs_start_delalloc_roots+0x174/0x280 [btrfs]
shrink_delalloc+0x114/0x280 [btrfs]
flush_space+0x250/0x2f8 [btrfs]
btrfs_async_reclaim_data_space+0x180/0x228 [btrfs]
process_one_work+0x164/0x408
worker_thread+0x25c/0x388
kthread+0x100/0x118
ret_from_fork+0x10/0x20
Code: 910a8021 a90363f7 a9046bf9 94012379 (d4210000)
---[ end trace 0000000000000000 ]---
```

### [CAUSE]

The first two lines of extra debug messages show the problem is caused by the error handling of `run_delalloc_nocow()`.

E.g. we have the following dirtied range (4K blocksize 4K page size):

```

0          16K          32K
|//////////|
| Pre-allocated |
```

And the range [0, 16K) has a preallocated extent.

- Enter `run_delalloc_nocow()` for range [0, 16K)  
Which found range [0, 16K) is preallocated, can do the proper NOCOW write.
- Enter `fallback_to_fow()` for range [16K, 32K)  
Since the range [16K, 32K) is not backed by preallocated extent, we have to go COW.
- `cow_file_range()` failed for range [16K, 32K)  
So `cow_file_range()` will do the clean up by clearing folio dirty, unlock the folios.

Now the folios in range [16K, 32K) is unlocked.

- Enter `extent_clear_unlock_delalloc()` from `run_delalloc_nocow()`  
Which is called with `PAGE_START_WRITEBACK` to start page writeback.  
But folios can only be marked writeback when it's properly locked, thus this triggered the `VM_BUG_ON_FOLIO()`.

Furthermore there is another hidden but common bug that `run_delalloc_nocow()` is not clearing the folio dirty flags in its error handling path.

This is the common bug shared between `run_delalloc_nocow()` and

cow\_file\_range().

[FIX]

- Clear folio dirty for range [@start, @cur\_offset)  
Introduce a helper, cleanup\_dirty\_folios(), which will find and lock the folio in the range, clear the dirty flag and start/end the writeback, with the extra handling for the @locked\_folio.
- Introduce a helper to clear folio dirty, start and end writeback
- Introduce a helper to record the last failed COW range end  
This is to trace which range we should skip, to avoid double unlocking.
- Skip the failed COW range for the e  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-57975>

**[CVE-2024-57976] kernel: btrfs: do proper folio cleanup when cow\_file\_range() failed (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: do proper folio cleanup when cow\_file\_range() failed

[BUG]

When testing with COW fixup marked as BUG\_ON() (this is involved with the new pin\_user\_pages\*() change, which should not result new out-of-band dirty pages), I hit a crash triggered by the BUG\_ON() from hitting COW fixup path.

This BUG\_ON() happens just after a failed btrfs\_run\_delalloc\_range():

BTRFS error (device dm-2): failed to run delalloc range, root 348 ino 405 folio 65536 submit\_bitmap 6-15 start 90112 len 106496: -28

-----[ cut here ]-----

kernel BUG at fs/btrfs/extent\_io.c:1444!

Internal error: Oops - BUG: 00000000f2000800 [#1] SMP

CPU: 0 UID: 0 PID: 434621 Comm: kworker/u24:8 Tainted: G OE 6.12.0-rc7-custom+ #86

Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022

Workqueue: events\_unbound btrfs\_async\_reclaim\_data\_space [btrfs]

pc : extent\_writepage\_io+0x2d4/0x308 [btrfs]

lr : extent\_writepage\_io+0x2d4/0x308 [btrfs]

Call trace:

extent\_writepage\_io+0x2d4/0x308 [btrfs]

extent\_writepage+0x218/0x330 [btrfs]

extent\_write\_cache\_pages+0x1d4/0x4b0 [btrfs]

```

btrfs_writepages+0x94/0x150 [btrfs]
do_writepages+0x74/0x190
filemap_fdatawrite_wbc+0x88/0xc8
start_delalloc_inodes+0x180/0x3b0 [btrfs]
btrfs_start_delalloc_roots+0x174/0x280 [btrfs]
shrink_delalloc+0x114/0x280 [btrfs]
flush_space+0x250/0x2f8 [btrfs]
btrfs_async_reclaim_data_space+0x180/0x228 [btrfs]
process_one_work+0x164/0x408
worker_thread+0x25c/0x388
kthread+0x100/0x118
ret_from_fork+0x10/0x20
Code: aa1403e1 9402f3ef aa1403e0 9402f36f (d4210000)
---[ end trace 0000000000000000 ]---

```

## [CAUSE]

That failure is mostly from `cow_file_range()`, where we can hit `-ENOSPC`.

Although the `-ENOSPC` is already a bug related to our space reservation code, let's just focus on the error handling.

For example, we have the following dirty range `[0, 64K)` of an inode, with 4K sector size and 4K page size:

```

0      16K      32K      48K      64K
|//////////|
|#####|

```

Where `////|` means page are still dirty, and `|####|` means the extent io tree has `EXTENT_DELALLOC` flag.

- Enter `extent_writepage()` for page 0
- Enter `btrfs_run_delalloc_range()` for range `[0, 64K)`
- Enter `cow_file_range()` for range `[0, 64K)`
- Function `btrfs_reserve_extent()` only reserved one 16K extent  
So we created extent map and ordered extent for range `[0, 16K)`

```

0      16K      32K      48K      64K
|////////|////////|
|<- OE ->|#####|

```

And range `[0, 16K)` has its delalloc flag cleared.  
But since we haven't yet submit any bio, involved 4 pages are still dirty.

- Function `btrfs_reserve_extent()` returns with `-ENOSPC`  
Now we have to run error cleanup, which will clear all `EXTENT_DELALLOC*` flags and clear the dirty flags for the remaining ranges:

0	16K	32K	48K	64K

Note that range [0, 16K) still has its pages dirty.

- Some time later, writeback is triggered again for the range [0, 16K) since the page range still has dirty flags.
- btrfs\_run\_delalloc\_range() will do nothing because there is no EXTENT\_DELALLOC flag.
- extent\_writepage\_io() finds page 0 has no ordered flag Which falls into the COW fixup path, triggering the BUG\_ON().

Unfortunately this error handling bug dates back to the introduction of btrfs. Thankfully with the abuse of COW fixup, at least it won't crash the kernel.

[FIX]  
Instead of immediately unlocking the extent and folios, we keep the extent and folios locked until either erroring out or the whole delalloc range finished.

When the whole delalloc range finished without error, we just unlock the whole range with PAGE\_SET\_ORDERED (and PAGE\_UNLOCK for !keep\_locked cases)  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-57976>

### [CVE-2024-57977] kernel: memcg: fix soft lockup in the OOM process (Severity: MEDIUM)

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

memcg: fix soft lockup in the OOM process

A soft lockup issue was found in the product with about 56,000 tasks were in the OOM cgroup, it was traversing them when the soft lockup was triggered.

```
watchdog: BUG: soft lockup - CPU#2 stuck for 23s! [VM Thread:1503066]
CPU: 2 PID: 1503066 Comm: VM Thread Kdump: loaded Tainted: G
Hardware name: Huawei Cloud OpenStack Nova, BIOS
RIP: 0010:console_unlock+0x343/0x540
RSP: 0000:ffffb751447db9a0 EFLAGS: 00000247 ORIG_RAX: ffffffff13
RAX: 0000000000000001 RBX: 0000000000000000 RCX: 00000000ffffff
RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000247
RBP: ffffffffa71f90 R08: 0000000000000000 R09: 0000000000000040
R10: 0000000000000080 R11: 0000000000000000 R12: ffffffffa71f90
```

R13: ffffffffaf60a220 R14: 0000000000000247 R15: 0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 00007f2fe6ad91f0 CR3: 00000004b2076003 CR4: 0000000000360ee0  
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000  
DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

Call Trace:

vprintk\_emit+0x193/0x280  
printk+0x52/0x6e  
dump\_task+0x114/0x130  
mem\_cgroup\_scan\_tasks+0x76/0x100  
dump\_header+0x1fe/0x210  
oom\_kill\_process+0xd1/0x100  
out\_of\_memory+0x125/0x570  
mem\_cgroup\_out\_of\_memory+0xb5/0xd0  
try\_charge+0x720/0x770  
mem\_cgroup\_try\_charge+0x86/0x180  
mem\_cgroup\_try\_charge\_delay+0x1c/0x40  
do\_anonymous\_page+0xb5/0x390  
handle\_mm\_fault+0xc4/0x1f0

This is because thousands of processes are in the OOM cgroup, it takes a long time to traverse all of them. As a result, this lead to soft lockup in the OOM process.

To fix this issue, call 'cond\_resched' in the 'mem\_cgroup\_scan\_tasks' function per 1000 iterations. For global OOM, call 'touch\_softlockup\_watchdog' per 1000 iterations to avoid this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57977>

### **[CVE-2024-57999] kernel: powerpc/pseries/iommu: IOMMU incorrectly marks MMIO range in DDW (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

powerpc/pseries/iommu: IOMMU incorrectly marks MMIO range in DDW

Power Hypervisor can possibly allocate MMIO window intersecting with Dynamic DMA Window (DDW) range, which is over 32-bit addressing.

These MMIO pages needs to be marked as reserved so that IOMMU doesn't map DMA buffers in this range.

The current code is not marking these pages correctly which is resulting in LPAR to OOPS while booting. The stack is at below

BUG: Unable to handle kernel data access on read at 0xc00800005cd40000  
Faulting instruction address: 0xc00000000005cdac  
Oops: Kernel access of bad area, sig: 11 [#1]



LE PAGE\_SIZE=64K MMU=Hash SMP NR\_CPUS=2048 NUMA pSeries  
Modules linked in: af\_packet rfkill ibmveth(X) lpfc(+) nvmet\_fc nvmet nvme\_keyring crct10dif\_vpmsum nvme\_fc  
nvme\_fabrics nvme\_core be2net(+) nvme\_auth rtc\_generic nfsd auth\_rpcgss nfs\_acl lockd grace sunrpc fuse configfs  
ip\_tables x\_tables xfs libcrc32c dm\_service\_time ibmvfc(X) scsi\_transport\_fc vmx\_crypto gf128mul crc32c\_vpmsum  
dm\_mirror dm\_region\_hash dm\_log dm\_multipath dm\_mod sd\_mod scsi\_dh\_emc scsi\_dh\_rdac scsi\_dh\_alua t10\_pi  
crc64\_rocksoft\_generic crc64\_rocksoft sg crc64 scsi\_mod  
Supported: Yes, External  
CPU: 8 PID: 241 Comm: kworker/8:1 Kdump: loaded Not tainted 6.4.0-150600.23.14-default #1 SLE15-SP6  
b44ee71c81261b9e4bab5e0cde1f2ed891d5359b  
Hardware name: IBM,9080-M9S POWER9 (raw) 0x4e2103 0xf000005 of:IBM,FW950.B0 (VH950\_149) hv:phyp pSeries  
Workqueue: events work\_for\_cpu\_fn  
NIP: c00000000005cdac LR: c00000000005e830 CTR: 0000000000000000  
REGS: c00001400c9ff770 TRAP: 0300 Not tainted (6.4.0-150600.23.14-default)  
MSR: 800000000280b033 <SF,VEC,VSX,EE,FP,ME,IR,DR,RI,LE> CR: 24228448 XER: 00000001  
CFAR: c00000000005cdd4 DAR: c00800005cd40000 DSISR: 40000000 IRQMASK: 0  
GPR00: c00000000005e830 c00001400c9ffa10 c000000001987d00 c00001400c4fe800  
GPR04: 0000080000000000 0000000000000001 0000000004000000 0000000000800000  
GPR08: 0000000004000000 0000000000000001 c00800005cd40000 ffffffff  
GPR12: 0000000084228882 c00000000a4c4f00 0000000000000010 0000080000000000  
GPR16: c00001400c4fe800 0000000004000000 0800000000000000 c00000006088b800  
GPR20: c00001401a7be980 c00001400eff3800 c000000002a2da68 000000000000002b  
GPR24: c0000000026793a8 c000000002679368 000000000000002a c0000000026793c8  
GPR28: 000008007efffff 0000080000000000 0000000000800000 c00001400c4fe800  
NIP [c00000000005cdac] iommu\_table\_reserve\_pages+0xac/0x100  
LR [c00000000005e830] iommu\_init\_table+0x80/0x1e0  
Call Trace:  
[c00001400c9ffa10] [c00000000005e810] iommu\_init\_table+0x60/0x1e0 (unreliable)  
[c00001400c9ffa90] [c00000000010356c] iommu\_bypass\_supported\_pSeriesLP+0x9cc/0xe40  
[c00001400c9ffc30] [c00000000005c300] dma\_iommu\_dma\_supported+0xf0/0x230  
[c00001400c9ffcb0] [c00000000024b0c4] dma\_supported+0x44/0x90  
[c00001400c9ffcd0] [c00000000024b14c] dma\_set\_mask+0x3c/0x80  
[c00001400c9ffd00] [c0080000555b715c] be\_probe+0xc4/0xb90 [be2net]  
[c00001400c9ffdc0] [c000000000986f3c] local\_pci\_probe+0x6c/0x110  
[c00001400c9ffe40] [c000000000188f28] work\_for\_cpu\_fn+0x38/0x60  
[c00001400c9ffe70] [c00000000018e454] process\_one\_work+0x314/0x620  
[c00001400c9fff10] [c00000000018f280] worker\_thread+0x2b0/0x620  
[c00001400c9fff90] [c00000000019bb18] kthread+0x148/0x150  
[c00001400c9fffe0] [c0000000000ded8] start\_kernel\_thread+0x14/0x18

There are 2 issues in the code

1. The index is "int" while the address is "unsigned long". This results in negative value when setting the bitmap.
2. The DMA offset is page shifted but the MMIO range is used as-is (64-bit address). MMIO address needs to be page shifted as well.

More Info: <https://avd.aquasec.com/nvd/cve-2024-57999>

### **[CVE-2024-58005] kernel: tpm: Change to kcalloc() in eventlog/acpi.c (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

tpm: Change to kcalloc() in eventlog/acpi.c

The following failure was reported on HPE ProLiant D320:

```
[ 10.693310][ T1] tpm_tis STM0925:00: 2.0 TPM (device-id 0x3, rev-id 0)
[ 10.848132][ T1] -----[ cut here ]-----
[ 10.853559][ T1] WARNING: CPU: 59 PID: 1 at mm/page_alloc.c:4727 __alloc_pages_noprof+0x2ca/0x330
[ 10.862827][ T1] Modules linked in:
[ 10.866671][ T1] CPU: 59 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.0-lp155.2.g52785e2-default #1
openSUSE Tumbleweed (unreleased) 588cd98293a7c9eba9013378d807364c088c9375
[ 10.882741][ T1] Hardware name: HPE ProLiant DL320 Gen12/ProLiant DL320 Gen12, BIOS 1.20 10/28/2024
[ 10.892170][ T1] RIP: 0010:__alloc_pages_noprof+0x2ca/0x330
[ 10.898103][ T1] Code: 24 08 e9 4a fe ff ff e8 34 36 fa ff e9 88 fe ff ff 83 fe 0a 0f 86 b3 fd ff ff 80 3d 01 e7 ce 01 00
75 09 c6 05 f8 e6 ce 01 01 <0f> 0b 45 31 ff e9 e5 fe ff ff f7 c2 00 00 08 00 75 42 89 d9 80 e1
[ 10.917750][ T1] RSP: 0000:ffffb7cf40077980 EFLAGS: 00010246
[ 10.923777][ T1] RAX: 0000000000000000 RBX: 0000000000040cc0 RCX: 0000000000000000
[ 10.931727][ T1] RDX: 0000000000000000 RSI: 000000000000000c RDI: 0000000000040cc0
```

The above transcript shows that ACPI pointed a 16 MiB buffer for the log events because RSI maps to the 'order' parameter of `__alloc_pages_noprof()`. Address the bug by moving from `devm_kmalloc()` to `devm_add_action()` and `kvmalloc()` and `devm_add_action()`.

More Info: <https://avd.aquasec.com/nvd/cve-2024-58005>

### **[CVE-2024-58006] kernel: PCI: dwc: ep: Prevent changing BAR size/flags in pci\_epc\_set\_bar() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

PCI: dwc: ep: Prevent changing BAR size/flags in pci\_epc\_set\_bar()

In commit 4284c88ff0e ("PCI: designware-ep: Allow pci\_epc\_set\_bar() update inbound map address") `set_bar()` was modified to support dynamically changing the backing physical address of a BAR that was already configured.

This means that `set_bar()` can be called twice, without ever calling `clear_bar()` (as calling `clear_bar()` would clear the BAR's PCI address assigned by the host).

This can only be done if the new BAR size/flags does not differ from the existing BAR configuration. Add these missing checks.

If we allow `set_bar()` to set e.g. a new BAR size that differs from the existing BAR size, the new address translation range will be smaller than the BAR size already determined by the host, which would mean that a read

past the new BAR size would pass the iATU untranslated, which could allow the host to read memory not belonging to the new struct pci\_epf\_bar.

While at it, add comments which clarifies the support for dynamically changing the physical address of a BAR. (Which was also missing.)

More Info: <https://avd.aquasec.com/nvd/cve-2024-58006>

**[CVE-2024-58012] kernel: ASoC: SOF: Intel: hda-dai: Ensure DAI widget is valid during params (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ASoC: SOF: Intel: hda-dai: Ensure DAI widget is valid during params

Each cpu DAI should associate with a widget. However, the topology might not create the right number of DAI widgets for aggregated amps. And it will cause NULL pointer deference.

Check that the DAI widget associated with the CPU DAI is valid to prevent NULL pointer deference due to missing DAI widgets in topologies with aggregated amps.

More Info: <https://avd.aquasec.com/nvd/cve-2024-58012>

**[CVE-2024-58053] kernel: rxrpc: Fix handling of received connection abort (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

rxrpc: Fix handling of received connection abort

Fix the handling of a connection abort that we've received. Though the abort is at the connection level, it needs propagating to the calls on that connection. Whilst the propagation bit is performed, the calls aren't then woken up to go and process their termination, and as no further input is forthcoming, they just hang.

Also add some tracing for the logging of connection aborts.

More Info: <https://avd.aquasec.com/nvd/cve-2024-58053>

**[CVE-2024-58079] kernel: media: uvcvideo: Fix crash during unbind if gpio unit is in use (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

media: uvcvideo: Fix crash during unbind if gpio unit is in use

We used the wrong device for the device managed functions. We used the usb device, when we should be using the interface device.

If we unbind the driver from the usb interface, the cleanup functions are never called. In our case, the IRQ is never disabled.

If an IRQ is triggered, it will try to access memory sections that are already free, causing an OOPS.

We cannot use the function `devm_request_threaded_irq` here. The `devm_*` clean functions may be called after the main structure is released by `uvc_delete`.

Luckily this bug has small impact, as it is only affected by devices with gpio units and the user has to unbind the device, a disconnect will not trigger this error.

More Info: <https://avd.aquasec.com/nvd/cve-2024-58079>

### **[CVE-2024-58089] kernel: btrfs: fix double accounting race when btrfs\_run\_delalloc\_range() failed (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: fix double accounting race when btrfs\_run\_delalloc\_range() failed

#### **[BUG]**

When running btrfs with block size (4K) smaller than page size (64K, aarch64), there is a very high chance to crash the kernel at generic/750, with the following messages:  
(before the call traces, there are 3 extra debug messages added)

```
BTRFS warning (device dm-3): read-write for sector size 4096 with page size 65536 is experimental
BTRFS info (device dm-3): checking UUID tree
hrtimer: interrupt took 5451385 ns
BTRFS error (device dm-3): cow_file_range failed, root=4957 inode=257 start=1605632 len=69632: -28
BTRFS error (device dm-3): run_delalloc_nocow failed, root=4957 inode=257 start=1605632 len=69632: -28
BTRFS error (device dm-3): failed to run delalloc range, root=4957 ino=257 folio=1572864 submit_bitmap=8-15
start=1605632 len=69632: -28
-----[ cut here ]-----
WARNING: CPU: 2 PID: 3020984 at ordered-data.c:360 can_finish_ordered_extent+0x370/0x3b8 [btrfs]
CPU: 2 UID: 0 PID: 3020984 Comm: kworker/u24:1 Tainted: G      OE   6.13.0-rc1-custom+ #89
Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE
Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022
Workqueue: events_unbound btrfs_async_reclaim_data_space [btrfs]
```

```

pc : can_finish_ordered_extent+0x370/0x3b8 [btrfs]
lr : can_finish_ordered_extent+0x1ec/0x3b8 [btrfs]
Call trace:
can_finish_ordered_extent+0x370/0x3b8 [btrfs] (P)
can_finish_ordered_extent+0x1ec/0x3b8 [btrfs] (L)
btrfs_mark_ordered_io_finished+0x130/0x2b8 [btrfs]
extent_writepage+0x10c/0x3b8 [btrfs]
extent_write_cache_pages+0x21c/0x4e8 [btrfs]
btrfs_writepages+0x94/0x160 [btrfs]
do_writepages+0x74/0x190
filemap_fdatawrite_wbc+0x74/0xa0
start_delalloc_inodes+0x17c/0x3b0 [btrfs]
btrfs_start_delalloc_roots+0x17c/0x288 [btrfs]
shrink_delalloc+0x11c/0x280 [btrfs]
flush_space+0x288/0x328 [btrfs]
btrfs_async_reclaim_data_space+0x180/0x228 [btrfs]
process_one_work+0x228/0x680
worker_thread+0x1bc/0x360
kthread+0x100/0x118
ret_from_fork+0x10/0x20
--[ end trace 0000000000000000 ]---
BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1605632 OE len=16384
to_dec=16384 left=0
BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1622016 OE len=12288
to_dec=12288 left=0
Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008
BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1634304 OE len=8192
to_dec=4096 left=0
CPU: 1 UID: 0 PID: 3286940 Comm: kworker/u24:3 Tainted: G      W OE      6.13.0-rc1-custom+ #89
Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022
Workqueue: btrfs_work_helper [btrfs] (btrfs-endio-write)
pstate: 404000c5 (nZcv daIF +PAN -UAO -TCO -DIT -SSBS BTYPE=--)
pc : process_one_work+0x110/0x680
lr : worker_thread+0x1bc/0x360
Call trace:
process_one_work+0x110/0x680 (P)
worker_thread+0x1bc/0x360 (L)
worker_thread+0x1bc/0x360
kthread+0x100/0x118
ret_from_fork+0x10/0x20
Code: f84086a1 f9000fe1 53041c21 b9003361 (f9400661)
---[ end trace 0000000000000000 ]---
Kernel panic - not syncing: Oops: Fatal exception
SMP: stopping secondary CPUs
SMP: failed to stop secondary CPUs 2-3
Dumping ftrace buffer:
(ftrace buffer empty)
Kernel Offset: 0x275bb9540000 from 0xffff800080000000
PHYS_OFFSET: 0xffff8fbba0000000
CPU features: 0x100,00000070,00801250,8201720b

```

#### [CAUSE]

The above warning is triggered immediately after the delalloc range

failure, this happens in the following sequence:

- Range [1568K, 1636K) is dirty

```
1536K 1568K 1600K 1636K 1664K
|  |////////|////////|  |
```

Where 1536K, 1600K and 1664K are page boundaries (64K page size)

- Enter extent\_writepage() for page 1536K

- Enter run\_delalloc\_nocow() with locke

---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-58089>

### **[CVE-2024-58090] kernel: sched/core: Prevent rescheduling when interrupts are disabled (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

sched/core: Prevent rescheduling when interrupts are disabled

David reported a warning observed while loop testing kexec jump:

Interrupts enabled after irqrouter\_resume+0x0/0x50

WARNING: CPU: 0 PID: 560 at drivers/base/syscore.c:103 syscore\_resume+0x18a/0x220

kernel\_kexec+0xf6/0x180

\_\_do\_sys\_reboot+0x206/0x250

do\_syscall\_64+0x95/0x180

The corresponding interrupt flag trace:

hardirqs last enabled at (15573): [<fffffffa8281b8e>] \_\_up\_console\_sem+0x7e/0x90

hardirqs last disabled at (15580): [<fffffffa8281b73>] \_\_up\_console\_sem+0x63/0x90

That means \_\_up\_console\_sem() was invoked with interrupts enabled. Further instrumentation revealed that in the interrupt disabled section of kexec jump one of the syscore\_suspend() callbacks woke up a task, which set the NEED\_RESCHED flag. A later callback in the resume path invoked cond\_resched() which in turn led to the invocation of the scheduler:

\_\_cond\_resched+0x21/0x60

down\_timeout+0x18/0x60

acpi\_os\_wait\_semaphore+0x4c/0x80

acpi\_ut\_acquire\_mutex+0x3d/0x100

acpi\_ns\_get\_node+0x27/0x60

acpi\_ns\_evaluate+0x1cb/0x2d0

acpi\_rs\_set\_srs\_method\_data+0x156/0x190

```
acpi_pci_link_set+0x11c/0x290
irqrouter_resume+0x54/0x60
syscore_resume+0x6a/0x200
kernel_kexec+0x145/0x1c0
__do_sys_reboot+0xeb/0x240
do_syscall_64+0x95/0x180
```

This is a long standing problem, which probably got more visible with the recent printk changes. Something does a task wakeup and the scheduler sets the NEED\_RESCHED flag. cond\_resched() sees it set and invokes schedule() from a completely bogus context. The scheduler enables interrupts after context switching, which causes the above warning at the end.

Quite some of the code paths in syscore\_suspend()/resume() can result in triggering a wakeup with the exactly same consequences. They might not have done so yet, but as they share a lot of code with normal operations it's just a question of time.

The problem only affects the PREEMPT\_NONE and PREEMPT\_VOLUNTARY scheduling models. Full preemption is not affected as cond\_resched() is disabled and the preemption check preemptible() takes the interrupt disabled flag into account.

Cure the problem by adding a corresponding check into cond\_resched().

More Info: <https://avd.aquasec.com/nvd/cve-2024-58090>

### **[CVE-2025-21634] kernel: cgroup/cpuset: remove kernfs active break (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

cgroup/cpuset: remove kernfs active break

A warning was found:

```
WARNING: CPU: 10 PID: 3486953 at fs/kernfs/file.c:828
CPU: 10 PID: 3486953 Comm: rmdir Kdump: loaded Tainted: G
RIP: 0010:kernfs_should_drain_open_files+0x1a1/0x1b0
RSP: 0018:ffff8881107ef9e0 EFLAGS: 00010202
RAX: 0000000080000002 RBX: ffff888154738c00 RCX: dffffc0000000000
RDX: 0000000000000007 RSI: 0000000000000004 RDI: ffff888154738c04
RBP: ffff888154738c04 R08: ffffffffaf27fa15 R09: ffffed102a8e7180
R10: ffff888154738c07 R11: 0000000000000000 R12: ffff888154738c08
R13: ffff888750f8c000 R14: ffff888750f8c0e8 R15: ffff888154738ca0
FS: 00007f84cd0be740(0000) GS:ffff8887ddc00000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 0000555f9fbe00c8 CR3: 0000000153eec001 CR4: 0000000000370ee0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
```

Call Trace:

```
kernfs_drain+0x15e/0x2f0
__kernfs_remove+0x165/0x300
kernfs_remove_by_name_ns+0x7b/0xc0
cgroup_rm_file+0x154/0x1c0
cgroup_addrm_files+0x1c2/0x1f0
css_clear_dir+0x77/0x110
kill_css+0x4c/0x1b0
cgroup_destroy_locked+0x194/0x380
cgroup_rmdir+0x2a/0x140
```

It can be explained by:

```
rmdir    echo 1 > cpuset.cpus
    kernfs_fop_write_iter // active=0
cgroup_rm_file
kernfs_remove_by_name_ns kernfs_get_active // active=1
__kernfs_remove    // active=0x80000002
kernfs_drain  cpuset_write_resmask
wait_event
//waiting (active == 0x80000001)
    kernfs_break_active_protection
    // active = 0x80000001
// continue
    kernfs_unbreak_active_protection
    // active = 0x80000002
...
kernfs_should_drain_open_files
// warning occurs
    kernfs_put_active
```

This warning is caused by 'kernfs\_break\_active\_protection' when it is writing to cpuset.cpus, and the cgroup is removed concurrently.

The commit 3a5a6d0c2b03 ("cpuset: don't nest cgroup\_mutex inside get\_online\_cpus()") made cpuset\_hotplug\_workfn asynchronous. This change involves calling flush\_work(), which can create a multiple processes circular locking dependency that involve cgroup\_mutex, potentially leading to a deadlock. To avoid deadlock. the commit 76bb5ab8f6e3 ("cpuset: break kernfs active protection in cpuset\_write\_resmask()") added 'kernfs\_break\_active\_protection' in the cpuset\_write\_resmask. This could lead to this warning.

After the commit 2125c0034c5d ("cgroup/cpuset: Make cpuset hotplug processing synchronous"), the cpuset\_write\_resmask no longer needs to wait the hotplug to finish, which means that concurrent hotplug and cpuset operations are no longer possible. Therefore, the deadlock doesn't exist anymore and it does not have to 'break active protection' now. To fix this warning, just remove kernfs\_break\_active\_protection operation in the 'cpuset\_write\_resmask'.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21634>

**[CVE-2025-21635] kernel: rds: sysctl: rds\_tcp\_{rcv,snd}buf: avoid using current->nsproxy (Severity:**



## MEDIUM)

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

rds: sysctl: rds\_tcp\_{rcv,snd}buf: avoid using current->nsproxy

As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons:

- Inconsistency: getting info from the reader's/writer's netns vs only from the opener's netns.
- current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2).

The per-netns structure can be obtained from the table->data using container\_of(), then the 'net' one can be retrieved from the listen socket (if available).

More Info: <https://avd.aquasec.com/nvd/cve-2025-21635>

## **[CVE-2025-21645] kernel: platform/x86/amd/pmc: Only disable IRQ1 wakeup where i8042 actually enabled it (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

platform/x86/amd/pmc: Only disable IRQ1 wakeup where i8042 actually enabled it

Wakeup for IRQ1 should be disabled only in cases where i8042 had actually enabled it, otherwise "wake\_depth" for this IRQ will try to drop below zero and there will be an unpleasant WARN() logged:

kernel: atkbd serio0: Disabling IRQ1 wakeup source to avoid platform firmware bug

kernel: -----[ cut here ]-----

kernel: Unbalanced IRQ 1 wake disable

kernel: WARNING: CPU: 10 PID: 6431 at kernel/irq/manage.c:920 irq\_set\_irq\_wake+0x147/0x1a0

The PMC driver uses DEFINE\_SIMPLE\_DEV\_PM\_OPS() to define its dev\_pm\_ops which sets amd\_pmc\_suspend\_handler() to the .suspend, .freeze, and .poweroff handlers. i8042\_pm\_suspend(), however, is only set as the .suspend handler.

Fix the issue by call PMC suspend handler only from the same set of dev\_pm\_ops handlers as i8042\_pm\_suspend(), which currently means just the .suspend handler.

To reproduce this issue try hibernating (S4) the machine after a fresh boot without putting it into s2idle first.

[ij: edited the commit message.]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21645>

**[CVE-2025-21649] kernel: net: hns3: fix kernel crash when 1588 is sent on HIP08 devices (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: hns3: fix kernel crash when 1588 is sent on HIP08 devices

Currently, HIP08 devices does not register the ptp devices, so the hdev->ptp is NULL. But the tx process would still try to set hardware time stamp info with SKBTX\_HW\_TSTAMP flag and cause a kernel crash.

[ 128.087798] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000018

...

[ 128.280251] pc : hclge\_ptp\_set\_tx\_info+0x2c/0x140 [hclge]  
[ 128.286600] lr : hclge\_ptp\_set\_tx\_info+0x20/0x140 [hclge]  
[ 128.292938] sp : ffff800059b93140  
[ 128.297200] x29: ffff800059b93140 x28: 00000000000003280  
[ 128.303455] x27: ffff800020d48280 x26: ffff0cb9dc814080  
[ 128.309715] x25: ffff0cb9cde93fa0 x24: 0000000000000001  
[ 128.315969] x23: 0000000000000000 x22: 0000000000000194  
[ 128.322219] x21: ffff0cd94f986000 x20: 0000000000000000  
[ 128.328462] x19: ffff0cb9d2a166c0 x18: 0000000000000000  
[ 128.334698] x17: 0000000000000000 x16: ffffcf1fc523ed24  
[ 128.340934] x15: 0000ffffd530a518 x14: 0000000000000000  
[ 128.347162] x13: ffff0cd6bdb31310 x12: 0000000000000368  
[ 128.353388] x11: ffff0cb9cfbc7070 x10: ffff2cf55dd11e02  
[ 128.359606] x9 : ffffcf1f85a212b4 x8 : ffff0cd7cf27dab0  
[ 128.365831] x7 : 00000000000000a20 x6 : ffff0cd7cf27d000  
[ 128.372040] x5 : 0000000000000000 x4 : 000000000000ffff  
[ 128.378243] x3 : 0000000000000400 x2 : ffffcf1f85a21294  
[ 128.384437] x1 : ffff0cb9db520080 x0 : ffff0cb9db500080  
[ 128.390626] Call trace:  
[ 128.393964] hclge\_ptp\_set\_tx\_info+0x2c/0x140 [hclge]  
[ 128.399893] hns3\_nic\_net\_xmit+0x39c/0x4c4 [hns3]  
[ 128.405468] xmit\_one.constprop.0+0xc4/0x200  
[ 128.410600] dev\_hard\_start\_xmit+0x54/0xf0  
[ 128.415556] sch\_direct\_xmit+0xe8/0x634  
[ 128.420246] \_\_dev\_queue\_xmit+0x224/0xc70  
[ 128.425101] dev\_queue\_xmit+0x1c/0x40  
[ 128.429608] ovs\_vport\_send+0xac/0x1a0 [openvswitch]  
[ 128.435409] do\_output+0x60/0x17c [openvswitch]

```
[ 128.440770] do_execute_actions+0x898/0x8c4 [openvswitch]
[ 128.446993] ovs_execute_actions+0x64/0xf0 [openvswitch]
[ 128.453129] ovs_dp_process_packet+0xa0/0x224 [openvswitch]
[ 128.459530] ovs_vport_receive+0x7c/0xfc [openvswitch]
[ 128.465497] internal_dev_xmit+0x34/0xb0 [openvswitch]
[ 128.471460] xmit_one.constprop.0+0xc4/0x200
[ 128.476561] dev_hard_start_xmit+0x54/0xf0
[ 128.481489] __dev_queue_xmit+0x968/0xc70
[ 128.486330] dev_queue_xmit+0x1c/0x40
[ 128.490856] ip_finish_output2+0x250/0x570
[ 128.495810] __ip_finish_output+0x170/0x1e0
[ 128.500832] ip_finish_output+0x3c/0xf0
[ 128.505504] ip_output+0xbc/0x160
[ 128.509654] ip_send_skb+0x58/0xd4
[ 128.513892] udp_send_skb+0x12c/0x354
[ 128.518387] udp_sendmsg+0x7a8/0x9c0
[ 128.522793] inet_sendmsg+0x4c/0x8c
[ 128.527116] __sock_sendmsg+0x48/0x80
[ 128.531609] __sys_sendto+0x124/0x164
[ 128.536099] __arm64_sys_sendto+0x30/0x5c
[ 128.540935] invoke_syscall+0x50/0x130
[ 128.545508] el0_svc_common.constprop.0+0x10c/0x124
[ 128.551205] do_el0_svc+0x34/0xdc
[ 128.555347] el0_svc+0x20/0x30
[ 128.559227] el0_sync_handler+0xb8/0xc0
[ 128.563883] el0_sync+0x160/0x180
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21649>

### **[CVE-2025-21651] kernel: net: hns3: don't auto enable misc vector (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: hns3: don't auto enable misc vector

Currently, there is a time window between misc irq enabled and service task inited. If an interrupt is reported at this time, it will cause warning like below:

```
[ 16.324639] Call trace:
[ 16.324641] __queue_delayed_work+0xb8/0xe0
[ 16.324643] mod_delayed_work_on+0x78/0xd0
[ 16.324655] hclge_errhand_task_schedule+0x58/0x90 [hclge]
[ 16.324662] hclge_misc_irq_handle+0x168/0x240 [hclge]
[ 16.324666] __handle_irq_event_percpu+0x64/0x1e0
[ 16.324667] handle_irq_event+0x80/0x170
[ 16.324670] handle_fasteoi_edge_irq+0x110/0x2bc
[ 16.324671] __handle_domain_irq+0x84/0xfc
[ 16.324673] gic_handle_irq+0x88/0x2c0
[ 16.324674] el1_irq+0xb8/0x140
```

```
[ 16.324677] arch_cpu_idle+0x18/0x40
[ 16.324679] default_idle_call+0x5c/0x1bc
[ 16.324682] cpuidle_idle_call+0x18c/0x1c4
[ 16.324684] do_idle+0x174/0x17c
[ 16.324685] cpu_startup_entry+0x30/0x6c
[ 16.324687] secondary_start_kernel+0x1a4/0x280
[ 16.324688] ---[ end trace 6aa0bff672a964aa ]---
```

So don't auto enable misc vector when request irq..

More Info: <https://avd.aquasec.com/nvd/cve-2025-21651>

### **[CVE-2025-21656] kernel: hwmon: (drivetemp) Fix driver producing garbage data when SCSI errors occur (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

hwmon: (drivetemp) Fix driver producing garbage data when SCSI errors occur

scsi\_execute\_cmd() function can return both negative (linux codes) and positive (scsi\_cmnd result field) error codes.

Currently the driver just passes error codes of scsi\_execute\_cmd() to hwmon core, which is incorrect because hwmon only checks for negative error codes. This leads to hwmon reporting uninitialized data to userspace in case of SCSI errors (for example if the disk drive was disconnected).

This patch checks scsi\_execute\_cmd() output and returns -EIO if it's error code is positive.

[groeck: Avoid inline variable declaration for portability]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21656>

### **[CVE-2025-21658] kernel: btrfs: avoid NULL pointer dereference if no valid extent tree (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

btrfs: avoid NULL pointer dereference if no valid extent tree

[BUG]

Syzbot reported a crash with the following call trace:

BTRFS info (device loop0): scrub: started on devid 1

BUG: kernel NULL pointer dereference, address: 0000000000000208  
#PF: supervisor read access in kernel mode  
#PF: error\_code(0x0000) - not-present page  
PGD 106e70067 P4D 106e70067 PUD 107143067 PMD 0  
Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI  
CPU: 1 UID: 0 PID: 689 Comm: repro Kdump: loaded Tainted: G O 6.13.0-rc4-custom+ #206  
Tainted: [O]=OOT\_MODULE  
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS unknown 02/02/2022  
RIP: 0010:find\_first\_extent\_item+0x26/0x1f0 [btrfs]  
Call Trace:  
<TASK>  
scrub\_find\_fill\_first\_stripe+0x13d/0x3b0 [btrfs]  
scrub\_simple\_mirror+0x175/0x260 [btrfs]  
scrub\_stripe+0x5d4/0x6c0 [btrfs]  
scrub\_chunk+0xbb/0x170 [btrfs]  
scrub\_enumerate\_chunks+0x2f4/0x5f0 [btrfs]  
btrfs\_scrub\_dev+0x240/0x600 [btrfs]  
btrfs\_ioctl+0x1dc8/0x2fa0 [btrfs]  
? do\_sys\_openat2+0xa5/0xf0  
\_\_x64\_sys\_ioctl+0x97/0xc0  
do\_syscall\_64+0x4f/0x120  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e  
</TASK>

#### [CAUSE]

The reproducer is using a corrupted image where extent tree root is corrupted, thus forcing to use "rescue=all,ro" mount option to mount the image.

Then it triggered a scrub, but since scrub relies on extent tree to find where the data/metadata extents are, scrub\_find\_fill\_first\_stripe() relies on an non-empty extent root.

But unfortunately scrub\_find\_fill\_first\_stripe() doesn't really expect an NULL pointer for extent root, it use extent\_root to grab fs\_info and triggered a NULL pointer dereference.

#### [FIX]

Add an extra check for a valid extent root at the beginning of scrub\_find\_fill\_first\_stripe().

The new error path is introduced by 42437a6386ff ("btrfs: introduce mount option rescue=ignorebadroots"), but that's pretty old, and later commit b979547513ff ("btrfs: scrub: introduce helper to find and fill sector info for a scrub\_stripe") changed how we do scrub.

So for kernels older than 6.6, the fix will need manual backport.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21658>

**[CVE-2025-21673] kernel: smb: client: fix double free of TCP\_Server\_Info::hostname (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

smb: client: fix double free of TCP\_Server\_Info::hostname

When shutting down the server in `cifs_put_tcp_session()`, `cifs` thread might be reconnecting to multiple DFS targets before it realizes it should exit the loop, so `@server->hostname` can't be freed as long as `cifs` thread isn't done. Otherwise the following can happen:

```
RIP: 0010:__slab_free+0x223/0x3c0
Code: 5e 41 5f c3 cc cc cc cc 4c 89 de 4c 89 cf 44 89 44 24 08 4c 89
1c 24 e8 fb cf 8e 00 44 8b 44 24 08 4c 8b 1c 24 e9 5f fe ff ff <0f>
0b 41 f7 45 08 00 0d 21 00 0f 85 2d ff ff ff e9 1f ff ff ff 80
RSP: 0018:ffffb26180dbfd08 EFLAGS: 00010246
RAX: ffff8ea34728e510 RBX: ffff8ea34728e500 RCX: 0000000000800068
RDX: 0000000000800068 RSI: 0000000000000000 RDI: ffff8ea340042400
RBP: fffe112041ca380 R08: 0000000000000001 R09: 0000000000000000
R10: 6170732e31303000 R11: 70726f632e786563 R12: ffff8ea34728e500
R13: ffff8ea340042400 R14: ffff8ea34728e500 R15: 0000000000800068
FS: 0000000000000000(0000) GS:ffff8ea66fd80000(0000)
000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007ffc25376080 CR3: 000000012a2ba001 CR4:
PKRU: 55555554
Call Trace:
<TASK>
? show_trace_log_lvl+0x1c4/0x2df
? show_trace_log_lvl+0x1c4/0x2df
? __reconnect_target_unlocked+0x3e/0x160 [cifs]
? __die_body.cold+0x8/0xd
? die+0x2b/0x50
? do_trap+0xce/0x120
? __slab_free+0x223/0x3c0
? do_error_trap+0x65/0x80
? __slab_free+0x223/0x3c0
? exc_invalid_op+0x4e/0x70
? __slab_free+0x223/0x3c0
? asm_exc_invalid_op+0x16/0x20
? __slab_free+0x223/0x3c0
? extract_hostname+0x5c/0xa0 [cifs]
? extract_hostname+0x5c/0xa0 [cifs]
? __kmalloc+0x4b/0x140
__reconnect_target_unlocked+0x3e/0x160 [cifs]
reconnect_dfs_server+0x145/0x430 [cifs]
cifs_handle_standard+0x1ad/0x1d0 [cifs]
cifs_demultiplex_thread+0x592/0x730 [cifs]
? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs]
kthread+0xdd/0x100
? __pfx_kthread+0x10/0x10
```

```
ret_from_fork+0x29/0x50
</TASK>
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21673>

**[CVE-2025-21676] kernel: net: fec: handle page\_pool\_dev\_alloc\_pages error (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: fec: handle page\_pool\_dev\_alloc\_pages error

The fec\_enet\_update\_cbd function calls page\_pool\_dev\_alloc\_pages but did not handle the case when it returned NULL. There was a WARN\_ON(!new\_page) but it would still proceed to use the NULL pointer and then crash.

This case does seem somewhat rare but when the system is under memory pressure it can happen. One case where I can duplicate this with some frequency is when writing over a smb share to a SATA HDD attached to an imx6q.

Setting /proc/sys/vm/min\_free\_kbytes to higher values also seems to solve the problem for my test case. But it still seems wrong that the fec driver ignores the memory allocation error and can crash.

This commit handles the allocation error by dropping the current packet.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21676>

**[CVE-2025-21682] kernel: eth: bnxt: always recalculate features after XDP clearing, fix null-deref (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

eth: bnxt: always recalculate features after XDP clearing, fix null-deref

Recalculate features when XDP is detached.

Before:

```
# ip li set dev eth0 xdp obj xdp_dummy.bpf.o sec xdp
# ip li set dev eth0 xdp off
# ethtool -k eth0 | grep gro
rx-gro-hw: off [requested on]
```

After:

```
# ip li set dev eth0 xdp obj xdp_dummy.bpf.o sec xdp
# ip li set dev eth0 xdp off
```

```
# ethtool -k eth0 | grep gro
rx-gro-hw: on
```

The fact that HW-GRO doesn't get re-enabled automatically is just a minor annoyance. The real issue is that the features will randomly come back during another reconfiguration which just happens to invoke `netdev_update_features()`. The driver doesn't handle reconfiguring two things at a time very robustly.

Starting with commit 98ba1d931f61 ("bnxt\_en: Fix RSS logic in `__bnxt_reserve_rings()`") we only reconfigure the RSS hash table if the "effective" number of Rx rings has changed. If HW-GRO is enabled "effective" number of rings is 2x what user sees. So if we are in the bad state, with HW-GRO re-enablement "pending" after XDP off, and we lower the rings by / 2 - the HW-GRO rings doing 2x and the `ethtool -L` doing / 2 may cancel each other out, and the:

```
if (old_rx_rings != bp->hw_resc.resv_rx_rings &&
```

condition in `__bnxt_reserve_rings()` will be false.  
The RSS map won't get updated, and we'll crash with:

```
BUG: kernel NULL pointer dereference, address: 0000000000000168
RIP: 0010:__bnxt_hwrn_vnic_set_rss+0x13a/0x1a0
      bnxt_hwrn_vnic_rss_cfg_p5+0x47/0x180
      __bnxt_setup_vnic_p5+0x58/0x110
      bnxt_init_nic+0xb72/0xf50
      __bnxt_open_nic+0x40d/0xab0
      bnxt_open_nic+0x2b/0x60
      ethtool_set_channels+0x18c/0x1d0
```

As we try to access a freed ring.

The issue is present since XDP support was added, really, but prior to commit 98ba1d931f61 ("bnxt\_en: Fix RSS logic in `__bnxt_reserve_rings()`") it wasn't causing major issues.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21682>

### **[CVE-2025-21693] kernel: mm: zswap: properly synchronize freeing resources during CPU hotunplug (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm: zswap: properly synchronize freeing resources during CPU hotunplug

In `zswap_compress()` and `zswap_decompress()`, the per-CPU `acomp_ctx` of the current CPU at the beginning of the operation is retrieved and used



throughout. However, since neither preemption nor migration are disabled, it is possible that the operation continues on a different CPU.

If the original CPU is hotunplugged while the `acomp_ctx` is still in use, we run into a UAF bug as some of the resources attached to the `acomp_ctx` are freed during hotunplug in `zswap_cpu_comp_dead()` (i.e. `acomp_ctx.buffer`, `acomp_ctx.req`, or `acomp_ctx.acomp`).

The problem was introduced in commit `1ec3b5fe6eec` ("mm/zswap: move to use `crypto_acomp` API for hardware acceleration") when the switch to the `crypto_acomp` API was made. Prior to that, the per-CPU `crypto_comp` was retrieved using `get_cpu_ptr()` which disables preemption and makes sure the CPU cannot go away from under us. Preemption cannot be disabled with the `crypto_acomp` API as a sleepable context is needed.

Use the `acomp_ctx.mutex` to synchronize CPU hotplug callbacks allocating and freeing resources with compression/decompression paths. Make sure that `acomp_ctx.req` is NULL when the resources are freed. In the compression/decompression paths, check if `acomp_ctx.req` is NULL after acquiring the mutex (meaning the CPU was offlined) and retry on the new CPU.

The initialization of `acomp_ctx.mutex` is moved from the CPU hotplug callback to the pool initialization where it belongs (where the mutex is allocated). In addition to adding clarity, this makes sure that CPU hotplug cannot reinitialize a mutex that is already locked by compression/decompression.

Previously a fix was attempted by holding `cpus_read_lock()` [1]. This would have caused a potential deadlock as it is possible for code already holding the lock to fall into reclaim and enter `zswap` (causing a deadlock). A fix was also attempted using SRCU for synchronization, but Johannes pointed out that `synchronize_srcu()` cannot be used in CPU hotplug notifiers [2].

Alternative fixes that were considered/attempted and could have worked:

- Refcounting the per-CPU `acomp_ctx`. This involves complexity in handling the race between the refcount dropping to zero in `zswap_[de]compress()` and the refcount being re-initialized when the CPU is online.
- Disabling migration before getting the per-CPU `acomp_ctx` [3], but that's discouraged and is a much bigger hammer than needed, and could result in subtle performance issues.

[1]<https://lkml.kernel.org/20241219212437.2714151-1-yosryahmed@google.com/>

[2]<https://lkml.kernel.org/20250107074724.1756696-2-yosryahmed@google.com/>

[3]<https://lkml.kernel.org/20250107222236.2715883-2-yosryahmed@google.com/>

[yosryahmed@google.com: remove comment]

Link: <https://lkml.kernel.org/r/CAJD7tkaxS1wjn+swugt8QCvQ-rVF5RZnjxwPGX17k8x9zSManA@mail.gmail.com>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21693>

## **[CVE-2025-21696] kernel: mm: clear uffd-wp PTE/PMD state on mremap() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm: clear uffd-wp PTE/PMD state on mremap()

When mremap()ing a memory region previously registered with userfaultfd as write-protected but without UFFD\_FEATURE\_EVENT\_REMAP, an inconsistency in flag clearing leads to a mismatch between the vma flags (which have uffd-wp cleared) and the pte/pmd flags (which do not have uffd-wp cleared). This mismatch causes a subsequent mprotect(PROT\_WRITE) to trigger a warning in page\_table\_check\_pte\_flags() due to setting the pte to writable while uffd-wp is still set.

Fix this by always explicitly clearing the uffd-wp pte/pmd flags on any such mremap() so that the values are consistent with the existing clearing of VM\_UFFD\_WP. Be careful to clear the logical flag regardless of its physical form; a PTE bit, a swap PTE bit, or a PTE marker. Cover PTE, huge PMD and hugetlb paths.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21696>

## **[CVE-2025-21712] kernel: md/md-bitmap: Synchronize bitmap\_get\_stats() with bitmap lifetime (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

md/md-bitmap: Synchronize bitmap\_get\_stats() with bitmap lifetime

After commit ec6bb299c7c3 ("md/md-bitmap: add 'sync\_size' into struct md\_bitmap\_stats"), following panic is reported:

Oops: general protection fault, probably for non-canonical address

RIP: 0010:bitmap\_get\_stats+0x2b/0xa0

Call Trace:

<TASK>

md\_seq\_show+0x2d2/0x5b0

seq\_read\_iter+0x2b9/0x470

seq\_read+0x12f/0x180

proc\_reg\_read+0x57/0xb0

vfs\_read+0xf6/0x380

ksys\_read+0x6c/0xf0

do\_syscall\_64+0x82/0x170

entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Root cause is that bitmap\_get\_stats() can be called at anytime if mddev

is still there, even if bitmap is destroyed, or not fully initialized.  
Deferencing bitmap in this case can crash the kernel. Meanwhile, the  
above commit start to deferencing bitmap->storage, make the problem  
easier to trigger.

Fix the problem by protecting bitmap\_get\_stats() with bitmap\_info.mutex.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21712>

### **[CVE-2025-21714] kernel: RDMA/mlx5: Fix implicit ODP use after free (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/mlx5: Fix implicit ODP use after free

Prevent double queueing of implicit ODP mr destroy work by using  
\_\_xa\_cmpxchg() to make sure this is the only time we are destroying this  
specific mr.

Without this change, we could try to invalidate this mr twice, which in  
turn could result in queuing a MR work destroy twice, and eventually the  
second work could execute after the MR was freed due to the first work,  
causing a user after free and trace below.

refcount\_t: underflow; use-after-free.

WARNING: CPU: 2 PID: 12178 at lib/refcount.c:28 refcount\_warn\_saturate+0x12b/0x130

Modules linked in: bonding ib\_ipoib vfio\_pci ip\_gre geneve nf\_tables ip6\_gre gre ip6\_tunnel tunnel6 ipip tunnel4  
ib\_umad rdma\_ucm mlx5\_vfio\_pci vfio\_pci\_core vfio\_iommu\_type1 mlx5\_ib vfio ib\_uverbs mlx5\_core iptable\_raw  
openvswitch nsh rpcrdma ib\_iser libiscsi scsi\_transport\_iscsi rdma\_cm iw\_cm ib\_cm ib\_core xt\_contrack  
xt\_MASQUERADE nf\_contrack\_netlink nfnetlink xt\_addrtype iptable\_nat nf\_nat br\_netfilter rpcsec\_gss\_krb5  
auth\_rpcgss oid\_registry overlay zram zsmalloc fuse [last unloaded: ib\_uverbs]

CPU: 2 PID: 12178 Comm: kworker/u20:5 Not tainted 6.5.0-rc1\_net\_next\_mlx5\_58c644e #1

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org  
04/01/2014

Workqueue: events\_unbound free\_implicit\_child\_mr\_work [mlx5\_ib]

RIP: 0010:refcount\_warn\_saturate+0x12b/0x130

Code: 48 c7 c7 38 95 2a 82 c6 05 bc c6 fe 00 01 e8 0c 66 aa ff 0f 0b 5b c3 48 c7 c7 e0 94 2a 82 c6 05 a7 c6 fe 00 01  
e8 f5 65 aa ff <0f> 0b 5b c3 90 8b 07 3d 00 00 00 c0 74 12 83 f8 01 74 13 8d 50 ff

RSP: 0018:ffff8881008e3e40 EFLAGS: 00010286

RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000027

RDX: ffff88852c91b5c8 RSI: 0000000000000001 RDI: ffff88852c91b5c0

RBP: ffff8881dacd4e00 R08: 00000000ffffff R09: 0000000000000019

R10: 0000000000000072e R11: 0000000063666572 R12: ffff88812bfd9e00

R13: ffff8881c792d200 R14: ffff88810011c005 R15: ffff8881002099c0

FS: 0000000000000000(0000) GS:ffff88852c900000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 00007f5694b5e000 CR3: 00000001153f6003 CR4: 0000000000370ea0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

Call Trace:

```
<TASK>
? refcount_warn_saturate+0x12b/0x130
free_implicit_child_mr_work+0x180/0x1b0 [mlx5_ib]
process_one_work+0x1cc/0x3c0
worker_thread+0x218/0x3c0
kthread+0xc6/0xf0
ret_from_fork+0x1f/0x30
</TASK>
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21714>

**[CVE-2025-21721] kernel: nilfs2: handle errors that nilfs\_prepare\_chunk() may return (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

nilfs2: handle errors that nilfs\_prepare\_chunk() may return

Patch series "nilfs2: fix issues with rename operations".

This series fixes BUG\_ON check failures reported by syzbot around rename operations, and a minor behavioral issue where the mtime of a child directory changes when it is renamed instead of moved.

This patch (of 2):

The directory manipulation routines nilfs\_set\_link() and nilfs\_delete\_entry() rewrite the directory entry in the folio/page previously read by nilfs\_find\_entry(), so error handling is omitted on the assumption that nilfs\_prepare\_chunk(), which prepares the buffer for rewriting, will always succeed for these. And if an error is returned, it triggers the legacy BUG\_ON() checks in each routine.

This assumption is wrong, as proven by syzbot: the buffer layer called by nilfs\_prepare\_chunk() may call nilfs\_get\_block() if necessary, which may fail due to metadata corruption or other reasons. This has been there all along, but improved sanity checks and error handling may have made it more reproducible in fuzzing tests.

Fix this issue by adding missing error paths in nilfs\_set\_link(), nilfs\_delete\_entry(), and their caller nilfs\_rename().

More Info: <https://avd.aquasec.com/nvd/cve-2025-21721>

**[CVE-2025-21723] kernel: scsi: mpi3mr: Fix possible crash when setting up bsg fails (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: mpi3mr: Fix possible crash when setting up bsg fails

If bsg\_setup\_queue() fails, the bsg\_queue is assigned a non-NULL value. Consequently, in mpi3mr\_bsg\_exit(), the condition "if(!mrioc->bsg\_queue)" will not be satisfied, preventing execution from entering bsg\_remove\_queue(), which could lead to the following crash:

BUG: kernel NULL pointer dereference, address: 000000000000041c

Call Trace:

<TASK>

mpi3mr\_bsg\_exit+0x1f/0x50 [mpi3mr]

mpi3mr\_remove+0x6f/0x340 [mpi3mr]

pci\_device\_remove+0x3f/0xb0

device\_release\_driver\_internal+0x19d/0x220

unbind\_store+0xa4/0xb0

kernfs\_fop\_write\_iter+0x11f/0x200

vfs\_write+0x1fc/0x3e0

ksys\_write+0x67/0xe0

do\_syscall\_64+0x38/0x80

entry\_SYSCALL\_64\_after\_hwframe+0x78/0xe2

More Info: <https://avd.aquasec.com/nvd/cve-2025-21723>

### **[CVE-2025-21729] kernel: wifi: rtw89: fix race between cancel\_hw\_scan and hw\_scan completion (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

wifi: rtw89: fix race between cancel\_hw\_scan and hw\_scan completion

The rtwdev->scanning flag isn't protected by mutex originally, so cancel\_hw\_scan can pass the condition, but suddenly hw\_scan completion unset the flag and calls ieee80211\_scan\_completed() that will free local->hw\_scan\_req. Then, cancel\_hw\_scan raises null-ptr-deref and use-after-free. Fix it by moving the check condition to where protected by mutex.

KASAN: null-ptr-deref in range [0x0000000000000088-0x000000000000008f]

CPU: 2 PID: 6922 Comm: kworker/2:2 Tainted: G OE

Hardware name: LENOVO 2356AD1/2356AD1, BIOS G7ETB6WW (2.76 ) 09/10/2019

Workqueue: events cfg80211\_conn\_work [cfg80211]

RIP: 0010:rtw89\_fw\_h2c\_scan\_offload\_be+0xc33/0x13c3 [rtw89\_core]

Code: 00 45 89 6c 24 1c 0f 85 23 01 00 00 48 8b 85 20 ff ff ff 48 8d

RSP: 0018:ffff88811fd9f068 EFLAGS: 00010206

RAX: dffffc0000000000 RBX: ffff88811fd9f258 RCX: 0000000000000001

RDX: 0000000000000011 RSI: 0000000000000001 RDI: 0000000000000089  
RBP: ffff88811fd9f170 R08: 0000000000000000 R09: 0000000000000000  
R10: ffff88811fd9f108 R11: 0000000000000000 R12: ffff88810e47f960  
R13: 0000000000000000 R14: 0000000000000fff R15: 0000000000000000  
FS: 0000000000000000(0000) GS:ffff8881d6f00000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 00007531dfca55b0 CR3: 00000001be296004 CR4: 0000000001706e0

Call Trace:

<TASK>

? show\_regs+0x61/0x73  
? \_\_die\_body+0x20/0x73  
? die\_addr+0x4f/0x7b  
? exc\_general\_protection+0x191/0x1db  
? asm\_exc\_general\_protection+0x27/0x30  
? rtw89\_fw\_h2c\_scan\_offload\_be+0xc33/0x13c3 [rtw89\_core]  
? rtw89\_fw\_h2c\_scan\_offload\_be+0x458/0x13c3 [rtw89\_core]  
? \_\_pfx\_rtw89\_fw\_h2c\_scan\_offload\_be+0x10/0x10 [rtw89\_core]  
? do\_raw\_spin\_lock+0x75/0xdb  
? \_\_pfx\_do\_raw\_spin\_lock+0x10/0x10  
rtw89\_hw\_scan\_offload+0xb5e/0xbf7 [rtw89\_core]  
? \_raw\_spin\_unlock+0xe/0x24  
? \_\_mutex\_lock.constprop.0+0x40c/0x471  
? \_\_pfx\_rtw89\_hw\_scan\_offload+0x10/0x10 [rtw89\_core]  
? \_\_mutex\_lock\_slowpath+0x13/0x1f  
? mutex\_lock+0xa2/0xdc  
? \_\_pfx\_mutex\_lock+0x10/0x10  
rtw89\_hw\_scan\_abort+0x58/0xb7 [rtw89\_core]  
rtw89\_ops\_cancel\_hw\_scan+0x120/0x13b [rtw89\_core]  
ieee80211\_scan\_cancel+0x468/0x4d0 [mac80211]  
ieee80211\_prep\_connection+0x858/0x899 [mac80211]  
ieee80211\_mgd\_auth+0xbea/0xdde [mac80211]  
? \_\_pfx\_ieee80211\_mgd\_auth+0x10/0x10 [mac80211]  
? cfg80211\_find\_elem+0x15/0x29 [cfg80211]  
? is\_bss+0x1b7/0x1d7 [cfg80211]  
ieee80211\_auth+0x18/0x27 [mac80211]  
cfg80211\_mlme\_auth+0x3bb/0x3e7 [cfg80211]  
cfg80211\_conn\_do\_work+0x410/0xb81 [cfg80211]  
? \_\_pfx\_cfg80211\_conn\_do\_work+0x10/0x10 [cfg80211]  
? \_\_kasan\_check\_read+0x11/0x1f  
? psi\_group\_change+0x8bc/0x944  
? \_\_kasan\_check\_write+0x14/0x22  
? mutex\_lock+0x8e/0xdc  
? \_\_pfx\_mutex\_lock+0x10/0x10  
? \_\_pfx\_\_radix\_tree\_lookup+0x10/0x10  
cfg80211\_conn\_work+0x245/0x34d [cfg80211]  
? \_\_pfx\_cfg80211\_conn\_work+0x10/0x10 [cfg80211]  
? update\_cfs\_rq\_load\_avg+0x3bc/0x3d7  
? sched\_clock\_noinstr+0x9/0x1a  
? sched\_clock+0x10/0x24  
? sched\_clock\_cpu+0x7e/0x42e  
? newidle\_balance+0x796/0x937  
? \_\_pfx\_sched\_clock\_cpu+0x10/0x10  
? \_\_pfx\_newidle\_balance+0x10/0x10

```
? __kasan_check_read+0x11/0x1f
? psi_group_change+0x8bc/0x944
? _raw_spin_unlock+0xe/0x24
? raw_spin_rq_unlock+0x47/0x54
? raw_spin_rq_unlock_irq+0x9/0x1f
? finish_task_switch.isra.0+0x347/0x586
? __schedule+0x27bf/0x2892
? mutex_unlock+0x80/0xd0
? do_raw_spin_lock+0x75/0xdb
? __pfx__schedule+0x10/0x10
process_scheduled_works+0x58c/0x821
worker_thread+0x4c7/0x586
? __kasan_check_read+0x11/0x1f
kthread+0x285/0x294
? __pfx_worker_thread+0x10/0x10
? __pfx_kthread+0x10/0x10
ret_from_fork+0x29/0x6f
? __pfx_kthread+0x10/0x10
ret_from_fork_asm+0x1b/0x30
</TASK>
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21729>

**[CVE-2025-21732] kernel: RDMA/mlx5: Fix a race for an ODP MR which leads to CQE with error (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/mlx5: Fix a race for an ODP MR which leads to CQE with error

This patch addresses a race condition for an ODP MR that can result in a CQE with an error on the UMR QP.

During the \_\_mlx5\_ib\_dereg\_mr() flow, the following sequence of calls occurs:

```
mlx5_revoke_mr()
mlx5r_umr_revoke_mr()
mlx5r_umr_post_send_wait()
```

At this point, the lkey is freed from the hardware's perspective.

However, concurrently, mlx5\_ib\_invalidate\_range() might be triggered by another task attempting to invalidate a range for the same freed lkey.

This task will:

- Acquire the umem\_odp->umem\_mutex lock.
- Call mlx5r\_umr\_update\_xlt() on the UMR QP.
- Since the lkey has already been freed, this can lead to a CQE error,

causing the UMR QP to enter an error state [1].

To resolve this race condition, the `umem_odp->umem_mutex` lock is now also acquired as part of the `mlx5_revoke_mr()` scope. Upon successful revoke, we set `umem_odp->private` which points to that MR to NULL, preventing any further invalidation attempts on its lkey.

[1] From dmesg:

```
infiniband rocep8s0f0: dump_cqe:277:(pid 0): WC error: 6, Message: memory bind operation error
cqe_dump: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
cqe_dump: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
cqe_dump: 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
cqe_dump: 00000030: 00 00 00 00 08 00 78 06 25 00 11 b9 00 0e dd d2
```

WARNING: CPU: 15 PID: 1506 at drivers/infiniband/hw/mlx5/umr.c:394 `mlx5r_umr_post_send_wait+0x15a/0x2b0` [`mlx5_ib`]

Modules linked in: `ip6table_mangle ip6table_nat ip6table_filter ip6_tables iptable_mangle xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcsec_gss_krb5 auth_rpcgss oid_registry overlay rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi rdma_cm iw_cm ib_umad ib_ipoib ib_cm mlx5_ib ib_uverbs ib_core fuse mlx5_core`

CPU: 15 UID: 0 PID: 1506 Comm: `ibv_rc_pingpong` Not tainted 6.12.0-rc7+ #1626

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014

RIP: `0010:mlx5r_umr_post_send_wait+0x15a/0x2b0` [`mlx5_ib`]

[..]

Call Trace:

<TASK>

`mlx5r_umr_update_xlt+0x23c/0x3e0` [`mlx5_ib`]

`mlx5_ib_invalidate_range+0x2e1/0x330` [`mlx5_ib`]

`__mmu_notifier_invalidate_range_start+0x1e1/0x240`

`zap_page_range_single+0xf1/0x1a0`

`madvise_vma_behavior+0x677/0x6e0`

`do_madvise+0x1a2/0x4b0`

`__x64_sys_madvise+0x25/0x30`

`do_syscall_64+0x6b/0x140`

`entry_SYSCALL_64_after_hwframe+0x76/0x7e`

More Info: <https://avd.aquasec.com/nvd/cve-2025-21732>

### **[CVE-2025-21739] kernel: scsi: ufs: core: Fix use-after free in init error and remove paths (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

scsi: ufs: core: Fix use-after free in init error and remove paths

`devm_blk_crypto_profile_init()` registers a cleanup handler to run when the associated (platform-) device is being released. For UFS, the



crypto private data and pointers are stored as part of the ufs\_hba's data structure 'struct ufs\_hba::crypto\_profile'. This structure is allocated as part of the underlying ufshcd and therefore Scsi\_host allocation.

During driver release or during error handling in ufshcd\_pltfrm\_init(), this structure is released as part of ufshcd\_dealloc\_host() before the (platform-) device associated with the crypto call above is released. Once this device is released, the crypto cleanup code will run, using the just-released 'struct ufs\_hba::crypto\_profile'. This causes a use-after-free situation:

Call trace:

```
kfree+0x60/0x2d8 (P)
kvfree+0x44/0x60
blk_crypto_profile_destroy_callback+0x28/0x70
devm_action_release+0x1c/0x30
release_nodes+0x6c/0x108
devres_release_all+0x98/0x100
device_unbind_cleanup+0x20/0x70
really_probe+0x218/0x2d0
```

In other words, the initialisation code flow is:

```
platform-device probe
ufshcd_pltfrm_init()
ufshcd_alloc_host()
scsi_host_alloc()
    allocation of struct ufs_hba
    creation of scsi-host devices
devm_blk_crypto_profile_init()
devm registration of cleanup handler using platform-device
```

and during error handling of ufshcd\_pltfrm\_init() or during driver removal:

```
ufshcd_dealloc_host()
scsi_host_put()
put_device(scsi-host)
    release of struct ufs_hba
put_device(platform-device)
crypto cleanup handler
```

To fix this use-after free, change ufshcd\_alloc\_host() to register a devres action to automatically cleanup the underlying SCSI device on ufshcd destruction, without requiring explicit calls to ufshcd\_dealloc\_host(). This way:

- \* the crypto profile and all other ufs\_hba-owned resources are destroyed before SCSI (as they've been registered after)
- \* a memleak is plugged in tc-dwc-g210-pci.c remove() as a side-effect
- \* EXPORT\_SYMBOL\_GPL(ufshcd\_dealloc\_host) can be removed fully as

it's not needed anymore

\* no future drivers using `ufshcd_alloc_host()` could ever forget adding the cleanup

More Info: <https://avd.aquasec.com/nvd/cve-2025-21739>

**[CVE-2025-21751] kernel: net/mlx5: HWS, change error flow on matcher disconnect (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5: HWS, change error flow on matcher disconnect

Currently, when firmware failure occurs during matcher disconnect flow, the error flow of the function reconnects the matcher back and returns an error, which continues running the calling function and eventually frees the matcher that is being disconnected.

This leads to a case where we have a freed matcher on the matchers list, which in turn leads to use-after-free and eventual crash.

This patch fixes that by not trying to reconnect the matcher back when some FW command fails during disconnect.

Note that we're dealing here with FW error. We can't overcome this problem. This might lead to bad steering state (e.g. wrong connection between matchers), and will also lead to resource leakage, as it is the case with any other error handling during resource destruction.

However, the goal here is to allow the driver to continue and not crash the machine with use-after-free error.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21751>

**[CVE-2025-21756] kernel: vsock: Keep the binding until socket destruction (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

vsock: Keep the binding until socket destruction

Preserve sockets bindings; this includes both resulting from an explicit `bind()` and those implicitly bound through `autobind` during `connect()`.

Prevents socket unbinding during a transport reassignment, which fixes a use-after-free:

1. `vsock_create()` (`refcnt=1`) calls `vsock_insert_unbound()` (`refcnt=2`)

2. transport->release() calls vsock\_remove\_bound() without checking if sk was bound and moved to bound list (refcnt=1)
3. vsock\_bind() assumes sk is in unbound list and before  
\_\_vsock\_insert\_bound(vsock\_bound\_sockets()) calls  
\_\_vsock\_remove\_bound() which does:  
list\_del\_init(&vsk->bound\_table); // nop  
sock\_put(&vsk->sk); // refcnt=0

BUG: KASAN: slab-use-after-free in \_\_vsock\_bind+0x62e/0x730

Read of size 4 at addr ffff88816b46a74c by task a.out/2057

dump\_stack\_lvl+0x68/0x90  
print\_report+0x174/0x4f6  
kasan\_report+0xb9/0x190  
\_\_vsock\_bind+0x62e/0x730  
vsock\_bind+0x97/0xe0  
\_\_sys\_bind+0x154/0x1f0  
\_\_x64\_sys\_bind+0x6e/0xb0  
do\_syscall\_64+0x93/0x1b0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Allocated by task 2057:

kasan\_save\_stack+0x1e/0x40  
kasan\_save\_track+0x10/0x30  
\_\_kasan\_slab\_alloc+0x85/0x90  
kmem\_cache\_alloc\_noprof+0x131/0x450  
sk\_prot\_alloc+0x5b/0x220  
sk\_alloc+0x2c/0x870  
\_\_vsock\_create.constprop.0+0x2e/0xb60  
vsock\_create+0xe4/0x420  
\_\_sock\_create+0x241/0x650  
\_\_sys\_socket+0xf2/0x1a0  
\_\_x64\_sys\_socket+0x6e/0xb0  
do\_syscall\_64+0x93/0x1b0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

Freed by task 2057:

kasan\_save\_stack+0x1e/0x40  
kasan\_save\_track+0x10/0x30  
kasan\_save\_free\_info+0x37/0x60  
\_\_kasan\_slab\_free+0x4b/0x70  
kmem\_cache\_free+0x1a1/0x590  
\_\_sk\_destruct+0x388/0x5a0  
\_\_vsock\_bind+0x5e1/0x730  
vsock\_bind+0x97/0xe0  
\_\_sys\_bind+0x154/0x1f0  
\_\_x64\_sys\_bind+0x6e/0xb0  
do\_syscall\_64+0x93/0x1b0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

refcount\_t: addition on 0; use-after-free.

WARNING: CPU: 7 PID: 2057 at lib/refcount.c:25 refcount\_warn\_saturate+0xce/0x150

RIP: 0010:refcount\_warn\_saturate+0xce/0x150

\_\_vsock\_bind+0x66d/0x730

vsock\_bind+0x97/0xe0  
\_\_sys\_bind+0x154/0x1f0  
\_\_x64\_sys\_bind+0x6e/0xb0  
do\_syscall\_64+0x93/0x1b0  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

refcount\_t: underflow; use-after-free.

WARNING: CPU: 7 PID: 2057 at lib/refcount.c:28 refcount\_warn\_saturate+0xee/0x150

RIP: 0010:refcount\_warn\_saturate+0xee/0x150

vsock\_remove\_bound+0x187/0x1e0

\_\_vsock\_release+0x383/0x4a0

vsock\_release+0x90/0x120

\_\_sock\_release+0xa3/0x250

sock\_close+0x14/0x20

\_\_fput+0x359/0xa80

task\_work\_run+0x107/0x1d0

do\_exit+0x847/0x2560

do\_group\_exit+0xb8/0x250

\_\_x64\_sys\_exit\_group+0x3a/0x50

x64\_sys\_call+0xfec/0x14f0

do\_syscall\_64+0x93/0x1b0

entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e

More Info: <https://avd.aquasec.com/nvd/cve-2025-21756>

### **[CVE-2025-21759] kernel: ipv6: mcast: extend RCU protection in igmp6\_send() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ipv6: mcast: extend RCU protection in igmp6\_send()

igmp6\_send() can be called without RTNL or RCU being held.

Extend RCU protection so that we can safely fetch the net pointer and avoid a potential UAF.

Note that we no longer can use sock\_alloc\_send\_skb() because ipv6.igmp\_sk uses GFP\_KERNEL allocations which can sleep.

Instead use alloc\_skb() and charge the net->ipv6.igmp\_sk socket under RCU protection.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21759>

### **[CVE-2025-21768] kernel: net: ipv6: fix dst ref loops in rpl, seg6 and ioam6 lwtunnels (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: ipv6: fix dst ref loops in rpl, seg6 and ioam6 lwtunnels

Some lwtunnels have a dst cache for post-transformation dst. If the packet destination did not change we may end up recording a reference to the lwtunnel in its own cache, and the lwtunnel state will never be freed.

Discovered by the ioam6.sh test, kmemleak was recently fixed to catch per-cpu memory leaks. I'm not sure if rpl and seg6 can actually hit this, but in principle I don't see why not.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21768>

**[CVE-2025-21801] kernel: net: ravb: Fix missing rtnl lock in suspend/resume path (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: ravb: Fix missing rtnl lock in suspend/resume path

Fix the suspend/resume path by ensuring the rtnl lock is held where required. Calls to ravb\_open, ravb\_close and wol operations must be performed under the rtnl lock to prevent conflicts with ongoing ndo operations.

Without this fix, the following warning is triggered:

```
[ 39.032969] =====
[ 39.032983] WARNING: suspicious RCU usage
[ 39.033019] -----
[ 39.033033] drivers/net/phy/phy_device.c:2004 suspicious
rcu_dereference_protected() usage!
...
[ 39.033597] stack backtrace:
[ 39.033613] CPU: 0 UID: 0 PID: 174 Comm: python3 Not tainted
6.13.0-rc7-next-20250116-arm64-renesas-00002-g35245dfdc62c #7
[ 39.033623] Hardware name: Renesas SMARC EVK version 2 based on
r9a08g045s33 (DT)
[ 39.033628] Call trace:
[ 39.033633] show_stack+0x14/0x1c (C)
[ 39.033652] dump_stack_lvl+0xb4/0xc4
[ 39.033664] dump_stack+0x14/0x1c
[ 39.033671] lockdep_rcu_suspicious+0x16c/0x22c
[ 39.033682] phy_detach+0x160/0x190
[ 39.033694] phy_disconnect+0x40/0x54
[ 39.033703] ravb_close+0x6c/0x1cc
[ 39.033714] ravb_suspend+0x48/0x120
[ 39.033721] dpm_run_callback+0x4c/0x14c
[ 39.033731] device_suspend+0x11c/0x4dc
```

```
[ 39.033740] dpm_suspend+0xdc/0x214
[ 39.033748] dpm_suspend_start+0x48/0x60
[ 39.033758] suspend_devices_and_enter+0x124/0x574
[ 39.033769] pm_suspend+0x1ac/0x274
[ 39.033778] state_store+0x88/0x124
[ 39.033788] kobj_attr_store+0x14/0x24
[ 39.033798] sysfs_kf_write+0x48/0x6c
[ 39.033808] kernfs_fop_write_iter+0x118/0x1a8
[ 39.033817] vfs_write+0x27c/0x378
[ 39.033825] ksys_write+0x64/0xf4
[ 39.033833] __arm64_sys_write+0x18/0x20
[ 39.033841] invoke_syscall+0x44/0x104
[ 39.033852] el0_svc_common.constprop.0+0xb4/0xd4
[ 39.033862] do_el0_svc+0x18/0x20
[ 39.033870] el0_svc+0x3c/0xf0
[ 39.033880] el0t_64_sync_handler+0xc0/0xc4
[ 39.033888] el0t_64_sync+0x154/0x158
[ 39.041274] ravb 11c30000.ethernet eth0: Link is Down
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21801>

## **[CVE-2025-21816] kernel: hrtimers: Force migrate away hrtimers queued after CPUHP\_AP\_HRTIMERS\_DYING (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

hrtimers: Force migrate away hrtimers queued after CPUHP\_AP\_HRTIMERS\_DYING

hrtimers are migrated away from the dying CPU to any online target at the CPUHP\_AP\_HRTIMERS\_DYING stage in order not to delay bandwidth timers handling tasks involved in the CPU hotplug forward progress.

However wakeups can still be performed by the outgoing CPU after CPUHP\_AP\_HRTIMERS\_DYING. Those can result again in bandwidth timers being armed. Depending on several considerations (crystal ball power management based election, earliest timer already enqueued, timer migration enabled or not), the target may eventually be the current CPU even if offline. If that happens, the timer is eventually ignored.

The most notable example is RCU which had to deal with each and every of those wake-ups by deferring them to an online CPU, along with related workarounds:

```
_ e787644caf76 (rcu: Defer RCU kthreads wakeup when CPU is dying)
_ 9139f93209d1 (rcu/nocb: Fix RT throttling hrtimer armed from offline CPU)
_ f7345ccc62a4 (rcu/nocb: Fix rcuog wake-up from offline softirq)
```

The problem isn't confined to RCU though as the stop machine kthread (which runs CPUHP\_AP\_HRTIMERS\_DYING) reports its completion at the end

of its work through `cpu_stop_signal_done()` and performs a wake up that eventually arms the deadline server timer:

```
WARNING: CPU: 94 PID: 588 at kernel/time/hrtimer.c:1086 hrtimer_start_range_ns+0x289/0x2d0
CPU: 94 UID: 0 PID: 588 Comm: migration/94 Not tainted
Stopper: multi_cpu_stop+0x0/0x120 <- stop_machine_cpuslocked+0x66/0xc0
RIP: 0010:hrtimer_start_range_ns+0x289/0x2d0
Call Trace:
<TASK>
  start_dl_timer
  enqueue_dl_entity
  dl_server_start
  enqueue_task_fair
  enqueue_task
  ttwu_do_activate
  try_to_wake_up
  complete
  cpu_stopper_thread
```

Instead of providing yet another bandaid to work around the situation, fix it in the hrtimers infrastructure instead: always migrate away a timer to an online target whenever it is enqueued from an offline CPU.

This will also allow to revert all the above RCU disgraceful hacks.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21816>

#### **[CVE-2025-21817] kernel: block: mark GFP\_NOIO around sysfs ->store() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

block: mark GFP\_NOIO around sysfs ->store()

sysfs ->store is called with queue freezed, meantime we have several ->store() callbacks(update\_nr\_requests, wbt, scheduler) to allocate memory with GFP\_KERNEL which may run into direct reclaim code path, then potential deadlock can be caused.

Fix the issue by marking NOIO around sysfs ->store()

More Info: <https://avd.aquasec.com/nvd/cve-2025-21817>

#### **[CVE-2025-21831] kernel: PCI: Avoid putting some root ports into D3 on TUXEDO Sirius Gen1 (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

PCI: Avoid putting some root ports into D3 on TUXEDO Sirius Gen1

commit 9d26d3a8f1b0 ("PCI: Put PCIe ports into D3 during suspend") sets the policy that all PCIe ports are allowed to use D3. When the system is suspended if the port is not power manageable by the platform and won't be used for wakeup via a PME this sets up the policy for these ports to go into D3hot.

This policy generally makes sense from an OSPM perspective but it leads to problems with wakeup from suspend on the TUXEDO Sirius 16 Gen 1 with a specific old BIOS. This manifests as a system hang.

On the affected Device + BIOS combination, add a quirk for the root port of the problematic controller to ensure that these root ports are not put into D3hot at suspend.

This patch is based on

<https://lore.kernel.org/linux-pci/20230708214457.1229-2-mario.limonciello@amd.com>

but with the added condition both in the documentation and in the code to apply only to the TUXEDO Sirius 16 Gen 1 with a specific old BIOS and only the affected root ports.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21831>

#### **[CVE-2025-21833] kernel: iommu/vt-d: Avoid use of NULL after WARN\_ON\_ONCE (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

iommu/vt-d: Avoid use of NULL after WARN\_ON\_ONCE

There is a WARN\_ON\_ONCE to catch an unlikely situation when domain\_remove\_dev\_pasid can't find the `pasid`. In case it nevertheless happens we must avoid using a NULL pointer.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21833>

#### **[CVE-2025-21836] kernel: io\_uring/kbuf: re-using old struct io\_buffer\_list may lead to a use-after-free situation (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

io\_uring/kbuf: reallocate buf lists on upgrade



IORING\_REGISTER\_PBUF\_RING can reuse an old struct io\_buffer\_list if it was created for legacy selected buffer and has been emptied. It violates the requirement that most of the field should stay stable after publish. Always reallocate it instead.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21836>

**[CVE-2025-21838] kernel: usb: gadget: core: flush gadget workqueue after device removal (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usb: gadget: core: flush gadget workqueue after device removal

device\_del() can lead to new work being scheduled in gadget->work workqueue. This is observed, for example, with the dwc3 driver with the following call stack:

```
device_del()
  gadget_unbind_driver()
    usb_gadget_disconnect_locked()
      dwc3_gadget_pullup()
dwc3_gadget_soft_disconnect()
  usb_gadget_set_state()
    schedule_work(&gadget->work)
```

Move flush\_work() after device\_del() to ensure the workqueue is cleaned up.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21838>

**[CVE-2025-21839] kernel: KVM: x86: Load DR6 with guest value only before entering .vcpu\_run() loop (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

KVM: x86: Load DR6 with guest value only before entering .vcpu\_run() loop

Move the conditional loading of hardware DR6 with the guest's DR6 value out of the core .vcpu\_run() loop to fix a bug where KVM can load hardware with a stale vcpu->arch.dr6.

When the guest accesses a DR and host userspace isn't debugging the guest, KVM disables DR interception and loads the guest's values into hardware on VM-Enter and saves them on VM-Exit. This allows the guest to access DRs at will, e.g. so that a sequence of DR accesses to configure a breakpoint only generates one VM-Exit.

For DR0-DR3, the logic/behavior is identical between VMX and SVM, and also identical between KVM\_DEBUGREG\_BP\_ENABLED (userspace debugging the guest) and KVM\_DEBUGREG\_WONT\_EXIT (guest using DRs), and so KVM handles loading DR0-DR3 in common code, `_outside_` of the core `kvm_x86_ops.vcpu_run()` loop.

But for DR6, the guest's value doesn't need to be loaded into hardware for KVM\_DEBUGREG\_BP\_ENABLED, and SVM provides a dedicated VMCB field whereas VMX requires software to manually load the guest value, and so loading the guest's value into DR6 is handled by `{svm,vmx}_vcpu_run()`, i.e. is done `_inside_` the core run loop.

Unfortunately, saving the guest values on VM-Exit is initiated by common x86, again outside of the core run loop. If the guest modifies DR6 (in hardware, when DR interception is disabled), and then the next VM-Exit is a fastpath VM-Exit, KVM will reload hardware DR6 with `vcpu->arch.dr6` and clobber the guest's actual value.

The bug shows up primarily with nested VMX because KVM handles the VMX preemption timer in the fastpath, and the window between hardware DR6 being modified (in guest context) and DR6 being read by guest software is orders of magnitude larger in a nested setup. E.g. in non-nested, the VMX preemption timer would need to fire precisely between #DB injection and the #DB handler's read of DR6, whereas with a KVM-on-KVM setup, the window where hardware DR6 is "dirty" extends all the way from L1 writing DR6 to VMRESUME (in L1).

L1's view:

=====

<L1 disables DR interception>

CPU 0/KVM-7289 [023] d.... 2925.640961: kvm\_entry: vcpu 0

A: L1 Writes DR6

CPU 0/KVM-7289 [023] d.... 2925.640963: <hack>: Set DRs, DR6 = 0xffff0ff1

B: CPU 0/KVM-7289 [023] d.... 2925.640967: kvm\_exit: vcpu 0 reason EXTERNAL\_INTERRUPT intr\_info 0x800000ec

D: L1 reads DR6, `arch.dr6 = 0`

CPU 0/KVM-7289 [023] d.... 2925.640969: <hack>: Sync DRs, DR6 = 0xffff0ff0

CPU 0/KVM-7289 [023] d.... 2925.640976: kvm\_entry: vcpu 0

L2 reads DR6, L1 disables DR interception

CPU 0/KVM-7289 [023] d.... 2925.640980: kvm\_exit: vcpu 0 reason DR\_ACCESS info1 0x00000000000000216

CPU 0/KVM-7289 [023] d.... 2925.640983: kvm\_entry: vcpu 0

CPU 0/KVM-7289 [023] d.... 2925.640983: <hack>: Set DRs, DR6 = 0xffff0ff0

L2 detects failure

CPU 0/KVM-7289 [023] d.... 2925.640987: kvm\_exit: vcpu 0 reason HLT

L1 reads DR6 (confirms failure)

CPU 0/KVM-7289 [023] d.... 2925.640990: <hack>: Sync DRs, DR6 = 0xffff0ff0

L0's view:

```

=====
L2 reads DR6, arch.dr6 = 0
      CPU 23/KVM-5046      [001] d.... 3410.005610: kvm_exit: vcpu 23 reason DR_ACCESS info1
0x00000000000000216
      CPU 23/KVM-5046      [001] ..... 3410.005610: kvm_nested_vmexit: vcpu 23 reason DR_ACCESS info1
0x00000000000000216

L2 => L1 nested VM-Exit
      CPU 23/KVM-5046      [001] ..... 3410.005610: kvm_nested_vmexit_inject: reason: DR_ACCESS ext_inf1:
0x00000000000000216

      CPU 23/KVM-5046      [001] d.... 3410.005610: kvm_entry: vcpu 23
      CPU 23/KVM-5046      [001] d.... 3410.005611: kvm_exit: vcpu 23 reason VMREAD
      CPU 23/KVM-5046      [001] d.... 3410.005611: kvm_entry: vcpu 23
      CPU 23/KVM-5046      [001] d.... 3410.
---truncated---

```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21839>

**[CVE-2025-21844] kernel: smb: client: Add check for next\_buffer in receive\_encrypted\_standard()  
(Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

smb: client: Add check for next\_buffer in receive\_encrypted\_standard()

Add check for the return value of cifs\_buf\_get() and cifs\_small\_buf\_get()  
in receive\_encrypted\_standard() to prevent null pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21844>

**[CVE-2025-21846] kernel: acct: perform last write from workqueue (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

acct: perform last write from workqueue

In [1] it was reported that the acct(2) system call can be used to trigger NULL deref in cases where it is set to write to a file that triggers an internal lookup. This can e.g., happen when pointing acct(2) to /sys/power/resume. At the point the where the write to this file happens the calling task has already exited and called exit\_fs(). A lookup will thus trigger a NULL-deref when accessing current->fs.

Reorganize the code so that the the final write happens from the workqueue but with the caller's credentials. This preserves the

(strange) permission model and has almost no regression risk.

This api should stop to exist though.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21846>

**[CVE-2025-21848] kernel: nfp: bpf: Add check for nfp\_app\_ctrl\_msg\_alloc() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

nfp: bpf: Add check for nfp\_app\_ctrl\_msg\_alloc()

Add check for the return value of nfp\_app\_ctrl\_msg\_alloc() in nfp\_bpf\_cmsg\_alloc() to prevent null pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21848>

**[CVE-2025-21853] kernel: bpf: avoid holding freeze\_mutex during mmap operation (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: avoid holding freeze\_mutex during mmap operation

We use map->freeze\_mutex to prevent races between map\_freeze() and memory mapping BPF map contents with writable permissions. The way we naively do this means we'll hold freeze\_mutex for entire duration of all the mm and VMA manipulations, which is completely unnecessary. This can potentially also lead to deadlocks, as reported by syzbot in [0].

So, instead, hold freeze\_mutex only during writeability checks, bump (proactively) "write active" count for the map, unlock the mutex and proceed with mmap logic. And only if something went wrong during mmap logic, then undo that "write active" counter increment.

[0] <https://lore.kernel.org/bpf/678dcbc9.050a0220.303755.0066.GAE@google.com/>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21853>

**[CVE-2025-21859] kernel: USB: gadget: f\_midi: f\_midi\_complete to call queue\_work (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

USB: gadget: f\_midi: f\_midi\_complete to call queue\_work

When using USB MIDI, a lock is attempted to be acquired twice through a re-entrant call to f\_midi\_transmit, causing a deadlock.

Fix it by using queue\_work() to schedule the inner f\_midi\_transmit() via a high priority work queue from the completion handler.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21859>

### **[CVE-2025-21861] kernel: mm/migrate\_device: don't add folio to be freed to LRU in migrate\_device\_finalize() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm/migrate\_device: don't add folio to be freed to LRU in migrate\_device\_finalize()

If migration succeeded, we called folio\_migrate\_flags()->mem\_cgroup\_migrate() to migrate the memcg from the old to the new folio. This will set memcg\_data of the old folio to 0.

Similarly, if migration failed, memcg\_data of the dst folio is left unset.

If we call folio\_putback\_lru() on such folios (memcg\_data == 0), we will add the folio to be freed to the LRU, making memcg code unhappy. Running the hmm selftests:

```
# ./hmm-tests
...
# RUN      hmm.hmm_device_private.migrate ...
[ 102.078007][T14893] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x7ff27d200 pfn:0x13cc00
[ 102.079974][T14893] anon flags: 0x17ff00000020018(uptodate|dirty|swapbacked|node=0|zone=2|lastcpupid=0x7ff)
[ 102.082037][T14893] raw: 017ff00000020018 dead0000000000100 dead0000000000122 ffff8881353896c9
[ 102.083687][T14893] raw: 00000007ff27d200 0000000000000000 00000001fffffff 0000000000000000
[ 102.085331][T14893] page dumped because: VM_WARN_ON_ONCE_FOLIO(!memcg && !mem_cgroup_disabled())
[ 102.087230][T14893] -----[ cut here ]-----
[ 102.088279][T14893] WARNING: CPU: 0 PID: 14893 at ./include/linux/memcontrol.h:726
folio_lruvec_lock_irqsave+0x10e/0x170
[ 102.090478][T14893] Modules linked in:
[ 102.091244][T14893] CPU: 0 UID: 0 PID: 14893 Comm: hmm-tests Not tainted 6.13.0-09623-g6c216bc522fd #151
[ 102.093089][T14893] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014
[ 102.094848][T14893] RIP: 0010:folio_lruvec_lock_irqsave+0x10e/0x170
[ 102.096104][T14893] Code: ...
[ 102.099908][T14893] RSP: 0018:ffffc900236c37b0 EFLAGS: 00010293
[ 102.101152][T14893] RAX: 0000000000000000 RBX: ffffea0004f30000 RCX: ffffffff8183f426
[ 102.102684][T14893] RDX: ffff8881063cb880 RSI: ffffffff81b8117f RDI: ffff8881063cb880
[ 102.104227][T14893] RBP: 0000000000000000 R08: 0000000000000005 R09: 0000000000000000
[ 102.105757][T14893] R10: 0000000000000001 R11: 0000000000000002 R12: ffffc900236c37d8
```

```

[ 102.107296][T14893] R13: ffff888277a2bcb0 R14: 000000000000001f R15: 0000000000000000
[ 102.108830][T14893] FS: 00007ff27dbdd740(0000) GS:ffff888277a00000(0000) knlGS:0000000000000000
[ 102.110643][T14893] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 102.111924][T14893] CR2: 00007ff27d400000 CR3: 000000010866e000 CR4: 0000000000750ef0
[ 102.113478][T14893] PKRU: 55555554
[ 102.114172][T14893] Call Trace:
[ 102.114805][T14893] <TASK>
[ 102.115397][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170
[ 102.116547][T14893] ? __warn.cold+0x110/0x210
[ 102.117461][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170
[ 102.118667][T14893] ? report_bug+0x1b9/0x320
[ 102.119571][T14893] ? handle_bug+0x54/0x90
[ 102.120494][T14893] ? exc_invalid_op+0x17/0x50
[ 102.121433][T14893] ? asm_exc_invalid_op+0x1a/0x20
[ 102.122435][T14893] ? __wake_up_klogd.part.0+0x76/0xd0
[ 102.123506][T14893] ? dump_page+0x4f/0x60
[ 102.124352][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170
[ 102.125500][T14893] folio_batch_move_lru+0xd4/0x200
[ 102.126577][T14893] ? __pfx_lru_add+0x10/0x10
[ 102.127505][T14893] __folio_batch_add_and_move+0x391/0x720
[ 102.128633][T14893] ? __pfx_lru_add+0x10/0x10
[ 102.129550][T14893] folio_putback_lru+0x16/0x80
[ 102.130564][T14893] migrate_device_finalize+0x9b/0x530
[ 102.131640][T14893] dmirror_migrate_to_device.constprop.0+0x7c5/0xad0
[ 102.133047][T14893] dmirror_fops_unlocked_ioctl+0x89b/0xc80

```

Likely, nothing else goes wrong: putting the last folio reference will remove the folio from the LRU again. So besides memcg complaining, adding the folio to be freed to the LRU is just an unnecessary step.

The new flow resembles what we have in migrate\_folio\_move(): add the dst to the lru, rem  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2025-21861>

**[CVE-2025-21862] kernel: drop\_monitor: fix incorrect initialization order (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drop\_monitor: fix incorrect initialization order

Syzkaller reports the following bug:

```

BUG: spinlock bad magic on CPU#1, syz-executor.0/7995
lock: 0xffff88805303f3e0, .magic: 00000000, .owner: <none>/-1, .owner_cpu: 0
CPU: 1 PID: 7995 Comm: syz-executor.0 Tainted: G      E   5.10.209+ #1
Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020
Call Trace:
__dump_stack lib/dump_stack.c:77 [inline]

```

```

dump_stack+0x119/0x179 lib/dump_stack.c:118
debug_spin_lock_before kernel/locking/spinlock_debug.c:83 [inline]
do_raw_spin_lock+0x1f6/0x270 kernel/locking/spinlock_debug.c:112
__raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:117 [inline]
__raw_spin_lock_irqsave+0x50/0x70 kernel/locking/spinlock.c:159
reset_per_cpu_data+0xe6/0x240 [drop_monitor]
net_dm_cmd_trace+0x43d/0x17a0 [drop_monitor]
genl_family_rcv_msg_doit+0x22f/0x330 net/netlink/genetlink.c:739
genl_family_rcv_msg net/netlink/genetlink.c:783 [inline]
genl_rcv_msg+0x341/0x5a0 net/netlink/genetlink.c:800
netlink_rcv_skb+0x14d/0x440 net/netlink/af_netlink.c:2497
genl_rcv+0x29/0x40 net/netlink/genetlink.c:811
netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline]
netlink_unicast+0x54b/0x800 net/netlink/af_netlink.c:1348
netlink_sendmsg+0x914/0xe00 net/netlink/af_netlink.c:1916
sock_sendmsg_nosec net/socket.c:651 [inline]
__sock_sendmsg+0x157/0x190 net/socket.c:663
__sys_sendmsg+0x712/0x870 net/socket.c:2378
__sys_sendmsg+0xf8/0x170 net/socket.c:2432
__sys_sendmsg+0xea/0x1b0 net/socket.c:2461
do_syscall_64+0x30/0x40 arch/x86/entry/common.c:46
entry_SYSCALL_64_after_hwframe+0x62/0xc7
RIP: 0033:0x7f3f9815aee9
Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 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 b0 ff ff ff f7 d8 64 89 01 48
RSP: 002b:00007f3f972bf0c8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e
RAX: ffffffffda RBX: 00007f3f9826d050 RCX: 00007f3f9815aee9
RDX: 0000000020000000 RSI: 0000000020001300 RDI: 0000000000000007
RBP: 00007f3f981b63bd R08: 0000000000000000 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000
R13: 0000000000000006 R14: 00007f3f9826d050 R15: 00007ffe01ee6768

```

If `drop_monitor` is built as a kernel module, `syzkaller` may have time to send a netlink `NET_DM_CMD_START` message during the module loading. This will call the `net_dm_monitor_start()` function that uses a spinlock that has not yet been initialized.

To fix this, let's place resource initialization above the registration of a generic netlink family.

Found by InfoTeCS on behalf of Linux Verification Center ([linuxtesting.org](https://linuxtesting.org)) with Syzkaller.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21862>

**[CVE-2025-21864] kernel: tcp: drop secpath at the same time as we currently drop dst (Severity: MEDIUM)**

Package: linux-libc-dev  
 Installed: 6.1.129-1  
 Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

tcp: drop secpath at the same time as we currently drop dst

Xiumei reported hitting the WARN in xfrm6\_tunnel\_net\_exit while running tests that boil down to:

- create a pair of netns
- run a basic TCP test over ipcomp6
- delete the pair of netns

The xfrm\_state found on spi\_byaddr was not deleted at the time we delete the netns, because we still have a reference on it. This lingering reference comes from a secpath (which holds a ref on the xfrm\_state), which is still attached to an skb. This skb is not leaked, it ends up on sk\_receive\_queue and then gets defer-free'd by skb\_attempt\_defer\_free.

The problem happens when we defer freeing an skb (push it on one CPU's defer\_list), and don't flush that list before the netns is deleted. In that case, we still have a reference on the xfrm\_state that we don't expect at this point.

We already drop the skb's dst in the TCP receive path when it's no longer needed, so let's also drop the secpath. At this point, tcp\_filter has already called into the LSM hooks that may require the secpath, so it should not be needed anymore. However, in some of those places, the MPTCP extension has just been attached to the skb, so we cannot simply drop all extensions.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21864>

**[CVE-2025-21865] kernel: gtp: Suppress list corruption splat in gtp\_net\_exit\_batch\_rtnl(). (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

gtp: Suppress list corruption splat in gtp\_net\_exit\_batch\_rtnl().

Brad Spengler reported the list\_del() corruption splat in gtp\_net\_exit\_batch\_rtnl(). [0]

Commit eb28fd76c0a0 ("gtp: Destroy device along with udp socket's netns dismantle.") added the for\_each\_netdev() loop in gtp\_net\_exit\_batch\_rtnl() to destroy devices in each netns as done in geneve and ip tunnels.

However, this could trigger ->dellink() twice for the same device during ->exit\_batch\_rtnl().

Say we have two netns A & B and gtp device B that resides in netns B but whose UDP socket is in netns A.



1. cleanup\_net() processes netns A and then B.
2. gtp\_net\_exit\_batch\_rtnl() finds the device B while iterating netns A's gn->gtp\_dev\_list and calls ->dellink().

[ device B is not yet unlinked from netns B  
as unregister\_netdevice\_many() has not been called. ]

3. gtp\_net\_exit\_batch\_rtnl() finds the device B while iterating netns B's for\_each\_netdev() and calls ->dellink().

gtp\_dellink() cleans up the device's hash table, unlinks the dev from gn->gtp\_dev\_list, and calls unregister\_netdevice\_queue().

Basically, calling gtp\_dellink() multiple times is fine unless CONFIG\_DEBUG\_LIST is enabled.

Let's remove for\_each\_netdev() in gtp\_net\_exit\_batch\_rtnl() and delegate the destruction to default\_device\_exit\_batch() as done in bareudp.

[0]:  
list\_del corruption, ffff8880aaa62c00->next  
(autoslab\_size\_M\_dev\_P\_net\_core\_dev\_11127\_8\_1328\_8\_S\_4096\_A\_64\_n\_139+0xc00/0x1000 [slab object]) is  
LIST\_POISON1 (ffffffffffff02) (prev is 0xffffffffffff04)  
kernel BUG at lib/list\_debug.c:58!  
Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN  
CPU: 1 UID: 0 PID: 1804 Comm: kworker/u8:7 Tainted: G T 6.12.13-grsec-full-20250211091339 #1  
Tainted: [T]=RANDSTRUCT  
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014  
Workqueue: netns cleanup\_net  
RIP: 0010:[<ffffffff84947381>] \_\_list\_del\_entry\_valid\_or\_report+0x141/0x200 lib/list\_debug.c:58  
Code: c2 76 91 31 c0 e8 9f b1 f7 fc 0f 0b 4d 89 f0 48 c7 c1 02 ff ff 48 89 ea 48 89 ee 48 c7 c7 e0 c2 76 91 31 c0 e8 7f  
b1 f7 fc <0f> 0b 4d 89 e8 48 c7 c1 04 ff ff 48 89 ea 48 89 ee 48 c7 c7 60  
RSP: 0018:ffffe8040b4fbd0 EFLAGS: 00010283  
RAX: 0000000000000000cc RBX: dffffc0000000000 RCX: ffffffff818c4054  
RDX: ffffffff84947381 RSI: ffffffff818d1512 RDI: 0000000000000000  
RBP: ffff8880aaa62c00 R08: 0000000000000001 R09: fffffbd008169f32  
R10: fffffe8040b4f997 R11: 0000000000000001 R12: a1988d84f24943e4  
R13: ffffffffffff02 R14: ffffffffffff04 R15: ffff8880aaa62c08  
RBX: kasan shadow of 0x0  
RCX: \_\_wake\_up\_klogd.part.0+0x74/0xe0 kernel/printk/printk.c:4554  
RDX: \_\_list\_del\_entry\_valid\_or\_report+0x141/0x200 lib/list\_debug.c:58  
RSI: vprintk+0x72/0x100 kernel/printk/printk\_safe.c:71  
RBP: autoslab\_size\_M\_dev\_P\_net\_core\_dev\_11127\_8\_1328\_8\_S\_4096\_A\_64\_n\_139+0xc00/0x1000 [slab object]  
RSP: process kstack fffffe8040b4fbd0+0x7bd0/0x8000 [kworker/u8:7+netns 1804 ]  
R09: kasan shadow of process kstack fffffe8040b4f990+0x7990/0x8000 [kworker/u8:7+netns 1804 ]  
R10: process kstack fffffe8040b4f997+0x7997/0x8000 [kworker/u8:7+netns 1804 ]  
R15: autoslab\_size\_M\_dev\_P\_net\_core\_dev\_11127\_8\_1328\_8\_S\_4096\_A\_64\_n\_139+0xc08/0x1000 [slab object]  
FS: 0000000000000000(0000) GS:ffff888116000000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 0000748f5372c000 CR3: 0000000015408000 CR4: 00000000003406f0 shadow CR4: 00000000003406f0

Stack:

```
0000000000000000 ffffffff8a0c35e7 ffffffff8a0c3603 ffff8880aaa62c00
fff8880aaa62c00 0000000000000004 ffff88811145311c 0000000000000005
0000000000000001 ffff8880aaa62000 ffffe8040b4fd40 ffffffff8a0c360d
```

Call Trace:

<TASK>

```
[<fffffff8a0c360d>] __list_del_entry_valid include/linux/list.h:131 [inline] ffffe8040b4fc28
[<fffffff8a0c360d>] __list_del_entry include/linux/list.h:248 [inline] ffffe8040b4fc28
[<fffffff8a0c360d>] list_del include/linux/list.h:262 [inl
```

---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2025-21865>

## **[CVE-2025-21866] kernel: powerpc/code-patching: Fix KASAN hit by not flagging text patching area as VM\_ALLOC (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

powerpc/code-patching: Fix KASAN hit by not flagging text patching area as VM\_ALLOC

Erhard reported the following KASAN hit while booting his PowerMac G4 with a KASAN-enabled kernel 6.13-rc6:

BUG: KASAN: vmalloc-out-of-bounds in copy\_to\_kernel\_nofault+0xd8/0x1c8  
Write of size 8 at addr f1000000 by task chronyd/1293

CPU: 0 UID: 123 PID: 1293 Comm: chronyd Tainted: G W 6.13.0-rc6-PMacG4 #2

Tainted: [W]=WARN

Hardware name: PowerMac3,6 7455 0x80010303 PowerMac

Call Trace:

```
[c2437590] [c1631a84] dump_stack_lvl+0x70/0x8c (unreliable)
[c24375b0] [c0504998] print_report+0xdc/0x504
[c2437610] [c050475c] kasan_report+0xf8/0x108
[c2437690] [c0505a3c] kasan_check_range+0x24/0x18c
[c24376a0] [c03fb5e4] copy_to_kernel_nofault+0xd8/0x1c8
[c24376c0] [c004c014] patch_instructions+0x15c/0x16c
[c2437710] [c00731a8] bpf_arch_text_copy+0x60/0x7c
[c2437730] [c0281168] bpf_jit_binary_pack_finalize+0x50/0xac
[c2437750] [c0073cf4] bpf_int_jit_compile+0xb30/0xdec
[c2437880] [c0280394] bpf_prog_select_runtime+0x15c/0x478
[c24378d0] [c1263428] bpf_prepare_filter+0xbf8/0xc14
[c2437990] [c12677ec] bpf_prog_create_from_user+0x258/0x2b4
[c24379d0] [c027111c] do_seccomp+0x3dc/0x1890
[c2437ac0] [c001d8e0] system_call_exception+0x2dc/0x420
[c2437f30] [c00281ac] ret_from_syscall+0x0/0x2c
```

--- interrupt: c00 at 0x5a1274

NIP: 005a1274 LR: 006a3b3c CTR: 005296c8

REGS: c2437f40 TRAP: 0c00 Tainted: G W (6.13.0-rc6-PMacG4)

MSR: 0200f932 <VEC,EE,PR,FP,ME,IR,DR,RI> CR: 24004422 XER: 00000000

GPR00: 00000166 af8f3fa0 a7ee3540 00000001 00000000 013b6500 005a5858 0200f932  
GPR08: 00000000 00001fe9 013d5fc8 005296c8 2822244c 00b2fcd8 00000000 af8f4b57  
GPR16: 00000000 00000001 00000000 00000000 00000000 00000001 00000000 00000002  
GPR24: 00afdbb0 00000000 00000000 00000000 006e0004 013ce060 006e7c1c 00000001  
NIP [005a1274] 0x5a1274  
LR [006a3b3c] 0x6a3b3c  
--- interrupt: c00

The buggy address belongs to the virtual mapping at  
[f1000000, f1002000) created by:  
text\_area\_cpu\_up+0x20/0x190

The buggy address belongs to the physical page:  
page: refcount:1 mapcount:0 mapping:00000000 index:0x0 pfn:0x76e30  
flags: 0x80000000(zone=2)  
raw: 80000000 00000000 00000122 00000000 00000000 00000000 ffffffff 00000001  
raw: 00000000  
page dumped because: kasan: bad access detected

Memory state around the buggy address:  
f0fff00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
f0fff80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
>f1000000: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8  
          ^  
f1000080: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8  
f1000100: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8

f8 corresponds to KASAN\_VMAALLOC\_INVALID which means the area is not initialised hence not supposed to be used yet.

Powerpc text patching infrastructure allocates a virtual memory area using get\_vm\_area() and flags it as VM\_ALLOC. But that flag is meant to be used for vmalloc() and vmalloc() allocated memory is not supposed to be used before a call to \_\_vmalloc\_node\_range() which is never called for that area.

That went undetected until commit e4137f08816b ("mm, kasan, kmsan: instrument copy\_from/to\_kernel\_nofault")

The area allocated by text\_area\_cpu\_up() is not vmalloc memory, it is mapped directly on demand when needed by map\_kernel\_page(). There is no VM flag corresponding to such usage, so just pass no flag. That way the area will be unpoisonned and usable immediately.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21866>

**[CVE-2025-21867] kernel: bpf, test\_run: Fix use-after-free issue in eth\_skb\_pkt\_type() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

bpf, test\_run: Fix use-after-free issue in eth\_skb\_pkt\_type()

KMSAN reported a use-after-free issue in eth\_skb\_pkt\_type()[1]. The cause of the issue was that eth\_skb\_pkt\_type() accessed skb's data that didn't contain an Ethernet header. This occurs when bpf\_prog\_test\_run\_xdp() passes an invalid value as the user\_data argument to bpf\_test\_init().

Fix this by returning an error when user\_data is less than ETH\_HLEN in bpf\_test\_init(). Additionally, remove the check for "if (user\_size > size)" as it is unnecessary.

[1]

BUG: KMSAN: use-after-free in eth\_skb\_pkt\_type include/linux/etherdevice.h:627 [inline]

BUG: KMSAN: use-after-free in eth\_type\_trans+0x4ee/0x980 net/ethernet/eth.c:165

eth\_skb\_pkt\_type include/linux/etherdevice.h:627 [inline]

eth\_type\_trans+0x4ee/0x980 net/ethernet/eth.c:165

\_\_xdp\_build\_skb\_from\_frame+0x5a8/0xa50 net/core/xdp.c:635

xdp\_rcv\_frames net/bpf/test\_run.c:272 [inline]

xdp\_test\_run\_batch net/bpf/test\_run.c:361 [inline]

bpf\_test\_run\_xdp\_live+0x2954/0x3330 net/bpf/test\_run.c:390

bpf\_prog\_test\_run\_xdp+0x148e/0x1b10 net/bpf/test\_run.c:1318

bpf\_prog\_test\_run+0x5b7/0xa30 kernel/bpf/syscall.c:4371

\_\_sys\_bpf+0x6a6/0xe20 kernel/bpf/syscall.c:5777

\_\_do\_sys\_bpf kernel/bpf/syscall.c:5866 [inline]

\_\_se\_sys\_bpf kernel/bpf/syscall.c:5864 [inline]

\_\_x64\_sys\_bpf+0xa4/0xf0 kernel/bpf/syscall.c:5864

x64\_sys\_call+0x2ea0/0x3d90 arch/x86/include/generated/asm/syscalls\_64.h:322

do\_syscall\_x64 arch/x86/entry/common.c:52 [inline]

do\_syscall\_64+0xd9/0x1d0 arch/x86/entry/common.c:83

entry\_SYSCALL\_64\_after\_hwframe+0x77/0x7f

Uninit was created at:

free\_pages\_prepare mm/page\_alloc.c:1056 [inline]

free\_unref\_page+0x156/0x1320 mm/page\_alloc.c:2657

\_\_free\_pages+0xa3/0x1b0 mm/page\_alloc.c:4838

bpf\_ringbuf\_free kernel/bpf/ringbuf.c:226 [inline]

ringbuf\_map\_free+0xff/0x1e0 kernel/bpf/ringbuf.c:235

bpf\_map\_free kernel/bpf/syscall.c:838 [inline]

bpf\_map\_free\_deferred+0x17c/0x310 kernel/bpf/syscall.c:862

process\_one\_work kernel/workqueue.c:3229 [inline]

process\_scheduled\_works+0xa2b/0x1b60 kernel/workqueue.c:3310

worker\_thread+0xedf/0x1550 kernel/workqueue.c:3391

kthread+0x535/0x6b0 kernel/kthread.c:389

ret\_from\_fork+0x6e/0x90 arch/x86/kernel/process.c:147

ret\_from\_fork\_asm+0x1a/0x30 arch/x86/entry/entry\_64.S:244

CPU: 1 UID: 0 PID: 17276 Comm: syz.1.16450 Not tainted 6.12.0-05490-g9bb88c659673 #8

Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014

More Info: <https://avd.aquasec.com/nvd/cve-2025-21867>

**[CVE-2025-21870] kernel: ASoC: SOF: ipc4-topology: Harden loops for looking up ALH copiers (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ASoC: SOF: ipc4-topology: Harden loops for looking up ALH copiers

Other, non DAI copier widgets could have the same stream name (sname) as the ALH copier and in that case the copier->data is NULL, no alh\_data is attached, which could lead to NULL pointer dereference.

We could check for this NULL pointer in sof\_ipc4\_prepare\_copier\_module() and avoid the crash, but a similar loop in sof\_ipc4\_widget\_setup\_comp\_dai() will miscalculate the ALH device count, causing broken audio.

The correct fix is to harden the matching logic by making sure that the

1. widget is a DAI widget - so dai = w->private is valid
2. the dai (and thus the copier) is ALH copier

More Info: <https://avd.aquasec.com/nvd/cve-2025-21870>

**[CVE-2025-21871] kernel: tee: optee: Fix supplicant wait loop (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

tee: optee: Fix supplicant wait loop

OP-TEE supplicant is a user-space daemon and it's possible for it be hung or crashed or killed in the middle of processing an OP-TEE RPC call. It becomes more complicated when there is incorrect shutdown ordering of the supplicant process vs the OP-TEE client application which can eventually lead to system hang-up waiting for the closure of the client application.

Allow the client process waiting in kernel for supplicant response to be killed rather than indefinitely waiting in an unkillable state. Also, a normal uninterruptible wait should not have resulted in the hung-task watchdog getting triggered, but the endless loop would.

This fixes issues observed during system reboot/shutdown when supplicant got hung for some reason or gets crashed/killed which lead to client getting hung in an unkillable state. It in turn lead to system being in hung up state requiring hard power off/on to recover.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21871>

**[CVE-2025-21872] kernel: efi: Don't map the entire mokvar table to determine its size (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

efi: Don't map the entire mokvar table to determine its size

Currently, when validating the mokvar table, we (re)map the entire table on each iteration of the loop, adding space as we discover new entries.

If the table grows over a certain size, this fails due to limitations of `early_memmap()`, and we get a failure and traceback:

```
-----[ cut here ]-----
```

```
WARNING: CPU: 0 PID: 0 at mm/early_ioremap.c:139 __early_ioremap+0xef/0x220
```

```
...
```

```
Call Trace:
```

```
<TASK>
```

```
? __early_ioremap+0xef/0x220
```

```
? __warn.cold+0x93/0xfa
```

```
? __early_ioremap+0xef/0x220
```

```
? report_bug+0xff/0x140
```

```
? early_fixup_exception+0x5d/0xb0
```

```
? early_idt_handler_common+0x2f/0x3a
```

```
? __early_ioremap+0xef/0x220
```

```
? efi_mokvar_table_init+0xce/0x1d0
```

```
? setup_arch+0x864/0xc10
```

```
? start_kernel+0x6b/0xa10
```

```
? x86_64_start_reservations+0x24/0x30
```

```
? x86_64_start_kernel+0xed/0xf0
```

```
? common_startup_64+0x13e/0x141
```

```
</TASK>
```

```
---[ end trace 0000000000000000 ]---
```

```
mokvar: Failed to map EFI MOKvar config table pa=0x7c4c3000, size=265187.
```

Mapping the entire structure isn't actually necessary, as we don't ever need more than one entry header mapped at once.

Changes `efi_mokvar_table_init()` to only map each entry header, not the entire table, when determining the table size. Since we're not mapping any data past the variable name, it also changes the code to enforce that each variable name is NUL terminated, rather than attempting to verify it in place.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21872>

**[CVE-2025-21875] kernel: mptcp: always handle address removal under msk socket lock (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

mptcp: always handle address removal under msk socket lock

Syzkaller reported a lockdep splat in the PM control path:

```
WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 sock_owned_by_me include/net/sock.h:1711 [inline]
WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 msk_owned_by_me net/mptcp/protocol.h:363 [inline]
WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 mptcp_pm_nl_addr_send_ack+0x57c/0x610
net/mptcp/pm_netlink.c:788
Modules linked in:
CPU: 0 UID: 0 PID: 6693 Comm: syz.0.205 Not tainted 6.14.0-rc2-syzkaller-00303-gad1b832bf1cf #0
Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 12/27/2024
RIP: 0010:sock_owned_by_me include/net/sock.h:1711 [inline]
RIP: 0010:msk_owned_by_me net/mptcp/protocol.h:363 [inline]
RIP: 0010:mptcp_pm_nl_addr_send_ack+0x57c/0x610 net/mptcp/pm_netlink.c:788
Code: 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc e8 ca 7b d3 f5 eb b9 e8 c3 7b d3 f5 90 0f 0b 90 e9 dd fb ff ff e8 b5
7b d3 f5 90 <0f> 0b 90 e9 3e fb ff ff 44 89 f1 80 e1 07 38 c1 0f 8c eb fb ff ff
RSP: 0000:ffffc900034f6f60 EFLAGS: 00010283
RAX: ffffffff8bee3c2b RBX: 0000000000000001 RCX: 0000000000008000
RDX: ffff90004d42000 RSI: 000000000000a407 RDI: 000000000000a408
RBP: ffff900034f7030 R08: ffffffff8bee37f6 R09: 0100000000000000
R10: dffffc0000000000 R11: ffffed100bcc62e4 R12: ffff88805e6316e0
R13: ffff88805e630c00 R14: dffffc0000000000 R15: ffff88805e630c00
FS: 00007f7e9a7e96c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00000001b2fd18ff8 CR3: 0000000032c24000 CR4: 00000000003526f0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400
Call Trace:
<TASK>
mptcp_pm_remove_addr+0x103/0x1d0 net/mptcp/pm.c:59
mptcp_pm_remove_anno_addr+0x1f4/0x2f0 net/mptcp/pm_netlink.c:1486
mptcp_nl_remove_subflow_and_signal_addr net/mptcp/pm_netlink.c:1518 [inline]
mptcp_pm_nl_del_addr_doit+0x118d/0x1af0 net/mptcp/pm_netlink.c:1629
genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline]
genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline]
genl_rcv_msg+0xb1f/0xec0 net/netlink/genetlink.c:1210
netlink_rcv_skb+0x206/0x480 net/netlink/af_netlink.c:2543
genl_rcv+0x28/0x40 net/netlink/genetlink.c:1219
netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline]
netlink_unicast+0x7f6/0x990 net/netlink/af_netlink.c:1348
netlink_sendmsg+0x8de/0xcb0 net/netlink/af_netlink.c:1892
sock_sendmsg_nosec net/socket.c:718 [inline]
__sock_sendmsg+0x221/0x270 net/socket.c:733
__sys_sendmsg+0x53a/0x860 net/socket.c:2573
__sys_sendmsg net/socket.c:2627 [inline]
__sys_sendmsg+0x269/0x350 net/socket.c:2659
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: 0033:0x7f7e9998cde9
Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 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 a8 ff ff ff f7 d8 64 89 01 48
RSP: 002b:00007f7e9a7e9038 EFLAGS: 00000246 ORIG_RAX: 0000000000000002e
RAX: ffffffffda RBX: 00007f7e99ba5fa0 RCX: 00007f7e9998cde9
RDX: 000000002000c094 RSI: 0000400000000000 RDI: 0000000000000007
RBP: 00007f7e99a0e2a0 R08: 0000000000000000 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000
R13: 0000000000000000 R14: 00007f7e99ba5fa0 R15: 00007fff49231088
```

Indeed the PM can try to send a RM\_ADDR over a msk without acquiring first the msk socket lock.

The bugged code-path comes from an early optimization: when there are no subflows, the PM should (usually) not send RM\_ADDR notifications.

The above statement is incorrect, as without locks another process could concur  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2025-21875>

### **[CVE-2025-21877] kernel: usbnet: gl620a: fix endpoint checking in genelink\_bind() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usbnet: gl620a: fix endpoint checking in genelink\_bind()

Syzbot reports [1] a warning in usb\_submit\_urb() triggered by inconsistencies between expected and actually present endpoints in gl620a driver. Since genelink\_bind() does not properly verify whether specified eps are in fact provided by the device, in this case, an artificially manufactured one, one may get a mismatch.

Fix the issue by resorting to a usbnet utility function usbnet\_get\_endpoints(), usually reserved for this very problem. Check for endpoints and return early before proceeding further if any are missing.

[1] Syzbot report:

```
usb 5-1: Manufacturer: syz
usb 5-1: SerialNumber: syz
usb 5-1: config 0 descriptor??
gl620a 5-1:0.23 usb0: register 'gl620a' at usb-dummy_hcd.0-1, ...
```



-----[ cut here ]-----

usb 5-1: BOGUS urb xfer, pipe 3 != type 1

WARNING: CPU: 2 PID: 1841 at drivers/usb/core/urb.c:503 usb\_submit\_urb+0xe4b/0x1730 drivers/usb/core/urb.c:503

Modules linked in:

CPU: 2 UID: 0 PID: 1841 Comm: kworker/2:2 Not tainted 6.12.0-syzkaller-07834-g06afb0f36106 #0

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014

Workqueue: mld mld\_ifc\_work

RIP: 0010:usb\_submit\_urb+0xe4b/0x1730 drivers/usb/core/urb.c:503

...

Call Trace:

<TASK>

usbnet\_start\_xmit+0x6be/0x2780 drivers/net/usb/usbnet.c:1467

\_\_netdev\_start\_xmit include/linux/netdevice.h:5002 [inline]

netdev\_start\_xmit include/linux/netdevice.h:5011 [inline]

xmit\_one net/core/dev.c:3590 [inline]

dev\_hard\_start\_xmit+0x9a/0x7b0 net/core/dev.c:3606

sch\_direct\_xmit+0x1ae/0xc30 net/sched/sch\_generic.c:343

\_\_dev\_xmit\_skb net/core/dev.c:3827 [inline]

\_\_dev\_queue\_xmit+0x13d4/0x43e0 net/core/dev.c:4400

dev\_queue\_xmit include/linux/netdevice.h:3168 [inline]

neigh\_resolve\_output net/core/neighbour.c:1514 [inline]

neigh\_resolve\_output+0x5bc/0x950 net/core/neighbour.c:1494

neigh\_output include/net/neighbour.h:539 [inline]

ip6\_finish\_output2+0xb1b/0x2070 net/ipv6/ip6\_output.c:141

\_\_ip6\_finish\_output net/ipv6/ip6\_output.c:215 [inline]

ip6\_finish\_output+0x3f9/0x1360 net/ipv6/ip6\_output.c:226

NF\_HOOK\_COND include/linux/netfilter.h:303 [inline]

ip6\_output+0x1f8/0x540 net/ipv6/ip6\_output.c:247

dst\_output include/net/dst.h:450 [inline]

NF\_HOOK include/linux/netfilter.h:314 [inline]

NF\_HOOK include/linux/netfilter.h:308 [inline]

mld\_sendpack+0x9f0/0x11d0 net/ipv6/mcast.c:1819

mld\_send\_cr net/ipv6/mcast.c:2120 [inline]

mld\_ifc\_work+0x740/0xca0 net/ipv6/mcast.c:2651

process\_one\_work+0x9c5/0x1ba0 kernel/workqueue.c:3229

process\_scheduled\_works kernel/workqueue.c:3310 [inline]

worker\_thread+0x6c8/0xf00 kernel/workqueue.c:3391

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

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21877>

## **[CVE-2025-21878] kernel: i2c: npcm: disable interrupt enable bit before devm\_request\_irq (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

i2c: npcm: disable interrupt enable bit before devm\_request\_irq

The customer reports that there is a soft lockup issue related to the i2c driver. After checking, the i2c module was doing a tx transfer and the bmc machine reboots in the middle of the i2c transaction, the i2c module keeps the status without being reset.

Due to such an i2c module status, the i2c irq handler keeps getting triggered since the i2c irq handler is registered in the kernel booting process after the bmc machine is doing a warm rebooting. The continuous triggering is stopped by the soft lockup watchdog timer.

Disable the interrupt enable bit in the i2c module before calling devm\_request\_irq to fix this issue since the i2c relative status bit is read-only.

Here is the soft lockup log.

```
[ 28.176395] watchdog: BUG: soft lockup - CPU#0 stuck for 26s! [swapper/0:1]
[ 28.183351] Modules linked in:
[ 28.186407] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.15.120-yocto-s-dirty-bbebc78 #1
[ 28.201174] pstate: 40000005 (nZcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--)
[ 28.208128] pc : __do_softirq+0xb0/0x368
[ 28.212055] lr : __do_softirq+0x70/0x368
[ 28.215972] sp : ffffff8035ebca00
[ 28.219278] x29: ffffff8035ebca00 x28: 0000000000000002 x27: ffffff80071a3780
[ 28.226412] x26: ffffffc008bdc000 x25: ffffffc008bcc640 x24: ffffffc008be50c0
[ 28.233546] x23: ffffffc00800200c x22: 0000000000000000 x21: 000000000000001b
[ 28.240679] x20: 0000000000000000 x19: ffffff80001c3200 x18: ffffffff
[ 28.247812] x17: ffffffc02d2e0000 x16: ffffff8035eb8b40 x15: 00001e8480000000
[ 28.254945] x14: 02c3647e37dbfcb6 x13: 02c364f2ab14200c x12: 0000000002c364f2
[ 28.262078] x11: 00000000fa83b2da x10: 000000000000b67e x9 : ffffffc008010250
[ 28.269211] x8 : 000000009d983d00 x7 : 7fffffff x6 : 0000036d74732434
[ 28.276344] x5 : 00ffffffff x4 : 0000000000000015 x3 : 00000000000000198
[ 28.283476] x2 : ffffffc02d2e0000 x1 : 00000000000000e0 x0 : ffffffc008bdc40
[ 28.290611] Call trace:
[ 28.293052] __do_softirq+0xb0/0x368
[ 28.296625] __irq_exit_rcu+0xe0/0x100
[ 28.300374] irq_exit+0x14/0x20
[ 28.303513] handle_domain_irq+0x68/0x90
[ 28.307440] gic_handle_irq+0x78/0xb0
[ 28.311098] call_on_irq_stack+0x20/0x38
[ 28.315019] do_interrupt_handler+0x54/0x5c
[ 28.319199] el1_interrupt+0x2c/0x4c
[ 28.322777] el1h_64_irq_handler+0x14/0x20
[ 28.326872] el1h_64_irq+0x74/0x78
[ 28.330269] __setup_irq+0x454/0x780
[ 28.333841] request_threaded_irq+0xd0/0x1b4
[ 28.338107] devm_request_threaded_irq+0x84/0x100
[ 28.342809] npcm_i2c_probe_bus+0x188/0x3d0
[ 28.346990] platform_probe+0x6c/0xc4
[ 28.350653] really_probe+0xcc/0x45c
[ 28.354227] __driver_probe_device+0x8c/0x160
[ 28.358578] driver_probe_device+0x44/0xe0
```

```
[ 28.362670] __driver_attach+0x124/0x1d0
[ 28.366589] bus_for_each_dev+0x7c/0xe0
[ 28.370426] driver_attach+0x28/0x30
[ 28.373997] bus_add_driver+0x124/0x240
[ 28.377830] driver_register+0x7c/0x124
[ 28.381662] __platform_driver_register+0x2c/0x34
[ 28.386362] npcm_i2c_init+0x3c/0x5c
[ 28.389937] do_one_initcall+0x74/0x230
[ 28.393768] kernel_init_freeable+0x24c/0x2b4
[ 28.398126] kernel_init+0x28/0x130
[ 28.401614] ret_from_fork+0x10/0x20
[ 28.405189] Kernel panic - not syncing: softlockup: hung tasks
[ 28.411011] SMP: stopping secondary CPUs
[ 28.414933] Kernel Offset: disabled
[ 28.418412] CPU features: 0x00000000,00000802
[ 28.427644] Rebooting in 20 seconds..
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21878>

### **[CVE-2025-21881] kernel: uprobes: Reject the shared zeropage in uprobe\_write\_opcode() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

uprobes: Reject the shared zeropage in uprobe\_write\_opcode()

We triggered the following crash in syzkaller tests:

```
BUG: Bad page state in process syz.7.38 pfn:1eff3
page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x1eff3
flags: 0x3ffff00004004(referenced|reserved|node=0|zone=1|lastcpupid=0x1ffff)
raw: 003ffff00004004 ffffe6c6c07bfcc8 ffffe6c6c07bfcc8 0000000000000000
raw: 0000000000000000 0000000000000000 00000000fffffffe 0000000000000000
page dumped because: PAGE_FLAGS_CHECK_AT_FREE flag(s) set
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014
Call Trace:
<TASK>
dump_stack_lvl+0x32/0x50
bad_page+0x69/0xf0
free_unref_page_prepare+0x401/0x500
free_unref_page+0x6d/0x1b0
uprobe_write_opcode+0x460/0x8e0
install_breakpoint.part.0+0x51/0x80
register_for_each_vma+0x1d9/0x2b0
__uprobe_register+0x245/0x300
bpf_uprobe_multi_link_attach+0x29b/0x4f0
link_create+0x1e2/0x280
__sys_bpf+0x75f/0xac0
__x64_sys_bpf+0x1a/0x30
```

```
do_syscall_64+0x56/0x100
entry_SYSCALL_64_after_hwframe+0x78/0xe2
```

BUG: Bad rss-counter state mm:00000000452453e0 type:MM\_FILEPAGES val:-1

The following syzkaller test case can be used to reproduce:

```
r2 = creat(&(0x7f0000000000)='./file0\x00', 0x8)
write$nbfd(r2, &(0x7f0000000580)=ANY=[], 0x10)
r4 = openat(0xffffffffffff9c, &(0x7f0000000040)='./file0\x00', 0x42, 0x0)
mmap$IORING_OFF_SQ_RING(&(0x7f0000ffd000/0x3000)=nil, 0x3000, 0x0, 0x12, r4, 0x0)
r5 = userfaultfd(0x80801)
ioctl$UFFDIO_API(r5, 0xc018aa3f, &(0x7f0000000040)={0xaa, 0x20})
r6 = userfaultfd(0x80801)
ioctl$UFFDIO_API(r6, 0xc018aa3f, &(0x7f0000000140))
ioctl$UFFDIO_REGISTER(r6, 0xc020aa00, &(0x7f0000000100)={{&(0x7f0000ffc000/0x4000)=nil, 0x4000}, 0x2})
ioctl$UFFDIO_ZEROPAGE(r5, 0xc020aa04, &(0x7f0000000000)={{&(0x7f0000ffd000/0x1000)=nil, 0x1000}})
r7 = bpf$PROG_LOAD(0x5, &(0x7f0000000140)={0x2, 0x3,
&(0x7f0000000200)=ANY=[@ANYBLOB="180000000012000000000000000000000095"], &(0x7f0000000000)='GPL\x00',
0x7, 0x0, 0x0, 0x0, 0x0, 'x00', 0x0, @fallback=0x30, 0xffffffffffff, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
0x10, 0x0, @void, @value}, 0x94)
bpf$BPF_LINK_CREATE_XDP(0x1c, &(0x7f0000000040)={r7, 0x0, 0x30, 0x1e,
@val=@uprobe_multi={&(0x7f0000000080)='./file0\x00', &(0x7f0000000100)=[0x2], 0x0, 0x0, 0x1}}, 0x40)
```

The cause is that zero pfn is set to the PTE without increasing the RSS count in `mfill_atomic_pte_zeropage()` and the refcount of zero folio does not increase accordingly. Then, the operation on the same pfn is performed in `uprobe_write_opcode()->__replace_page()` to unconditional decrease the RSS count and old\_folio's refcount.

Therefore, two bugs are introduced:

1. The RSS count is incorrect, when process exit, the `check_mm()` report error "Bad rss-count".
2. The reserved folio (zero folio) is freed when `folio->refcount` is zero, then `free_pages_prepare->free_page_is_bad()` report error "Bad page state".

There is more, the following warning could also theoretically be triggered:

```
__replace_page()
-> ...
-> folio_remove_rmap_pte()
-> VM_WARN_ON_FOLIO(is_zero_folio(folio), folio)
```

Considering that uprobe hit on the zero folio is a very rare case, just reject zero old folio immediately after `get_user_page_vma_remote()`.

[ mingo: Cleaned up the changelog ]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21881>

**[CVE-2025-21885] kernel: RDMA/bnxt\_re: Fix the page details for the srq created by kernel consumers**  
**(Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/bnxt\_re: Fix the page details for the srq created by kernel consumers

While using nvme target with use\_srq on, below kernel panic is noticed.

```
[ 549.698111] bnxt_en 0000:41:00.0 enp65s0np0: FEC autoneg off encoding: Clause 91 RS(544,514)
[ 566.393619] Oops: divide error: 0000 [#1] PREEMPT SMP NOPTI
..
[ 566.393799] <TASK>
[ 566.393807] ? __die_body+0x1a/0x60
[ 566.393823] ? die+0x38/0x60
[ 566.393835] ? do_trap+0xe4/0x110
[ 566.393847] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re]
[ 566.393867] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re]
[ 566.393881] ? do_error_trap+0x7c/0x120
[ 566.393890] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re]
[ 566.393911] ? exc_divide_error+0x34/0x50
[ 566.393923] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re]
[ 566.393939] ? asm_exc_divide_error+0x16/0x20
[ 566.393966] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re]
[ 566.393997] bnxt_qplib_create_srq+0xc9/0x340 [bnxt_re]
[ 566.394040] bnxt_re_create_srq+0x335/0x3b0 [bnxt_re]
[ 566.394057] ? srso_return_thunk+0x5/0x5f
[ 566.394068] ? __init_swait_queue_head+0x4a/0x60
[ 566.394090] ib_create_srq_user+0xa7/0x150 [ib_core]
[ 566.394147] nvmet_rdma_queue_connect+0x7d0/0xbe0 [nvmet_rdma]
[ 566.394174] ? lock_release+0x22c/0x3f0
[ 566.394187] ? srso_return_thunk+0x5/0x5f
```

Page size and shift info is set only for the user space SRQs.  
Set page size and page shift for kernel space SRQs also.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21885>

**[CVE-2025-21887] kernel: ovl: fix UAF in ovl\_dentry\_update\_reval by moving dput() in ovl\_link\_up**  
**(Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ovl: fix UAF in ovl\_dentry\_update\_reval by moving dput() in ovl\_link\_up

The issue was caused by dput(upper) being called before

ovl\_dentry\_update\_reval(), while upper->d\_flags was still accessed in ovl\_dentry\_remote().

Move dput(upper) after its last use to prevent use-after-free.

BUG: KASAN: slab-use-after-free in ovl\_dentry\_remote fs/overlayfs/util.c:162 [inline]

BUG: KASAN: slab-use-after-free in ovl\_dentry\_update\_reval+0xd2/0xf0 fs/overlayfs/util.c:167

Call Trace:

<TASK>

\_\_dump\_stack lib/dump\_stack.c:88 [inline]  
dump\_stack\_lvl+0x116/0x1f0 lib/dump\_stack.c:114  
print\_address\_description mm/kasan/report.c:377 [inline]  
print\_report+0xc3/0x620 mm/kasan/report.c:488  
kasan\_report+0xd9/0x110 mm/kasan/report.c:601  
ovl\_dentry\_remote fs/overlayfs/util.c:162 [inline]  
ovl\_dentry\_update\_reval+0xd2/0xf0 fs/overlayfs/util.c:167  
ovl\_link\_up fs/overlayfs/copy\_up.c:610 [inline]  
ovl\_copy\_up\_one+0x2105/0x3490 fs/overlayfs/copy\_up.c:1170  
ovl\_copy\_up\_flags+0x18d/0x200 fs/overlayfs/copy\_up.c:1223  
ovl\_rename+0x39e/0x18c0 fs/overlayfs/dir.c:1136  
vfs\_rename+0xf84/0x20a0 fs/namei.c:4893

...

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21887>

### **[CVE-2025-21888] kernel: RDMA/mlx5: Fix a WARN during dereg\_mr for DM type (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/mlx5: Fix a WARN during dereg\_mr for DM type

Memory regions (MR) of type DM (device memory) do not have an associated umem.

In the \_\_mlx5\_ib\_dereg\_mr() -> mlx5\_free\_priv\_descs() flow, the code incorrectly takes the wrong branch, attempting to call dma\_unmap\_single() on a DMA address that is not mapped.

This results in a WARN [1], as shown below.

The issue is resolved by properly accounting for the DM type and ensuring the correct branch is selected in mlx5\_free\_priv\_descs().

[1]

WARNING: CPU: 12 PID: 1346 at drivers/iommu/dma-iommu.c:1230 iommu\_dma\_unmap\_page+0x79/0x90  
Modules linked in: ip6table\_mangle ip6table\_nat ip6table\_filter ip6\_tables iptable\_mangle xt\_conntrack xt\_MASQUERADE nf\_conntrack\_netlink nfnetlink xt\_addrtype iptable\_nat nf\_nat br\_netfilter rpcsec\_gss\_krb5 auth\_rpcgss oid\_registry ovelay rprdma rdma\_ucm ib\_iser libiscsi scsi\_transport\_iscsi ib\_umad rdma\_cm ib\_ipoib

iw\_cm ib\_cm mlx5\_ib ib\_uverbs ib\_core fuse mlx5\_core  
CPU: 12 UID: 0 PID: 1346 Comm: ibv\_rc\_pingpong Not tainted 6.12.0-rc7+ #1631  
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org  
04/01/2014  
RIP: 0010:iommu\_dma\_unmap\_page+0x79/0x90  
Code: 2b 49 3b 29 72 26 49 3b 69 08 73 20 4d 89 f0 44 89 e9 4c 89 e2 48 89 ee 48 89 df 5b 5d 41 5c 41 5d 41 5e 41 5f  
e9 07 b8 88 ff <0f> 0b 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc 66 0f 1f 44 00  
RSP: 0018:ffff90001913a10 EFLAGS: 00010246  
RAX: 0000000000000000 RBX: ffff88810194b0a8 RCX: 0000000000000000  
RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000001  
RBP: ffff88810194b0a8 R08: 0000000000000000 R09: 0000000000000000  
R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000000  
R13: 0000000000000001 R14: 0000000000000000 R15: 0000000000000000  
FS: 00007f537abdd740(0000) GS:ffff8885fb000000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 00007f537aeb8000 CR3: 000000010c248001 CR4: 0000000000372eb0  
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000  
DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400  
Call Trace:  
<TASK>  
? \_\_warn+0x84/0x190  
? iommu\_dma\_unmap\_page+0x79/0x90  
? report\_bug+0xf8/0x1c0  
? handle\_bug+0x55/0x90  
? exc\_invalid\_op+0x13/0x60  
? asm\_exc\_invalid\_op+0x16/0x20  
? iommu\_dma\_unmap\_page+0x79/0x90  
dma\_unmap\_page\_attrs+0xe6/0x290  
mlx5\_free\_priv\_descs+0xb0/0xe0 [mlx5\_ib]  
\_\_mlx5\_ib\_dereg\_mr+0x37e/0x520 [mlx5\_ib]  
? \_raw\_spin\_unlock\_irq+0x24/0x40  
? wait\_for\_completion+0xfe/0x130  
? rdma\_restrack\_put+0x63/0xe0 [ib\_core]  
ib\_dereg\_mr\_user+0x5f/0x120 [ib\_core]  
? lock\_release+0xc6/0x280  
destroy\_hw\_idr\_uobject+0x1d/0x60 [ib\_uverbs]  
uverbs\_destroy\_uobject+0x58/0x1d0 [ib\_uverbs]  
uobj\_destroy+0x3f/0x70 [ib\_uverbs]  
ib\_uverbs\_cmd\_verbs+0x3e4/0xbb0 [ib\_uverbs]  
? \_\_pfx\_uverbs\_destroy\_def\_handler+0x10/0x10 [ib\_uverbs]  
? lock\_acquire+0xc1/0x2f0  
? ib\_uverbs\_ioctl+0xcb/0x170 [ib\_uverbs]  
? ib\_uverbs\_ioctl+0x116/0x170 [ib\_uverbs]  
? lock\_release+0xc6/0x280  
ib\_uverbs\_ioctl+0xe7/0x170 [ib\_uverbs]  
? ib\_uverbs\_ioctl+0xcb/0x170 [ib\_uverbs]  
\_\_x64\_sys\_ioctl+0x1b0/0xa70  
do\_syscall\_64+0x6b/0x140  
entry\_SYSCALL\_64\_after\_hwframe+0x76/0x7e  
RIP: 0033:0x7f537adaf17b  
Code: 0f 1e fa 48 8b 05 1d ad 0c 00 64 c7 00 26 00 00 00 48 c7 c0 ff ff ff c3 66 0f 1f 44 00 00 f3 0f 1e fa b8 10 00 00  
00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d ed ac 0c 00 f7 d8 64 89 01 48  
RSP: 002b:00007ffff218f0b8 EFLAGS: 00000246 ORIG\_RAX: 0000000000000010

RAX: ffffffffda RBX: 00007ffff218f1d8 RCX: 00007f537adaf17b  
RDX: 00007ffff218f1c0 RSI: 00000000c0181b01 RDI: 0000000000000003  
RBP: 00007ffff218f1a0 R08: 00007f537aa8d010 R09: 0000561ee2e4f270  
R10: 00007f537aace3a8 R11: 00000000000000246 R12: 00007ffff218f190  
R13: 0000000000000001c R14: 0000561ee2e4d7c0 R15: 00007ffff218f450  
</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21888>

### **[CVE-2025-21891] kernel: ipvlan: ensure network headers are in skb linear part (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ipvlan: ensure network headers are in skb linear part

syzbot found that `ipvlan_process_v6_outbound()` was assuming the IPv6 network header is present in `skb->head` [1]

Add the needed `pskb_network_may_pull()` calls for both IPv4 and IPv6 handlers.

[1]  
BUG: KMSAN: uninit-value in `__ipv6_addr_type+0xa2/0x490` net/ipv6/addrconf\_core.c:47  
`__ipv6_addr_type+0xa2/0x490` net/ipv6/addrconf\_core.c:47  
`ipv6_addr_type` include/net/ipv6.h:555 [inline]  
`ip6_route_output_flags_noref` net/ipv6/route.c:2616 [inline]  
`ip6_route_output_flags+0x51/0x720` net/ipv6/route.c:2651  
`ip6_route_output` include/net/ip6\_route.h:93 [inline]  
`ipvlan_route_v6_outbound+0x24e/0x520` drivers/net/ipvlan/ipvlan\_core.c:476  
`ipvlan_process_v6_outbound` drivers/net/ipvlan/ipvlan\_core.c:491 [inline]  
`ipvlan_process_outbound` drivers/net/ipvlan/ipvlan\_core.c:541 [inline]  
`ipvlan_xmit_mode_l3` drivers/net/ipvlan/ipvlan\_core.c:605 [inline]  
`ipvlan_queue_xmit+0xd72/0x1780` drivers/net/ipvlan/ipvlan\_core.c:671  
`ipvlan_start_xmit+0x5b/0x210` drivers/net/ipvlan/ipvlan\_main.c:223  
`__netdev_start_xmit` include/linux/netdevice.h:5150 [inline]  
`netdev_start_xmit` include/linux/netdevice.h:5159 [inline]  
`xmit_one` net/core/dev.c:3735 [inline]  
`dev_hard_start_xmit+0x247/0xa20` net/core/dev.c:3751  
`sch_direct_xmit+0x399/0xd40` net/sched/sch\_generic.c:343  
`qdisc_restart` net/sched/sch\_generic.c:408 [inline]  
`__qdisc_run+0x14da/0x35d0` net/sched/sch\_generic.c:416  
`qdisc_run+0x141/0x4d0` include/net/pkt\_sched.h:127  
`net_tx_action+0x78b/0x940` net/core/dev.c:5484  
`handle_softirqs+0x1a0/0x7c0` kernel/softirq.c:561  
`__do_softirq+0x14/0x1a` kernel/softirq.c:595  
`do_softirq+0x9a/0x100` kernel/softirq.c:462  
`__local_bh_enable_ip+0x9f/0xb0` kernel/softirq.c:389  
`local_bh_enable` include/linux/bottom\_half.h:33 [inline]  
`rcu_read_unlock_bh` include/linux/rcupdate.h:919 [inline]  
`__dev_queue_xmit+0x2758/0x57d0` net/core/dev.c:4611



```
dev_queue_xmit include/linux/netdevice.h:3311 [inline]
packet_xmit+0x9c/0x6c0 net/packet/af_packet.c:276
packet_snd net/packet/af_packet.c:3132 [inline]
packet_sendmsg+0x93e0/0xa7e0 net/packet/af_packet.c:3164
sock_sendmsg_nosec net/socket.c:718 [inline]
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21891>

## **[CVE-2025-21892] kernel: RDMA/mlx5: Fix the recovery flow of the UMR QP (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

RDMA/mlx5: Fix the recovery flow of the UMR QP

This patch addresses an issue in the recovery flow of the UMR QP, ensuring tasks do not get stuck, as highlighted by the call trace [1].

During recovery, before transitioning the QP to the RESET state, the software must wait for all outstanding WRs to complete.

Failing to do so can cause the firmware to skip sending some flushed CQEs with errors and simply discard them upon the RESET, as per the IB specification.

This race condition can result in lost CQEs and tasks becoming stuck.

To resolve this, the patch sends a final WR which serves only as a barrier before moving the QP state to RESET.

Once a CQE is received for that final WR, it guarantees that no outstanding WRs remain, making it safe to transition the QP to RESET and subsequently back to RTS, restoring proper functionality.

Note:

For the barrier WR, we simply reuse the failed and ready WR.

Since the QP is in an error state, it will only receive

IB\_WC\_WR\_FLUSH\_ERR. However, as it serves only as a barrier we don't care about its status.

[1]

INFO: task rdma\_resource\_l:1922 blocked for more than 120 seconds.

Tainted: G W 6.12.0-rc7+ #1626

"echo 0 > /proc/sys/kernel/hung\_task\_timeout\_secs" disables this message.

task:rdma\_resource\_l state:D stack:0 pid:1922 tgid:1922 ppid:1369

flags:0x00004004

Call Trace:

<TASK>

\_\_schedule+0x420/0xd30

schedule+0x47/0x130

schedule\_timeout+0x280/0x300

```

? mark_held_locks+0x48/0x80
? lockdep_hardirqs_on_prepare+0xe5/0x1a0
wait_for_completion+0x75/0x130
mlx5r_umr_post_send_wait+0x3c2/0x5b0 [mlx5_ib]
? __pfx_mlx5r_umr_done+0x10/0x10 [mlx5_ib]
mlx5r_umr_revoke_mr+0x93/0xc0 [mlx5_ib]
__mlx5_ib_dereg_mr+0x299/0x520 [mlx5_ib]
? _raw_spin_unlock_irq+0x24/0x40
? wait_for_completion+0xfe/0x130
? rdma_restrack_put+0x63/0xe0 [ib_core]
ib_dereg_mr_user+0x5f/0x120 [ib_core]
? lock_release+0xc6/0x280
destroy_hw_idr_uobject+0x1d/0x60 [ib_uverbs]
uverbs_destroy_uobject+0x58/0x1d0 [ib_uverbs]
uobj_destroy+0x3f/0x70 [ib_uverbs]
ib_uverbs_cmd_verbs+0x3e4/0xbb0 [ib_uverbs]
? __pfx_uverbs_destroy_def_handler+0x10/0x10 [ib_uverbs]
? __lock_acquire+0x64e/0x2080
? mark_held_locks+0x48/0x80
? find_held_lock+0x2d/0xa0
? lock_acquire+0xc1/0x2f0
? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs]
? __fget_files+0xc3/0x1b0
ib_uverbs_ioctl+0xe7/0x170 [ib_uverbs]
? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs]
__x64_sys_ioctl+0x1b0/0xa70
do_syscall_64+0x6b/0x140
entry_SYSCALL_64_after_hwframe+0x76/0x7e
RIP: 0033:0x7f99c918b17b
RSP: 002b:00007ffc766d0468 EFLAGS: 00000246 ORIG_RAX:
0000000000000010
RAX: ffffffffda RBX: 00007ffc766d0578 RCX:
00007f99c918b17b
RDX: 00007ffc766d0560 RSI: 00000000c0181b01 RDI:
0000000000000003
RBP: 00007ffc766d0540 R08: 00007f99c8f99010 R09:
000000000000bd7e
R10: 00007f99c94c1c70 R11: 0000000000000246 R12:
00007ffc766d0530
R13: 000000000000001c R14: 0000000040246a80 R15:
0000000000000000
</TASK>

```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21892>

### **[CVE-2025-21894] kernel: net: enetc: VFs do not support HWTSTAMP\_TX\_ONESTEP\_SYNC (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: enetc: VFs do not support HWTSTAMP\_TX\_ONESTEP\_SYNC

Actually ENETC VFs do not support HWTSTAMP\_TX\_ONESTEP\_SYNC because only ENETC PF can access PMA\_SINGLE\_STEP registers. And there will be a crash if VFs are used to test one-step timestamp, the crash log as follows.

```
[ 129.110909] Unable to handle kernel paging request at virtual address 000000000000080c0
[ 129.287769] Call trace:
[ 129.290219] enetc_port_mac_wr+0x30/0xec (P)
[ 129.294504] enetc_start_xmit+0xda4/0xe74
[ 129.298525] enetc_xmit+0x70/0xec
[ 129.301848] dev_hard_start_xmit+0x98/0x118
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21894>

### **[CVE-2025-21898] kernel: ftrace: Avoid potential division by zero in function\_stat\_show() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ftrace: Avoid potential division by zero in function\_stat\_show()

Check whether denominator expression  $x * (x - 1) * 1000 \bmod \{2^{32}, 2^{64}\}$  produce zero and skip stddev computation in that case.

For now don't care about  $\text{rec} \rightarrow \text{counter} * \text{rec} \rightarrow \text{counter}$  overflow because  $\text{rec} \rightarrow \text{time} * \text{rec} \rightarrow \text{time}$  overflow will likely happen earlier.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21898>

### **[CVE-2025-21899] kernel: tracing: Fix bad hist from corrupting named\_triggers list (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

tracing: Fix bad hist from corrupting named\_triggers list

The following commands causes a crash:

```
~# cd /sys/kernel/tracing/events/rcu/rcu_callback
~# echo 'hist:name=bad:keys=common_pid:onmax(bogus).save(common_pid)' > trigger
bash: echo: write error: Invalid argument
~# echo 'hist:name=bad:keys=common_pid' > trigger
```

Because the following occurs:

```

event_trigger_write() {
    trigger_process_regex() {
        event_hist_trigger_parse() {

            data = event_trigger_alloc(..);

            event_trigger_register(.., data) {
                cmd_ops->reg(.., data, ..) [hist_register_trigger()] {
                    data->ops->init() [event_hist_trigger_init()] {
                        save_named_trigger(name, data) {
                            list_add(&data->named_list, &named_triggers);
                        }
                    }
                }
            }

            ret = create_actions(); (return -EINVAL)
            if (ret)
                goto out_unreg;
[.]
            ret = hist_trigger_enable(data, ...) {
                list_add_tail_rcu(&data->list, &file->triggers); <<<---- SKIPPED!!! (this is important!)
[.]
            out_unreg:
                event_hist_unregister(.., data) {
                    cmd_ops->unreg(.., data, ..) [hist_unregister_trigger()] {
                        list_for_each_entry(iter, &file->triggers, list) {
                            if (!hist_trigger_match(data, iter, named_data, false)) <- never matches
                                continue;
                            [.]
                            test = iter;
                        }
                        if (test && test->ops->free) <<<-- test is NULL

                            test->ops->free(test) [event_hist_trigger_free()] {
                                [.]
                                if (data->name)
                                    del_named_trigger(data) {
                                        list_del(&data->named_list); <<<<-- NEVER gets removed!
                                    }
                                }
                            }
                        }

                        [.]
                        kfree(data); <<<-- frees item but it is still on list

```

The next time a hist with name is registered, it causes an u-a-f bug and the kernel can crash.

Move the code around such that if event\_trigger\_register() succeeds, the next thing called is hist\_trigger\_enable() which adds it to the list.

A bunch of actions is called if `get_named_trigger_data()` returns false. But that doesn't need to be called after `event_trigger_register()`, so it can be moved up, allowing `event_trigger_register()` to be called just before `hist_trigger_enable()` keeping them together and allowing the file->triggers to be properly populated.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21899>

**[CVE-2025-21904] kernel: caif\_virtio: fix wrong pointer check in cfv\_probe() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

caif\_virtio: fix wrong pointer check in cfv\_probe()

`del_vqs()` frees virtqueues, therefore `cfv->vq_tx` pointer should be checked for NULL before calling it, not `cfv->vdev`. Also the current implementation is redundant because the pointer `cfv->vdev` is dereferenced before it is checked for NULL.

Fix this by checking `cfv->vq_tx` for NULL instead of `cfv->vdev` before calling `del_vqs()`.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21904>

**[CVE-2025-21905] kernel: wifi: iwlwifi: limit printed string from FW file (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

wifi: iwlwifi: limit printed string from FW file

There's no guarantee here that the file is always with a NUL-termination, so reading the string may read beyond the end of the TLV. If that's the last TLV in the file, it can perhaps even read beyond the end of the file buffer.

Fix that by limiting the print format to the size of the buffer we have.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21905>

**[CVE-2025-21907] kernel: mm: memory-failure: update ttu flag inside unmap\_poisoned\_folio (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

mm: memory-failure: update ttu flag inside unmap\_poisoned\_folio

Patch series "mm: memory\_failure: unmap poisoned folio during migrate properly", v3.

Fix two bugs during folio migration if the folio is poisoned.

This patch (of 3):

Commit 6da6b1d4a7df ("mm/hwpoison: convert TTU\_IGNORE\_HWPOISON to TTU\_HWPOISON") introduce TTU\_HWPOISON to replace TTU\_IGNORE\_HWPOISON in order to stop send SIGBUS signal when accessing an error page after a memory error on a clean folio. However during page migration, anon folio must be set with TTU\_HWPOISON during unmap\_\*. For pagecache we need some policy just like the one in hwpoison\_user\_mappings to set this flag. So move this policy from hwpoison\_user\_mappings to unmap\_poisoned\_folio to handle this warning properly.

Warning will be produced during unmap poison folio with the following log:

-----[ cut here ]-----

WARNING: CPU: 1 PID: 365 at mm/rmap.c:1847 try\_to\_unmap\_one+0x8fc/0xd3c

Modules linked in:

CPU: 1 UID: 0 PID: 365 Comm: bash Tainted: G W 6.13.0-rc1-00018-gacdb4bbda7ab #42

Tainted: [W]=WARN

Hardware name: QEMU QEMU Virtual Machine, BIOS 0.0.0 02/06/2015

pstate: 20400005 (nzCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--)

pc : try\_to\_unmap\_one+0x8fc/0xd3c

lr : try\_to\_unmap\_one+0x3dc/0xd3c

Call trace:

try\_to\_unmap\_one+0x8fc/0xd3c (P)

try\_to\_unmap\_one+0x3dc/0xd3c (L)

rmap\_walk\_anon+0xdc/0x1f8

rmap\_walk+0x3c/0x58

try\_to\_unmap+0x88/0x90

unmap\_poisoned\_folio+0x30/0xa8

do\_migrate\_range+0x4a0/0x568

offline\_pages+0x5a4/0x670

memory\_block\_action+0x17c/0x374

memory\_subsys\_offline+0x3c/0x78

device\_offline+0xa4/0xd0

state\_store+0x8c/0xf0

dev\_attr\_store+0x18/0x2c

sysfs\_kf\_write+0x44/0x54

kernfs\_fop\_write\_iter+0x118/0x1a8

vfs\_write+0x3a8/0x4bc

ksys\_write+0x6c/0xf8

\_\_arm64\_sys\_write+0x1c/0x28

invoke\_syscall+0x44/0x100

```
el0_svc_common.constprop.0+0x40/0xe0
do_el0_svc+0x1c/0x28
el0_svc+0x30/0xd0
el0t_64_sync_handler+0xc8/0xcc
el0t_64_sync+0x198/0x19c
---[ end trace 0000000000000000 ]---
```

[mawupeng1@huawei.com: unmap\_poisoned\_folio(): remove shadowed local `mapping`, per Miaohe]  
Link: <https://lkml.kernel.org/r/20250219060653.3849083-1-mawupeng1@huawei.com>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21907>

**[CVE-2025-21909] kernel: wifi: nl80211: reject cooked mode if it is set along with other flags (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

wifi: nl80211: reject cooked mode if it is set along with other flags

It is possible to set both MONITOR\_FLAG\_COOK\_FRAMES and MONITOR\_FLAG\_ACTIVE flags simultaneously on the same monitor interface from the userspace. This causes a sub-interface to be created with no IEEE80211\_SDATA\_IN\_DRIVER bit set because the monitor interface is in the cooked state and it takes precedence over all other states. When the interface is then being deleted the kernel calls WARN\_ONCE() from check\_sdata\_in\_driver() because of missing that bit.

Fix this by rejecting MONITOR\_FLAG\_COOK\_FRAMES if it is set along with other flags.

Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21909>

**[CVE-2025-21910] kernel: wifi: cfg80211: regulatory: improve invalid hints checking (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

wifi: cfg80211: regulatory: improve invalid hints checking

Syzbot keeps reporting an issue [1] that occurs when erroneous symbols sent from userspace get through into user\_alpha2[] via regulatory\_hint\_user() call. Such invalid regulatory hints should be rejected.

While a sanity check from commit 47caf685a685 ("cfg80211: regulatory: reject invalid hints") looks to be enough to deter these very cases, there is a way to get around it due to 2 reasons.

1) The way `isalpha()` works, symbols other than latin lower and upper letters may be used to determine a country/domain. For instance, greek letters will also be considered upper/lower letters and for such characters `isalpha()` will return true as well. However, ISO-3166-1 alpha2 codes should only hold latin characters.

2) While processing a user regulatory request, between `reg_process_hint_user()` and `regulatory_hint_user()` there happens to be a call to `queue_regulatory_request()` which modifies letters in `request->alpha2[]` with `toupper()`. This works fine for latin symbols, less so for weird letter characters from the second part of `_ctype[]`.

Syzbot triggers a warning in `is_user_regdom_saved()` by first sending over an unexpected non-latin letter that gets malformed by `toupper()` into a character that ends up failing `isalpha()` check.

Prevent this by enhancing `is_an_alpha2()` to ensure that incoming symbols are latin letters and nothing else.

[1] Syzbot report:

-----[ cut here ]-----

Unexpected user alpha2: Aï¿½

WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 is\_user\_regdom\_saved net/wireless/reg.c:440 [inline]

WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 restore\_alpha2 net/wireless/reg.c:3424 [inline]

WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 restore\_regulatory\_settings+0x3c0/0x1e50 net/wireless/reg.c:3516

Modules linked in:

CPU: 1 UID: 0 PID: 964 Comm: kworker/1:2 Not tainted 6.12.0-rc5-syzkaller-00044-gc1e939a21eb1 #0

Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024

Workqueue: events\_power\_efficient crda\_timeout\_work

RIP: 0010:is\_user\_regdom\_saved net/wireless/reg.c:440 [inline]

RIP: 0010:restore\_alpha2 net/wireless/reg.c:3424 [inline]

RIP: 0010:restore\_regulatory\_settings+0x3c0/0x1e50 net/wireless/reg.c:3516

...

Call Trace:

<TASK>

crda\_timeout\_work+0x27/0x50 net/wireless/reg.c:542

process\_one\_work kernel/workqueue.c:3229 [inline]

process\_scheduled\_works+0xa65/0x1850 kernel/workqueue.c:3310

worker\_thread+0x870/0xd30 kernel/workqueue.c:3391

kthread+0x2f2/0x390 kernel/kthread.c:389

ret\_from\_fork+0x4d/0x80 arch/x86/kernel/process.c:147

ret\_from\_fork\_asm+0x1a/0x30 arch/x86/entry/entry\_64.S:244

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21910>

**[CVE-2025-21912] kernel: gpio: rcar: Use raw\_spinlock to protect register access (Severity: MEDIUM)**



Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

gpio: rcar: Use raw\_spinlock to protect register access

Use raw\_spinlock in order to fix spurious messages about invalid context when spinlock debugging is enabled. The lock is only used to serialize register access.

```
[ 4.239592] =====
[ 4.239595] [ BUG: Invalid wait context ]
[ 4.239599] 6.13.0-rc7-arm64-renesas-05496-gd088502a519f #35 Not tainted
[ 4.239603] -----
[ 4.239606] kworker/u8:5/76 is trying to lock:
[ 4.239609] ffff0000091898a0 (&p->lock){....}-{3:3}, at: gpio_rcar_config_interrupt_input_mode+0x34/0x164
[ 4.239641] other info that might help us debug this:
[ 4.239643] context-{5:5}
[ 4.239646] 5 locks held by kworker/u8:5/76:
[ 4.239651] #0: ffff0000080fb148 ((wq_completion)async){+..+}-{0:0}, at: process_one_work+0x190/0x62c
[ 4.250180] OF: /soc/sound@ec500000/ports/port@0/endpoint: Read of boolean property 'frame-master' with a
value.
[ 4.254094] #1: ffff80008299bd80 ((work_completion)(&entry->work)){+..+}-{0:0}, at:
process_one_work+0x1b8/0x62c
[ 4.254109] #2: ffff00000920c8f8
[ 4.258345] OF: /soc/sound@ec500000/ports/port@1/endpoint: Read of boolean property 'bitclock-master' with a
value.
[ 4.264803] (&dev->mutex){....}-{4:4}, at: __device_attach_async_helper+0x3c/0xdc
[ 4.264820] #3: ffff00000a50ca40 (request_class#2){+..+}-{4:4}, at: __setup_irq+0xa0/0x690
[ 4.264840] #4:
[ 4.268872] OF: /soc/sound@ec500000/ports/port@1/endpoint: Read of boolean property 'frame-master' with a
value.
[ 4.273275] ffff00000a50c8c8 (lock_class){....}-{2:2}, at: __setup_irq+0xc4/0x690
[ 4.296130] renesas_sdhi_internal_dmac ee100000.mmc: mmc1 base at 0x00000000ee100000, max clock rate 200
MHz
[ 4.304082] stack backtrace:
[ 4.304086] CPU: 1 UID: 0 PID: 76 Comm: kworker/u8:5 Not tainted
6.13.0-rc7-arm64-renesas-05496-gd088502a519f #35
[ 4.304092] Hardware name: Renesas Salvator-X 2nd version board based on r8a77965 (DT)
[ 4.304097] Workqueue: async async_run_entry_fn
[ 4.304106] Call trace:
[ 4.304110] show_stack+0x14/0x20 (C)
[ 4.304122] dump_stack_lvl+0x6c/0x90
[ 4.304131] dump_stack+0x14/0x1c
[ 4.304138] __lock_acquire+0xdfc/0x1584
[ 4.426274] lock_acquire+0x1c4/0x33c
[ 4.429942] _raw_spin_lock_irqsave+0x5c/0x80
[ 4.434307] gpio_rcar_config_interrupt_input_mode+0x34/0x164
[ 4.440061] gpio_rcar_irq_set_type+0xd4/0xd8
[ 4.444422] __irq_set_trigger+0x5c/0x178
[ 4.448435] __setup_irq+0x2e4/0x690
```

```
[ 4.452012] request_threaded_irq+0xc4/0x190
[ 4.456285] devm_request_threaded_irq+0x7c/0xf4
[ 4.459398] ata1: link resume succeeded after 1 retries
[ 4.460902] mmc_gpiod_request_cd_irq+0x68/0xe0
[ 4.470660] mmc_start_host+0x50/0xac
[ 4.474327] mmc_add_host+0x80/0xe4
[ 4.477817] tmio_mmc_host_probe+0x2b0/0x440
[ 4.482094] renesas_sdhi_probe+0x488/0x6f4
[ 4.486281] renesas_sdhi_internal_dmac_probe+0x60/0x78
[ 4.491509] platform_probe+0x64/0xd8
[ 4.495178] really_probe+0xb8/0x2a8
[ 4.498756] __driver_probe_device+0x74/0x118
[ 4.503116] driver_probe_device+0x3c/0x154
[ 4.507303] __device_attach_driver+0xd4/0x160
[ 4.511750] bus_for_each_drv+0x84/0xe0
[ 4.515588] __device_attach_async_helper+0xb0/0xdc
[ 4.520470] async_run_entry_fn+0x30/0xd8
[ 4.524481] process_one_work+0x210/0x62c
[ 4.528494] worker_thread+0x1ac/0x340
[ 4.532245] kthread+0x10c/0x110
[ 4.535476] ret_from_fork+0x10/0x20
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21912>

### **[CVE-2025-21913] kernel: x86/amd\_nb: Use rdmsr\_safe() in amd\_get\_mmconfig\_range() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

x86/amd\_nb: Use rdmsr\_safe() in amd\_get\_mmconfig\_range()

Xen doesn't offer MSR\_FAM10H\_MMIO\_CONF\_BASE to all guests. This results in the following warning:

unchecked MSR access error: RDMSR from 0xc0010058 at rIP: 0xffffffff8101d19f (xen\_do\_read\_msr+0x7f/0xa0)

Call Trace:

```
xen_read_msr+0x1e/0x30
amd_get_mmconfig_range+0x2b/0x80
quirk_amd_mmconfig_area+0x28/0x100
pnp_fixup_device+0x39/0x50
__pnp_add_device+0xf/0x150
pnp_add_device+0x3d/0x100
pnpacpi_add_device_handler+0x1f9/0x280
acpi_ns_get_device_callback+0x104/0x1c0
acpi_ns_walk_namespace+0x1d0/0x260
acpi_get_devices+0x8a/0xb0
pnpacpi_init+0x50/0x80
do_one_initcall+0x46/0x2e0
kernel_init_freeable+0x1da/0x2f0
```

```
kernel_init+0x16/0x1b0
ret_from_fork+0x30/0x50
ret_from_fork_asm+0x1b/0x30
```

based on quirks for a "PNP0c01" device. Treating MMCFG as disabled is the right course of action, so no change is needed there.

This was most likely exposed by fixing the Xen MSR accessors to not be silently-safe.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21913>

**[CVE-2025-21914] kernel: slimbus: messaging: Free transaction ID in delayed interrupt scenario (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

slimbus: messaging: Free transaction ID in delayed interrupt scenario

In case of interrupt delay for any reason, slim\_do\_transfer() returns timeout error but the transaction ID (TID) is not freed. This results into invalid memory access inside qcom\_slim\_ngd\_rx\_msgq\_cb() due to invalid TID.

Fix the issue by freeing the TID in slim\_do\_transfer() before returning timeout error to avoid invalid memory access.

Call trace:

```
__memcpy_fromio+0x20/0x190
qcom_slim_ngd_rx_msgq_cb+0x130/0x290 [slim_qcom_ngd_ctrl]
vchan_complete+0x2a0/0x4a0
tasklet_action_common+0x274/0x700
tasklet_action+0x28/0x3c
_stext+0x188/0x620
run_ksoftirqd+0x34/0x74
smpboot_thread_fn+0x1d8/0x464
kthread+0x178/0x238
ret_from_fork+0x10/0x20
Code: aa0003e8 91000429 f100044a 3940002b (3800150b)
---[ end trace 0fe00bec2b975c99 ]---
```

Kernel panic - not syncing: Oops: Fatal exception in interrupt.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21914>

**[CVE-2025-21916] kernel: usb: atm: cxacru: fix a flaw in existing endpoint checks (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usb: atm: cxacru: fix a flaw in existing endpoint checks

Syzbot once again identified a flaw in usb endpoint checking, see [1]. This time the issue stems from a commit authored by me (2eabb655a968 ("usb: atm: cxacru: fix endpoint checking in cxacru\_bind()")).

While using `usb_find_common_endpoints()` may usually be enough to discard devices with wrong endpoints, in this case one needs more than just finding and identifying the sufficient number of endpoints of correct types - one needs to check the endpoint's address as well.

Since `cxacru_bind()` fills URBs with `CXACRU_EP_CMD` address in mind, switch the endpoint verification approach to `usb_check_XXX_endpoints()` instead to fix incomplete ep testing.

[1] Syzbot report:

usb 5-1: BOGUS urb xfer, pipe 3 != type 1

WARNING: CPU: 0 PID: 1378 at drivers/usb/core/urb.c:504 usb\_submit\_urb+0xc4e/0x18c0 drivers/usb/core/urb.c:503

...

RIP: 0010:usb\_submit\_urb+0xc4e/0x18c0 drivers/usb/core/urb.c:503

...

Call Trace:

<TASK>

cxacru\_cm+0x3c8/0xe50 drivers/usb/atm/cxacru.c:649

cxacru\_card\_status drivers/usb/atm/cxacru.c:760 [inline]

cxacru\_bind+0xcf9/0x1150 drivers/usb/atm/cxacru.c:1223

usbatm\_usb\_probe+0x314/0x1d30 drivers/usb/atm/usbatm.c:1058

cxacru\_usb\_probe+0x184/0x220 drivers/usb/atm/cxacru.c:1377

usb\_probe\_interface+0x641/0xbb0 drivers/usb/core/driver.c:396

really\_probe+0x2b9/0xad0 drivers/base/dd.c:658

\_\_driver\_probe\_device+0x1a2/0x390 drivers/base/dd.c:800

driver\_probe\_device+0x50/0x430 drivers/base/dd.c:830

...

More Info: <https://avd.aquasec.com/nvd/cve-2025-21916>

### **[CVE-2025-21917] kernel: usb: renesas\_usbhs: Flush the notify\_hotplug\_work (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usb: renesas\_usbhs: Flush the notify\_hotplug\_work

When performing continuous unbind/bind operations on the USB drivers available on the Renesas RZ/G2L SoC, a kernel crash with the message "Unable to handle kernel NULL pointer dereference at virtual address" may occur. This issue points to the `usbhsc_notify_hotplug()` function.

Flush the delayed work to avoid its execution when driver resources are

unavailable.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21917>

**[CVE-2025-21918] kernel: usb: typec: ucsi: Fix NULL pointer access (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

usb: typec: ucsi: Fix NULL pointer access

Resources should be released only after all threads that utilize them have been destroyed.

This commit ensures that resources are not released prematurely by waiting for the associated workqueue to complete before deallocating them.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21918>

**[CVE-2025-21922] kernel: ppp: Fix KMSAN uninit-value warning with bpf (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ppp: Fix KMSAN uninit-value warning with bpf

Syzbot caught an "KMSAN: uninit-value" warning [1], which is caused by the ppp driver not initializing a 2-byte header when using socket filter.

The following code can generate a PPP filter BPF program:

```
'''
struct bpf_program fp;
pcap_t *handle;
handle = pcap_open_dead(DLT_PPP_PPPD, 65535);
pcap_compile(handle, &fp, "ip and outbound", 0, 0);
bpf_dump(&fp, 1);
'''
```

Its output is:

```
'''
(000) ldh [2]
(001) jeq #0x21 jt 2 jf 5
(002) ldb [0]
(003) jeq #0x1 jt 4 jf 5
(004) ret #65535
(005) ret #0
'''
```

We can find similar code at the following link:

<https://github.com/ppp-project/ppp/blob/master/pppd/options.c#L1680>

The maintainer of this code repository is also the original maintainer

of the ppp driver.

As you can see the BPF program skips 2 bytes of data and then reads the 'Protocol' field to determine if it's an IP packet. Then it read the first byte of the first 2 bytes to determine the direction.

The issue is that only the first byte indicating direction is initialized in current ppp driver code while the second byte is not initialized.

For normal BPF programs generated by libpcap, uninitialized data won't be used, so it's not a problem. However, for carefully crafted BPF programs, such as those generated by syzkaller [2], which start reading from offset 0, the uninitialized data will be used and caught by KMSAN.

[1] <https://syzkaller.appspot.com/bug?extid=853242d9c9917165d791>

[2] <https://syzkaller.appspot.com/text?tag=ReproC&x=11994913980000>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21922>

### **[CVE-2025-21924] kernel: net: hns3: make sure ptp clock is unregister and freed if hclge\_ptp\_get\_cycle returns an error (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net: hns3: make sure ptp clock is unregister and freed if hclge\_ptp\_get\_cycle returns an error

During the initialization of ptp, hclge\_ptp\_get\_cycle might return an error and returned directly without unregister clock and free it. To avoid that, call hclge\_ptp\_destroy\_clock to unregister and free clock if hclge\_ptp\_get\_cycle failed.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21924>

### **[CVE-2025-21925] kernel: llc: do not use skb\_get() before dev\_queue\_xmit() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

llc: do not use skb\_get() before dev\_queue\_xmit()

syzbot is able to crash hosts [1], using llc and devices not supporting IFF\_TX\_SKB\_SHARING.

In this case, e1000 driver calls eth\_skb\_pad(), while the skb is shared.

Simply replace skb\_get() by skb\_clone() in net/llc/llc\_s\_ac.c

Note that e1000 driver might have an issue with pktgen, because it does not clear IFF\_TX\_SKB\_SHARING, this is an orthogonal change.

We need to audit other skb\_get() uses in net/llc.

[1]

kernel BUG at net/core/skbuff.c:2178 !

Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI

CPU: 0 UID: 0 PID: 16371 Comm: syz.2.2764 Not tainted 6.14.0-rc4-syzkaller-00052-gac9c34d1e45a #0

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014

RIP: 0010:pskb\_expand\_head+0x6ce/0x1240 net/core/skbuff.c:2178

Call Trace:

<TASK>

\_\_skb\_pad+0x18a/0x610 net/core/skbuff.c:2466

\_\_skb\_put\_padto include/linux/skbuff.h:3843 [inline]

skb\_put\_padto include/linux/skbuff.h:3862 [inline]

eth\_skb\_pad include/linux/etherdevice.h:656 [inline]

e1000\_xmit\_frame+0x2d99/0x5800 drivers/net/ethernet/intel/e1000/e1000\_main.c:3128

\_\_netdev\_start\_xmit include/linux/netdevice.h:5151 [inline]

netdev\_start\_xmit include/linux/netdevice.h:5160 [inline]

xmit\_one net/core/dev.c:3806 [inline]

dev\_hard\_start\_xmit+0x9a/0x7b0 net/core/dev.c:3822

sch\_direct\_xmit+0x1ae/0xc30 net/sched/sch\_generic.c:343

\_\_dev\_xmit\_skb net/core/dev.c:4045 [inline]

\_\_dev\_queue\_xmit+0x13d4/0x43e0 net/core/dev.c:4621

dev\_queue\_xmit include/linux/netdevice.h:3313 [inline]

llc\_sap\_action\_send\_test\_c+0x268/0x320 net/llc/llc\_s\_ac.c:144

llc\_exec\_sap\_trans\_actions net/llc/llc\_sap.c:153 [inline]

llc\_sap\_next\_state net/llc/llc\_sap.c:182 [inline]

llc\_sap\_state\_process+0x239/0x510 net/llc/llc\_sap.c:209

llc\_ui\_sendmsg+0xd0d/0x14e0 net/llc/af\_llc.c:993

sock\_sendmsg\_nosec net/socket.c:718 [inline]

More Info: <https://avd.aquasec.com/nvd/cve-2025-21925>

### **[CVE-2025-21931] kernel: hwpoison, memory\_hotplug: lock folio before unmap hwpoisoned folio (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

hwpoison, memory\_hotplug: lock folio before unmap hwpoisoned folio

Commit b15c87263a69 ("hwpoison, memory\_hotplug: allow hwpoisoned pages to be offlined) add page poison checks in do\_migrate\_range in order to make offline hwpoisoned page possible by introducing isolate\_lru\_page and try\_to\_unmap for hwpoisoned page. However folio lock must be held before

calling try\_to\_unmap. Add it to fix this problem.

Warning will be produced if folio is not locked during unmap:

```
-----[ cut here ]-----
kernel BUG at ./include/linux/swapops.h:400!
Internal error: Oops - BUG: 00000000f2000800 [#1] PREEMPT SMP
Modules linked in:
CPU: 4 UID: 0 PID: 411 Comm: bash Tainted: G      W      6.13.0-rc1-00016-g3c434c7ee82a-dirty #41
Tainted: [W]=WARN
Hardware name: QEMU QEMU Virtual Machine, BIOS 0.0.0 02/06/2015
pstate: 40400005 (nZcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--)
pc : try_to_unmap_one+0xb08/0xd3c
lr : try_to_unmap_one+0x3dc/0xd3c
Call trace:
try_to_unmap_one+0xb08/0xd3c (P)
try_to_unmap_one+0x3dc/0xd3c (L)
rmap_walk_anon+0xdc/0x1f8
rmap_walk+0x3c/0x58
try_to_unmap+0x88/0x90
unmap_poisoned_folio+0x30/0xa8
do_migrate_range+0x4a0/0x568
offline_pages+0x5a4/0x670
memory_block_action+0x17c/0x374
memory_subsys_offline+0x3c/0x78
device_offline+0xa4/0xd0
state_store+0x8c/0xf0
dev_attr_store+0x18/0x2c
sysfs_kf_write+0x44/0x54
kernfs_fop_write_iter+0x118/0x1a8
vfs_write+0x3a8/0x4bc
ksys_write+0x6c/0xf8
__arm64_sys_write+0x1c/0x28
invoke_syscall+0x44/0x100
el0_svc_common.constprop.0+0x40/0xe0
do_el0_svc+0x1c/0x28
el0_svc+0x30/0xd0
el0t_64_sync_handler+0xc8/0xcc
el0t_64_sync+0x198/0x19c
Code: f9407be0 b5fff320 d4210000 17ffff97 (d4210000)
---[ end trace 0000000000000000 ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21931>

**[CVE-2025-21935] kernel: rapidio: add check for rio\_add\_net() in rio\_scan\_alloc\_net() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:



rapidio: add check for rio\_add\_net() in rio\_scan\_alloc\_net()

The return value of rio\_add\_net() should be checked. If it fails, put\_device() should be called to free the memory and give up the reference initialized in rio\_add\_net().

More Info: <https://avd.aquasec.com/nvd/cve-2025-21935>

**[CVE-2025-21936] kernel: Bluetooth: Add check for mgmt\_alloc\_skb() in mgmt\_device\_connected() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: Add check for mgmt\_alloc\_skb() in mgmt\_device\_connected()

Add check for the return value of mgmt\_alloc\_skb() in mgmt\_device\_connected() to prevent null pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21936>

**[CVE-2025-21937] kernel: Bluetooth: Add check for mgmt\_alloc\_skb() in mgmt\_remote\_name() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: Add check for mgmt\_alloc\_skb() in mgmt\_remote\_name()

Add check for the return value of mgmt\_alloc\_skb() in mgmt\_remote\_name() to prevent null pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21937>

**[CVE-2025-21938] kernel: mptcp: fix 'scheduling while atomic' in mptcp\_pm\_nl\_append\_new\_local\_addr (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

mptcp: fix 'scheduling while atomic' in mptcp\_pm\_nl\_append\_new\_local\_addr

If multiple connection requests attempt to create an implicit mptcp endpoint in parallel, more than one caller may end up in mptcp\_pm\_nl\_append\_new\_local\_addr because none found the address in local\_addr\_list during their call to mptcp\_pm\_nl\_get\_local\_id. In this

case, the concurrent new\_local\_addr calls may delete the address entry created by the previous caller. These deletes use synchronize\_rcu, but this is not permitted in some of the contexts where this function may be called. During packet recv, the caller may be in a rcu read critical section and have preemption disabled.

An example stack:

BUG: scheduling while atomic: swapper/2/0/0x00000302

Call Trace:

```
<IRQ>
dump_stack_lvl (lib/dump_stack.c:117 (discriminator 1))
dump_stack (lib/dump_stack.c:124)
__schedule_bug (kernel/sched/core.c:5943)
schedule_debug.constprop.0 (arch/x86/include/asm/preempt.h:33 kernel/sched/core.c:5970)
__schedule (arch/x86/include/asm/jump_label.h:27 include/linux/jump_label.h:207 kernel/sched/features.h:29
kernel/sched/core.c:6621)
schedule (arch/x86/include/asm/preempt.h:84 kernel/sched/core.c:6804 kernel/sched/core.c:6818)
schedule_timeout (kernel/time/timer.c:2160)
wait_for_completion (kernel/sched/completion.c:96 kernel/sched/completion.c:116 kernel/sched/completion.c:127
kernel/sched/completion.c:148)
__wait_rcu_gp (include/linux/rcupdate.h:311 kernel/rcu/update.c:444)
synchronize_rcu (kernel/rcu/tree.c:3609)
mptcp_pm_nl_append_new_local_addr (net/mptcp/pm_netlink.c:966 net/mptcp/pm_netlink.c:1061)
mptcp_pm_nl_get_local_id (net/mptcp/pm_netlink.c:1164)
mptcp_pm_get_local_id (net/mptcp/pm.c:420)
subflow_check_req (net/mptcp/subflow.c:98 net/mptcp/subflow.c:213)
subflow_v4_route_req (net/mptcp/subflow.c:305)
tcp_conn_request (net/ipv4/tcp_input.c:7216)
subflow_v4_conn_request (net/mptcp/subflow.c:651)
tcp_rcv_state_process (net/ipv4/tcp_input.c:6709)
tcp_v4_do_rcv (net/ipv4/tcp_ipv4.c:1934)
tcp_v4_rcv (net/ipv4/tcp_ipv4.c:2334)
ip_protocol_deliver_rcu (net/ipv4/ip_input.c:205 (discriminator 1))
ip_local_deliver_finish (include/linux/rcupdate.h:813 net/ipv4/ip_input.c:234)
ip_local_deliver (include/linux/netfilter.h:314 include/linux/netfilter.h:308 net/ipv4/ip_input.c:254)
ip_sublist_rcv_finish (include/net/dst.h:461 net/ipv4/ip_input.c:580)
ip_sublist_rcv (net/ipv4/ip_input.c:640)
ip_list_rcv (net/ipv4/ip_input.c:675)
__netif_receive_skb_list_core (net/core/dev.c:5583 net/core/dev.c:5631)
netif_receive_skb_list_internal (net/core/dev.c:5685 net/core/dev.c:5774)
napi_complete_done (include/linux/list.h:37 include/net/gro.h:449 include/net/gro.h:444 net/core/dev.c:6114)
igb_poll (drivers/net/ethernet/intel/igb/igb_main.c:8244) igb
__napi_poll (net/core/dev.c:6582)
net_rx_action (net/core/dev.c:6653 net/core/dev.c:6787)
handle_softirqs (kernel/softirq.c:553)
__irq_exit_rcu (kernel/softirq.c:588 kernel/softirq.c:427 kernel/softirq.c:636)
irq_exit_rcu (kernel/softirq.c:651)
common_interrupt (arch/x86/kernel/irq.c:247 (discriminator 14))
</IRQ>
```

This problem seems particularly prevalent if the user advertises an

endpoint that has a different external vs internal address. In the case where the external address is advertised and multiple connections already exist, multiple subflow SYNs arrive in parallel which tends to trigger the race during creation of the first local\_addr\_list entries which have the internal address instead.

Fix by skipping the replacement of an existing implicit local address if called via mptcp\_pm\_nl\_get\_local\_id.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21938>

**[CVE-2025-21941] kernel: drm/amd/display: Fix null check for pipe\_ctx->plane\_state in resource\_build\_scaling\_params (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix null check for pipe\_ctx->plane\_state in resource\_build\_scaling\_params

Null pointer dereference 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.

Found by code review.

(cherry picked from commit 63e6a77ccf239337baa9b1e7787cde9fa0462092)

More Info: <https://avd.aquasec.com/nvd/cve-2025-21941>

**[CVE-2025-21943] kernel: gpio: aggregator: protect driver attr handlers against module unload (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

gpio: aggregator: protect driver attr handlers against module unload

Both new\_device\_store and delete\_device\_store touch module global resources (e.g. gpio\_aggregator\_lock). To prevent race conditions with module unload, a reference needs to be held.

Add try\_module\_get() in these handlers.

For new\_device\_store, this eliminates what appears to be the most dangerous scenario: if an id is allocated from gpio\_aggregator\_idr but platform\_device\_register has not yet been called or completed, a concurrent module unload could fail to unregister/delete the device, leaving behind a dangling platform device/GPIO forwarder. This can result in various issues.

The following simple reproducer demonstrates these problems:

```
#!/bin/bash
while ;; do
    # note: whether 'gpiochip0 0' exists or not does not matter.
    echo 'gpiochip0 0' > /sys/bus/platform/drivers/gpio-aggregator/new_device
done &
while ;; do
    modprobe gpio-aggregator
    modprobe -r gpio-aggregator
done &
wait
```

Starting with the following warning, several kinds of warnings will appear and the system may become unstable:

```
-----[ cut here ]-----
list_del corruption, ffff888103e2e980->next is LIST_POISON1 (dead000000000100)
WARNING: CPU: 1 PID: 1327 at lib/list_debug.c:56 __list_del_entry_valid_or_report+0xa3/0x120
[...]
RIP: 0010:__list_del_entry_valid_or_report+0xa3/0x120
[...]
Call Trace:
<TASK>
? __list_del_entry_valid_or_report+0xa3/0x120
? __warn.cold+0x93/0xf2
? __list_del_entry_valid_or_report+0xa3/0x120
? report_bug+0xe6/0x170
? __irq_work_queue_local+0x39/0xe0
? handle_bug+0x58/0x90
? exc_invalid_op+0x13/0x60
? asm_exc_invalid_op+0x16/0x20
? __list_del_entry_valid_or_report+0xa3/0x120
gpiod_remove_lookup_table+0x22/0x60
new_device_store+0x315/0x350 [gpio_aggregator]
kernfs_fop_write_iter+0x137/0x1f0
vfs_write+0x262/0x430
ksys_write+0x60/0xd0
do_syscall_64+0x6c/0x180
entry_SYSCALL_64_after_hwframe+0x76/0x7e
[...]
</TASK>
---[ end trace 0000000000000000 ]---
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21943>

### **[CVE-2025-21944] kernel: ksmbd: fix bug on trap in smb2\_lock (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix bug on trap in smb2\_lock

If lock count is greater than 1, flags could be old value.  
It should be checked with flags of smb\_lock, not flags.  
It will cause bug-on trap from locks\_free\_lock in error handling routine.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21944>

**[CVE-2025-21946] kernel: ksmbd: fix out-of-bounds in parse\_sec\_desc() (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix out-of-bounds in parse\_sec\_desc()

If osidoffset, gsidoffset and dacloffset could be greater than smb\_ntsd struct size. If it is smaller, It could cause slab-out-of-bounds.  
And when validating sid, It need to check it included subauth array size.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21946>

**[CVE-2025-21947] kernel: ksmbd: fix type confusion via race condition when using ipc\_msg\_send\_request (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix type confusion via race condition when using ipc\_msg\_send\_request

req->handle is allocated using ksmbd\_acquire\_id(&ipc\_ida), based on ida\_alloc. req->handle from ksmbd\_ipc\_login\_request and FSCTL\_PIPE\_TRANSCEIVE ioctl can be same and it could lead to type confusion between messages, resulting in access to unexpected parts of memory after an incorrect delivery. ksmbd check type of ipc response but missing add continue to check next ipc reponse.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21947>

**[CVE-2025-21948] kernel: HID: appleir: Fix potential NULL dereference at raw event handle (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

HID: appleir: Fix potential NULL dereference at raw event handle

Syzkaller reports a NULL pointer dereference issue in input\_event().

BUG: KASAN: null-ptr-deref in instrument\_atomic\_read include/linux/instrumented.h:68 [inline]

BUG: KASAN: null-ptr-deref in \_test\_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline]

BUG: KASAN: null-ptr-deref in is\_event\_supported drivers/input/input.c:67 [inline]

BUG: KASAN: null-ptr-deref in input\_event+0x42/0xa0 drivers/input/input.c:395

Read of size 8 at addr 0000000000000028 by task syz-executor199/2949

CPU: 0 UID: 0 PID: 2949 Comm: syz-executor199 Not tainted 6.13.0-rc4-syzkaller-00076-gf097a36ef88d #0

Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024

Call Trace:

<IRQ>

\_\_dump\_stack lib/dump\_stack.c:94 [inline]

dump\_stack\_lvl+0x116/0x1f0 lib/dump\_stack.c:120

kasan\_report+0xd9/0x110 mm/kasan/report.c:602

check\_region\_inline mm/kasan/generic.c:183 [inline]

kasan\_check\_range+0xef/0x1a0 mm/kasan/generic.c:189

instrument\_atomic\_read include/linux/instrumented.h:68 [inline]

\_test\_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline]

is\_event\_supported drivers/input/input.c:67 [inline]

input\_event+0x42/0xa0 drivers/input/input.c:395

input\_report\_key include/linux/input.h:439 [inline]

key\_down drivers/hid/hid-appleir.c:159 [inline]

appleir\_raw\_event+0x3e5/0x5e0 drivers/hid/hid-appleir.c:232

\_\_hid\_input\_report.constprop.0+0x312/0x440 drivers/hid/hid-core.c:2111

hid\_ctrl+0x49f/0x550 drivers/hid/usbhid/hid-core.c:484

\_\_usb\_hcd\_giveback\_urb+0x389/0x6e0 drivers/usb/core/hcd.c:1650

usb\_hcd\_giveback\_urb+0x396/0x450 drivers/usb/core/hcd.c:1734

dummy\_timer+0x17f7/0x3960 drivers/usb/gadget/udc/dummy\_hcd.c:1993

\_\_run\_hrtimer kernel/time/hrtimer.c:1739 [inline]

\_\_hrtimer\_run\_queues+0x20a/0xae0 kernel/time/hrtimer.c:1803

hrtimer\_run\_softirq+0x17d/0x350 kernel/time/hrtimer.c:1820

handle\_softirqs+0x206/0x8d0 kernel/softirq.c:561

\_\_do\_softirq kernel/softirq.c:595 [inline]

invoke\_softirq kernel/softirq.c:435 [inline]

\_\_irq\_exit\_rcu+0xfa/0x160 kernel/softirq.c:662

irq\_exit\_rcu+0x9/0x30 kernel/softirq.c:678

instr\_sysvec\_apic\_timer\_interrupt arch/x86/kernel/apic/apic.c:1049 [inline]

sysvec\_apic\_timer\_interrupt+0x90/0xb0 arch/x86/kernel/apic/apic.c:1049

</IRQ>

<TASK>

asm\_sysvec\_apic\_timer\_interrupt+0x1a/0x20 arch/x86/include/asm/identry.h:702

\_\_mod\_timer+0x8f6/0xdc0 kernel/time/timer.c:1185

add\_timer+0x62/0x90 kernel/time/timer.c:1295

schedule\_timeout+0x11f/0x280 kernel/time/sleep\_timeout.c:98

usbhid\_wait\_io+0x1c7/0x380 drivers/hid/usbhid/hid-core.c:645

usbhid\_init\_reports+0x19f/0x390 drivers/hid/usbhid/hid-core.c:784

hiddev\_ioctl+0x1133/0x15b0 drivers/hid/usbhid/hiddev.c:794

vfs\_ioctl fs/ioctl.c:51 [inline]

\_\_do\_sys\_ioctl fs/ioctl.c:906 [inline]

\_\_se\_sys\_ioctl fs/ioctl.c:892 [inline]

\_\_x64\_sys\_ioctl+0x190/0x200 fs/ioctl.c:892

```
do_syscall_x64 arch/x86/entry/common.c:52 [inline]
do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83
entry_SYSCALL_64_after_hwframe+0x77/0x7f
</TASK>
```

This happens due to the malformed report items sent by the emulated device which results in a report, that has no fields, being added to the report list. Due to this appleir\_input\_configured() is never called, hidinput\_connect() fails which results in the HID\_CLAIMED\_INPUT flag is not being set. However, it does not make appleir\_probe() fail and lets the event callback to be called without the associated input device.

Thus, add a check for the HID\_CLAIMED\_INPUT flag and leave the event hook early if the driver didn't claim any input\_dev for some reason. Moreover, some other hid drivers accessing input\_dev in their event callbacks do have similar checks, too.

Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21948>

### **[CVE-2025-21949] kernel: LoongArch: Set hugetlb mmap base address aligned with pmd size (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

LoongArch: Set hugetlb mmap base address aligned with pmd size

With ltp test case "testcases/bin/hugefork02", there is a dmesg error report message such as:

```
kernel BUG at mm/hugetlb.c:5550!
Oops - BUG[#1]:
CPU: 0 UID: 0 PID: 1517 Comm: hugefork02 Not tainted 6.14.0-rc2+ #241
Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 2/2/2022
pc 90000000004eaf1c ra 9000000000485538 tp 900000010edbc000 sp 900000010edbf940
a0 900000010edbf900 a1 9000000108d20280 a2 00007fff9474000 a3 00007fff3474000
a4 0000000000000000 a5 0000000000000003 a6 00000000003cadd3 a7 0000000000000000
t0 0000000001ffffff t1 0000000001474000 t2 900000010ecd7900 t3 00007fff9474000
t4 00007fff9474000 t5 0000000000000040 t6 900000010edbf900 t7 0000000000000001
t8 0000000000000005 u0 90000000004849d0 s9 900000010edbf900 s0 9000000108d20280
s1 00007fff9474000 s2 0000000002000000 s3 9000000108d20280 s4 9000000002b38b10
s5 900000010edbf900 s6 00007fff3474000 s7 00000000000000406 s8 900000010edbf908
ra: 9000000000485538 unmap_vmas+0x130/0x218
ERA: 90000000004eaf1c __unmap_hugepage_range+0x6f4/0x7d0
PRMD: 00000004 (PPLV0 +PIE -PWE)
EUE: 00000007 (+FPE +SXE +ASXE -BTE)
ECFG: 00071c1d (LIE=0,2-4,10-12 VS=7)
ESTAT: 000c0000 [BRK] (IS= ECode=12 EsubCode=0)
```

PRID: 0014c010 (Loongson-64bit, Loongson-3A5000)

Process hugefork02 (pid: 1517, threadinfo=00000000a670eaf4, task=000000007a95fc64)

Call Trace:

```
[<90000000004eaf1c>] __unmap_hugepage_range+0x6f4/0x7d0
[<9000000000485534>] unmap_vmas+0x12c/0x218
[<9000000000494068>] exit_mmap+0xe0/0x308
[<900000000025fdc4>] mmpu+0x74/0x180
[<900000000026a284>] do_exit+0x294/0x898
[<900000000026aa30>] do_group_exit+0x30/0x98
[<900000000027bed4>] get_signal+0x83c/0x868
[<90000000002457b4>] arch_do_signal_or_restart+0x54/0xfa0
[<900000000015795e8>] irqentry_exit_to_user_mode+0xb8/0x138
[<90000000002572d0>] tlb_do_page_fault_1+0x114/0x1b4
```

The problem is that base address allocated from hugetlbfs is not aligned with pmd size. Here add a checking for hugetlbfs and align base address with pmd size. After this patch the test case "testcases/bin/hugefork02" passes to run.

This is similar to the commit 7f24cbc9c4d42db8a3c8484d1 ("mm/mmap: teach generic\_get\_unmapped\_area{\_topdown} to handle hugetlb mappings").

More Info: <https://avd.aquasec.com/nvd/cve-2025-21949>

### **[CVE-2025-21950] kernel: drivers: virt: acrn: hsm: Use kzalloc to avoid info leak in pmcmd\_ioctl (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drivers: virt: acrn: hsm: Use kzalloc to avoid info leak in pmcmd\_ioctl

In the "pmcmd\_ioctl" function, three memory objects allocated by kmalloc are initialized by "hcall\_get\_cpu\_state", which are then copied to user space. The initializer is indeed implemented in "acrn\_hypervcall2" (arch/x86/include/asm/acrn.h). There is a risk of information leakage due to uninitialized bytes.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21950>

### **[CVE-2025-21951] kernel: bus: mhi: host: pci\_generic: Use pci\_try\_reset\_function() to avoid deadlock (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

bus: mhi: host: pci\_generic: Use pci\_try\_reset\_function() to avoid deadlock



There are multiple places from where the recovery work gets scheduled asynchronously. Also, there are multiple places where the caller waits synchronously for the recovery to be completed. One such place is during the PM shutdown() callback.

If the device is not alive during recovery\_work, it will try to reset the device using pci\_reset\_function(). This function internally will take the device\_lock() first before resetting the device. By this time, if the lock has already been acquired, then recovery\_work will get stalled while waiting for the lock. And if the lock was already acquired by the caller which waits for the recovery\_work to be completed, it will lead to deadlock.

This is what happened on the X1E80100 CRD device when the device died before shutdown() callback. Driver core calls the driver's shutdown() callback while holding the device\_lock() leading to deadlock.

And this deadlock scenario can occur on other paths as well, like during the PM suspend() callback, where the driver core would hold the device\_lock() before calling driver's suspend() callback. And if the recovery\_work was already started, it could lead to deadlock. This is also observed on the X1E80100 CRD.

So to fix both issues, use pci\_try\_reset\_function() in recovery\_work. This function first checks for the availability of the device\_lock() before trying to reset the device. If the lock is available, it will acquire it and reset the device. Otherwise, it will return -EAGAIN. If that happens, recovery\_work will fail with the error message "Recovery failed" as not much could be done.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21951>

### **[CVE-2025-21955] kernel: ksmbd: prevent connection release during oplock break notification (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: prevent connection release during oplock break notification

ksmbd\_work could be freed when after connection release.  
Increment r\_count of ksmbd\_conn to indicate that requests are not finished yet and to not release the connection.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21955>

### **[CVE-2025-21956] kernel: drm/amd/display: Assign normalized\_pix\_clk when color depth = 14 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Assign normalized\_pix\_clk when color depth = 14

[WHY & HOW]

A warning message "WARNING: CPU: 4 PID: 459 at ... /dc\_resource.c:3397 calculate\_phy\_pix\_clks+0xef/0x100 [amdgpu]" occurs because the display\_color\_depth == COLOR\_DEPTH\_141414 is not handled. This is observed in Radeon RX 6600 XT.

It is fixed by assigning pix\_clk \* (14 \* 3) / 24 - same as the rests.

Also fixes the indentation in get\_norm\_pix\_clk.

(cherry picked from commit 274a87eb389f58eddc5659ab0b180b37e92775)

More Info: <https://avd.aquasec.com/nvd/cve-2025-21956>

### **[CVE-2025-21957] kernel: scsi: qla1280: Fix kernel oops when debug level > 2 (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

scsi: qla1280: Fix kernel oops when debug level > 2

A null dereference or oops exception will eventually occur when qla1280.c driver is compiled with DEBUG\_QLA1280 enabled and ql\_debug\_level > 2. I think its clear from the code that the intention here is sg\_dma\_len(s) not length of sg\_next(s) when printing the debug info.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21957>

### **[CVE-2025-21959] kernel: netfilter: nf\_conncount: Fully initialize struct nf\_conncount\_tuple in insert\_tree() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

netfilter: nf\_conncount: Fully initialize struct nf\_conncount\_tuple in insert\_tree()

Since commit b36e4523d4d5 ("netfilter: nf\_conncount: fix garbage collection confirm race"), `cpu` and `jiffies32` were introduced to the struct nf\_conncount\_tuple.

The commit made nf\_conncount\_add() initialize `conn->cpu` and `conn->jiffies32` when allocating the struct.

In contrast, `count_tree()` was not changed to initialize them.

By commit 34848d5c896e ("netfilter: nf\_conncount: Split insert and traversal"), `count_tree()` was split and the relevant allocation code now resides in `insert_tree()`.

Initialize ``conn->cpu`` and ``conn->jiffies32`` in `insert_tree()`.

BUG: KMSAN: uninit-value in `find_or_evict` net/netfilter/nf\_conncount.c:117 [inline]

BUG: KMSAN: uninit-value in `__nf_conncount_add+0xd9c/0x2850` net/netfilter/nf\_conncount.c:143

`find_or_evict` net/netfilter/nf\_conncount.c:117 [inline]

`__nf_conncount_add+0xd9c/0x2850` net/netfilter/nf\_conncount.c:143

`count_tree` net/netfilter/nf\_conncount.c:438 [inline]

`nf_conncount_count+0x82f/0x1e80` net/netfilter/nf\_conncount.c:521

`connlimit_mt+0x7f6/0xbd0` net/netfilter/xt\_connlimit.c:72

`__nft_match_eval` net/netfilter/nft\_compat.c:403 [inline]

`nft_match_eval+0x1a5/0x300` net/netfilter/nft\_compat.c:433

`expr_call_ops_eval` net/netfilter/nf\_tables\_core.c:240 [inline]

`nft_do_chain+0x426/0x2290` net/netfilter/nf\_tables\_core.c:288

`nft_do_chain_ipv4+0x1a5/0x230` net/netfilter/nft\_chain\_filter.c:23

`nf_hook_entry_hookfn` include/linux/netfilter.h:154 [inline]

`nf_hook_slow+0xf4/0x400` net/netfilter/core.c:626

`nf_hook_slow_list+0x24d/0x860` net/netfilter/core.c:663

`NF_HOOK_LIST` include/linux/netfilter.h:350 [inline]

`ip_sublist_rcv+0x17b7/0x17f0` net/ipv4/ip\_input.c:633

`ip_list_rcv+0x9ef/0xa40` net/ipv4/ip\_input.c:669

`__netif_receive_skb_list_ptype` net/core/dev.c:5936 [inline]

`__netif_receive_skb_list_core+0x15c5/0x1670` net/core/dev.c:5983

`__netif_receive_skb_list` net/core/dev.c:6035 [inline]

`netif_receive_skb_list_internal+0x1085/0x1700` net/core/dev.c:6126

`netif_receive_skb_list+0x5a/0x460` net/core/dev.c:6178

`xdp_rcv_frames` net/bpf/test\_run.c:280 [inline]

`xdp_test_run_batch` net/bpf/test\_run.c:361 [inline]

`bpf_test_run_xdp_live+0x2e86/0x3480` net/bpf/test\_run.c:390

`bpf_prog_test_run_xdp+0xf1d/0x1ae0` net/bpf/test\_run.c:1316

`bpf_prog_test_run+0x5e5/0xa30` kernel/bpf/syscall.c:4407

`__sys_bpf+0x6aa/0xd90` kernel/bpf/syscall.c:5813

`__do_sys_bpf` kernel/bpf/syscall.c:5902 [inline]

`__se_sys_bpf` kernel/bpf/syscall.c:5900 [inline]

`__ia32_sys_bpf+0xa0/0xe0` kernel/bpf/syscall.c:5900

`ia32_sys_call+0x394d/0x4180` arch/x86/include/generated/asm/syscalls\_32.h:358

`do_syscall_32_irqs_on` arch/x86/entry/common.c:165 [inline]

`__do_fast_syscall_32+0xb0/0x110` arch/x86/entry/common.c:387

`do_fast_syscall_32+0x38/0x80` arch/x86/entry/common.c:412

`do_SYSENTER_32+0x1f/0x30` arch/x86/entry/common.c:450

`entry_SYSENTER_compat_after_hwframe+0x84/0x8e`

Uninit was created at:

`slab_post_alloc_hook` mm/slub.c:4121 [inline]

`slab_alloc_node` mm/slub.c:4164 [inline]

`kmem_cache_alloc_noprof+0x915/0xe10` mm/slub.c:4171

`insert_tree` net/netfilter/nf\_conncount.c:372 [inline]

`count_tree` net/netfilter/nf\_conncount.c:450 [inline]

`nf_conncount_count+0x1415/0x1e80` net/netfilter/nf\_conncount.c:521

```
connlimit_mt+0x7f6/0xbd0 net/netfilter/xt_connlimit.c:72
__nft_match_eval net/netfilter/nft_compat.c:403 [inline]
nft_match_eval+0x1a5/0x300 net/netfilter/nft_compat.c:433
expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline]
nft_do_chain+0x426/0x2290 net/netfilter/nf_tables_core.c:288
nft_do_chain_ipv4+0x1a5/0x230 net/netfilter/nft_chain_filter.c:23
nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline]
nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626
nf_hook_slow_list+0x24d/0x860 net/netfilter/core.c:663
NF_HOOK_LIST include/linux/netfilter.h:350 [inline]
ip_sublist_rcv+0x17b7/0x17f0 net/ipv4/ip_input.c:633
ip_list_rcv+0x9ef/0xa40 net/ip
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21959>

### **[CVE-2025-21960] kernel: eth: bnxt: do not update checksum in bnxt\_xdp\_build\_skb() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

eth: bnxt: do not update checksum in bnxt\_xdp\_build\_skb()

The bnxt\_rx\_pkt() updates ip\_summed value at the end if checksum offload is enabled.

When the XDP-MB program is attached and it returns XDP\_PASS, the bnxt\_xdp\_build\_skb() is called to update skb\_shared\_info.

The main purpose of bnxt\_xdp\_build\_skb() is to update skb\_shared\_info, but it updates ip\_summed value too if checksum offload is enabled.

This is actually duplicate work.

When the bnxt\_rx\_pkt() updates ip\_summed value, it checks if ip\_summed is CHECKSUM\_NONE or not.

It means that ip\_summed should be CHECKSUM\_NONE at this moment.

But ip\_summed may already be updated to CHECKSUM\_UNNECESSARY in the XDP-MB-PASS path.

So the by skb\_checksum\_none\_assert() WARNS about it.

This is duplicate work and updating ip\_summed in the bnxt\_xdp\_build\_skb() is not needed.

Splat looks like:

```
WARNING: CPU: 3 PID: 5782 at ./include/linux/skbuff.h:5155 bnxt_rx_pkt+0x479b/0x7610 [bnxt_en]
Modules linked in: bnxt_re bnxt_en rdma_ucm rdma_cm iw_cm ib_cm ib_uverbs veth xt_nat xt_tcpudp xt_conntrack
nft_chain_nat xt_MASQUERADE nf_]
CPU: 3 UID: 0 PID: 5782 Comm: socat Tainted: G      W      6.14.0-rc4+ #27
Tainted: [W]=WARN
Hardware name: ASUS System Product Name/PRIME Z690-P D4, BIOS 0603 11/01/2021
RIP: 0010:bnxt_rx_pkt+0x479b/0x7610 [bnxt_en]
```

Code: 54 24 0c 4c 89 f1 4c 89 ff c1 ea 1f ff d3 0f 1f 00 49 89 c6 48 85 c0 0f 84 4c e5 ff ff 48 89 c7 e8 ca 3d a0 c8 e9 8f f4 ff ff <0f> 0b f

RSP: 0018:ffff88881ba09928 EFLAGS: 00010202

RAX: 0000000000000000 RBX: 00000000c7590303 RCX: 0000000000000000

RDX: 1ffff1104e7d1610 RSI: 0000000000000001 RDI: ffff8881c91300b8

RBP: ffff88881ba09b28 R08: ffff888273e8b0d0 R09: ffff888273e8b070

R10: ffff888273e8b010 R11: ffff888278b0f000 R12: ffff888273e8b080

R13: ffff8881c9130e00 R14: ffff8881505d3800 R15: ffff888273e8b000

FS: 00007f5a2e7be080(0000) GS:ffff88881ba00000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 00007fff2e708ff8 CR3: 000000013e3b0000 CR4: 00000000007506f0

PKRU: 55555554

Call Trace:

<IRQ>

? \_\_warn+0xcd/0x2f0

? bnxt\_rx\_pkt+0x479b/0x7610

? report\_bug+0x326/0x3c0

? handle\_bug+0x53/0xa0

? exc\_invalid\_op+0x14/0x50

? asm\_exc\_invalid\_op+0x16/0x20

? bnxt\_rx\_pkt+0x479b/0x7610

? bnxt\_rx\_pkt+0x3e41/0x7610

? \_\_pfx\_bnxt\_rx\_pkt+0x10/0x10

? napi\_complete\_done+0x2cf/0x7d0

\_\_bnxt\_poll\_work+0x4e8/0x1220

? \_\_pfx\_\_bnxt\_poll\_work+0x10/0x10

? \_\_pfx\_mark\_lock.part.0+0x10/0x10

bnxt\_poll\_p5+0x36a/0xfa0

? \_\_pfx\_bnxt\_poll\_p5+0x10/0x10

\_\_napi\_poll.constprop.0+0xa0/0x440

net\_rx\_action+0x899/0xd00

...

Following ping.py patch adds xdp-mb-pass case. so ping.py is going to be able to reproduce this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21960>

### **[CVE-2025-21961] kernel: eth: bnxt: fix truesize for mb-xdp-pass case (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

eth: bnxt: fix truesize for mb-xdp-pass case

When mb-xdp is set and return is XDP\_PASS, packet is converted from xdp\_buff to sk\_buff with xdp\_update\_skb\_shared\_info() in bnxt\_xdp\_build\_skb().

bnxt\_xdp\_build\_skb() passes incorrect truesize argument to xdp\_update\_skb\_shared\_info().

The truesize is calculated as BNXT\_RX\_PAGE\_SIZE \* sinfo->nr\_frags but

the skb\_shared\_info was wiped by napi\_build\_skb() before.  
So it stores sinfo->nr\_frags before bnxt\_xdp\_build\_skb() and use it  
instead of getting skb\_shared\_info from xdp\_get\_shared\_info\_from\_buff().

Splat looks like:

```
-----[ cut here ]-----  
WARNING: CPU: 2 PID: 0 at net/core/skbuff.c:6072 skb_try_coalesce+0x504/0x590  
Modules linked in: xt_nat xt_tcpudp veth af_packet xt_contrack nft_chain_nat xt_MASQUERADE nf_contrack_netlink  
xfrm_user xt_addrtype nft_coms  
CPU: 2 UID: 0 PID: 0 Comm: swapper/2 Not tainted 6.14.0-rc2+ #3  
RIP: 0010:skb_try_coalesce+0x504/0x590  
Code: 4b fd ff ff 49 8b 34 24 40 80 e6 40 0f 84 3d fd ff ff 49 8b 74 24 48 40 f6 c6 01 0f 84 2e fd ff ff 48 8d 4e ff e9 25 fd  
ff ff <0f> 0b e99  
RSP: 0018:ffffb62c4120caa8 EFLAGS: 00010287  
RAX: 0000000000000003 RBX: fffffb62c4120cb14 RCX: 00000000000000ec0  
RDX: 00000000000001000 RSI: fffffa06e5d7dc000 RDI: 0000000000000003  
RBP: fffffa06e5d7ddec0 R08: fffffa06e6120a800 R09: fffffa06e7a119900  
R10: 00000000000002310 R11: fffffa06e5d7dcec0 R12: fffffe4360575f740  
R13: fffffe43600000000 R14: 0000000000000002 R15: 0000000000000002  
FS: 0000000000000000(0000) GS:ffffa0755f700000(0000) knlGS:0000000000000000  
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033  
CR2: 00007f147b76b0f8 CR3: 00000001615d4000 CR4: 00000000007506f0  
PKRU: 55555554  
Call Trace:  
<IRQ>  
? __warn+0x84/0x130  
? skb_try_coalesce+0x504/0x590  
? report_bug+0x18a/0x1a0  
? handle_bug+0x53/0x90  
? exc_invalid_op+0x14/0x70  
? asm_exc_invalid_op+0x16/0x20  
? skb_try_coalesce+0x504/0x590  
inet_frag_reasm_finish+0x11f/0x2e0  
ip_defrag+0x37a/0x900  
ip_local_deliver+0x51/0x120  
ip_sublist_rcv_finish+0x64/0x70  
ip_sublist_rcv+0x179/0x210  
ip_list_rcv+0xf9/0x130
```

How to reproduce:

<Node A>

```
ip link set $interface1 xdp obj xdp_pass.o
```

```
ip link set $interface1 mtu 9000 up
```

```
ip a a 10.0.0.1/24 dev $interface1
```

<Node B>

```
ip link set $interfac2 mtu 9000 up
```

```
ip a a 10.0.0.2/24 dev $interface2
```

```
ping 10.0.0.1 -s 65000
```

Following ping.py patch adds xdp-mb-pass case. so ping.py is going to be  
able to reproduce this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21961>

**[CVE-2025-21962] kernel: cifs: Fix integer overflow while processing closetimeo mount option (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

cifs: Fix integer overflow while processing closetimeo mount option

User-provided mount parameter closetimeo of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow.

Found by Linux Verification Center (linuxtesting.org) with SVACE.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21962>

**[CVE-2025-21963] kernel: cifs: Fix integer overflow while processing acdirmax mount option (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

cifs: Fix integer overflow while processing acdirmax mount option

User-provided mount parameter acdirmax of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow.

Found by Linux Verification Center (linuxtesting.org) with SVACE.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21963>

**[CVE-2025-21964] kernel: cifs: Fix integer overflow while processing acregmax mount option (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

cifs: Fix integer overflow while processing acregmax mount option

User-provided mount parameter acregmax of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow.

Found by Linux Verification Center (linuxtesting.org) with SVACE.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21964>

**[CVE-2025-21967] kernel: ksmbd: fix use-after-free in ksmbd\_free\_work\_struct (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix use-after-free in ksmbd\_free\_work\_struct

->interim\_entry of ksmbd\_work could be deleted after oplock is freed.

We don't need to manage it with linked list. The interim request could be immediately sent whenever a oplock break wait is needed.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21967>

**[CVE-2025-21968] kernel: drm/amd/display: Fix slab-use-after-free on hdcp\_work (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix slab-use-after-free on hdcp\_work

[Why]

A slab-use-after-free is reported when HDCP is destroyed but the property\_validate\_dwork queue is still running.

[How]

Cancel the delayed work when destroying workqueue.

(cherry picked from commit 725a04ba5a95e89c89633d4322430cfbca7ce128)

More Info: <https://avd.aquasec.com/nvd/cve-2025-21968>

**[CVE-2025-21969] kernel: Bluetooth: L2CAP: Fix slab-use-after-free Read in l2cap\_send\_cmd (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: L2CAP: Fix slab-use-after-free Read in l2cap\_send\_cmd

After the hci sync command releases l2cap\_conn, the hci receive data work queue references the released l2cap\_conn when sending to the upper layer. Add hci dev lock to the hci receive data work queue to synchronize the two.



[1]

BUG: KASAN: slab-use-after-free in l2cap\_send\_cmd+0x187/0x8d0 net/bluetooth/l2cap\_core.c:954  
Read of size 8 at addr ffff8880271a4000 by task kworker/u9:2/5837

CPU: 0 UID: 0 PID: 5837 Comm: kworker/u9:2 Not tainted 6.13.0-rc5-syzkaller-00163-gab75170520d4 #0

Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024

Workqueue: hci1 hci\_rx\_work

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:378 [inline]  
print\_report+0x169/0x550 mm/kasan/report.c:489  
kasan\_report+0x143/0x180 mm/kasan/report.c:602  
l2cap\_build\_cmd net/bluetooth/l2cap\_core.c:2964 [inline]  
l2cap\_send\_cmd+0x187/0x8d0 net/bluetooth/l2cap\_core.c:954  
l2cap\_sig\_send\_rej net/bluetooth/l2cap\_core.c:5502 [inline]  
l2cap\_sig\_channel net/bluetooth/l2cap\_core.c:5538 [inline]  
l2cap\_rcv\_frame+0x221f/0x10db0 net/bluetooth/l2cap\_core.c:6817  
hci\_acldata\_packet net/bluetooth/hci\_core.c:3797 [inline]  
hci\_rx\_work+0x508/0xdb0 net/bluetooth/hci\_core.c:4040  
process\_one\_work kernel/workqueue.c:3229 [inline]  
process\_scheduled\_works+0xa66/0x1840 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 5837:

kasan\_save\_stack mm/kasan/common.c:47 [inline]  
kasan\_save\_track+0x3f/0x80 mm/kasan/common.c:68  
poison\_kmalloc\_redzone mm/kasan/common.c:377 [inline]  
\_\_kasan\_kmalloc+0x98/0xb0 mm/kasan/common.c:394  
kasan\_kmalloc include/linux/kasan.h:260 [inline]  
\_\_kmalloc\_cache\_noprof+0x243/0x390 mm/slub.c:4329  
kmalloc\_noprof include/linux/slab.h:901 [inline]  
kzalloc\_noprof include/linux/slab.h:1037 [inline]  
l2cap\_conn\_add+0xa9/0x8e0 net/bluetooth/l2cap\_core.c:6860  
l2cap\_connect\_cfm+0x115/0x1090 net/bluetooth/l2cap\_core.c:7239  
hci\_connect\_cfm include/net/bluetooth/hci\_core.h:2057 [inline]  
hci\_remote\_features\_evt+0x68e/0xac0 net/bluetooth/hci\_event.c:3726  
hci\_event\_func net/bluetooth/hci\_event.c:7473 [inline]  
hci\_event\_packet+0xac2/0x1540 net/bluetooth/hci\_event.c:7525  
hci\_rx\_work+0x3f3/0xdb0 net/bluetooth/hci\_core.c:4035  
process\_one\_work kernel/workqueue.c:3229 [inline]  
process\_scheduled\_works+0xa66/0x1840 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

Freed by task 54:

```
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:582
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:233 [inline]
slab_free_hook mm/slub.c:2353 [inline]
slab_free mm/slub.c:4613 [inline]
kfree+0x196/0x430 mm/slub.c:4761
l2cap_connect_cfm+0xcc/0x1090 net/bluetooth/l2cap_core.c:7235
hci_connect_cfm include/net/bluetooth/hci_core.h:2057 [inline]
hci_conn_failed+0x287/0x400 net/bluetooth/hci_conn.c:1266
hci_abort_conn_sync+0x56c/0x11f0 net/bluetooth/hci_sync.c:5603
hci_cmd_sync_work+0x22b/0x400 net/bluetooth/hci_sync.c:332
process_one_work kernel/workqueue.c:3229 [inline]
process_scheduled_works+0xa66/0x1840 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/entr
---truncated---
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21969>

### **[CVE-2025-21970] kernel: net/mlx5: Bridge, fix the crash caused by LAG state check (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5: Bridge, fix the crash caused by LAG state check

When removing LAG device from bridge, NETDEV\_CHANGEUPPER event is triggered. Driver finds the lower devices (PFs) to flush all the offloaded entries. And `mlx5_lag_is_shared_fdb` is checked, it returns false if one of PF is unloaded. In such case, `mlx5_esw_bridge_lag_rep_get()` and its caller return NULL, instead of the alive PF, and the flush is skipped.

Besides, the bridge fdb entry's `lastuse` is updated in `mlx5` bridge event handler. But this `SWITCHDEV_FDB_ADD_TO_BRIDGE` event can be ignored in this case because the upper interface for bond is deleted, and the entry will never be aged because `lastuse` is never updated.

To make things worse, as the entry is alive, `mlx5` bridge workqueue keeps sending that event, which is then handled by kernel bridge notifier. It causes the following crash when accessing the passed bond `netdev` which is already destroyed.

To fix this issue, remove such checks. LAG state is already checked in

commit 15f8f168952f ("net/mlx5: Bridge, verify LAG state when adding bond to bridge"), driver still need to skip offload if LAG becomes invalid state after initialization.

Oops: stack segment: 0000 [#1] SMP

CPU: 3 UID: 0 PID: 23695 Comm: kworker/u40:3 Tainted: G OE 6.11.0\_mlnx #1

Tainted: [O]=OOT\_MODULE, [E]=UNSIGNED\_MODULE

Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014

Workqueue: mlx5\_bridge\_wq mlx5\_esw\_bridge\_update\_work [mlx5\_core]

RIP: 0010:br\_switchdev\_event+0x2c/0x110 [bridge]

Code: 44 00 00 48 8b 02 48 f7 00 00 02 00 00 74 69 41 54 55 53 48 83 ec 08 48 8b a8 08 01 00 00 48 85 ed 74 4a 48 83 fe 02 48 89 d3 <4c> 8b 65 00 74 23 76 49 48 83 fe 05 74 7e 48 83 fe 06 75 2f 0f b7

RSP: 0018:ffff900092cfd0 EFLAGS: 00010297

RAX: ffff888123bfe000 RBX: ffff900092cfe08 RCX: 00000000ffffffff

RDX: ffff900092cfe08 RSI: 0000000000000001 RDI: ffffffff0c585f0

RBP: 6669746f6e690a30 R08: 0000000000000000 R09: ffff888123ae92c8

R10: 0000000000000000 R11: fefefefefefeff R12: ffff888123ae9c60

R13: 0000000000000001 R14: ffff900092cfe08 R15: 0000000000000000

FS: 0000000000000000(0000) GS:ffff88852c980000(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033

CR2: 00007f15914c8734 CR3: 0000000002830005 CR4: 0000000000770ef0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400

PKRU: 55555554

Call Trace:

<TASK>

? \_\_die\_body+0x1a/0x60

? die+0x38/0x60

? do\_trap+0x10b/0x120

? do\_error\_trap+0x64/0xa0

? exc\_stack\_segment+0x33/0x50

? asm\_exc\_stack\_segment+0x22/0x30

? br\_switchdev\_event+0x2c/0x110 [bridge]

? sched\_balance\_newidle.isra.149+0x248/0x390

notifier\_call\_chain+0x4b/0xa0

atomic\_notifier\_call\_chain+0x16/0x20

mlx5\_esw\_bridge\_update+0xec/0x170 [mlx5\_core]

mlx5\_esw\_bridge\_update\_work+0x19/0x40 [mlx5\_core]

process\_scheduled\_works+0x81/0x390

worker\_thread+0x106/0x250

? bh\_worker+0x110/0x110

kthread+0xb7/0xe0

? kthread\_park+0x80/0x80

ret\_from\_fork+0x2d/0x50

? kthread\_park+0x80/0x80

ret\_from\_fork\_asm+0x11/0x20

</TASK>

More Info: <https://avd.aquasec.com/nvd/cve-2025-21970>

**[CVE-2025-21971] kernel: net\_sched: Prevent creation of classes with TC\_H\_ROOT (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net\_sched: Prevent creation of classes with TC\_H\_ROOT

The function `qdisc_tree_reduce_backlog()` uses `TC_H_ROOT` as a termination condition when traversing up the qdisc tree to update parent backlog counters. However, if a class is created with classid `TC_H_ROOT`, the traversal terminates prematurely at this class instead of reaching the actual root qdisc, causing parent statistics to be incorrectly maintained. In case of DRR, this could lead to a crash as reported by Mingi Cho.

Prevent the creation of any Qdisc class with classid `TC_H_ROOT` (`0xFFFFFFFF`) across all qdisc types, as suggested by Jamal.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21971>

### **[CVE-2025-21972] kernel: net: mctp: unshare packets when reassembling (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net: mctp: unshare packets when reassembling

Ensure that the `frag_list` used for reassembly isn't shared with other packets. This avoids incorrect reassembly when packets are cloned, and prevents a memory leak due to circular references between fragments and their `skb_shared_info`.

The upcoming MCTP-over-USB driver uses `skb_clone` which can trigger the problem - other MCTP drivers don't share SKBs.

A kunit test is added to reproduce the issue.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21972>

### **[CVE-2025-21975] kernel: net/mlx5: handle errors in `mlx5_chains_create_table()` (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net/mlx5: handle errors in `mlx5_chains_create_table()`

In `mlx5_chains_create_table()`, the return value of `mlx5_get_fdb_sub_ns()` and `mlx5_get_flow_namespace()` must be checked to prevent NULL pointer

dereferences. If either function fails, the function should log error message with `mlx5_core_warn()` and return error pointer.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21975>

### **[CVE-2025-21976] kernel: fbdev: hyperv\_fb: Allow graceful removal of framebuffer (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

fbdev: hyperv\_fb: Allow graceful removal of framebuffer

When a Hyper-V framebuffer device is unbind, hyperv\_fb driver tries to release the framebuffer forcefully. If this framebuffer is in use it produce the following WARN and hence this framebuffer is never released.

```
[ 44.111220] WARNING: CPU: 35 PID: 1882 at drivers/video/fbdev/core/fb_info.c:70 framebuffer_release+0x2c/0x40
< snip >
[ 44.111289] Call Trace:
[ 44.111290] <TASK>
[ 44.111291] ? show_regs+0x6c/0x80
[ 44.111295] ? __warn+0x8d/0x150
[ 44.111298] ? framebuffer_release+0x2c/0x40
[ 44.111300] ? report_bug+0x182/0x1b0
[ 44.111303] ? handle_bug+0x6e/0xb0
[ 44.111306] ? exc_invalid_op+0x18/0x80
[ 44.111308] ? asm_exc_invalid_op+0x1b/0x20
[ 44.111311] ? framebuffer_release+0x2c/0x40
[ 44.111313] ? hvfb_remove+0x86/0xa0 [hyperv_fb]
[ 44.111315] vmbus_remove+0x24/0x40 [hv_vmbus]
[ 44.111323] device_remove+0x40/0x80
[ 44.111325] device_release_driver_internal+0x20b/0x270
[ 44.111327] ? bus_find_device+0xb3/0xf0
```

Fix this by moving the release of framebuffer and associated memory to `fb_ops.fb_destroy` function, so that framebuffer framework handles it gracefully.

While we fix this, also replace manual registrations/unregistration of framebuffer with `devm_register_framebuffer`.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21976>

### **[CVE-2025-21978] kernel: drm/hyperv: Fix address space leak when Hyper-V DRM device is removed (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

drm/hyperv: Fix address space leak when Hyper-V DRM device is removed

When a Hyper-V DRM device is probed, the driver allocates MMIO space for the vram, and maps it cacheable. If the device removed, or in the error path for device probing, the MMIO space is released but no unmap is done. Consequently the kernel address space for the mapping is leaked.

Fix this by adding iounmap() calls in the device removal path, and in the error path during device probing.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21978>

**[CVE-2025-21980] kernel: sched: address a potential NULL pointer dereference in the GRED scheduler. (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

sched: address a potential NULL pointer dereference in the GRED scheduler.

If kzalloc in gred\_init returns a NULL pointer, the code follows the error handling path, invoking gred\_destroy. This, in turn, calls gred\_offload, where memset could receive a NULL pointer as input, potentially leading to a kernel crash.

When table->opt is NULL in gred\_init(), gred\_change\_table\_def() is not called yet, so it is not necessary to call ->ndo\_setup\_tc() in gred\_offload().

More Info: <https://avd.aquasec.com/nvd/cve-2025-21980>

**[CVE-2025-21981] kernel: ice: fix memory leak in aRFS after reset (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ice: fix memory leak in aRFS after reset

Fix aRFS (accelerated Receive Flow Steering) structures memory leak by adding a checker to verify if aRFS memory is already allocated while configuring VSI. aRFS objects are allocated in two cases:

- as part of VSI initialization (at probe), and
- as part of reset handling

However, VSI reconfiguration executed during reset involves memory allocation one more time, without prior releasing already allocated resources. This led to the memory leak with the following signature:

```
[root@os-delivery ~]# cat /sys/kernel/debug/kmemleak
unreferenced object 0xff3c1ca7252e6000 (size 8192):
  comm "kworker/0:0", pid 8, jiffies 4296833052
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
  backtrace (crc 0):
    [<ffffffff991ec485>] __kmalloc_cache_noprof+0x275/0x340
    [<ffffffffc0a6e06a>] ice_init_arfs+0x3a/0xe0 [ice]
    [<ffffffffc09f1027>] ice_vsi_cfg_def+0x607/0x850 [ice]
    [<ffffffffc09f244b>] ice_vsi_setup+0x5b/0x130 [ice]
    [<ffffffffc09c2131>] ice_init+0x1c1/0x460 [ice]
    [<ffffffffc09c64af>] ice_probe+0x2af/0x520 [ice]
    [<ffffffff994fbcd3>] local_pci_probe+0x43/0xa0
    [<ffffffff98f07103>] work_for_cpu_fn+0x13/0x20
    [<ffffffff98f0b6d9>] process_one_work+0x179/0x390
    [<ffffffff98f0c1e9>] worker_thread+0x239/0x340
    [<ffffffff98f14abc>] kthread+0xcc/0x100
    [<ffffffff98e45a6d>] ret_from_fork+0x2d/0x50
    [<ffffffff98e083ba>] ret_from_fork_asm+0x1a/0x30
    ...
```

More Info: <https://avd.aquasec.com/nvd/cve-2025-21981>

### **[CVE-2025-21985] kernel: drm/amd/display: Fix out-of-bound accesses (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

drm/amd/display: Fix out-of-bound accesses

#### **[WHAT & HOW]**

hpo\_stream\_to\_link\_encoder\_mapping has size MAX\_HPO\_DP2\_ENCODERS(=4), but location can have size up to 6. As a result, it is necessary to check location against MAX\_HPO\_DP2\_ENCODERS.

Similiarly, disp\_cfg\_stream\_location can be used as an array index which should be 0..5, so the ASSERT's conditions should be less without equal.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21985>

### **[CVE-2025-21986] kernel: net: switchdev: Convert blocking notification chain to a raw one (Severity: MEDIUM)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net: switchdev: Convert blocking notification chain to a raw one

A blocking notification chain uses a read-write semaphore to protect the integrity of the chain. The semaphore is acquired for writing when adding / removing notifiers to / from the chain and acquired for reading when traversing the chain and informing notifiers about an event.

In case of the blocking switchdev notification chain, recursive notifications are possible which leads to the semaphore being acquired twice for reading and to lockdep warnings being generated [1].

Specifically, this can happen when the bridge driver processes a SWITCHDEV\_BRPORT\_UNOFFLOADED event which causes it to emit notifications about deferred events when calling switchdev\_deferred\_process().

Fix this by converting the notification chain to a raw notification chain in a similar fashion to the netdev notification chain. Protect the chain using the RTNL mutex by acquiring it when modifying the chain. Events are always informed under the RTNL mutex, but add an assertion in call\_switchdev\_blocking\_notifiers() to make sure this is not violated in the future.

Maintain the "blocking" prefix as events are always emitted from process context and listeners are allowed to block.

[1]:

WARNING: possible recursive locking detected

6.14.0-rc4-custom-g079270089484 #1 Not tainted

-----

ip/52731 is trying to acquire lock:

ffffffff850918d8 ((switchdev\_blocking\_notif\_chain).rwsem){++++}-{4:4}, at: blocking\_notifier\_call\_chain+0x58/0xa0

but task is already holding lock:

ffffffff850918d8 ((switchdev\_blocking\_notif\_chain).rwsem){++++}-{4:4}, at: blocking\_notifier\_call\_chain+0x58/0xa0

other info that might help us debug this:

Possible unsafe locking scenario:

CPU0

----

lock((switchdev\_blocking\_notif\_chain).rwsem);

lock((switchdev\_blocking\_notif\_chain).rwsem);

\*\*\* DEADLOCK \*\*\*

May be due to missing lock nesting notation

3 locks held by ip/52731:

#0: ffffffff84f795b0 (rtnl\_mutex){+.+.}-{4:4}, at: rtnl\_newlink+0x727/0x1dc0

#1: ffffffff8731f628 (&net->rtnl\_mutex){+.+.}-{4:4}, at: rtnl\_newlink+0x790/0x1dc0

#2: ffffffff850918d8 ((switchdev\_blocking\_notif\_chain).rwsem){++++}-{4:4}, at: blocking\_notifier\_call\_chain+0x58/0xa0

stack backtrace:

...

? \_\_pfx\_down\_read+0x10/0x10

? \_\_pfx\_mark\_lock+0x10/0x10



? \_\_pfx\_switchdev\_port\_attr\_set\_deferred+0x10/0x10  
blocking\_notifier\_call\_chain+0x58/0xa0  
switchdev\_port\_attr\_notify.constprop.0+0xb3/0x1b0  
? \_\_pfx\_switchdev\_port\_attr\_notify.constprop.0+0x10/0x10  
? mark\_held\_locks+0x94/0xe0  
? switchdev\_deferred\_process+0x11a/0x340  
switchdev\_port\_attr\_set\_deferred+0x27/0xd0  
switchdev\_deferred\_process+0x164/0x340  
br\_switchdev\_port\_unoffload+0xc8/0x100 [bridge]  
br\_switchdev\_blocking\_event+0x29f/0x580 [bridge]  
notifier\_call\_chain+0xa2/0x440  
blocking\_notifier\_call\_chain+0x6e/0xa0  
switchdev\_bridge\_port\_unoffload+0xde/0x1a0  
...

More Info: <https://avd.aquasec.com/nvd/cve-2025-21986>

### **[CVE-2025-21992] kernel: HID: ignore non-functional sensor in HP 5MP Camera (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

HID: ignore non-functional sensor in HP 5MP Camera

The HP 5MP Camera (USB ID 0408:5473) reports a HID sensor interface that is not actually implemented. Attempting to access this non-functional sensor via iio\_info causes system hangs as runtime PM tries to wake up an unresponsive sensor.

[453] hid-sensor-hub 0003:0408:5473.0003: Report latency attributes: ffffffff:fffffff

[453] hid-sensor-hub 0003:0408:5473.0003: common attributes: 5:1, 2:1, 3:1 ffffffff:fffffff

Add this device to the HID ignore list since the sensor interface is non-functional by design and should not be exposed to userspace.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21992>

### **[CVE-2025-21994] kernel: ksmbd: fix incorrect validation for num\_aces field of smb\_acl (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ksmbd: fix incorrect validation for num\_aces field of smb\_acl

parse\_dcal() validate num\_aces to allocate posix\_ace\_state\_array.

if (num\_aces > ULONG\_MAX / sizeof(struct smb\_ace \*))

It is an incorrect validation that we can create an array of size `ULONG_MAX`.  
`smb_acl` has `->size` field to calculate actual number of aces in request buffer size. Use this to check invalid `num_aces`.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21994>

**[CVE-2025-21996] kernel: drm/radeon: fix uninitialized size issue in radeon\_vce\_cs\_parse() (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

`drm/radeon: fix uninitialized size issue in radeon_vce_cs_parse()`

On the off chance that command stream passed from userspace via `ioctl()` call to `radeon_vce_cs_parse()` is weirdly crafted and first command to execute is to encode (case `0x03000001`), the function in question will attempt to call `radeon_vce_cs_reloc()` with size argument that has not been properly initialized. Specifically, `'size'` will point to `'tmp'` variable before the latter had a chance to be assigned any value.

Play it safe and init `'tmp'` with 0, thus ensuring that `radeon_vce_cs_reloc()` will catch an early error in cases like these.

Found by Linux Verification Center ([linuxtesting.org](https://linuxtesting.org)) with static analysis tool SVACE.

(cherry picked from commit `2d52de55f9ee7aaee0e09ac443f77855989c6b68`)

More Info: <https://avd.aquasec.com/nvd/cve-2025-21996>

**[CVE-2025-21997] kernel: xsk: fix an integer overflow in xp\_create\_and\_assign\_umem() (Severity: MEDIUM)**

Package: `linux-libc-dev`

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

`xsk: fix an integer overflow in xp_create_and_assign_umem()`

Since the `i` and `pool->chunk_size` variables are of type `'u32'`, their product can wrap around and then be cast to `'u64'`. This can lead to two different XDP buffers pointing to the same memory area.

Found by InfoTeCS on behalf of Linux Verification Center ([linuxtesting.org](https://linuxtesting.org)) with SVACE.

More Info: <https://avd.aquasec.com/nvd/cve-2025-21997>

**[CVE-2025-22004] kernel: net: atm: fix use after free in lec\_send() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

net: atm: fix use after free in lec\_send()

The ->send() operation frees skb so save the length before calling ->send() to avoid a use after free.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22004>

**[CVE-2025-22005] kernel: ipv6: Fix memleak of nhc\_pcpu\_rth\_output in fib\_check\_nh\_v6\_gw(). (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

ipv6: Fix memleak of nhc\_pcpu\_rth\_output in fib\_check\_nh\_v6\_gw().

fib\_check\_nh\_v6\_gw() expects that fib6\_nh\_init() cleans up everything when it fails.

Commit 7dd73168e273 ("ipv6: Always allocate pcpu memory in a fib6\_nh") moved fib\_nh\_common\_init() before alloc\_percpu\_gfp() within fib6\_nh\_init() but forgot to add cleanup for fib6\_nh->nh\_common.nhc\_pcpu\_rth\_output in case it fails to allocate fib6\_nh->rt6i\_pcpu, resulting in memleak.

Let's call fib\_nh\_common\_release() and clear nhc\_pcpu\_rth\_output in the error path.

Note that we can remove the fib6\_nh\_release() call in nh\_create\_ipv6() later in net-next.git.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22005>

**[CVE-2025-22007] kernel: Bluetooth: Fix error code in chan\_alloc\_skb\_cb() (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

Bluetooth: Fix error code in chan\_alloc\_skb\_cb()

The `chan_alloc_skb_cb()` function is supposed to return error pointers on error. Returning NULL will lead to a NULL dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22007>

**[CVE-2025-22008] kernel: regulator: check that dummy regulator has been probed before using it (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

regulator: check that dummy regulator has been probed before using it

Due to asynchronous driver probing there is a chance that the dummy regulator hasn't already been probed when first accessing it.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22008>

**[CVE-2025-22010] kernel: RDMA/hns: Fix soft lockup during bt pages loop (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

RDMA/hns: Fix soft lockup during bt pages loop

Driver runs a for-loop when allocating bt pages and mapping them with buffer pages. When a large buffer (e.g. MR over 100GB) is being allocated, it may require a considerable loop count. This will lead to soft lockup:

```
watchdog: BUG: soft lockup - CPU#27 stuck for 22s!
```

```
...
```

```
Call trace:
```

```
hem_list_alloc_mid_bt+0x124/0x394 [hns_roce_hw_v2]
hns_roce_hem_list_request+0xf8/0x160 [hns_roce_hw_v2]
hns_roce_mtr_create+0x2e4/0x360 [hns_roce_hw_v2]
alloc_mr_pbl+0xd4/0x17c [hns_roce_hw_v2]
hns_roce_reg_user_mr+0xf8/0x190 [hns_roce_hw_v2]
ib_uverbs_reg_mr+0x118/0x290
```

```
watchdog: BUG: soft lockup - CPU#35 stuck for 23s!
```

```
...
```

```
Call trace:
```

```
hns_roce_hem_list_find_mtt+0x7c/0xb0 [hns_roce_hw_v2]
mtr_map_bufs+0xc4/0x204 [hns_roce_hw_v2]
hns_roce_mtr_create+0x31c/0x3c4 [hns_roce_hw_v2]
alloc_mr_pbl+0xb0/0x160 [hns_roce_hw_v2]
hns_roce_reg_user_mr+0x108/0x1c0 [hns_roce_hw_v2]
ib_uverbs_reg_mr+0x120/0x2bc
```

Add a `cond_resched()` to fix soft lockup during these loops. In order not to affect the allocation performance of normal-size buffer, set the loop count of a 100GB MR as the threshold to call `cond_resched()`.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22010>

### **[CVE-2025-22014] kernel: soc: qcom: pdr: Fix the potential deadlock (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

soc: qcom: pdr: Fix the potential deadlock

When some client process A call `pdr_add_lookup()` to add the look up for the service and does schedule locator work, later a process B got a new server packet indicating locator is up and call `pdr_locator_new_server()` which eventually sets `pdr->locator_init_complete` to true which process A sees and takes list lock and queries domain list but it will timeout due to deadlock as the response will queued to the same `qmi->wq` and it is ordered workqueue and process B is not able to complete new server request work due to deadlock on list lock.

Fix it by removing the unnecessary list iteration as the list iteration is already being done inside locator work, so avoid it here and just call `schedule_work()` here.

Process A	Process B
	<code>process_scheduled_works()</code>
<code>pdr_add_lookup()</code>	<code>qmi_data_ready_work()</code>
<code>process_scheduled_works()</code>	<code>pdr_locator_new_server()</code>
	<code>pdr-&gt;locator_init_complete=true;</code>
<code>pdr_locator_work()</code>	
<code>mutex_lock(&amp;pdr-&gt;list_lock);</code>	
<code>pdr_locate_service()</code>	<code>mutex_lock(&amp;pdr-&gt;list_lock);</code>
<code>pdr_get_domain_list()</code>	
<code>pr_err("PDR: %s get domain list</code>	
<code>txn wait failed: %d\n",</code>	
<code>req-&gt;service_name,</code>	
<code>ret);</code>	

Timeout error log due to deadlock:

```
"
PDR: tms/servreg get domain list txn wait failed: -110
PDR: service lookup for msm/adsp/sensor_pd:tms/servreg failed: -110
"
```

Thanks to Bjorn and Johan for letting me know that this commit also fixes an audio regression when using the in-kernel pd-mapper as that makes it easier to hit this race. [1]

More Info: <https://avd.aquasec.com/nvd/cve-2025-22014>

### **[CVE-2025-22015] kernel: mm/migrate: fix shmem xarray update during migration (Severity: MEDIUM)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: 6.1.133-1

In the Linux kernel, the following vulnerability has been resolved:

mm/migrate: fix shmem xarray update during migration

A shmem folio can be either in page cache or in swap cache, but not at the same time. Namely, once it is in swap cache, folio->mapping should be NULL, and the folio is no longer in a shmem mapping.

In `__folio_migrate_mapping()`, to determine the number of xarray entries to update, `folio_test_swapbacked()` is used, but that conflates shmem in page cache case and shmem in swap cache case. It leads to xarray multi-index entry corruption, since it turns a sibling entry to a normal entry during `xas_store()` (see [1] for a userspace reproduction). Fix it by only using `folio_test_swapcache()` to determine whether xarray is storing swap cache entries or not to choose the right number of xarray entries to update.

[1] <https://lore.kernel.org/linux-mm/Z8idPCkaJW1lChjT@casper.infradead.org/>

Note:

In `__split_huge_page()`, `folio_test_anon()` && `folio_test_swapcache()` is used to get swap\_cache address space, but that ignores the shmem folio in swap cache case. It could lead to NULL pointer dereferencing when a in-swap-cache shmem folio is split at `__xa_store()`, since `!folio_test_anon()` is true and `folio->mapping` is NULL. But fortunately, its caller `split_huge_page_to_list_to_order()` bails out early with EBUSY when `folio->mapping` is NULL. So no need to take care of it here.

More Info: <https://avd.aquasec.com/nvd/cve-2025-22015>

### **[CVE-2004-0230] TCP, when using a large Window Size, makes it easier for remote attack ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

TCP, when using a large Window Size, makes it easier for remote attackers to guess sequence numbers and cause a denial of service (connection loss) to persistent TCP connections by repeatedly injecting a TCP RST packet, especially in protocols that use long-lived connections, such as BGP.

More Info: <https://avd.aquasec.com/nvd/cve-2004-0230>

**[CVE-2005-3660] Linux kernel 2.4 and 2.6 allows attackers to cause a denial of service ... (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

Linux kernel 2.4 and 2.6 allows attackers to cause a denial of service (memory exhaustion and panic) by creating a large number of connected file descriptors or socketpairs and setting a large data transfer buffer, then preventing Linux from being able to finish the transfer by causing the process to become a zombie, or closing the file descriptor without closing an associated reference.

More Info: <https://avd.aquasec.com/nvd/cve-2005-3660>

**[CVE-2007-3719] kernel: secretly Monopolizing the CPU Without Superuser Privileges (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

The process scheduler in the Linux kernel 2.6.16 gives preference to "interactive" processes that perform voluntary sleeps, which allows local users to cause a denial of service (CPU consumption), as described in "Secretly Monopolizing the CPU Without Superuser Privileges."

More Info: <https://avd.aquasec.com/nvd/cve-2007-3719>

**[CVE-2008-2544] kernel: mounting proc readonly on a different mount point silently mounts it rw if the /proc mount is rw (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

Mounting /proc filesystem via chroot command silently mounts it in read-write mode. The user could bypass the chroot environment and gain write access to files, he would never have otherwise.

More Info: <https://avd.aquasec.com/nvd/cve-2008-2544>

**[CVE-2008-4609] kernel: TCP protocol vulnerabilities from Outpost24 (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

The TCP implementation in (1) Linux, (2) platforms based on BSD Unix, (3) Microsoft Windows, (4) Cisco products, and probably other operating systems allows remote attackers to cause a denial of service (connection queue exhaustion) via multiple vectors that manipulate information in the TCP state table, as demonstrated by sockstress.

More Info: <https://avd.aquasec.com/nvd/cve-2008-4609>

**[CVE-2010-4563] kernel: ipv6: sniffer detection (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

The Linux kernel, when using IPv6, allows remote attackers to determine whether a host is sniffing the network by sending an ICMPv6 Echo Request to a multicast address and determining whether an Echo Reply is sent, as demonstrated by thcping.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4563>

**[CVE-2010-5321] kernel: v4l: videobuf: hotfix a bug on multiple calls to mmap() (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

Memory leak in drivers/media/video/videobuf-core.c in the videobuf subsystem in the Linux kernel 2.6.x through 4.x allows local users to cause a denial of service (memory consumption) by leveraging /dev/video access for a series of mmap calls that require new allocations, a different vulnerability than CVE-2007-6761. NOTE: as of 2016-06-18, this affects only 11 drivers that have not been updated to use videobuf2 instead of videobuf.

More Info: <https://avd.aquasec.com/nvd/cve-2010-5321>

**[CVE-2011-4915] fs/proc/base.c in the Linux kernel through 3.1 allows local users to o ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

fs/proc/base.c in the Linux kernel through 3.1 allows local users to obtain sensitive keystroke information via access to /proc/interrupts.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4915>

**[CVE-2011-4916] Linux kernel through 3.1 allows local users to obtain sensitive keystr ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

Linux kernel through 3.1 allows local users to obtain sensitive keystroke information via access to /dev/pts/ and /dev/tty\*.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4916>

**[CVE-2011-4917] In the Linux kernel through 3.1 there is an information disclosure iss ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel through 3.1 there is an information disclosure issue via /proc/stat.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4917>

**[CVE-2012-4542] kernel: block: default SCSI command filter does not accomodate commands overlap across device classes (Severity: LOW)**



Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

block/scsi\_ioctls.c in the Linux kernel through 3.8 does not properly consider the SCSI device class during authorization of SCSI commands, which allows local users to bypass intended access restrictions via an SG\_IO ioctl call that leverages overlapping opcodes.

More Info: <https://avd.aquasec.com/nvd/cve-2012-4542>

**[CVE-2014-9892] The snd\_compr\_tstamp function in sound/core/compress\_offload.c in the ... (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

The snd\_compr\_tstamp function in sound/core/compress\_offload.c in the Linux kernel through 4.7, as used in Android before 2016-08-05 on Nexus 5 and 7 (2013) devices, does not properly initialize a timestamp data structure, which allows attackers to obtain sensitive information via a crafted application, aka Android internal bug 28770164 and Qualcomm internal bug CR568717.

More Info: <https://avd.aquasec.com/nvd/cve-2014-9892>

**[CVE-2014-9900] kernel: Info leak in uninitialized structure ethtool\_wolinfo in ethtool\_get\_wol() (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

The ethtool\_get\_wol function in net/core/ethtool.c in the Linux kernel through 4.7, as used in Android before 2016-08-05 on Nexus 5 and 7 (2013) devices, does not initialize a certain data structure, which allows local users to obtain sensitive information via a crafted application, aka Android internal bug 28803952 and Qualcomm internal bug CR570754.

More Info: <https://avd.aquasec.com/nvd/cve-2014-9900>

**[CVE-2015-2877] Kernel: Cross-VM ASL INTrospection (CAIN) (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %!s(<nil>)

Kernel Samepage Merging (KSM) in the Linux kernel 2.6.32 through 4.x does not prevent use of a write-timing side channel, which allows guest OS users to defeat the ASLR protection mechanism on other guest OS instances via a Cross-VM ASL INTrospection (CAIN) attack. NOTE: the vendor states "Basically if you care about this attack vector, disable deduplication." Share-until-written approaches for memory conservation among mutually untrusting tenants are inherently detectable for information disclosure, and can be classified as potentially misunderstood behaviors rather than vulnerabilities

More Info: <https://avd.aquasec.com/nvd/cve-2015-2877>

**[CVE-2016-10723] An issue was discovered in the Linux kernel through 4.17.2. Since the ... (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 4.17.2. Since the page allocator does not yield CPU resources to the owner of the oom\_lock mutex, a local unprivileged user can trivially lock up the system forever by wasting CPU resources from the page allocator (e.g., via concurrent page fault events) when the global OOM killer is invoked. NOTE: the software maintainer has not accepted certain proposed patches, in part because of a viewpoint that "the underlying problem is non-trivial to handle."

More Info: <https://avd.aquasec.com/nvd/cve-2016-10723>

**[CVE-2016-8660] kernel: xfs: local DoS due to a page lock order bug in the XFS seek hole/data implementation (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

The XFS subsystem in the Linux kernel through 4.8.2 allows local users to cause a denial of service (fdatasync failure and system hang) by using the vfs syscall group in the trinity program, related to a "page lock order bug in the XFS seek hole/data implementation."

More Info: <https://avd.aquasec.com/nvd/cve-2016-8660>

**[CVE-2017-0630] kernel: Information disclosure vulnerability in kernel trace subsystem (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An information disclosure vulnerability in the kernel trace subsystem could enable a local malicious application to access data outside of its permission levels. This issue is rated as Moderate because it first requires compromising a privileged process. Product: Android. Versions: Kernel-3.10, Kernel-3.18. Android ID: A-34277115.

More Info: <https://avd.aquasec.com/nvd/cve-2017-0630>

**[CVE-2017-13693] kernel: ACPI operand cache leak in dsutils.c (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

The acpi\_ds\_create\_operands() function in drivers/acpi/acpica/dsutils.c in the Linux kernel through 4.12.9 does not flush the operand cache and causes a kernel stack dump, which allows local users to obtain sensitive information from kernel memory and bypass the KASLR protection mechanism (in the kernel through 4.9) via a crafted ACPI table.

More Info: <https://avd.aquasec.com/nvd/cve-2017-13693>

**[CVE-2017-13694] kernel: ACPI node and node\_ext cache leak (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

The `acpi_ps_complete_final_op()` function in `drivers/acpi/acpica/psobject.c` in the Linux kernel through 4.12.9 does not flush the node and node\_ext caches and causes a kernel stack dump, which allows local users to obtain sensitive information from kernel memory and bypass the KASLR protection mechanism (in the kernel through 4.9) via a crafted ACPI table.

More Info: <https://avd.aquasec.com/nvd/cve-2017-13694>

### **[CVE-2018-1121] procs: process hiding through race condition enumerating /proc (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

procs-ng, procs is vulnerable to a process hiding through race condition. Since the kernel's `proc_pid_readdir()` returns PID entries in ascending numeric order, a process occupying a high PID can use inotify events to determine when the process list is being scanned, and fork/exec to obtain a lower PID, thus avoiding enumeration. An unprivileged attacker can hide a process from procs-ng's utilities by exploiting a race condition in reading `/proc/PID` entries. This vulnerability affects procs and procs-ng up to version 3.3.15, newer versions might be affected also.

More Info: <https://avd.aquasec.com/nvd/cve-2018-1121>

### **[CVE-2018-12928] kernel: NULL pointer dereference in hfs\_ext\_read\_extent in hfs.ko (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel 4.15.0, a NULL pointer dereference was discovered in `hfs_ext_read_extent` in `hfs.ko`. This can occur during a mount of a crafted hfs filesystem.

More Info: <https://avd.aquasec.com/nvd/cve-2018-12928>

### **[CVE-2018-17977] kernel: Mishandled interactions among XFRM Netlink messages, IPPROTO\_AH packets, and IPPROTO\_IP packets resulting in a denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

The Linux kernel 4.14.67 mishandles certain interaction among XFRM Netlink messages, IPPROTO\_AH packets, and IPPROTO\_IP packets, which allows local users to cause a denial of service (memory consumption and system hang) by leveraging root access to execute crafted applications, as demonstrated on CentOS 7.

More Info: <https://avd.aquasec.com/nvd/cve-2018-17977>

### **[CVE-2019-11191] kernel: race condition in load\_aout\_binary() allows local users to bypass ASLR on setuid a.out programs (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

The Linux kernel through 5.0.7, when `CONFIG_IA32_AOUT` is enabled and `ia32_aout` is loaded, allows local users to bypass ASLR on setuid a.out programs (if any exist) because `install_exec_creds()` is called too late in `load_aout_binary()` in `fs/binfmt_aout.c`, and thus the `ptrace_may_access()` check has a race condition when reading `/proc/pid/stat`. NOTE: the software maintainer disputes that this is a vulnerability because ASLR for a.out format

executables has never been supported

More Info: <https://avd.aquasec.com/nvd/cve-2019-11191>

**[CVE-2019-12378] kernel: unchecked kmalloc of new\_ra in ip6\_ra\_control leads to denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in ip6\_ra\_control in net/ipv6/ipv6\_sockglue.c in the Linux kernel through 5.1.5. There is an unchecked kmalloc of new\_ra, which might allow an attacker to cause a denial of service (NULL pointer dereference and system crash). NOTE: This has been disputed as not an issue

More Info: <https://avd.aquasec.com/nvd/cve-2019-12378>

**[CVE-2019-12379] kernel: memory leak in con\_insert\_unipair in drivers/tty/vt/consolemap.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in con\_insert\_unipair in drivers/tty/vt/consolemap.c in the Linux kernel through 5.1.5. There is a memory leak in a certain case of an ENOMEM outcome of kmalloc. NOTE: This id is disputed as not being an issue

More Info: <https://avd.aquasec.com/nvd/cve-2019-12379>

**[CVE-2019-12380] kernel: memory allocation failure in the efi subsystem leads to denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

**\*\*DISPUTED\*\*** An issue was discovered in the efi subsystem in the Linux kernel through 5.1.5. phys\_efi\_set\_virtual\_address\_map in arch/x86/platform/efi/efi.c and efi\_call\_phys\_prolog in arch/x86/platform/efi/efi\_64.c mishandle memory allocation failures. NOTE: This id is disputed as not being an issue because "All the code touched by the referenced commit runs only at boot, before any user processes are started. Therefore, there is no possibility for an unprivileged user to control it."

More Info: <https://avd.aquasec.com/nvd/cve-2019-12380>

**[CVE-2019-12381] kernel: unchecked kmalloc of new\_ra in ip\_ra\_control leads to denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

An issue was discovered in ip\_ra\_control in net/ipv4/ip\_sockglue.c in the Linux kernel through 5.1.5. There is an unchecked kmalloc of new\_ra, which might allow an attacker to cause a denial of service (NULL pointer dereference and system crash). NOTE: this is disputed because new\_ra is never used if it is NULL

More Info: <https://avd.aquasec.com/nvd/cve-2019-12381>

**[CVE-2019-12382] kernel: unchecked kstrdup of fwstr in drm\_load\_edid\_firmware leads to denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in `drm_load_edid_firmware` in `drivers/gpu/drm/drm_edid_load.c` in the Linux kernel through 5.1.5. There is an unchecked `kstrdup` of `fwstr`, which might allow an attacker to cause a denial of service (NULL pointer dereference and system crash). NOTE: The vendor disputes this issues as not being a vulnerability because `kstrdup()` returning NULL is handled sufficiently and there is no chance for a NULL pointer dereference

More Info: <https://avd.aquasec.com/nvd/cve-2019-12382>

**[CVE-2019-12455] kernel: null pointer dereference in sunxi\_divs\_clk\_setup in drivers/clk/sunxi/clk-sunxi.c causing denial of service (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in `sunxi_divs_clk_setup` in `drivers/clk/sunxi/clk-sunxi.c` in the Linux kernel through 5.1.5. There is an unchecked `kstrndup` of `derived_name`, which might allow an attacker to cause a denial of service (NULL pointer dereference and system crash). NOTE: This id is disputed as not being an issue because “The memory allocation that was not checked is part of a code that only runs at boot time, before user processes are started. Therefore, there is no possibility for an unprivileged user to control it, and no denial of service.”

More Info: <https://avd.aquasec.com/nvd/cve-2019-12455>

**[CVE-2019-12456] kernel: double fetch in the MPT3COMMAND case in \_ctl\_ioctl\_main in drivers/scsi/mpt3sas/mpt3sas\_ctl.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in the MPT3COMMAND case in `_ctl_ioctl_main` in `drivers/scsi/mpt3sas/mpt3sas_ctl.c` in the Linux kernel through 5.1.5. It allows local users to cause a denial of service or possibly have unspecified other impact by changing the value of `ioc_number` between two kernel reads of that value, aka a "double fetch" vulnerability. NOTE: a third party reports that this is unexploitable because the doubly fetched value is not used

More Info: <https://avd.aquasec.com/nvd/cve-2019-12456>

**[CVE-2019-16229] kernel: null pointer dereference in drivers/gpu/drm/amd/amdkfd/kfd\_interrupt.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

`drivers/gpu/drm/amd/amdkfd/kfd_interrupt.c` in the Linux kernel 5.2.14 does not check the `alloc_workqueue` return value, leading to a NULL pointer dereference. NOTE: The security community disputes this issues as not being serious enough to be deserving a CVE id

More Info: <https://avd.aquasec.com/nvd/cve-2019-16229>

**[CVE-2019-16230] kernel: null pointer dereference in drivers/gpu/drm/radeon/radeon\_display.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

drivers/gpu/drm/radeon/radeon\_display.c in the Linux kernel 5.2.14 does not check the alloc\_workqueue return value, leading to a NULL pointer dereference. NOTE: A third-party software maintainer states that the work queue allocation is happening during device initialization, which for a graphics card occurs during boot. It is not attacker controllable and OOM at that time is highly unlikely

More Info: <https://avd.aquasec.com/nvd/cve-2019-16230>

**[CVE-2019-16231] kernel: null-pointer dereference in drivers/net/fjes/fjes\_main.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

drivers/net/fjes/fjes\_main.c in the Linux kernel 5.2.14 does not check the alloc\_workqueue return value, leading to a NULL pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2019-16231>

**[CVE-2019-16232] kernel: null-pointer dereference in drivers/net/wireless/marvell/libertas/if\_sdio.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

drivers/net/wireless/marvell/libertas/if\_sdio.c in the Linux kernel 5.2.14 does not check the alloc\_workqueue return value, leading to a NULL pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2019-16232>

**[CVE-2019-16233] kernel: null pointer dereference in drivers/scsi/qla2xxx/qla\_os.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

drivers/scsi/qla2xxx/qla\_os.c in the Linux kernel 5.2.14 does not check the alloc\_workqueue return value, leading to a NULL pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2019-16233>

**[CVE-2019-16234] kernel: null pointer dereference in drivers/net/wireless/intel/iwlwifi/pcie/trans.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

drivers/net/wireless/intel/iwlwifi/pcie/trans.c in the Linux kernel 5.2.14 does not check the alloc\_workqueue return value, leading to a NULL pointer dereference.

More Info: <https://avd.aquasec.com/nvd/cve-2019-16234>

**[CVE-2019-19070] kernel: A memory leak in the spi\_gpio\_probe() function in drivers/spi/spi-gpio.c allows for a DoS (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A memory leak in the spi\_gpio\_probe() function in drivers/spi/spi-gpio.c in the Linux kernel through 5.3.11 allows attackers to cause a denial of service (memory consumption) by triggering devm\_add\_action\_or\_reset() failures, aka CID-d3b0ffa1d75d. NOTE: third parties dispute the relevance of this because the system must have already been out of memory before the probe began

More Info: <https://avd.aquasec.com/nvd/cve-2019-19070>

**[CVE-2019-19378] kernel: out-of-bounds write in index\_rbio\_pages in fs/btrfs/raid56.c (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel 5.0.21, mounting a crafted btrfs filesystem image can lead to slab-out-of-bounds write access in index\_rbio\_pages in fs/btrfs/raid56.c.

More Info: <https://avd.aquasec.com/nvd/cve-2019-19378>

**[CVE-2020-11725] kernel: improper handling of private\_size\*count multiplication due to count=info->owner typo (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

snd\_ctl\_elem\_add in sound/core/control.c in the Linux kernel through 5.6.3 has a count=info->owner line, which later affects a private\_size\*count multiplication for unspecified "interesting side effects." NOTE: kernel engineers dispute this finding, because it could be relevant only if new callers were added that were unfamiliar with the misuse of the info->owner field to represent data unrelated to the "owner" concept. The existing callers, SNDRV\_CTL\_IOCTL\_ELEM\_ADD and SNDRV\_CTL\_IOCTL\_ELEM\_REPLACE, have been designed to misuse the info->owner field in a safe way

More Info: <https://avd.aquasec.com/nvd/cve-2020-11725>

**[CVE-2020-35501] kernel: audit not logging access to syscall open\_by\_handle\_at for users with CAP\_DAC\_READ\_SEARCH capability (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

A flaw was found in the Linux kernels implementation of audit rules, where a syscall can unexpectedly not be correctly not be logged by the audit subsystem

More Info: <https://avd.aquasec.com/nvd/cve-2020-35501>

**[CVE-2021-26934] An issue was discovered in the Linux kernel 4.18 through 5.10.16, as u ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

An issue was discovered in the Linux kernel 4.18 through 5.10.16, as used by Xen. The backend allocation (aka be-alloc) mode of the drm\_xen\_front drivers was not meant to be a supported configuration, but this wasn't stated accordingly in its support status entry.

More Info: <https://avd.aquasec.com/nvd/cve-2021-26934>

**[CVE-2021-3714] kernel: Remote Page Deduplication Attacks (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

A flaw was found in the Linux kernels memory deduplication mechanism. Previous work has shown that memory deduplication can be attacked via a local exploitation mechanism. The same technique can be used if an attacker can upload page sized files and detect the change in access time from a networked service to determine if the page has been merged.

More Info: <https://avd.aquasec.com/nvd/cve-2021-3714>

**[CVE-2022-0400] kernel: Out of bounds read in the smc protocol stack (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

An out-of-bounds read vulnerability was discovered in linux kernel in the smc protocol stack, causing remote dos.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0400>

**[CVE-2022-1247] kernel: A race condition bug in rose\_connect() (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %s(<nil>)

An issue found in linux-kernel that leads to a race condition in rose\_connect(). The rose driver uses rose\_neigh->use to represent how many objects are using the rose\_neigh. When a user wants to delete a rose\_route via rose\_ioctl(), the rose driver calls rose\_del\_node() and removes neighbours only if their 'count' and 'use' are zero.

More Info: <https://avd.aquasec.com/nvd/cve-2022-1247>

**[CVE-2022-25265] kernel: Executable Space Protection Bypass (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1



Fixed: %ls(<nil>)

In the Linux kernel through 5.16.10, certain binary files may have the exec-all attribute if they were built in approximately 2003 (e.g., with GCC 3.2.2 and Linux kernel 2.4.20). This can cause execution of bytes located in supposedly non-executable regions of a file.

More Info: <https://avd.aquasec.com/nvd/cve-2022-25265>

#### **[CVE-2022-2961] kernel: race condition in rose\_bind() (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A use-after-free flaw was found in the Linux kernel's PLP Rose functionality in the way a user triggers a race condition by calling bind while simultaneously triggering the rose\_bind() function. This flaw allows a local user to crash or potentially escalate their privileges on the system.

More Info: <https://avd.aquasec.com/nvd/cve-2022-2961>

#### **[CVE-2022-3238] kernel: ntfs3 local privilege escalation if NTFS character set and remount and umount called simultaneously (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A double-free flaw was found in the Linux kernel's NTFS3 subsystem in how a user triggers remount and umount simultaneously. This flaw allows a local user to crash or potentially escalate their privileges on the system.

More Info: <https://avd.aquasec.com/nvd/cve-2022-3238>

#### **[CVE-2022-41848] kernel: Race condition between mgslpc\_ioctl and mgslpc\_detach (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

drivers/char/pcmcia/synclink\_cs.c in the Linux kernel through 5.19.12 has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling ioctl, aka a race condition between mgslpc\_ioctl and mgslpc\_detach.

More Info: <https://avd.aquasec.com/nvd/cve-2022-41848>

#### **[CVE-2022-44032] Kernel: Race between cmm\_open() and cm4000\_detach() result in UAF (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/cm4000\_cs.c has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling open(), aka a race condition between cmm\_open() and cm4000\_detach().

More Info: <https://avd.aquasec.com/nvd/cve-2022-44032>

**[CVE-2022-44033] Kernel: A race condition between cm4040\_open() and reader\_detach() may result in UAF (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/cm4040\_cs.c has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling open(), aka a race condition between cm4040\_open() and reader\_detach().

More Info: <https://avd.aquasec.com/nvd/cve-2022-44033>

**[CVE-2022-44034] Kernel: A use-after-free due to race between scr24x\_open() and scr24x\_remove() (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/scr24x\_cs.c has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling open(), aka a race condition between scr24x\_open() and scr24x\_remove().

More Info: <https://avd.aquasec.com/nvd/cve-2022-44034>

**[CVE-2022-4543] kernel: KASLR Prefetch Bypass Breaks KPTI (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

A flaw named "EntryBleed" was found in the Linux Kernel Page Table Isolation (KPTI). This issue could allow a local attacker to leak KASLR base via prefetch side-channels based on TLB timing for Intel systems.

More Info: <https://avd.aquasec.com/nvd/cve-2022-4543>

**[CVE-2022-45884] kernel: use-after-free due to race condition occurring in dvb\_register\_device() (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.0.9. drivers/media/dvb-core/dvbdev.c has a use-after-free, related to dvb\_register\_device dynamically allocating fops.

More Info: <https://avd.aquasec.com/nvd/cve-2022-45884>

**[CVE-2022-45885] kernel: use-after-free due to race condition occurring in dvb\_frontend.c (Severity: LOW)**

Package: linux-libc-dev  
Installed: 6.1.129-1  
Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.0.9. drivers/media/dvb-core/dvb\_frontend.c has a race condition that can cause a use-after-free when a device is disconnected.

More Info: <https://avd.aquasec.com/nvd/cve-2022-45885>

**[CVE-2023-23039] kernel: tty: vcc: race condition leading to use-after-free in vcc\_open() (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in the Linux kernel through 6.2.0-rc2. drivers/tty/vcc.c has a race condition and resultant use-after-free if a physically proximate attacker removes a VCC device while calling open(), aka a race condition between vcc\_open() and vcc\_remove().

More Info: <https://avd.aquasec.com/nvd/cve-2023-23039>

**[CVE-2023-26242] afu\_mmio\_region\_get\_by\_offset in drivers/fpga/dfi-afu-region.c in the ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

afu\_mmio\_region\_get\_by\_offset in drivers/fpga/dfi-afu-region.c in the Linux kernel through 6.1.12 has an integer overflow.

More Info: <https://avd.aquasec.com/nvd/cve-2023-26242>

**[CVE-2023-31081] An issue was discovered in drivers/media/test-drivers/vidtv/vidtv\_brid ... (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in drivers/media/test-drivers/vidtv/vidtv\_bridge.c in the Linux kernel 6.2. There is a NULL pointer dereference in vidtv\_mux\_stop\_thread. In vidtv\_stop\_streaming, after dvb->mux=NULL occurs, it executes vidtv\_mux\_stop\_thread(dvb->mux).

More Info: <https://avd.aquasec.com/nvd/cve-2023-31081>

**[CVE-2023-31085] kernel: divide-by-zero error in ctrl\_cdev\_ioctl when do\_div happens and erasesize is 0 (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An issue was discovered in drivers/mtd/ubi/cdev.c in the Linux kernel 6.2. There is a divide-by-zero error in do\_div(sz,mtd->erasesize), used indirectly by ctrl\_cdev\_ioctl, when mtd->erasesize is 0.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31085>

**[CVE-2023-3640] Kernel: x86/mm: a per-cpu entry area leak was identified through the init\_cea\_offsets function when prefetchnta and prefetcht2 instructions being used for the per-cpu entry area mapping to the user space (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A possible unauthorized memory access flaw was found in the Linux kernel's cpu\_entry\_area mapping of X86 CPU data to memory, where a user may guess the location of exception stacks or other important data. Based on the previous CVE-2023-0597, the 'Randomize per-cpu entry area' feature was implemented in /arch/x86/mm/cpu\_entry\_area.c, which works through the init\_cea\_offsets() function when KASLR is enabled. However, despite this feature, there is still a risk of per-cpu entry area leaks. This issue could allow a local user to gain access to some important data with memory in an expected location and potentially escalate their privileges on the system.

More Info: <https://avd.aquasec.com/nvd/cve-2023-3640>

**[CVE-2023-39191] kernel: eBPF: insufficient stack type checks in dynptr (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

An improper input validation flaw was found in the eBPF subsystem in the Linux kernel. The issue occurs due to a lack of proper validation of dynamic pointers within user-supplied eBPF programs prior to executing them. This may allow an attacker with CAP\_BPF privileges to escalate privileges and execute arbitrary code in the context of the kernel.

More Info: <https://avd.aquasec.com/nvd/cve-2023-39191>

**[CVE-2023-4134] kernel: cyttsp4\_core: use-after-free in cyttsp4\_watchdog\_work() (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A use-after-free vulnerability was found in the cyttsp4\_core driver in the Linux kernel. This issue occurs in the device cleanup routine due to a possible rearming of the watchdog\_timer from the workqueue. This could allow a local user to crash the system, causing a denial of service.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4134>

**[CVE-2024-0564] kernel: max page sharing of Kernel Samepage Merging (KSM) may cause memory deduplication (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

A flaw was found in the Linux kernel's memory deduplication mechanism. The max page sharing of Kernel Samepage Merging (KSM), added in Linux kernel version 4.4.0-96.119, can create a side channel. When the attacker and the victim share the same host and the default setting of KSM is "max page sharing=256", it is possible for the attacker to time the unmap to merge with the victim's page. The unmapping time depends on whether it merges with the victim's page and additional physical pages are created beyond the KSM's "max page share". Through these operations, the attacker can leak the victim's page.

More Info: <https://avd.aquasec.com/nvd/cve-2024-0564>

## **[CVE-2024-40918] kernel: parisc: Try to fix random segmentation faults in package builds (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

parisc: Try to fix random segmentation faults in package builds

PA-RISC systems with PA8800 and PA8900 processors have had problems with random segmentation faults for many years. Systems with earlier processors are much more stable.

Systems with PA8800 and PA8900 processors have a large L2 cache which needs per page flushing for decent performance when a large range is flushed. The combined cache in these systems is also more sensitive to non-equivalent aliases than the caches in earlier systems.

The majority of random segmentation faults that I have looked at appear to be memory corruption in memory allocated using mmap and malloc.

My first attempt at fixing the random faults didn't work. On reviewing the cache code, I realized that there were two issues which the existing code didn't handle correctly. Both relate to cache move-in. Another issue is that the present bit in PTEs is racy.

1) PA-RISC caches have a mind of their own and they can speculatively load data and instructions for a page as long as there is a entry in the TLB for the page which allows move-in. TLBs are local to each CPU. Thus, the TLB entry for a page must be purged before flushing the page. This is particularly important on SMP systems.

In some of the flush routines, the flush routine would be called and then the TLB entry would be purged. This was because the flush routine needed the TLB entry to do the flush.

2) My initial approach to trying the fix the random faults was to try and use flush\_cache\_page\_if\_present for all flush operations. This actually made things worse and led to a couple of hardware lockups. It finally dawned on me that some lines weren't being flushed because the pte check code was racy. This resulted in random inequivalent mappings to physical pages.

The \_\_flush\_cache\_page tmpalias flush sets up its own TLB entry and it doesn't need the existing TLB entry. As long as we can find the pte pointer for the vm page, we can get the pfn and physical address of the page. We can also purge the TLB entry for the page

before doing the flush. Further, `__flush_cache_page` uses a special TLB entry that inhibits cache move-in.

When switching page mappings, we need to ensure that lines are removed from the cache. It is not sufficient to just flush the lines to memory as they may come back.

This made it clear that we needed to implement all the required flush operations using `tmpalias` routines. This includes flushes for user and kernel pages.

After modifying the code to use `tmpalias` flushes, it became clear that the random segmentation faults were not fully resolved. The frequency of faults was worse on systems with a 64 MB L2 (PA8900) and systems with more CPUs (rp4440).

The warning that I added to `flush_cache_page_if_present` to detect pages that couldn't be flushed triggered frequently on some systems.

Helge and I looked at the pages that couldn't be flushed and found that the PTE was either cleared or for a swap page. Ignoring pages that were swapped out seemed okay but pages with cleared PTEs seemed problematic.

I looked at routines related to `pte_clear` and noticed `ptep_clear_flush`. The default implementation just flushes the TLB entry. However, it was obvious that on `parisc` we need to flush the cache page as well. If we don't flush the cache page, stale lines will be left in the cache and cause random corruption. Once a PTE is cleared, there is no way to find the physical address associated with the PTE and flush the associated page at a later time.

I implemented an updated change with a `parisc` specific version of `ptep_clear_flush`. It fixed the random data corruption on Helge's `rp4440` and `rp3440`, as well as on my `c8000`.

At this point, I realized that I could restore the code where we only flush in `flush_cache_page_if_present` if the page has been accessed. However, for this, we also need to flush the cache when the accessed bit is cleared in  
---truncated---

More Info: <https://avd.aquasec.com/nvd/cve-2024-40918>

### **[CVE-2024-42155] kernel: s390/pkey: Wipe copies of protected- and secure-keys (Severity: LOW)**

Package: `linux-libc-dev`  
Installed: 6.1.129-1  
Fixed: `%ls(<nil>)`

In the Linux kernel, the following vulnerability has been resolved:

`s390/pkey: Wipe copies of protected- and secure-keys`

Although the clear-key of neither protected- nor secure-keys is accessible, this key material should only be visible to the calling process. So wipe all copies of protected- or secure-keys from stack, even in case of an error.

More Info: <https://avd.aquasec.com/nvd/cve-2024-42155>

**[CVE-2024-50057] kernel: usb: typec: tipd: Free IRQ only if it was requested before (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

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
```

More Info: <https://avd.aquasec.com/nvd/cve-2024-50057>

**[CVE-2024-50211] kernel: udf: refactor inode\_bmap() to handle error (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

udf: refactor inode\_bmap() to handle error

Refactor inode\_bmap() to handle error since udf\_next\_aext() can return error now. On situations like ftruncate, udf\_extend\_file() can now detect errors and bail out early without resorting to checking for particular offsets and assuming internal behavior of these functions.

More Info: <https://avd.aquasec.com/nvd/cve-2024-50211>

**[CVE-2024-56641] kernel: net/smc: initialize close\_work early to avoid warning (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %ls(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

net/smc: initialize close\_work early to avoid warning

We encountered a warning that close\_work was canceled before initialization.

```
WARNING: CPU: 7 PID: 111103 at kernel/workqueue.c:3047 __flush_work+0x19e/0x1b0
Workqueue: events smc_lgr_terminate_work [smc]
RIP: 0010: __flush_work+0x19e/0x1b0
Call Trace:
? __wake_up_common+0x7a/0x190
? work_busy+0x80/0x80
__cancel_work_timer+0xe3/0x160
smc_close_cancel_work+0x1a/0x70 [smc]
smc_close_active_abort+0x207/0x360 [smc]
__smc_lgr_terminate.part.38+0xc8/0x180 [smc]
process_one_work+0x19e/0x340
worker_thread+0x30/0x370
? process_one_work+0x340/0x340
kthread+0x117/0x130
? __kthread_cancel_work+0x50/0x50
ret_from_fork+0x22/0x30
```

This is because when smc\_close\_cancel\_work is triggered, e.g. the RDMA driver is rmmmod and the LGR is terminated, the conn->close\_work is flushed before initialization, resulting in WARN\_ON(!work->func).

```
__smc_lgr_terminate      | smc_connect_{rdma|ism}
-----
                        | smc_conn_create
                        | \- smc_lgr_register_conn
for conn in lgr->conns_all |
\-\ smc_conn_kill        |
  \-\ smc_close_active_abort |
    \-\ smc_close_cancel_work |
      \-\ cancel_work_sync |
        \-\ __flush_work |
          (close_work) |
            | smc_close_init
```



```
| \- INIT_WORK(&close_work)
```

So fix this by initializing close\_work before establishing the connection.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56641>

## **[CVE-2025-21825] kernel: bpf: Cancel the running bpf\_timer through kworker for PREEMPT\_RT (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

In the Linux kernel, the following vulnerability has been resolved:

bpf: Cancel the running bpf\_timer through kworker for PREEMPT\_RT

During the update procedure, when overwrite element in a pre-allocated htab, the freeing of old\_element is protected by the bucket lock. The reason why the bucket lock is necessary is that the old\_element has already been stashed in htab->extra\_elems after alloc\_htab\_elem() returns. If freeing the old\_element after the bucket lock is unlocked, the stashed element may be reused by concurrent update procedure and the freeing of old\_element will run concurrently with the reuse of the old\_element. However, the invocation of check\_and\_free\_fields() may acquire a spin-lock which violates the lockdep rule because its caller has already held a raw-spin-lock (bucket lock). The following warning will be reported when such race happens:

BUG: scheduling while atomic: test\_progs/676/0x00000003

3 locks held by test\_progs/676:

#0: ffffffff864b0240 (rcu\_read\_lock\_trace){...}-{0:0}, at: bpf\_prog\_test\_run\_syscall+0x2c0/0x830

#1: ffff88810e961188 (&htab->lockdep\_key){...}-{2:2}, at: htab\_map\_update\_elem+0x306/0x1500

#2: ffff8881f4eac1b8 (&base->softirq\_expiry\_lock){...}-{2:2}, at: hrtimer\_cancel\_wait\_running+0xe9/0x1b0

Modules linked in: bpf\_testmod(O)

Preemption disabled at:

[<fffffff817837a3>] htab\_map\_update\_elem+0x293/0x1500

CPU: 0 UID: 0 PID: 676 Comm: test\_progs Tainted: G ... 6.12.0+ #11

Tainted: [W]=WARN, [O]=OOT\_MODULE

Hardware name: QEMU Standard PC (i440FX + PIIX, 1996)...

Call Trace:

<TASK>

dump\_stack\_lvl+0x57/0x70

dump\_stack+0x10/0x20

\_\_schedule\_bug+0x120/0x170

\_\_schedule+0x300c/0x4800

schedule\_rtlock+0x37/0x60

rtlock\_slowlock\_locked+0x6d9/0x54c0

rt\_spin\_lock+0x168/0x230

hrtimer\_cancel\_wait\_running+0xe9/0x1b0

hrtimer\_cancel+0x24/0x30

bpf\_timer\_delete\_work+0x1d/0x40

```

bpf_timer_cancel_and_free+0x5e/0x80
bpf_obj_free_fields+0x262/0x4a0
check_and_free_fields+0x1d0/0x280
htab_map_update_elem+0x7fc/0x1500
bpf_prog_9f90bc20768e0cb9_overwrite_cb+0x3f/0x43
bpf_prog_ea601c4649694dbd_overwrite_timer+0x5d/0x7e
bpf_prog_test_run_syscall+0x322/0x830
__sys_bpf+0x135d/0x3ca0
__x64_sys_bpf+0x75/0xb0
x64_sys_call+0x1b5/0xa10
do_syscall_64+0x3b/0xc0
entry_SYSCALL_64_after_hwframe+0x4b/0x53
...
</TASK>

```

It seems feasible to break the reuse and refill of per-cpu extra\_elems into two independent parts: reuse the per-cpu extra\_elems with bucket lock being held and refill the old\_element as per-cpu extra\_elems after the bucket lock is unlocked. However, it will make the concurrent overwrite procedures on the same CPU return unexpected -E2BIG error when the map is full.

Therefore, the patch fixes the lock problem by breaking the cancelling of bpf\_timer into two steps for PREEMPT\_RT:

- 1) use hrtimer\_try\_to\_cancel() and check its return value
- 2) if the timer is running, use hrtimer\_cancel() through a kworker to cancel it again

Considering that the current implementation of hrtimer\_cancel() will try to acquire a being held softirq\_expiry\_lock when the current timer is running, these steps above are reasonable. However, it also has downside. When the timer is running, the cancelling of the timer is delayed when releasing the last map uref. The delay is also fixable (e.g., break the cancelling of bpf timer into two parts: one part in locked scope, another one in unlocked scope), it can be revised later if necessary.

It is a bit hard to decide the right fix tag. One reason is that the problem depends on PREEMPT\_RT which is enabled in v6.12. Considering the softirq\_expiry\_lock lock exists since v5.4 and bpf\_timer is introduced in v5.15, the bpf\_timer commit is used in the fixes tag and an extra depends-on tag is added to state the dependency on PREEMPT\_RT.

Depends-on: v6.12+ with PREEMPT\_RT enabled

More Info: <https://avd.aquasec.com/nvd/cve-2025-21825>

## **[TEMP-0000000-F7A20F] [Kernel: Unprivileged user can freeze journald] (Severity: LOW)**

Package: linux-libc-dev

Installed: 6.1.129-1

Fixed: %!s(<nil>)

%!s(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0000000-F7A20F>

**[CVE-2023-4641] shadow-utils: possible password leak during passwd(1) change (Severity: MEDIUM)**

Package: login

Installed: 1:4.13+dfsg1-1+b1

Fixed: %!s(<nil>)

A flaw was found in shadow-utils. When asking for a new password, shadow-utils asks the password twice. If the password fails on the second attempt, shadow-utils fails in cleaning the buffer used to store the first entry. This may allow an attacker with enough access to retrieve the password from the memory.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4641>

**[CVE-2007-5686] initscripts in rPath Linux 1 sets insecure permissions for the /var/lo ... (Severity: LOW)**

Package: login

Installed: 1:4.13+dfsg1-1+b1

Fixed: %!s(<nil>)

initscripts in rPath Linux 1 sets insecure permissions for the /var/log/btmp file, which allows local users to obtain sensitive information regarding authentication attempts. NOTE: because sshd detects the insecure permissions and does not log certain events, this also prevents sshd from logging failed authentication attempts by remote attackers.

More Info: <https://avd.aquasec.com/nvd/cve-2007-5686>

**[CVE-2023-29383] shadow: Improper input validation in shadow-utils package utility chfn (Severity: LOW)**

Package: login

Installed: 1:4.13+dfsg1-1+b1

Fixed: %!s(<nil>)

In Shadow 4.13, it is possible to inject control characters into fields provided to the SUID program chfn (change finger). Although it is not possible to exploit this directly (e.g., adding a new user fails because \n is in the block list), it is possible to misrepresent the /etc/passwd file when viewed. Use of \r manipulations and Unicode characters to work around blocking of the : character make it possible to give the impression that a new user has been added. In other words, an adversary may be able to convince a system administrator to take the system offline (an indirect, social-engineered denial of service) by demonstrating that "cat /etc/passwd" shows a rogue user account.

More Info: <https://avd.aquasec.com/nvd/cve-2023-29383>

**[CVE-2024-56433] shadow-utils: Default subordinate ID configuration in /etc/login.defs could lead to compromise (Severity: LOW)**

Package: login

Installed: 1:4.13+dfsg1-1+b1

Fixed: %!s(<nil>)

shadow-utils (aka shadow) 4.4 through 4.17.0 establishes a default /etc/subuid behavior (e.g., uid 100000 through 165535 for the first user account) that can realistically conflict with the uids of users defined on locally administered networks, potentially leading to account takeover, e.g., by leveraging newuidmap for access to an NFS home directory (or same-host resources in the case of remote logins by these local network users). NOTE: it may also be argued that system administrators should not have assigned uids, within local networks, that are within the range that can occur in

/etc/subuid.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56433>

**[TEMP-0628843-DBAD28] [more related to CVE-2005-4890] (Severity: LOW)**

Package: login

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0628843-DBAD28>

**[CVE-2008-1687] m4: unquoted output of maketemp and mkstemp (Severity: LOW)**

Package: m4

Installed: 1.4.19-3

Fixed: %ls(<nil>)

The (1) maketemp and (2) mkstemp builtin functions in GNU m4 before 1.4.11 do not quote their output when a file is created, which might allow context-dependent attackers to trigger a macro expansion, leading to unspecified use of an incorrect filename.

More Info: <https://avd.aquasec.com/nvd/cve-2008-1687>

**[CVE-2008-1688] m4: code execution via -F argument (Severity: LOW)**

Package: m4

Installed: 1.4.19-3

Fixed: %ls(<nil>)

Unspecified vulnerability in GNU m4 before 1.4.11 might allow context-dependent attackers to execute arbitrary code, related to improper handling of filenames specified with the -F option. NOTE: it is not clear when this issue crosses privilege boundaries.

More Info: <https://avd.aquasec.com/nvd/cve-2008-1688>

**[CVE-2023-52969] mariadb: MariaDB Server Crash Due to Empty Backtrace Log (Severity: MEDIUM)**

Package: mariadb-common

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, and 11.0 through 11.0.\* can sometimes crash with an empty backtrace log. This may be related to make\_aggr\_tables\_info and optimize\_stage2.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52969>

**[CVE-2023-52970] mariadb: MariaDB Server Crash via Item\_direct\_view\_ref (Severity: MEDIUM)**

Package: mariadb-common

Installed: 1:10.11.11-0+deb12u1

Fixed: %ls(<nil>)

MariaDB Server 10.4 through 10.5.\*, 10.6 through 10.6.\*, 10.7 through 10.11.\*, 11.0 through 11.0.\*, and 11.1 through 11.4.\* crashes in Item\_direct\_view\_ref::derived\_field\_transformer\_for\_where.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52970>

**[CVE-2023-52971] mariadb: MariaDB Server Crash (Severity: MEDIUM)**

Package: mariadb-common  
Installed: 1:10.11.11-0+deb12u1  
Fixed: %ls(<nil>)

MariaDB Server 10.10 through 10.11.\* and 11.0 through 11.4.\* crashes in JOIN::fix\_allSplittings\_in\_plan.

More Info: <https://avd.aquasec.com/nvd/cve-2023-52971>

**[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: mount  
Installed: 2.38.1-5+deb12u3  
Fixed: %ls(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: ncurses-base  
Installed: 6.4-4  
Fixed: %ls(<nil>)

NCurses v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2023-50495] ncurses: segmentation fault via \_nc\_wrap\_entry() (Severity: MEDIUM)**

Package: ncurses-bin  
Installed: 6.4-4  
Fixed: %ls(<nil>)

NCurses v6.4-20230418 was discovered to contain a segmentation fault via the component \_nc\_wrap\_entry().

More Info: <https://avd.aquasec.com/nvd/cve-2023-50495>

**[CVE-2025-32728] openssh: OpenSSH SSHD Agent Forwarding and X11 Forwarding (Severity: MEDIUM)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

In sshd in OpenSSH before 10.0, the DisableForwarding directive does not adhere to the documentation stating that it disables X11 and agent forwarding.

More Info: <https://avd.aquasec.com/nvd/cve-2025-32728>

**[CVE-2007-2243] OpenSSH 4.6 and earlier, when ChallengeResponseAuthentication is enabl ...**  
**(Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

OpenSSH 4.6 and earlier, when ChallengeResponseAuthentication is enabled, allows remote attackers to determine the existence of user accounts by attempting to authenticate via S/KEY, which displays a different response if the user account exists, a similar issue to CVE-2001-1483.

More Info: <https://avd.aquasec.com/nvd/cve-2007-2243>

**[CVE-2007-2768] OpenSSH, when using OPIE (One-Time Passwords in Everything) for PAM, a ...**  
**(Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

OpenSSH, when using OPIE (One-Time Passwords in Everything) for PAM, allows remote attackers to determine the existence of certain user accounts, which displays a different response if the user account exists and is configured to use one-time passwords (OTP), a similar issue to CVE-2007-2243.

More Info: <https://avd.aquasec.com/nvd/cve-2007-2768>

**[CVE-2008-3234] sshd in OpenSSH 4 on Debian GNU/Linux, and the 20070303 OpenSSH snapsh ...**  
**(Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

sshd in OpenSSH 4 on Debian GNU/Linux, and the 20070303 OpenSSH snapshot, allows remote authenticated users to obtain access to arbitrary SELinux roles by appending a :/ (colon slash) sequence, followed by the role name, to the username.

More Info: <https://avd.aquasec.com/nvd/cve-2008-3234>

**[CVE-2016-20012] openssh: Public key information leak (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and public key is known to an SSH server, to test whether this suspicion is correct. This occurs because a challenge is sent only when that combination could be valid for a login session. NOTE: the vendor does not recognize user enumeration as a vulnerability for this product

More Info: <https://avd.aquasec.com/nvd/cve-2016-20012>

**[CVE-2018-15919] openssh: User enumeration via malformed packets in authentication requests (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect existence of users on a target system when GSS2 is in use. NOTE: the discoverer states 'We understand that the OpenSSH developers do not want to treat such a username enumeration (or "oracle") as a vulnerability.'

More Info: <https://avd.aquasec.com/nvd/cve-2018-15919>

**[CVE-2019-6110] openssh: Acceptance and display of arbitrary stderr allows for spoofing of scp client output (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

In OpenSSH 7.9, due to accepting and displaying arbitrary stderr output from the server, a malicious server (or Man-in-The-Middle attacker) can manipulate the client output, for example to use ANSI control codes to hide additional files being transferred.

More Info: <https://avd.aquasec.com/nvd/cve-2019-6110>

**[CVE-2020-14145] openssh: Observable discrepancy leading to an information leak in the algorithm negotiation (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client). NOTE: some reports state that 8.5 and 8.6 are also affected.

More Info: <https://avd.aquasec.com/nvd/cve-2020-14145>

**[CVE-2020-15778] openssh: scp allows command injection when using backtick characters in the destination argument (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5  
Fixed: %ls(<nil>)

scp in OpenSSH through 8.3p1 allows command injection in the scp.c toremote function, as demonstrated by backtick characters in the destination argument. NOTE: the vendor reportedly has stated that they intentionally omit validation of "anomalous argument transfers" because that could "stand a great chance of breaking existing workflows."

More Info: <https://avd.aquasec.com/nvd/cve-2020-15778>

**[CVE-2023-51767] openssh: authentication bypass via row hammer attack (Severity: LOW)**

Package: openssh-client  
Installed: 1:9.2p1-2+deb12u5

Fixed: %ls(<nil>)

OpenSSH through 9.6, when common types of DRAM are used, might allow row hammer attacks (for authentication bypass) because the integer value of authenticated in mm\_answer\_authpassword does not resist flips of a single bit. NOTE: this is applicable to a certain threat model of attacker-victim co-location in which the attacker has user privileges.

More Info: <https://avd.aquasec.com/nvd/cve-2023-51767>

### **[CVE-2024-13176] openssl: Timing side-channel in ECDSA signature computation (Severity: MEDIUM)**

Package: openssl

Installed: 3.0.15-1~deb12u1

Fixed: %ls(<nil>)

Issue summary: A timing side-channel which could potentially allow recovering the private key exists in the ECDSA signature computation.

Impact summary: A timing side-channel in ECDSA signature computations could allow recovering the private key by an attacker. However, measuring the timing would require either local access to the signing application or a very fast network connection with low latency.

There is a timing signal of around 300 nanoseconds when the top word of the inverted ECDSA nonce value is zero. This can happen with significant probability only for some of the supported elliptic curves. In particular the NIST P-521 curve is affected. To be able to measure this leak, the attacker process must either be located in the same physical computer or must have a very fast network connection with low latency. For that reason the severity of this vulnerability is Low.

The FIPS modules in 3.4, 3.3, 3.2, 3.1 and 3.0 are affected by this issue.

More Info: <https://avd.aquasec.com/nvd/cve-2024-13176>

### **[CVE-2023-4641] shadow-utils: possible password leak during passwd(1) change (Severity: MEDIUM)**

Package: passwd

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

A flaw was found in shadow-utils. When asking for a new password, shadow-utils asks the password twice. If the password fails on the second attempt, shadow-utils fails in cleaning the buffer used to store the first entry. This may allow an attacker with enough access to retrieve the password from the memory.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4641>

### **[CVE-2007-5686] initscripts in rPath Linux 1 sets insecure permissions for the /var/lo ... (Severity: LOW)**

Package: passwd

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

initscripts in rPath Linux 1 sets insecure permissions for the /var/log/btmp file, which allows local users to obtain sensitive information regarding authentication attempts. NOTE: because sshd detects the insecure permissions and



does not log certain events, this also prevents sshd from logging failed authentication attempts by remote attackers.

More Info: <https://avd.aquasec.com/nvd/cve-2007-5686>

**[CVE-2023-29383] shadow: Improper input validation in shadow-utils package utility chfn (Severity: LOW)**

Package: passwd

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

In Shadow 4.13, it is possible to inject control characters into fields provided to the SUID program chfn (change finger). Although it is not possible to exploit this directly (e.g., adding a new user fails because \n is in the block list), it is possible to misrepresent the /etc/passwd file when viewed. Use of \r manipulations and Unicode characters to work around blocking of the : character make it possible to give the impression that a new user has been added. In other words, an adversary may be able to convince a system administrator to take the system offline (an indirect, social-engineered denial of service) by demonstrating that "cat /etc/passwd" shows a rogue user account.

More Info: <https://avd.aquasec.com/nvd/cve-2023-29383>

**[CVE-2024-56433] shadow-utils: Default subordinate ID configuration in /etc/login.defs could lead to compromise (Severity: LOW)**

Package: passwd

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

shadow-utils (aka shadow) 4.4 through 4.17.0 establishes a default /etc/subuid behavior (e.g., uid 100000 through 165535 for the first user account) that can realistically conflict with the uids of users defined on locally administered networks, potentially leading to account takeover, e.g., by leveraging newuidmap for access to an NFS home directory (or same-host resources in the case of remote logins by these local network users). NOTE: it may also be argued that system administrators should not have assigned uids, within local networks, that are within the range that can occur in /etc/subuid.

More Info: <https://avd.aquasec.com/nvd/cve-2024-56433>

**[TEMP-0628843-DBAD28] [more related to CVE-2005-4890] (Severity: LOW)**

Package: passwd

Installed: 1:4.13+dfsg1-1+b1

Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0628843-DBAD28>

**[CVE-2010-4651] patch: directory traversal flaw allows for arbitrary file creation (Severity: LOW)**

Package: patch

Installed: 2.7.6-7

Fixed: %ls(<nil>)

Directory traversal vulnerability in util.c in GNU patch 2.6.1 and earlier allows user-assisted remote attackers to create or overwrite arbitrary files via a filename that is specified with a .. (dot dot) or full pathname, a related issue to CVE-2010-1679.

More Info: <https://avd.aquasec.com/nvd/cve-2010-4651>

**[CVE-2018-6951] patch: NULL pointer dereference in pch.c:intuit\_diff\_type() causes a crash (Severity: LOW)**

Package: patch  
Installed: 2.7.6-7  
Fixed: %!s(<nil>)

An issue was discovered in GNU patch through 2.7.6. There is a segmentation fault, associated with a NULL pointer dereference, leading to a denial of service in the intuit\_diff\_type function in pch.c, aka a "mangled rename" issue.

More Info: <https://avd.aquasec.com/nvd/cve-2018-6951>

**[CVE-2018-6952] patch: Double free of memory in pch.c:another\_hunk() causes a crash (Severity: LOW)**

Package: patch  
Installed: 2.7.6-7  
Fixed: %!s(<nil>)

A double free exists in the another\_hunk function in pch.c in GNU patch through 2.7.6.

More Info: <https://avd.aquasec.com/nvd/cve-2018-6952>

**[CVE-2021-45261] patch: Invalid Pointer via another\_hunk function (Severity: LOW)**

Package: patch  
Installed: 2.7.6-7  
Fixed: %!s(<nil>)

An Invalid Pointer vulnerability exists in GNU patch 2.7 via the another\_hunk function, which causes a Denial of Service.

More Info: <https://avd.aquasec.com/nvd/cve-2021-45261>

**[CVE-2023-31484] perl: CPAN.pm does not verify TLS certificates when downloading distributions over HTTPS (Severity: HIGH)**

Package: perl  
Installed: 5.36.0-7+deb12u1  
Fixed: %!s(<nil>)

CPAN.pm before 2.35 does not verify TLS certificates when downloading distributions over HTTPS.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31484>

**[CVE-2011-4116] perl: File::Temp insecure temporary file handling (Severity: LOW)**

Package: perl  
Installed: 5.36.0-7+deb12u1  
Fixed: %!s(<nil>)

\_is\_safe in the File::Temp module for Perl does not properly handle symlinks.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4116>

**[CVE-2023-31486] http-tiny: insecure TLS cert default (Severity: LOW)**

Package: perl

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default TLS configuration where users must opt in to verify certificates.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31486>

**[CVE-2023-31484] perl: CPAN.pm does not verify TLS certificates when downloading distributions over HTTPS (Severity: HIGH)**

Package: perl-base

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

CPAN.pm before 2.35 does not verify TLS certificates when downloading distributions over HTTPS.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31484>

**[CVE-2011-4116] perl: File::Temp insecure temporary file handling (Severity: LOW)**

Package: perl-base

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

\_is\_safe in the File::Temp module for Perl does not properly handle symlinks.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4116>

**[CVE-2023-31486] http-tiny: insecure TLS cert default (Severity: LOW)**

Package: perl-base

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default TLS configuration where users must opt in to verify certificates.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31486>

**[CVE-2023-31484] perl: CPAN.pm does not verify TLS certificates when downloading distributions over HTTPS (Severity: HIGH)**

Package: perl-modules-5.36

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

CPAN.pm before 2.35 does not verify TLS certificates when downloading distributions over HTTPS.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31484>

**[CVE-2011-4116] perl: File::Temp insecure temporary file handling (Severity: LOW)**

Package: perl-modules-5.36

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

\_is\_safe in the File::Temp module for Perl does not properly handle symlinks.

More Info: <https://avd.aquasec.com/nvd/cve-2011-4116>

### **[CVE-2023-31486] http-tiny: insecure TLS cert default (Severity: LOW)**

Package: perl-modules-5.36

Installed: 5.36.0-7+deb12u1

Fixed: %!s(<nil>)

HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default TLS configuration where users must opt in to verify certificates.

More Info: <https://avd.aquasec.com/nvd/cve-2023-31486>

### **[CVE-2023-4016] procps: ps buffer overflow (Severity: LOW)**

Package: procps

Installed: 2:4.0.2-3

Fixed: %!s(<nil>)

Under some circumstances, this weakness allows a user who has access to run the `ps` utility on a machine, the ability to write almost unlimited amounts of unfiltered data into the process heap.

More Info: <https://avd.aquasec.com/nvd/cve-2023-4016>

### **[CVE-2025-0938] python: cpython: URL parser allowed square brackets in domain names (Severity: MEDIUM)**

Package: python3.11

Installed: 3.11.2-6+deb12u5

Fixed: %!s(<nil>)

The Python standard library functions `urllib.parse.urlsplit` and `urlparse` accepted domain names that included square brackets which isn't valid according to RFC 3986. Square brackets are only meant to be used as delimiters for specifying IPv6 and IPvFuture hosts in URLs. This could result in differential parsing across the Python URL parser and other specification-compliant URL parsers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0938>

### **[CVE-2025-1795] python: Mishandling of comma during folding and unicode-encoding of email headers (Severity: LOW)**

Package: python3.11

Installed: 3.11.2-6+deb12u5

Fixed: %!s(<nil>)

During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-encoded then the separator itself is also unicode-encoded. Expected behavior is that the separating comma remains a plain comma. This can result in the address header being misinterpreted by some mail servers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1795>

**[CVE-2025-0938] python: cpython: URL parser allowed square brackets in domain names (Severity: MEDIUM)**

Package: python3.11-minimal  
Installed: 3.11.2-6+deb12u5  
Fixed: %ls(<nil>)

The Python standard library functions ``urllib.parse.urlsplit`` and ``urlparse`` accepted domain names that included square brackets which isn't valid according to RFC 3986. Square brackets are only meant to be used as delimiters for specifying IPv6 and IPvFuture hosts in URLs. This could result in differential parsing across the Python URL parser and other specification-compliant URL parsers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-0938>

**[CVE-2025-1795] python: Mishandling of comma during folding and unicode-encoding of email headers (Severity: LOW)**

Package: python3.11-minimal  
Installed: 3.11.2-6+deb12u5  
Fixed: %ls(<nil>)

During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-encoded then the separator itself is also unicode-encoded. Expected behavior is that the separating comma remains a plain comma. This can result in the address header being misinterpreted by some mail servers.

More Info: <https://avd.aquasec.com/nvd/cve-2025-1795>

**[TEMP-0517018-A83CE6] [sysvinit: no-root option in expert installer exposes locally exploitable security flaw] (Severity: LOW)**

Package: sysvinit-utils  
Installed: 3.06-4  
Fixed: %ls(<nil>)

%ls(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0517018-A83CE6>

**[CVE-2005-2541] tar: does not properly warn the user when extracting setuid or setgid files (Severity: LOW)**

Package: tar  
Installed: 1.34+dfsg-1.2+deb12u1  
Fixed: %ls(<nil>)

Tar 1.15.1 does not properly warn the user when extracting setuid or setgid files, which may allow local users or remote attackers to gain privileges.

More Info: <https://avd.aquasec.com/nvd/cve-2005-2541>

**[TEMP-0290435-0B57B5] [tar's rmt command may have undesired side effects] (Severity: LOW)**

Package: tar  
Installed: 1.34+dfsg-1.2+deb12u1  
Fixed: %ls(<nil>)

%!s(<nil>)

More Info: <https://security-tracker.debian.org/tracker/TEMP-0290435-0B57B5>

### **[CVE-2021-4217] unzip: Null pointer dereference in Unicode strings code (Severity: LOW)**

Package: unzip  
Installed: 6.0-28  
Fixed: %!s(<nil>)

A flaw was found in unzip. The vulnerability occurs due to improper handling of Unicode strings, which can lead to a null pointer dereference. This flaw allows an attacker to input a specially crafted zip file, leading to a crash or code execution.

More Info: <https://avd.aquasec.com/nvd/cve-2021-4217>

### **[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: util-linux  
Installed: 2.38.1-5+deb12u3  
Fixed: %!s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

### **[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: util-linux-extra  
Installed: 2.38.1-5+deb12u3  
Fixed: %!s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

### **[CVE-2022-0563] util-linux: partial disclosure of arbitrary files in chfn and chsh when compiled with libreadline (Severity: LOW)**

Package: uuid-dev  
Installed: 2.38.1-5+deb12u3  
Fixed: %!s(<nil>)

A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.

More Info: <https://avd.aquasec.com/nvd/cve-2022-0563>

**[CVE-2021-31879] wget: authorization header disclosure on redirect (Severity: MEDIUM)**

Package: wget

Installed: 1.21.3-1+deb12u1

Fixed: %ls(<nil>)

GNU Wget through 1.21.1 does not omit the Authorization header upon a redirect to a different origin, a related issue to CVE-2018-1000007.

More Info: <https://avd.aquasec.com/nvd/cve-2021-31879>

**[CVE-2024-10524] wget: GNU Wget is vulnerable to an SSRF attack when accessing partially-user-controlled shorthand URLs (Severity: MEDIUM)**

Package: wget

Installed: 1.21.3-1+deb12u1

Fixed: %ls(<nil>)

Applications that use Wget to access a remote resource using shorthand URLs and pass arbitrary user credentials in the URL are vulnerable. In these cases attackers can enter crafted credentials which will cause Wget to access an arbitrary host.

More Info: <https://avd.aquasec.com/nvd/cve-2024-10524>

**[CVE-2023-45853] zlib: integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZip4\_6 (Severity: CRITICAL)**

Package: zlib1g

Installed: 1:1.2.13.dfsg-1

Fixed: %ls(<nil>)

MiniZip in zlib through 1.3 has an integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZip4\_64 via a long filename, comment, or extra field. NOTE: MiniZip is not a supported part of the zlib product. NOTE: pyminizip through 0.2.6 is also vulnerable because it bundles an affected zlib version, and exposes the applicable MiniZip code through its compress API.

More Info: <https://avd.aquasec.com/nvd/cve-2023-45853>

**[CVE-2023-45853] zlib: integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZip4\_6 (Severity: CRITICAL)**

Package: zlib1g-dev

Installed: 1:1.2.13.dfsg-1

Fixed: %ls(<nil>)

MiniZip in zlib through 1.3 has an integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZip4\_64 via a long filename, comment, or extra field. NOTE: MiniZip is not a supported part of the zlib product. NOTE: pyminizip through 0.2.6 is also vulnerable because it bundles an affected zlib version, and exposes the applicable MiniZip code through its compress API.

More Info: <https://avd.aquasec.com/nvd/cve-2023-45853>

Target: Node.js