```
commit fc8d83d62cbe3850bc0ae794111cf9bf17c3798a
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Apr 26 14:30:09 2023 +0200

    Linux 6.2.13

    Link: https://lore.kernel.org/r/20230424131136.142490414@linuxfoundation.org
    Tested-by: Markus Reichelt <lkt+2023@mareichelt.com>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Conor Dooley <conor.dooley@microchip.com>
    Tested-by: Chris Paterson (CIP) <chris.paterson2@renesas.com>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0a75851daf4115df3532499a2f496dc09191c459
Author: Ekaterina Orlova <vorobushek.ok@gmail.com>
Date:   Fri Apr 21 15:35:39 2023 +0100

    ASN.1: Fix check for strdup() success

    commit 5a43001c01691dcbd396541e6faa2c0077378f48 upstream.

    It seems there is a misprint in the check of strdup() return code that
    can lead to NULL pointer dereference.

    Found by Linux Verification Center (linuxtesting.org) with SVACE.

    Fixes: 4520c6a49af8 ("X.509: Add simple ASN.1 grammar compiler")
    Signed-off-by: Ekaterina Orlova <vorobushek.ok@gmail.com>
    Cc: David Woodhouse <dwmw2@infradead.org>
    Cc: James Bottomley <jejb@linux.ibm.com>
    Cc: Jarkko Sakkinen <jarkko@kernel.org>
    Cc: keyrings@vger.kernel.org
    Cc: linux-kbuild@vger.kernel.org
    Link: https://lore.kernel.org/r/20230315172130.140-1-vorobushek.ok@gmail.com/
    Signed-off-by: David Howells <dhowells@redhat.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3ec6943ea567cbbe491d5fb3de8495b03897e80e
Author: Chancel Liu <chancel.liu@nxp.com>
Date:   Tue Apr 18 17:42:59 2023 +0800

    ASoC: fsl_sai: Fix pins setting for i.MX8QM platform

    commit 238787157d83969e5149c8e99787d5d90e85fbe5 upstream.

    SAI on i.MX8QM platform supports the data lines up to 4. So the pins
    setting should be corrected to 4.

    Fixes: eba0f0077519 ("ASoC: fsl_sai: Enable combine mode soft")
    Signed-off-by: Chancel Liu <chancel.liu@nxp.com>
    Acked-by: Shengjiu Wang <shengjiu.wang@gmail.com>
    Reviewed-by: Iuliana Prodan <iuliana.prodan@nxp.com>
    Link: https://lore.kernel.org/r/20230418094259.4150771-1-chancel.liu@nxp.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit d866221cbb45c1f53cad31f093ee1e8d0256127b
Author: Nikita Zhandarovich <n.zhandarovich@fintech.ru>
Date:   Mon Apr 17 06:32:42 2023 -0700

    ASoC: fsl_asrc_dma: fix potential null-ptr-deref

    commit 86a24e99c97234f87d9f70b528a691150e145197 upstream.

    dma_request_slave_channel() may return NULL which will lead to
    NULL pointer dereference error in 'tmp_chan->private'.

    Correct this behaviour by, first, switching from deprecated function
    dma_request_slave_channel() to dma_request_chan(). Secondly, enable
    sanity check for the resuling value of dma_request_chan().
    Also, fix description that follows the enacted changes and that
    concerns the use of dma_request_slave_channel().

    Fixes: 706e2c881158 ("ASoC: fsl_asrc_dma: Reuse the dma channel if available in Back-
End")
    Co-developed-by: Natalia Petrova <n.petrova@fintech.ru>
    Signed-off-by: Nikita Zhandarovich <n.zhandarovich@fintech.ru>
    Acked-by: Shengjiu Wang <shengjiu.wang@gmail.com>
    Link: https://lore.kernel.org/r/20230417133242.53339-1-n.zhandarovich@fintech.ru
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fedd67003d6fd0efc43e085d5e0e8b4552272502
Author: Daniel Baluta <daniel.baluta@nxp.com>
Date:   Wed Apr 5 12:26:55 2023 +0300

    ASoC: SOF: pm: Tear down pipelines only if DSP was active

    commit 0b186bb06198653d74a141902a7739e0bde20cf4 upstream.

    With PCI if the device was suspended it is brought back to full
    power and then suspended again.

    This doesn't happen when device is described via DT.

    We need to make sure that we tear down pipelines only if the device
    was previously active (thus the pipelines were setup).

    Otherwise, we can break the use_count:

    [  219.009743] sof-audio-of-imx8m 3b6e8000.dsp:
    sof_ipc3_tear_down_all_pipelines: widget PIPELINE.2.SAI3.IN is still in use: count -1

    and after this everything stops working.

    Fixes: d185e0689abc ("ASoC: SOF: pm: Always tear down pipelines before DSP suspend")
    Reviewed-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Reviewed-by: Ranjani Sridharan <ranjani.sridharan@linux.intel.com>
    Signed-off-by: Daniel Baluta <daniel.baluta@nxp.com>
    Link: https://lore.kernel.org/r/20230405092655.19587-1-daniel.baluta@oss.nxp.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 64634bac131dc4a0b1636938aceb883845f24472
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Tue Apr 4 23:31:58 2023 +0900

    mm/page_alloc: fix potential deadlock on zonelist_update_seq seqlock

    commit 1007843a91909a4995ee78a538f62d8665705b66 upstream.
```

```
    syzbot is reporting circular locking dependency which involves
    zonelist_update_seq seqlock [1], for this lock is checked by memory
    allocation requests which do not need to be retried.

    One deadlock scenario is kmalloc(GFP_ATOMIC) from an interrupt handler.

      CPU0
      ----
      __build_all_zonelists() {
        write_seqlock(&zonelist_update_seq); // makes zonelist_update_seq.seqcount odd
        // e.g. timer interrupt handler runs at this moment
          some_timer_func() {
            kmalloc(GFP_ATOMIC) {
              __alloc_pages_slowpath() {
                read_seqbegin(&zonelist_update_seq) {
                  // spins forever because zonelist_update_seq.seqcount is odd
                }
              }
            }
          }
        // e.g. timer interrupt handler finishes
        write_sequnlock(&zonelist_update_seq); // makes zonelist_update_seq.seqcount even
      }

    This deadlock scenario can be easily eliminated by not calling
    read_seqbegin(&zonelist_update_seq) from !__GFP_DIRECT_RECLAIM allocation
    requests, for retry is applicable to only __GFP_DIRECT_RECLAIM allocation
    requests.  But Michal Hocko does not know whether we should go with this
    approach.

    Another deadlock scenario which syzbot is reporting is a race between
    kmalloc(GFP_ATOMIC) from tty_insert_flip_string_and_push_buffer() with
    port->lock held and printk() from __build_all_zonelists() with
    zonelist_update_seq held.

      CPU0                                    CPU1
      ----                                    ----
      pty_write() {
        tty_insert_flip_string_and_push_buffer() {
                                              __build_all_zonelists() {
                                                write_seqlock(&zonelist_update_seq);
                                                build_zonelists() {
                                                  printk() {
                                                    vprintk() {
                                                      vprintk_default() {
                                                        vprintk_emit() {
                                                          console_unlock() {
                                                            console_flush_all() {
                                                              console_emit_next_record() {
                                                                con->write() =
  serial8250_console_write() {
            spin_lock_irqsave(&port->lock, flags);
            tty_insert_flip_string() {
              tty_insert_flip_string_fixed_flag() {
                __tty_buffer_request_room() {
                  tty_buffer_alloc() {
                    kmalloc(GFP_ATOMIC | __GFP_NOWARN) {
                      __alloc_pages_slowpath() {
                        zonelist_iter_begin() {
                          read_seqbegin(&zonelist_update_seq); // spins forever because
  zonelist_update_seq.seqcount is odd
                                                                  spin_lock_irqsave(&port-
  >lock, flags); // spins forever because port->lock is held
                          }
                        }
```

```
                    }
                  }
                }
              }
            }
          }
          spin_unlock_irqrestore(&port->lock, flags);
                                                              // message is printed to
console

spin_unlock_irqrestore(&port->lock, flags);
                                                                  }
                                                                }
                                                              }
                                                            }
                                                          }
                                                        }
                                                      }
                                                    }
                                                  }
                                                  write_sequnlock(&zonelist_update_seq);
                                                }
      }
    }
```

This deadlock scenario can be eliminated by

  preventing interrupt context from calling kmalloc(GFP_ATOMIC)

and

  preventing printk() from calling console_flush_all()

while zonelist_update_seq.seqcount is odd.

Since Petr Mladek thinks that __build_all_zonelists() can become a
candidate for deferring printk() [2], let's address this problem by

  disabling local interrupts in order to avoid kmalloc(GFP_ATOMIC)

and

  disabling synchronous printk() in order to avoid console_flush_all()

.

As a side effect of minimizing duration of zonelist_update_seq.seqcount
being odd by disabling synchronous printk(), latency at
read_seqbegin(&zonelist_update_seq) for both !__GFP_DIRECT_RECLAIM and
__GFP_DIRECT_RECLAIM allocation requests will be reduced. Although, from
lockdep perspective, not calling read_seqbegin(&zonelist_update_seq) (i.e.
do not record unnecessary locking dependency) from interrupt context is
still preferable, even if we don't allow calling kmalloc(GFP_ATOMIC)
inside
write_seqlock(&zonelist_update_seq)/write_sequnlock(&zonelist_update_seq)
section...

Link: https://lkml.kernel.org/r/8796b95c-3da3-5885-fddd-6ef55f30e4d3@I-love.SAKURA.ne.jp
Fixes: 3d36424b3b58 ("mm/page_alloc: fix race condition between build_all_zonelists and
page allocation")
Link: https://lkml.kernel.org/r/ZCrs+1cDqPWTDFNM@alley [2]
Reported-by: syzbot <syzbot+223c7461c58c58a4cb10@syzkaller.appspotmail.com>
  Link: https://syzkaller.appspot.com/bug?extid=223c7461c58c58a4cb10 [1]
Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Acked-by: Michal Hocko <mhocko@suse.com>

```
    Acked-by: Mel Gorman <mgorman@techsingularity.net>
    Cc: Petr Mladek <pmladek@suse.com>
    Cc: David Hildenbrand <david@redhat.com>
    Cc: Ilpo Järvinen <ilpo.jarvinen@linux.intel.com>
    Cc: John Ogness <john.ogness@linutronix.de>
    Cc: Patrick Daly <quic_pdaly@quicinc.com>
    Cc: Sergey Senozhatsky <senozhatsky@chromium.org>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit df23ce01d937af7c9d71618b5bc0e4514dafa150
Author: Alexis Lothoré <alexis.lothore@bootlin.com>
Date:   Tue Apr 4 15:31:02 2023 +0200

    fpga: bridge: properly initialize bridge device before populating children

    commit dc70eb868b9cd2ca01313e5a394e6ea001d513e9 upstream.

    The current code path can lead to warnings because of uninitialized device,
    which contains, as a consequence, uninitialized kobject. The uninitialized
    device is passed to of_platform_populate, which will at some point, while
    creating child device, try to get a reference on uninitialized parent,
    resulting in the following warning:

    kobject: '(null)' ((ptrval)): is not initialized, yet kobject_get() is
    being called.

    The warning is observed after migrating a kernel 5.10.x to 6.1.x.
    Reverting commit 0d70af3c2530 ("fpga: bridge: Use standard dev_release for
    class driver") seems to remove the warning.
    This commit aggregates device_initialize() and device_add() into
    device_register() but this new call is done AFTER of_platform_populate

    Fixes: 0d70af3c2530 ("fpga: bridge: Use standard dev_release for class driver")
    Signed-off-by: Alexis Lothoré <alexis.lothore@bootlin.com>
    Acked-by: Xu Yilun <yilun.xu@intel.com>
    Link: https://lore.kernel.org/r/20230404133102.2837535-2-alexis.lothore@bootlin.com
    Signed-off-by: Xu Yilun <yilun.xu@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a88ba08648f33dd420f06fc1a17e65f31eabad8
Author: Dan Carpenter <error27@gmail.com>
Date:   Wed Mar 29 07:35:32 2023 +0300

    iio: adc: at91-sama5d2_adc: fix an error code in at91_adc_allocate_trigger()

    commit 73a428b37b9b538f8f8fe61caa45e7f243bab87c upstream.

    The at91_adc_allocate_trigger() function is supposed to return error
    pointers.  Returning a NULL will cause an Oops.

    Fixes: 5e1a1da0f8c9 ("iio: adc: at91-sama5d2_adc: add hw trigger and buffer support")
    Signed-off-by: Dan Carpenter <error27@gmail.com>
    Link: https://lore.kernel.org/r/5d728f9d-31d1-410d-a0b3-df6a63a2c8ba@kili.mountain
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5d7ef03e05b8c42a81c344896aba7ede4c26056f
Author: Soumya Negi <soumya.negi97@gmail.com>
Date:   Sun Apr 9 19:12:04 2023 -0700

    Input: pegasus-notetaker - check pipe type when probing
```

```
    commit b3d80fd27a3c2d8715a40cbf876139b56195f162 upstream.

    Fix WARNING in pegasus_open/usb_submit_urb
    Syzbot bug: https://syzkaller.appspot.com/bug?
id=bbc107584dcf3262253ce93183e51f3612aaeb13

    Warning raised because pegasus_driver submits transfer request for
    bogus URB (pipe type does not match endpoint type). Add sanity check at
    probe time for pipe value extracted from endpoint descriptor. Probe
    will fail if sanity check fails.

    Reported-and-tested-by: syzbot+04ee0cb4caccaed12d78@syzkaller.appspotmail.com
    Signed-off-by: Soumya Negi <soumya.negi97@gmail.com>
    Link: https://lore.kernel.org/r/20230404074145.11523-1-soumya.negi97@gmail.com
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7ae0ea9aba84e1753fb37e5b7044daf16c1db6d5
Author: hrdl <git@hrdl.eu>
Date:   Thu Apr 13 23:41:13 2023 -0700

    Input: cyttsp5 - fix sensing configuration data structure

    commit 5dc63e56a9cf8df0b59c234a505a1653f1bdf885 upstream.

    Prior to this patch, the sensing configuration data was not parsed
    correctly, breaking detection of max_tch. The vendor driver includes
    this field. This change informs the driver about the correct maximum
    number of simultaneous touch inputs.

    Tested on a Pine64 PineNote with a modified touch screen controller
    firmware.

    Signed-off-by: hrdl <git@hrdl.eu>
    Reviewed-by: Alistair Francis <alistair@alistair23.me>
    Link: https://lore.kernel.org/r/20230411211651.3791304-1-git@hrdl.eu
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a857ff05a9324fbf42bdb8df1493606cb7cb9717
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Sun Apr 23 09:56:20 2023 -0700

    gcc: disable '-Warray-bounds' for gcc-13 too

    commit 0da6e5fd6c3726723e275603426e09178940dace upstream.

    We started disabling '-Warray-bounds' for gcc-12 originally on s390,
    because it resulted in some warnings that weren't realistically fixable
    (commit 8b202ee21839: "s390: disable -Warray-bounds").

    That s390-specific issue was then found to be less common elsewhere, but
    generic (see f0be87c42cbd: "gcc-12: disable '-Warray-bounds' universally
    for now"), and then later expanded the version check was expanded to
    gcc-11 (5a41237ad1d4: "gcc: disable -Warray-bounds for gcc-11 too").

    And it turns out that I was much too optimistic in thinking that it's
    all going to go away, and here we are with gcc-13 showing all the same
    issues.  So instead of expanding this one version at a time, let's just
    disable it for gcc-11+, and put an end limit to it only when we actually
    find a solution.

    Yes, I'm sure some of this is because the kernel just does odd things
    (like our "container_of()" use, but also knowingly playing games with
    things like linker tables and array layouts).
```

```
    And yes, some of the warnings are likely signs of real bugs, but when
    there are hundreds of false positives, that doesn't really help.

    Oh well.

    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 89da310571c0410794ab4c0c25fe9a31ebec57d3
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Mon Apr 10 21:14:45 2023 +0200

    PCI/MSI: Remove over-zealous hardware size check in pci_msix_validate_entries()

    commit e3c026be4d3ca046799fde55ccbae9d0f059fb93 upstream.

    pci_msix_validate_entries() validates the entries array which is handed in
    by the caller for a MSI-X interrupt allocation. Aside of consistency
    failures it also detects a failure when the size of the MSI-X hardware table
    in the device is smaller than the size of the entries array.

    That's wrong for the case of range allocations where the caller provides
    the minimum and the maximum number of vectors to allocate, when the
    hardware size is greater or equal than the mininum, but smaller than the
    maximum.

    Remove the hardware size check completely from that function and just
    ensure that the entires array up to the maximum size is consistent.

    The limitation and range checking versus the hardware size happens
    independently of that afterwards anyway because the entries array is
    optional.

    Fixes: 4644d22eb673 ("PCI/MSI: Validate MSI-X contiguous restriction early")
    Reported-by: David Laight <David.Laight@aculab.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/87v8i3sg62.ffs@tglx
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6402e1238e725139b17c3689c8a8566be9080ac5
Author: Alyssa Ross <hi@alyssa.is>
Date:   Sun Mar 26 18:21:21 2023 +0000

    purgatory: fix disabling debug info

    commit d83806c4c0cccc0d6d3c3581a11983a9c186a138 upstream.

    Since 32ef9e5054ec, -Wa,-gdwarf-2 is no longer used in KBUILD_AFLAGS.
    Instead, it includes -g, the appropriate -gdwarf-* flag, and also the
    -Wa versions of both of those if building with Clang and GNU as.  As a
    result, debug info was being generated for the purgatory objects, even
    though the intention was that it not be.

    Fixes: 32ef9e5054ec ("Makefile.debug: re-enable debug info for .S files")
    Signed-off-by: Alyssa Ross <hi@alyssa.is>
    Cc: stable@vger.kernel.org
    Acked-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9d9f5b437684d2e0d613332df711a4fdd1188885
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Tue Apr 18 19:38:58 2023 +0800
```

```
    LoongArch: Make WriteCombine configurable for ioremap()

    commit 16c52e503043aed1e2a2ce38d9249de5936c1f6b upstream.

    LoongArch maintains cache coherency in hardware, but when paired with
    LS7A chipsets the WUC attribute (Weak-ordered UnCached, which is similar
    to WriteCombine) is out of the scope of cache coherency machanism for
    PCIe devices (this is a PCIe protocol violation, which may be fixed in
    newer chipsets).

    This means WUC can only used for write-only memory regions now, so this
    option is disabled by default, making WUC silently fallback to SUC for
    ioremap(). You can enable this option if the kernel is ensured to run on
    hardware without this bug.

    Kernel parameter writecombine=on/off can be used to override the Kconfig
    option.

    Cc: stable@vger.kernel.org
    Suggested-by: WANG Xuerui <kernel@xen0n.name>
    Reviewed-by: WANG Xuerui <kernel@xen0n.name>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 13e28ed3bfbb474a511e9cfdb92180ec53568247
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Sat Feb 25 15:52:56 2023 +0800

    LoongArch: Make -mstrict-align configurable

    commit 41596803302d83a67a80dc1efef4e51ac46acabb upstream.

    Introduce Kconfig option ARCH_STRICT_ALIGN to make -mstrict-align be
    configurable.

    Not all LoongArch cores support h/w unaligned access, we can use the
    -mstrict-align build parameter to prevent unaligned accesses.

    CPUs with h/w unaligned access support:
    Loongson-2K2000/2K3000/3A5000/3C5000/3D5000.

    CPUs without h/w unaligned access support:
    Loongson-2K500/2K1000.

    This option is enabled by default to make the kernel be able to run on
    all LoongArch systems. But you can disable it manually if you want to
    run kernel only on systems with h/w unaligned access support in order to
    optimise for performance.

    Reviewed-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 587bc862f72f400653230f2074faf28de27c18d2
Author: Jiaxun Yang <jiaxun.yang@flygoat.com>
Date:   Sat Apr 8 21:33:48 2023 +0100

    MIPS: Define RUNTIME_DISCARD_EXIT in LD script

    commit 6dcbd0a69c84a8ae7a442840a8cf6b1379dc8f16 upstream.

    MIPS's exit sections are discarded at runtime as well.

    Fixes link error:
```

```
    `.exit.text' referenced in section `__jump_table' of fs/fuse/inode.o:
    defined in discarded section `.exit.text' of fs/fuse/inode.o

    Fixes: 99cb0d917ffa ("arch: fix broken BuildID for arm64 and riscv")
    Reported-by: "kernelci.org bot" <bot@kernelci.org>
    Signed-off-by: Jiaxun Yang <jiaxun.yang@flygoat.com>
    Signed-off-by: Thomas Bogendoerfer <tsbogend@alpha.franken.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9762e58cbda22d1ca360ecca5715a3424c1e3bd3
Author: Dan Carpenter <dan.carpenter@linaro.org>
Date:   Wed Apr 19 13:16:13 2023 +0300

    KVM: arm64: Fix buffer overflow in kvm_arm_set_fw_reg()

    commit a25bc8486f9c01c1af6b6c5657234b2eee2c39d6 upstream.

    The KVM_REG_SIZE() comes from the ioctl and it can be a power of two
    between 0-32768 but if it is more than sizeof(long) this will corrupt
    memory.

    Fixes: 99adb567632b ("KVM: arm/arm64: Add save/restore support for firmware workaround
state")
    Signed-off-by: Dan Carpenter <dan.carpenter@linaro.org>
    Reviewed-by: Steven Price <steven.price@arm.com>
    Reviewed-by: Eric Auger <eric.auger@redhat.com>
    Reviewed-by: Marc Zyngier <maz@kernel.org>
    Link: https://lore.kernel.org/r/4efbab8c-640f-43b2-8ac6-6d68e08280fe@kili.mountain
    Signed-off-by: Oliver Upton <oliver.upton@linux.dev>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7a7d89a089fce5607a9bcc0d9ebf581de017fda8
Author: Marc Zyngier <maz@kernel.org>
Date:   Tue Apr 18 13:57:37 2023 +0100

    KVM: arm64: Make vcpu flag updates non-preemptible

    commit 35dcb3ac663a16510afc27ba2725d70c15e012a5 upstream.

    Per-vcpu flags are updated using a non-atomic RMW operation.
    Which means it is possible to get preempted between the read and
    write operations.

    Another interesting thing to note is that preemption also updates
    flags, as we have some flag manipulation in both the load and put
    operations.

    It is thus possible to lose information communicated by either
    load or put, as the preempted flag update will overwrite the flags
    when the thread is resumed. This is specially critical if either
    load or put has stored information which depends on the physical
    CPU the vcpu runs on.

    This results in really elusive bugs, and kudos must be given to
    Mostafa for the long hours of debugging, and finally spotting
    the problem.

    Fix it by disabling preemption during the RMW operation, which
    ensures that the state stays consistent. Also upgrade vcpu_get_flag
    path to use READ_ONCE() to make sure the field is always atomically
    accessed.

    Fixes: e87abb73e594 ("KVM: arm64: Add helpers to manipulate vcpu flags among a set")
    Reported-by: Mostafa Saleh <smostafa@google.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
```

```
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230418125737.2327972-1-maz@kernel.org
    Signed-off-by: Oliver Upton <oliver.upton@linux.dev>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e91802036b3e6e6e803a17ddf5783a6354fe5380
Author: Paulo Alcantara <pc@manguebit.com>
Date:   Sun Apr 16 15:38:28 2023 -0300

    cifs: avoid dup prefix path in dfs_get_automount_devname()

    commit d5a863a153e90996ab2aef6b9e08d509f4d5662b upstream.

    @server->origin_fullpath already contains the tree name + optional
    prefix, so avoid calling __build_path_from_dentry_optional_prefix() as
    it might end up duplicating prefix path from @cifs_sb->prepath into
    final full path.

    Instead, generate DFS full path by simply merging
    @server->origin_fullpath with dentry's path.

    This fixes the following case

            mount.cifs //root/dfs/dir /mnt/ -o ...
            ls /mnt/link

    where cifs_dfs_do_automount() will call smb3_parse_devname() with
    @devname set to "//root/dfs/dir/link" instead of
    "//root/dfs/dir/dir/link".

    Fixes: 7ad54b98fc1f ("cifs: use origin fullpath for automounts")
    Cc: <stable@vger.kernel.org> # 6.2+
    Signed-off-by: Paulo Alcantara (SUSE) <pc@manguebit.com>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d30989fe9a176565d360376d4bc2ea1c61cbbac
Author: Liam R. Howlett <Liam.Howlett@oracle.com>
Date:   Fri Apr 14 14:59:19 2023 -0400

    mm/mmap: regression fix for unmapped_area{_topdown}

    commit 58c5d0d6d522112577c7eeb71d382ea642ed7be4 upstream.

    The maple tree limits the gap returned to a window that specifically fits
    what was asked.  This may not be optimal in the case of switching search
    directions or a gap that does not satisfy the requested space for other
    reasons.  Fix the search by retrying the operation and limiting the search
    window in the rare occasion that a conflict occurs.

    Link: https://lkml.kernel.org/r/20230414185919.4175572-1-Liam.Howlett@oracle.com
    Fixes: 3499a13168da ("mm/mmap: use maple tree for unmapped_area{_topdown}")
    Signed-off-by: Liam R. Howlett <Liam.Howlett@oracle.com>
    Reported-by: Rick Edgecombe <rick.p.edgecombe@intel.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit efcdda22002f6c550fdd71cb72df0f0befd7a1b2
Author: Mel Gorman <mgorman@techsingularity.net>
Date:   Fri Apr 14 15:14:29 2023 +0100

    mm: page_alloc: skip regions with hugetlbfs pages when allocating 1G pages

    commit 4d73ba5fa710fe7d432e0b271e6fecd252aef66e upstream.
```

A bug was reported by Yuanxi Liu where allocating 1G pages at runtime is
taking an excessive amount of time for large amounts of memory. Further
testing allocating huge pages that the cost is linear i.e. if allocating
1G pages in batches of 10 then the time to allocate nr_hugepages from
10->20->30->etc increases linearly even though 10 pages are allocated at
each step. Profiles indicated that much of the time is spent checking the
validity within already existing huge pages and then attempting a
migration that fails after isolating the range, draining pages and a whole
lot of other useless work.

Commit eb14d4eefdc4 ("mm,page_alloc: drop unnecessary checks from
pfn_range_valid_contig") removed two checks, one which ignored huge pages
for contiguous allocations as huge pages can sometimes migrate. While
there may be value on migrating a 2M page to satisfy a 1G allocation, it's
potentially expensive if the 1G allocation fails and it's pointless to try
moving a 1G page for a new 1G allocation or scan the tail pages for valid
PFNs.

Reintroduce the PageHuge check and assume any contiguous region with
hugetlbfs pages is unsuitable for a new 1G allocation.

The hpagealloc test allocates huge pages in batches and reports the
average latency per page over time. This test happens just after boot
when fragmentation is not an issue. Units are in milliseconds.

hpagealloc

		6.3.0-rc6 vanilla	6.3.0-rc6 hugeallocrevert-v1r1	6.3.0-rc6 hugeallocsimple-v1r2
Min	Latency	26.42 (0.00%)	5.07 (80.82%)	18.94 (28.30%)
1st-qrtle	Latency	356.61 (0.00%)	5.34 (98.50%)	19.85 (94.43%)
2nd-qrtle	Latency	697.26 (0.00%)	5.47 (99.22%)	20.44 (97.07%)
3rd-qrtle	Latency	972.94 (0.00%)	5.50 (99.43%)	20.81 (97.86%)
Max-1	Latency	26.42 (0.00%)	5.07 (80.82%)	18.94 (28.30%)
Max-5	Latency	82.14 (0.00%)	5.11 (93.78%)	19.31 (76.49%)
Max-10	Latency	150.54 (0.00%)	5.20 (96.55%)	19.43 (87.09%)
Max-90	Latency	1164.45 (0.00%)	5.53 (99.52%)	20.97 (98.20%)
Max-95	Latency	1223.06 (0.00%)	5.55 (99.55%)	21.06 (98.28%)
Max-99	Latency	1278.67 (0.00%)	5.57 (99.56%)	22.56 (98.24%)
Max	Latency	1310.90 (0.00%)	8.06 (99.39%)	26.62 (97.97%)
Amean	Latency	678.36 (0.00%)	5.44 * 99.20%*	20.44 * 96.99%*

	6.3.0-rc6 vanilla	6.3.0-rc6 revert-v1	6.3.0-rc6 hugeallocfix-v2
Duration User	0.28	0.27	0.30
Duration System	808.66	17.77	35.99
Duration Elapsed	830.87	18.08	36.33

The vanilla kernel is poor, taking up to 1.3 second to allocate a huge
page and almost 10 minutes in total to run the test. Reverting the
problematic commit reduces it to 8ms at worst and the patch takes 26ms.
This patch fixes the main issue with skipping huge pages but leaves the
page_count() out because a page with an elevated count potentially can
migrate.

BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=217022
Link: https://lkml.kernel.org/r/20230414141429.pwgieuwluxwez3rj@techsingularity.net
Fixes: eb14d4eefdc4 ("mm,page_alloc: drop unnecessary checks from
pfn_range_valid_contig")
Signed-off-by: Mel Gorman <mgorman@techsingularity.net>
Reported-by: Yuanxi Liu <y.liu@naruida.com>
Acked-by: Vlastimil Babka <vbabka@suse.cz>
Reviewed-by: David Hildenbrand <david@redhat.com>
Acked-by: Michal Hocko <mhocko@suse.com>
Reviewed-by: Oscar Salvador <osalvador@suse.de>

```
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cd0eacb7f87110fb664d0f6c7f4c583d081da297
Author: Alexander Potapenko <glider@google.com>
Date:   Thu Apr 13 15:12:20 2023 +0200

    mm: kmsan: handle alloc failures in kmsan_vmap_pages_range_noflush()

    commit 47ebd0310e89c087f56e58c103c44b72a2f6b216 upstream.

    As reported by Dipanjan Das, when KMSAN is used together with kernel fault
    injection (or, generally, even without the latter), calls to kcalloc() or
    __vmap_pages_range_noflush() may fail, leaving the metadata mappings for
    the virtual mapping in an inconsistent state.  When these metadata
    mappings are accessed later, the kernel crashes.

    To address the problem, we return a non-zero error code from
    kmsan_vmap_pages_range_noflush() in the case of any allocation/mapping
    failure inside it, and make vmap_pages_range_noflush() return an error if
    KMSAN fails to allocate the metadata.

    This patch also removes KMSAN_WARN_ON() from vmap_pages_range_noflush(),
    as these allocation failures are not fatal anymore.

    Link: https://lkml.kernel.org/r/20230413131223.4135168-1-glider@google.com
    Fixes: b073d7f8aee4 ("mm: kmsan: maintain KMSAN metadata for page operations")
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Reported-by: Dipanjan Das <mail.dipanjan.das@gmail.com>
      Link: https://lore.kernel.org/linux-
mm/CANX2M5ZRrRA64k0hOif02TjmY9kbbO2aCBPyq79es34RXZ=cAw@mail.gmail.com/
    Reviewed-by: Marco Elver <elver@google.com>
    Cc: Christoph Hellwig <hch@infradead.org>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Cc: Uladzislau Rezki (Sony) <urezki@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e68157b0d584a9e2f10dc09c9e7fdd9cd657d091
Author: Alexander Potapenko <glider@google.com>
Date:   Thu Apr 13 15:12:21 2023 +0200

    mm: kmsan: handle alloc failures in kmsan_ioremap_page_range()

    commit fdea03e12aa2a44a7bb34144208be97fc25dfd90 upstream.

    Similarly to kmsan_vmap_pages_range_noflush(), kmsan_ioremap_page_range()
    must also properly handle allocation/mapping failures.  In the case of
    such, it must clean up the already created metadata mappings and return an
    error code, so that the error can be propagated to ioremap_page_range().
    Without doing so, KMSAN may silently fail to bring the metadata for the
    page range into a consistent state, which will result in user-visible
    crashes when trying to access them.

    Link: https://lkml.kernel.org/r/20230413131223.4135168-2-glider@google.com
    Fixes: b073d7f8aee4 ("mm: kmsan: maintain KMSAN metadata for page operations")
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Reported-by: Dipanjan Das <mail.dipanjan.das@gmail.com>
      Link: https://lore.kernel.org/linux-
mm/CANX2M5ZRrRA64k0hOif02TjmY9kbbO2aCBPyq79es34RXZ=cAw@mail.gmail.com/
    Reviewed-by: Marco Elver <elver@google.com>
    Cc: Christoph Hellwig <hch@infradead.org>
```

```
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Cc: Uladzislau Rezki (Sony) <urezki@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 252b0ecee80fd0763d4393c3ff76c56fa18016db
Author: Naoya Horiguchi <naoya.horiguchi@nec.com>
Date:   Thu Apr 6 17:20:04 2023 +0900

    mm/huge_memory.c: warn with pr_warn_ratelimited instead of VM_WARN_ON_ONCE_FOLIO

    commit 4737edbbdd4958ae29ca6a310a6a2fa4e0684b01 upstream.

    split_huge_page_to_list() WARNs when called for huge zero pages, which
    sounds to me too harsh because it does not imply a kernel bug, but just
    notifies the event to admins.  On the other hand, this is considered as
    critical by syzkaller and makes its testing less efficient, which seems to
    me harmful.

    So replace the VM_WARN_ON_ONCE_FOLIO with pr_warn_ratelimited.

    Link: https://lkml.kernel.org/r/20230406082004.2185420-1-naoya.horiguchi@linux.dev
    Fixes: 478d134e9506 ("mm/huge_memory: do not overkill when splitting huge_zero_page")
    Signed-off-by: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Reported-by: syzbot+07a218429c8d19b1fb25@syzkaller.appspotmail.com
      Link: https://lore.kernel.org/lkml/000000000000a6f34a05e6efcd01@google.com/
    Reviewed-by: Yang Shi <shy828301@gmail.com>
    Cc: Miaohe Lin <linmiaohe@huawei.com>
    Cc: Tetsuo Handa <penguin-kernel@i-love.sakura.ne.jp>
    Cc: Xu Yu <xuyu@linux.alibaba.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 423886491291549 75f28e35c5e8960be514c2156
Author: Peter Xu <peterx@redhat.com>
Date:   Wed Apr 5 11:51:20 2023 -0400

    mm/khugepaged: check again on anon uffd-wp during isolation

    commit dd47ac428c3f5f3bcabe845f36be870fe6c20784 upstream.

    Khugepaged collapse an anonymous thp in two rounds of scans.  The 2nd
    round done in __collapse_huge_page_isolate() after
    hpage_collapse_scan_pmd(), during which all the locks will be released
    temporarily.  It means the pgtable can change during this phase before 2nd
    round starts.

    It's logically possible some ptes got wr-protected during this phase, and
    we can errornously collapse a thp without noticing some ptes are
    wr-protected by userfault.  e1e267c7928f wanted to avoid it but it only
    did that for the 1st phase, not the 2nd phase.

    Since __collapse_huge_page_isolate() happens after a round of small page
    swapins, we don't need to worry on any !present ptes - if it existed
    khugepaged will already bail out.  So we only need to check present ptes
    with uffd-wp bit set there.

    This is something I found only but never had a reproducer, I thought it
    was one caused a bug in Muhammad's recent pagemap new ioctl work, but it
    turns out it's not the cause of that but an userspace bug.  However this
    seems to still be a real bug even with a very small race window, still
    worth to have it fixed and copy stable.
```

```
    Link: https://lkml.kernel.org/r/20230405155120.3608140-1-peterx@redhat.com
    Fixes: e1e267c7928f ("khugepaged: skip collapse if uffd-wp detected")
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Yang Shi <shy828301@gmail.com>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: Mike Rapoport <rppt@linux.vnet.ibm.com>
    Cc: Nadav Amit <nadav.amit@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7c24603bc26aee35be535e2d9545888ffdeffe95
Author: David Hildenbrand <david@redhat.com>
Date:   Wed Apr 5 18:02:35 2023 +0200

    mm/userfaultfd: fix uffd-wp handling for THP migration entries

    commit 24bf08c4376be417f16ceb609188b16f461b0443 upstream.

    Looks like what we fixed for hugetlb in commit 44f86392bdd1 ("mm/hugetlb:
    fix uffd-wp handling for migration entries in
    hugetlb_change_protection()") similarly applies to THP.

    Setting/clearing uffd-wp on THP migration entries is not implemented
    properly.  Further, while removing migration PMDs considers the uffd-wp
    bit, inserting migration PMDs does not consider the uffd-wp bit.

    We have to set/clear independently of the migration entry type in
    change_huge_pmd() and properly copy the uffd-wp bit in
    set_pmd_migration_entry().

    Verified using a simple reproducer that triggers migration of a THP, that
    the set_pmd_migration_entry() no longer loses the uffd-wp bit.

    Link: https://lkml.kernel.org/r/20230405160236.587705-2-david@redhat.com
    Fixes: f45ec5ff16a7 ("userfaultfd: wp: support swap and page migration")
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Peter Xu <peterx@redhat.com>
    Cc: <stable@vger.kernel.org>
    Cc: Muhammad Usama Anjum <usama.anjum@collabora.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 18471cb160b2298908d2d8d99ab8c0b142c752ba
Author: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
Date:   Thu Mar 30 09:38:22 2023 -0400

    mm: fix memory leak on mm_init error handling

    commit b20b0368c614c609badfe16fbd113dfb4780acd9 upstream.

    commit f1a7941243c1 ("mm: convert mm's rss stats into percpu_counter")
    introduces a memory leak by missing a call to destroy_context() when a
    percpu_counter fails to allocate.

    Before introducing the per-cpu counter allocations, init_new_context() was
    the last call that could fail in mm_init(), and thus there was no need to
    ever invoke destroy_context() in the error paths.  Adding the following
    percpu counter allocations adds error paths after init_new_context(),
    which means its associated destroy_context() needs to be called when
    percpu counters fail to allocate.

    Link: https://lkml.kernel.org/r/20230330133822.66271-1-mathieu.desnoyers@efficios.com
```

```
    Fixes: f1a7941243c1 ("mm: convert mm's rss stats into percpu_counter")
    Signed-off-by: Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
    Acked-by: Shakeel Butt <shakeelb@google.com>
    Cc: Marek Szyprowski <m.szyprowski@samsung.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8a123157b7244267ffe8737c3b27ef74afd89f0b
Author: Sascha Hauer <s.hauer@pengutronix.de>
Date:   Mon Apr 17 14:37:47 2023 +0200

    drm/rockchip: vop2: Use regcache_sync() to fix suspend/resume

    commit b63a553e8f5aa6574eeb535a551817a93c426d8c upstream.

    afa965a45e01 ("drm/rockchip: vop2: fix suspend/resume") uses
    regmap_reinit_cache() to fix the suspend/resume issue with the VOP2
    driver. During discussion it came up that we should rather use
    regcache_sync() instead. As the original patch is already applied
    fix this up in this follow-up patch.

    Fixes: afa965a45e01 ("drm/rockchip: vop2: fix suspend/resume")
    Cc: stable@vger.kernel.org
    Signed-off-by: Sascha Hauer <s.hauer@pengutronix.de>
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230417123747.2179695-1-
s.hauer@pengutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 935e5b5bc8239cd8e1cc647c857bb866f3766e59
Author: Sascha Hauer <s.hauer@pengutronix.de>
Date:   Thu Apr 13 16:43:47 2023 +0200

    drm/rockchip: vop2: fix suspend/resume

    commit afa965a45e01e541cdbe5c8018226eff117610f0 upstream.

    During a suspend/resume cycle the VO power domain will be disabled and
    the VOP2 registers will reset to their default values. After that the
    cached register values will be out of sync and the read/modify/write
    operations we do on the window registers will result in bogus values
    written. Fix this by re-initializing the register cache each time we
    enable the VOP2. With this the VOP2 will show a picture after a
    suspend/resume cycle whereas without this the screen stays dark.

    Fixes: 604be85547ce4 ("drm/rockchip: Add VOP2 driver")
    Cc: stable@vger.kernel.org
    Signed-off-by: Sascha Hauer <s.hauer@pengutronix.de>
    Tested-by: Chris Morgan <macromorgan@hotmail.com>
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230413144347.3506023-1-
s.hauer@pengutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 89da3543185d001bf90a36c25b17a95a45d429ed
Author: Dmytro Laktyushkin <Dmytro.Laktyushkin@amd.com>
Date:   Mon Apr 3 10:13:12 2023 -0400

    drm/amd/display: set dcn315 lb bpp to 48

    commit 6d9240c46f7419aa3210353b5f52cc63da5a6440 upstream.

    [Why & How]
    Fix a typo for dcn315 line buffer bpp.
```

```
    Reviewed-by: Jun Lei <Jun.Lei@amd.com>
    Acked-by: Qingqing Zhuo <qingqing.zhuo@amd.com>
    Signed-off-by: Dmytro Laktyushkin <Dmytro.Laktyushkin@amd.com>
    Tested-by: Daniel Wheeler <daniel.wheeler@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 56a03f64fedf49a4f81c5605167b6e7bb0300a59
Author: Alan Liu <HaoPing.Liu@amd.com>
Date:   Fri Apr 14 18:39:52 2023 +0800

    drm/amdgpu: Fix desktop freezed after gpu-reset

    commit c8b5a95b570949536a2b75cd8fc4f1de0bc60629 upstream.

    [Why]
    After gpu-reset, sometimes the driver fails to enable vblank irq,
    causing flip_done timed out and the desktop freezed.

    During gpu-reset, we disable and enable vblank irq in dm_suspend() and
    dm_resume(). Later on in amdgpu_irq_gpu_reset_resume_helper(), we check
    irqs' refcount and decide to enable or disable the irqs again.

    However, we have 2 sets of API for controling vblank irq, one is
    dm_vblank_get/put() and another is amdgpu_irq_get/put(). Each API has
    its own refcount and flag to store the state of vblank irq, and they
    are not synchronized.

    In drm we use the first API to control vblank irq but in
    amdgpu_irq_gpu_reset_resume_helper() we use the second set of API.

    The failure happens when vblank irq was enabled by dm_vblank_get()
    before gpu-reset, we have vblank->enabled true. However, during
    gpu-reset, in amdgpu_irq_gpu_reset_resume_helper() vblank irq's state
    checked from amdgpu_irq_update() is DISABLED. So finally it disables
    vblank irq again. After gpu-reset, if there is a cursor plane commit,
    the driver will try to enable vblank irq by calling drm_vblank_enable(),
    but the vblank->enabled is still true, so it fails to turn on vblank
    irq and causes flip_done can't be completed in vblank irq handler and
    desktop become freezed.

    [How]
    Combining the 2 vblank control APIs by letting drm's API finally calls
    amdgpu_irq's API, so the irq's refcount and state of both APIs can be
    synchronized. Also add a check to prevent refcount from being less then
    0 in amdgpu_irq_put().

    v2:
    - Add warning in amdgpu_irq_enable() if the irq is already disabled.
    - Call dc_interrupt_set() in dm_set_vblank() to avoid refcount change
      if it is in gpu-reset.

    v3:
    - Improve commit message and code comments.

    Signed-off-by: Alan Liu <HaoPing.Liu@amd.com>
    Reviewed-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e2b789bc3dc34edc87ffb85634967d24ed351acb
Author: Ville Syrjälä <ville.syrjala@linux.intel.com>
```

```
Date:   Wed Mar 29 20:24:33 2023 +0300

    drm/i915: Fix fast wake AUX sync len

    commit e1c71f8f918047ce822dc19b42ab1261ed259fd1 upstream.

    Fast wake should use 8 SYNC pulses for the preamble
    and 10-16 SYNC pulses for the precharge. Reduce our
    fast wake SYNC count to match the maximum value.
    We also use the maximum precharge length for normal
    AUX transactions.

    Cc: stable@vger.kernel.org
    Cc: Jouni Högander <jouni.hogander@intel.com>
    Signed-off-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230329172434.18744-1-
ville.syrjala@linux.intel.com
    Reviewed-by: Jouni Högander <jouni.hogander@intel.com>
    (cherry picked from commit 605f7c73133341d4b762cbd9a22174cc22d4c38b)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5e82ffbc886da7b1bfac811d88bf9a48edb4062e
Author: Bhavya Kapoor <b-kapoor@ti.com>
Date:   Fri Mar 17 14:57:11 2023 +0530

    mmc: sdhci_am654: Set HIGH_SPEED_ENA for SDR12 and SDR25

    commit 2265098fd6a6272fde3fd1be5761f2f5895bd99a upstream.

    Timing Information in Datasheet assumes that HIGH_SPEED_ENA=1 should be
    set for SDR12 and SDR25 modes. But sdhci_am654 driver clears
    HIGH_SPEED_ENA register. Thus, Modify sdhci_am654 to not clear
    HIGH_SPEED_ENA (HOST_CONTROL[2]) bit for SDR12 and SDR25 speed modes.

    Fixes: e374e87538f4 ("mmc: sdhci_am654: Clear HISPD_ENA in some lower speed modes")
    Signed-off-by: Bhavya Kapoor <b-kapoor@ti.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230317092711.660897-1-b-kapoor@ti.com
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5d20dd4361593f9947b522b2838be5a62f752a63
Author: Baokun Li <libaokun1@huawei.com>
Date:   Mon Apr 10 21:08:26 2023 +0800

    writeback, cgroup: fix null-ptr-deref write in bdi_split_work_to_wbs

    commit 1ba1199ec5747f475538c0d25a32804e5ba1dfde upstream.

    KASAN report null-ptr-deref:
    ==================================================================
    BUG: KASAN: null-ptr-deref in bdi_split_work_to_wbs+0x5c5/0x7b0
    Write of size 8 at addr 0000000000000000 by task sync/943
    CPU: 5 PID: 943 Comm: sync Tainted: 6.3.0-rc5-next-20230406-dirty #461
    Call Trace:
     <TASK>
     dump_stack_lvl+0x7f/0xc0
     print_report+0x2ba/0x340
     kasan_report+0xc4/0x120
     kasan_check_range+0x1b7/0x2e0
     __kasan_check_write+0x24/0x40
     bdi_split_work_to_wbs+0x5c5/0x7b0
     sync_inodes_sb+0x195/0x630
     sync_inodes_one_sb+0x3a/0x50
```

```
 iterate_supers+0x106/0x1b0
 ksys_sync+0x98/0x160
[...]
==================================================================
```

The race that causes the above issue is as follows:

```
           cpu1                        cpu2
-------------------------|-------------------------
inode_switch_wbs
 INIT_WORK(&isw->work, inode_switch_wbs_work_fn)
 queue_rcu_work(isw_wq, &isw->work)
 // queue_work async
  inode_switch_wbs_work_fn
   wb_put_many(old_wb, nr_switched)
    percpu_ref_put_many
     ref->data->release(ref)
     cgwb_release
      queue_work(cgwb_release_wq, &wb->release_work)
      // queue_work async
       &wb->release_work
       cgwb_release_workfn
                              ksys_sync
                               iterate_supers
                                sync_inodes_one_sb
                                 sync_inodes_sb
                                  bdi_split_work_to_wbs
                                   kmalloc(sizeof(*work), GFP_ATOMIC)
                                   // alloc memory failed
        percpu_ref_exit
         ref->data = NULL
         kfree(data)
                                   wb_get(wb)
                                    percpu_ref_get(&wb->refcnt)
                                     percpu_ref_get_many(ref, 1)
                                      atomic_long_add(nr, &ref->data->count)
                                       atomic64_add(i, v)
                                       // trigger null-ptr-deref
```

bdi_split_work_to_wbs() traverses &bdi->wb_list to split work into all wbs. If the allocation of new work fails, the on-stack fallback will be used and the reference count of the current wb is increased afterwards. If cgroup writeback membership switches occur before getting the reference count and the current wb is released as old_wd, then calling wb_get() or wb_put() will trigger the null pointer dereference above.

This issue was introduced in v4.3-rc7 (see fix tag1). Both sync_inodes_sb() and __writeback_inodes_sb_nr() calls to bdi_split_work_to_wbs() can trigger this issue. For scenarios called via sync_inodes_sb(), originally commit 7fc5854f8c6e ("writeback: synchronize sync(2) against cgroup writeback membership switches") reduced the possibility of the issue by adding wb_switch_rwsem, but in v5.14-rc1 (see fix tag2) removed the "inode_io_list_del_locked(inode, old_wb)" from inode_switch_wbs_work_fn() so that wb->state contains WB_has_dirty_io, thus old_wb is not skipped when traversing wbs in bdi_split_work_to_wbs(), and the issue becomes easily reproducible again.

To solve this problem, percpu_ref_exit() is called under RCU protection to avoid race between cgwb_release_workfn() and bdi_split_work_to_wbs(). Moreover, replace wb_get() with wb_tryget() in bdi_split_work_to_wbs(), and skip the current wb if wb_tryget() fails because the wb has already been shutdown.

Link: https://lkml.kernel.org/r/20230410130826.1492525-1-libaokun1@huawei.com
Fixes: b817525a4a80 ("writeback: bdi_writeback iteration must not skip dying ones")

```
    Signed-off-by: Baokun Li <libaokun1@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Acked-by: Tejun Heo <tj@kernel.org>
    Cc: Alexander Viro <viro@zeniv.linux.org.uk>
    Cc: Andreas Dilger <adilger.kernel@dilger.ca>
    Cc: Christian Brauner <brauner@kernel.org>
    Cc: Dennis Zhou <dennis@kernel.org>
    Cc: Hou Tao <houtao1@huawei.com>
    Cc: yangerkun <yangerkun@huawei.com>
    Cc: Zhang Yi <yi.zhang@huawei.com>
    Cc: Jens Axboe <axboe@kernel.dk>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8fa5061d73e201248650281354135ba26ecf9e5
Author: Ondrej Mosnacek <omosnace@redhat.com>
Date:   Fri Feb 17 17:21:54 2023 +0100

    kernel/sys.c: fix and improve control flow in __sys_setres[ug]id()

    commit 659c0ce1cb9efc7f58d380ca4bb2a51ae9e30553 upstream.

    Linux Security Modules (LSMs) that implement the "capable" hook will
    usually emit an access denial message to the audit log whenever they
    "block" the current task from using the given capability based on their
    security policy.

    The occurrence of a denial is used as an indication that the given task
    has attempted an operation that requires the given access permission, so
    the callers of functions that perform LSM permission checks must take care
    to avoid calling them too early (before it is decided if the permission is
    actually needed to perform the requested operation).

    The __sys_setres[ug]id() functions violate this convention by first
    calling ns_capable_setid() and only then checking if the operation
    requires the capability or not.  It means that any caller that has the
    capability granted by DAC (task's capability set) but not by MAC (LSMs)
    will generate a "denied" audit record, even if is doing an operation for
    which the capability is not required.

    Fix this by reordering the checks such that ns_capable_setid() is checked
    last and -EPERM is returned immediately if it returns false.

    While there, also do two small optimizations:
    * move the capability check before prepare_creds() and
    * bail out early in case of a no-op.

    Link: https://lkml.kernel.org/r/20230217162154.837549-1-omosnace@redhat.com
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Ondrej Mosnacek <omosnace@redhat.com>
    Cc: Eric W. Biederman <ebiederm@xmission.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d299776014babe432be8d5a41923e79493fe8dde
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Sat Apr 1 22:03:27 2023 +0200

    memstick: fix memory leak if card device is never registered

    commit 4b6d621c9d859ff89e68cebf6178652592676013 upstream.

    When calling dev_set_name() memory is allocated for the name for the
```

```
    struct device.  Once that structure device is registered, or attempted
    to be registerd, with the driver core, the driver core will handle
    cleaning up that memory when the device is removed from the system.

    Unfortunatly for the memstick code, there is an error path that causes
    the struct device to never be registered, and so the memory allocated in
    dev_set_name will be leaked.  Fix that leak by manually freeing it right
    before the memory for the device is freed.

    Cc: Maxim Levitsky <maximlevitsky@gmail.com>
    Cc: Alex Dubov <oakad@yahoo.com>
    Cc: Ulf Hansson <ulf.hansson@linaro.org>
    Cc: "Rafael J. Wysocki" <rafael@kernel.org>
    Cc: Hans de Goede <hdegoede@redhat.com>
    Cc: Kay Sievers <kay.sievers@vrfy.org>
    Cc: linux-mmc@vger.kernel.org
    Fixes: 0252c3b4f018 ("memstick: struct device - replace bus_id with dev_name(),
dev_set_name()")
    Cc: stable <stable@kernel.org>
    Co-developed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Co-developed-by: Mirsad Goran Todorovac <mirsad.todorovac@alu.unizg.hr>
    Signed-off-by: Mirsad Goran Todorovac <mirsad.todorovac@alu.unizg.hr>
    Link: https://lore.kernel.org/r/20230401200327.16800-1-gregkh@linuxfoundation.org
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 32d92df24fd5d5080d165e03912d304a89c0cd42
Author: Steve Chou <steve_chou@pesi.com.tw>
Date:   Tue Apr 11 11:49:28 2023 +0800

    tools/mm/page_owner_sort.c: fix TGID output when cull=tg is used

    commit 9235756885e865070c4be2facda75262dbd85967 upstream.

    When using cull option with 'tg' flag, the fprintf is using pid instead
    of tgid. It should use tgid instead.

    Link: https://lkml.kernel.org/r/20230411034929.2071501-1-steve_chou@pesi.com.tw
    Fixes: 9c8a0a8e599f4a ("tools/vm/page_owner_sort.c: support for user-defined culling
rules")
    Signed-off-by: Steve Chou <steve_chou@pesi.com.tw>
    Cc: Jiajian Ye <yejiajian2018@email.szu.edu.cn>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 313a1822c7420f266c6c9ffbb053ad7c5b53f547
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Mon Apr 17 16:00:41 2023 +0200

    mptcp: fix accept vs worker race

    commit 63740448a32eb662e05894425b47bcc5814136f4 upstream.

    The mptcp worker and mptcp_accept() can race, as reported by Christoph:

    refcount_t: addition on 0; use-after-free.
    WARNING: CPU: 1 PID: 14351 at lib/refcount.c:25 refcount_warn_saturate+0x105/0x1b0
lib/refcount.c:25
    Modules linked in:
    CPU: 1 PID: 14351 Comm: syz-executor.2 Not tainted 6.3.0-rc1-gde5e8fd0123c #11
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.0-2.el7 04/01/2014
    RIP: 0010:refcount_warn_saturate+0x105/0x1b0 lib/refcount.c:25
    Code: 02 31 ff 89 de e8 1b f0 a7 ff 84 db 0f 85 6e ff ff ff e8 3e f5 a7 ff 48 c7 c7 d8
```

```
 c7 34 83 c6 05 6d 2d 0f 02 01 e8 cb 3d 90 ff <0f> 0b e9 4f ff ff ff e8 1f f5 a7 ff 0f b6 1d
 54 2d 0f 02 31 ff 89
    RSP: 0018:ffffc90000a47bf8 EFLAGS: 00010282
    RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000
    RDX: ffff88802eae98c0 RSI: ffffffff81097d4f RDI: 0000000000000001
    RBP: ffff88802e712180 R08: 0000000000000001 R09: 0000000000000000
    R10: 0000000000000001 R11: ffff88802eaea148 R12: ffff88802e712100
    R13: ffff88802e712a88 R14: ffff888005cb93a8 R15: ffff88802e712a88
    FS:  0000000000000000(0000) GS:ffff88803ed00000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007f277fd89120 CR3: 0000000035486002 CR4: 0000000000370ee0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     <TASK>
     __refcount_add include/linux/refcount.h:199 [inline]
     __refcount_inc include/linux/refcount.h:250 [inline]
     refcount_inc include/linux/refcount.h:267 [inline]
     sock_hold include/net/sock.h:775 [inline]
     __mptcp_close+0x4c6/0x4d0 net/mptcp/protocol.c:3051
     mptcp_close+0x24/0xe0 net/mptcp/protocol.c:3072
     inet_release+0x56/0xa0 net/ipv4/af_inet.c:429
     __sock_release+0x51/0xf0 net/socket.c:653
     sock_close+0x18/0x20 net/socket.c:1395
     __fput+0x113/0x430 fs/file_table.c:321
     task_work_run+0x96/0x100 kernel/task_work.c:179
     exit_task_work include/linux/task_work.h:38 [inline]
     do_exit+0x4fc/0x10c0 kernel/exit.c:869
     do_group_exit+0x51/0xf0 kernel/exit.c:1019
     get_signal+0x12b0/0x1390 kernel/signal.c:2859
     arch_do_signal_or_restart+0x25/0x260 arch/x86/kernel/signal.c:306
     exit_to_user_mode_loop kernel/entry/common.c:168 [inline]
     exit_to_user_mode_prepare+0x131/0x1a0 kernel/entry/common.c:203
     __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline]
     syscall_exit_to_user_mode+0x19/0x40 kernel/entry/common.c:296
     do_syscall_64+0x46/0x90 arch/x86/entry/common.c:86
     entry_SYSCALL_64_after_hwframe+0x72/0xdc
    RIP: 0033:0x7fec4b4926a9
    Code: Unable to access opcode bytes at 0x7fec4b49267f.
    RSP: 002b:00007fec49f9dd78 EFLAGS: 00000246 ORIG_RAX: 00000000000000ca
    RAX: fffffffffffffe00 RBX: 00000000006bc058 RCX: 00007fec4b4926a9
    RDX: 0000000000000000 RSI: 0000000000000080 RDI: 00000000006bc058
    RBP: 00000000006bc050 R08: 00000000007df998 R09: 00000000007df998
    R10: 0000000000000000 R11: 0000000000000246 R12: 00000000006bc05c
    R13: ffffffffffffffea8 R14: 000000000000000b R15: 000000000001fe40
     </TASK>

    The root cause is that the worker can force fallback to TCP the first
    mptcp subflow, actually deleting the unaccepted msk socket.

    We can explicitly prevent the race delaying the unaccepted msk deletion
    at listener shutdown time. In case the closed subflow is later accepted,
    just drop the mptcp context and let the user-space deal with the
    paired mptcp socket.

    Fixes: b6985b9b8295 ("mptcp: use the workqueue to destroy unaccepted sockets")
    Cc: stable@vger.kernel.org
    Reported-by: Christoph Paasch <cpaasch@apple.com>
    Link: https://github.com/multipath-tcp/mptcp_net-next/issues/375
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Reviewed-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Tested-by: Christoph Paasch <cpaasch@apple.com>
    Signed-off-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 64b66601308dae6105fbde964a339462a29c2a73
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Mon Apr 17 16:00:40 2023 +0200

    mptcp: stops worker on unaccepted sockets at listener close

    commit 2a6a870e44dd88f1a6a2893c65ef756a9edfb4c7 upstream.

    This is a partial revert of the blamed commit, with a relevant
    change: mptcp_subflow_queue_clean() now just change the msk
    socket status and stop the worker, so that the UaF issue addressed
    by the blamed commit is not re-introduced.

    The above prevents the mptcp worker from running concurrently with
    inet_csk_listen_stop(), as such race would trigger a warning, as
    reported by Christoph:

    RSP: 002b:00007f784fe09cd8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e
    WARNING: CPU: 0 PID: 25807 at net/ipv4/inet_connection_sock.c:1387
inet_csk_listen_stop+0x664/0x870 net/ipv4/inet_connection_sock.c:1387
    RAX: ffffffffffffffda RBX: 00000000006bc050 RCX: 00007f7850afd6a9
    RDX: 0000000000000000 RSI: 0000000020000340 RDI: 0000000000000004
    Modules linked in:
    RBP: 0000000000000002 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000246 R12: 00000000006bc05c
    R13: fffffffffffffea8 R14: 00000000006bc050 R15: 000000000001fe40

     </TASK>
    CPU: 0 PID: 25807 Comm: syz-executor.7 Not tainted 6.2.0-g778e54711659 #7
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.0-2.el7 04/01/2014
    RIP: 0010:inet_csk_listen_stop+0x664/0x870 net/ipv4/inet_connection_sock.c:1387
    RAX: 0000000000000000 RBX: ffff888100dfbd40 RCX: 0000000000000000
    RDX: ffff8881363aab80 RSI: ffffffff81c494f4 RDI: 0000000000000005
    RBP: ffff888126dad080 R08: 0000000000000005 R09: 0000000000000000
    R10: 0000000000000001 R11: 0000000000000000 R12: ffff888100dfe040
    R13: 0000000000000001 R14: 0000000000000000 R15: ffff888100dfbdd8
    FS:  00007f7850a2c800(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 0000001b32d26000 CR3: 000000012fdd8006 CR4: 0000000000770ef0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    PKRU: 55555554
    Call Trace:
     <TASK>
     __tcp_close+0x5b2/0x620 net/ipv4/tcp.c:2875
     __mptcp_close_ssk+0x145/0x3d0 net/mptcp/protocol.c:2427
     mptcp_destroy_common+0x8a/0x1c0 net/mptcp/protocol.c:3277
     mptcp_destroy+0x41/0x60 net/mptcp/protocol.c:3304
     __mptcp_destroy_sock+0x56/0x140 net/mptcp/protocol.c:2965
     __mptcp_close+0x38f/0x4a0 net/mptcp/protocol.c:3057
     mptcp_close+0x24/0xe0 net/mptcp/protocol.c:3072
     inet_release+0x53/0xa0 net/ipv4/af_inet.c:429
     __sock_release+0x4e/0xf0 net/socket.c:651
     sock_close+0x15/0x20 net/socket.c:1393
     __fput+0xff/0x420 fs/file_table.c:321
     task_work_run+0x8b/0xe0 kernel/task_work.c:179
     resume_user_mode_work include/linux/resume_user_mode.h:49 [inline]
     exit_to_user_mode_loop kernel/entry/common.c:171 [inline]
     exit_to_user_mode_prepare+0x113/0x120 kernel/entry/common.c:203
     __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline]
     syscall_exit_to_user_mode+0x1d/0x40 kernel/entry/common.c:296
     do_syscall_64+0x46/0x90 arch/x86/entry/common.c:86
     entry_SYSCALL_64_after_hwframe+0x72/0xdc
    RIP: 0033:0x7f7850af70dc
```

```
    RAX: 0000000000000000 RBX: 0000000000000004 RCX: 00007f7850af70dc
    RDX: 00007f7850a2c800 RSI: 0000000000000002 RDI: 0000000000000003
    RBP: 00000000006bd980 R08: 0000000000000000 R09: 00000000000018a0
    R10: 00000000316338a4 R11: 0000000000000293 R12: 0000000000211e31
    R13: 00000000006bc05c R14: 00007f785062c000 R15: 0000000000211af0

    Fixes: 0a3f4f1f9c27 ("mptcp: fix UaF in listener shutdown")
    Cc: stable@vger.kernel.org
    Reported-by: Christoph Paasch <cpaasch@apple.com>
    Link: https://github.com/multipath-tcp/mptcp_net-next/issues/371
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Reviewed-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5da6416d12d43f3d1710b0a30e42c624bd16649a
Author: Ryusuke Konishi <konishi.ryusuke@gmail.com>
Date:   Tue Apr 18 02:35:13 2023 +0900

    nilfs2: initialize unused bytes in segment summary blocks

    commit ef832747a82dfbc22a3702219cc716f449b24e4a upstream.

    Syzbot still reports uninit-value in nilfs_add_checksums_on_logs() for
    KMSAN enabled kernels after applying commit 7397031622e0 ("nilfs2:
    initialize "struct nilfs_binfo_dat"->bi_pad field").

    This is because the unused bytes at the end of each block in segment
    summaries are not initialized.  So this fixes the issue by padding the
    unused bytes with null bytes.

    Link: https://lkml.kernel.org/r/20230417173513.12598-1-konishi.ryusuke@gmail.com
    Signed-off-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
    Tested-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
    Reported-by: syzbot+048585f3f4227bb2b49b@syzkaller.appspotmail.com
      Link: https://syzkaller.appspot.com/bug?extid=048585f3f4227bb2b49b
    Cc: Alexander Potapenko <glider@google.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 354312d4db9354ac6d9a4e3d86cd0d7e70794fbd
Author: Peter Ujfalusi <peter.ujfalusi@linux.intel.com>
Date:   Mon Apr 3 12:09:09 2023 +0300

    ASoC: SOF: ipc4-topology: Clarify bind failure caused by missing fw_module

    commit de6aa72b265b72bca2b1897d5000c8f0147d3157 upstream.

    The original patch uses a feature in lib/vsprintf.c to handle the invalid
    address when tring to print *_fw_module->man4_module_entry.name when the
    *rc_fw_module is NULL.
    This case is handled by check_pointer_msg() internally and turns the
    invalid pointer to '(efault)' for printing but it is hiding useful
    information about the circumstances. Change the print to emmit the name
    of the widget and a note on which side's fw_module is missing.

    Fixes: e3720f92e023 ("ASoC: SOF: avoid a NULL dereference with unsupported widgets")
    Reported-by: Dan Carpenter <error27@gmail.com>
    Link: https://lore.kernel.org/alsa-devel/4826f662-42f0-4a82-ba32-
8bf5f8a03256@kili.mountain/
    Signed-off-by: Peter Ujfalusi <peter.ujfalusi@linux.intel.com>
    Rule: 'Cc: stable@vger.kernel.org' or 'commit <sha1> upstream.'
    Link: https://lore.kernel.org/r/20230403090909.18233-1-peter.ujfalusi@linux.intel.com
```

```
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e72b3194bbabf6dfae7ae882306f435188365b8f
Author: Peng Zhang <zhangpeng.00@bytedance.com>
Date:   Tue Apr 11 12:10:04 2023 +0800

    maple_tree: fix a potential memory leak, OOB access, or other unpredictable bug

    commit 1f5f12ece722aacea1769fb644f27790ede339dc upstream.

    In mas_alloc_nodes(), "node->node_count = 0" means to initialize the
    node_count field of the new node, but the node may not be a new node.  It
    may be a node that existed before and node_count has a value, setting it
    to 0 will cause a memory leak.  At this time, mas->alloc->total will be
    greater than the actual number of nodes in the linked list, which may
    cause many other errors.  For example, out-of-bounds access in
    mas_pop_node(), and mas_pop_node() may return addresses that should not be
    used.  Fix it by initializing node_count only for new nodes.

    Also, by the way, an if-else statement was removed to simplify the code.

    Link: https://lkml.kernel.org/r/20230411041005.26205-1-zhangpeng.00@bytedance.com
    Fixes: 54a611b60590 ("Maple Tree: add new data structure")
    Signed-off-by: Peng Zhang <zhangpeng.00@bytedance.com>
    Reviewed-by: Liam R. Howlett <Liam.Howlett@oracle.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3bbdd9e29fe83f945870c9e49392a3c80808da8e
Author: Liam R. Howlett <Liam.Howlett@oracle.com>
Date:   Fri Apr 14 10:57:27 2023 -0400

    maple_tree: fix mas_empty_area() search

    commit 06e8fd999334bcd76b4d72d7b9206d4aea89764e upstream.

    The internal function of mas_awalk() was incorrectly skipping the last
    entry in a node, which could potentially be NULL.  This is only a problem
    for the left-most node in the tree - otherwise that NULL would not exist.

    Fix mas_awalk() by using the metadata to obtain the end of the node for
    the loop and the logical pivot as apposed to the raw pivot value.

    Link: https://lkml.kernel.org/r/20230414145728.4067069-2-Liam.Howlett@oracle.com
    Fixes: 54a611b60590 ("Maple Tree: add new data structure")
    Signed-off-by: Liam R. Howlett <Liam.Howlett@oracle.com>
    Reported-by: Rick Edgecombe <rick.p.edgecombe@intel.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bd01732a17f60236ef00b9a877eb1bb73467c724
Author: Liam R. Howlett <Liam.Howlett@oracle.com>
Date:   Fri Apr 14 10:57:26 2023 -0400

    maple_tree: make maple state reusable after mas_empty_area_rev()

    commit fad8e4291da5e3243e086622df63cb952db444d8 upstream.

    Stop using maple state min/max for the range by passing through pointers
    for those values.  This will allow the maple state to be reused without
    resetting.
```

```
    Also add some logic to fail out early on searching with invalid
    arguments.

    Link: https://lkml.kernel.org/r/20230414145728.4067069-1-Liam.Howlett@oracle.com
    Fixes: 54a611b60590 ("Maple Tree: add new data structure")
    Signed-off-by: Liam R. Howlett <Liam.Howlett@oracle.com>
    Reported-by: Rick Edgecombe <rick.p.edgecombe@intel.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a839cd33ae23b97c8b649599f236b222679f980d
Author: Toke Høiland-Jørgensen <toke@toke.dk>
Date:   Thu Apr 13 23:41:18 2023 +0200

    wifi: ath9k: Don't mark channelmap stack variable read-only in
ath9k_mci_update_wlan_channels()

    commit 0f2a4af27b649c13ba76431552fe49c60120d0f6 upstream.

    This partially reverts commit e161d4b60ae3a5356e07202e0bfedb5fad82c6aa.

    Turns out the channelmap variable is not actually read-only, it's modified
    through the MCI_GPM_CLR_CHANNEL_BIT() macro further down in the function,
    so making it read-only causes page faults when that code is hit.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=217183
    Link: https://lore.kernel.org/r/20230413214118.153781-1-toke@toke.dk
    Fixes: e161d4b60ae3 ("wifi: ath9k: Make arrays prof_prio and channelmap static const")
    Cc: stable@vger.kernel.org
    Signed-off-by: Toke Høiland-Jørgensen <toke@toke.dk>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 60864decd6a55733f2cb7b951ffb481e4e90d51c
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Tue Apr 18 19:38:58 2023 +0800

    LoongArch: Mark 3 symbol exports as non-GPL

    commit dce5ea1d0f45fa612f5760b88614a3f32bc75e3f upstream.

    vm_map_base, empty_zero_page and invalid_pmd_table could be accessed
    widely by some out-of-tree non-GPL but important file systems or drivers
    (e.g. OpenZFS). Let's use EXPORT_SYMBOL() instead of EXPORT_SYMBOL_GPL()
    to export them, so as to avoid build errors.

    1, Details about vm_map_base:

    This is a LoongArch-specific symbol and may be referenced through macros
    PCI_IOBASE, VMALLOC_START and VMALLOC_END.

    2, Details about empty_zero_page:

    As it stands today, only 3 architectures export empty_zero_page as a GPL
    symbol: IA64, LoongArch and MIPS. LoongArch gets the GPL export by
    inheriting from MIPS, and the MIPS export was first introduced in commit
    497d2adcbf50b ("[MIPS] Export empty_zero_page for sake of the ext4
    module."). The IA64 export was similar: commit a7d57ecf4216e ("[IA64]
    Export three symbols for module use") did so for kvm.

    In both IA64 and MIPS, the export of empty_zero_page was done for
    satisfying some in-kernel component built as module (kvm and ext4
    respectively), and given its reasonably low-level nature, GPL is a
    reasonable choice. But looking at the bigger picture it is evident most
```

```
    other architectures do not regard it as GPL, so in effect the symbol
    probably should not be treated as such, in favor of consistency.

    3, Details about invalid_pmd_table:

    Keep consistency with invalid_pte_table and make it be possible by some
    modules.

    Cc: stable@vger.kernel.org
    Reviewed-by: WANG Xuerui <git@xen0n.name>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 737001c15c888925fc6ba784a5ed9ac04bf92514
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Tue Apr 18 19:38:58 2023 +0800

    LoongArch: Fix probing of the CRC32 feature

    commit df830336045db1246d3245d3737fee9939c5f731 upstream.

    Not all LoongArch processors support CRC32 instructions. This feature
    is indicated by CPUCFG1.CRC32 (Bit25) but it is wrongly defined in the
    previous versions of the ISA manual (and so does in loongarch.h). The
    CRC32 feature is set unconditionally now, so fix it.

    BTW, expose the CRC32 feature in /proc/cpuinfo.

    Cc: stable@vger.kernel.org
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f6711d9a35e77754ce80e6095c4055c00443b7e2
Author: Tiezhu Yang <yangtiezhu@loongson.cn>
Date:   Wed Apr 19 12:07:27 2023 +0800

    LoongArch: Check unwind_error() in arch_stack_walk()

    commit 370a3b8f58743eceb97c5256538d6048c26d2d03 upstream.

    We can see the following messages with CONFIG_PROVE_LOCKING=y on
    LoongArch:

      BUG: MAX_STACK_TRACE_ENTRIES too low!
      turning off the locking correctness validator.

    This is because stack_trace_save() returns a big value after call
    arch_stack_walk(), here is the call trace:

      save_trace()
        stack_trace_save()
          arch_stack_walk()
            stack_trace_consume_entry()

    arch_stack_walk() should return immediately if unwind_next_frame()
    failed, no need to do the useless loops to increase the value of c->len
    in stack_trace_consume_entry(), then we can fix the above problem.

    Cc: stable@vger.kernel.org
    Reported-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/all/8a44ad71-68d2-4926-892f-72bfc7a67e2a@roeck-us.net/
    Signed-off-by: Tiezhu Yang <yangtiezhu@loongson.cn>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 1ec1e1dcb2d84e2497c1800765809ce2223edb80
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Tue Apr 18 19:38:58 2023 +0800

    LoongArch: module: set section addresses to 0x0

    commit 93eb1215ed794a18ba8753e0654f069d58838966 upstream.

    These got*, plt* and .text.ftrace_trampoline sections specified for
    LoongArch have non-zero addressses. Non-zero section addresses in a
    relocatable ELF would confuse GDB when it tries to compute the section
    offsets and it ends up printing wrong symbol addresses. Therefore, set
    them to zero, which mirrors the change in commit 5d8591bc0fbaeb6ded
    ("arm64 module: set plt* section addresses to 0x0").

    Cc: stable@vger.kernel.org
    Reviewed-by: Guo Ren <guoren@kernel.org>
    Signed-off-by: Chong Qiao <qiaochong@loongson.cn>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2aac424cb232e7a4a74fce22cf0be3fe5a7659a5
Author: David Gow <davidgow@google.com>
Date:   Wed Feb 15 06:47:35 2023 +0800

    rust: kernel: Mark rust_fmt_argument as extern "C"

    commit c682e4c37d2b8ba3bde1125cbbea4ee88824b4e2 upstream.

    The rust_fmt_argument function is called from printk() to handle the %pA
    format specifier.

    Since it's called from C, we should mark it extern "C" to make sure it's
    ABI compatible.

    Cc: stable@vger.kernel.org
    Fixes: 247b365dc8dc ("rust: add `kernel` crate")
    Signed-off-by: David Gow <davidgow@google.com>
    Reviewed-by: Gary Guo <gary@garyguo.net>
    Reviewed-by: Björn Roy Baron <bjorn3_gh@protonmail.com>
    Reviewed-by: Vincenzo Palazzo <vincenzopalazzodev@gmail.com>
    [Applied `rustfmt`]
    Signed-off-by: Miguel Ojeda <ojeda@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 199384f3dabbd069adbe64ffec670e205a096212
Author: Boris Burkov <boris@bur.io>
Date:   Wed Apr 5 12:43:59 2023 -0700

    btrfs: reinterpret async discard iops_limit=0 as no delay

    commit ef9cddfe57d86aac6b509b550136395669159b30 upstream.

    Currently, a limit of 0 results in a hard coded metering over 6 hours.
    Since the default is a set limit, I suspect no one truly depends on this
    rather arbitrary setting. Repurpose it for an arguably more useful
    "unlimited" mode, where the delay is 0.

    Note that if block groups are too new, or go fully empty, there is still
    a delay associated with those conditions. Those delays implement
    heuristics for not trimming a region we are relatively likely to fully
    overwrite soon.

    CC: stable@vger.kernel.org # 6.2+
    Reviewed-by: Neal Gompa <neal@gompa.dev>
```

```
    Signed-off-by: Boris Burkov <boris@bur.io>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9427fb1931adc47929f209520f120af22507fa69
Author: Boris Burkov <boris@bur.io>
Date:   Wed Apr 5 12:43:58 2023 -0700

    btrfs: set default discard iops_limit to 1000

    commit e9f59429b87d35cf23ae9ca19629bd686a1c0304 upstream.

    Previously, the default was a relatively conservative 10. This results
    in a 100ms delay, so with ~300 discards in a commit, it takes the full
    30s till the next commit to finish the discards. On a workstation, this
    results in the disk never going idle, wasting power/battery, etc.

    Set the default to 1000, which results in using the smallest possible
    delay, currently, which is 1ms. This has shown to not pathologically
    keep the disk busy by the original reporter.

    Link: https://lore.kernel.org/linux-btrfs/Y%2F+n1wS%2F4XAH7X1p@nz/
    Link: https://bugzilla.redhat.com/show_bug.cgi?id=2182228
    CC: stable@vger.kernel.org # 6.2+
    Reviewed-by: Neal Gompa <neal@gompa.dev
    Signed-off-by: Boris Burkov <boris@bur.io>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 521e9f4c9820cbcdd795486c07d2102c9ce42bf9
Author: Andy Chi <andy.chi@canonical.com>
Date:   Thu Apr 20 11:59:41 2023 +0800

    ALSA: hda/realtek: fix mute/micmute LEDs for a HP ProBook

    commit 2ae147d643d326f74d93ba4f72a405f25f2677ea upstream.

    There is a HP ProBook 455 G10 which using ALC236 codec and need the
    ALC236_FIXUP_HP_MUTE_LED_MICMUTE_VREF quirk to make mute LED and
    micmute LED work.

    Signed-off-by: Andy Chi <andy.chi@canonical.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20230420035942.66817-1-andy.chi@canonical.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 986739dd1d2d479fc104ed6f9ec449e31668a81c
Author: Brian Masney <bmasney@redhat.com>
Date:   Mon Apr 3 21:14:55 2023 -0400

    iio: light: tsl2772: fix reading proximity-diodes from device tree

    commit b1cb00d51e361cf5af93649917d9790e1623647e upstream.

    tsl2772_read_prox_diodes() will correctly parse the properties from
    device tree to determine which proximity diode(s) to read from, however
    it didn't actually set this value on the struct tsl2772_settings. Let's
    go ahead and fix that.

    Reported-by: Tom Rix <trix@redhat.com>
    Link: https://lore.kernel.org/lkml/20230327120823.1369700-1-trix@redhat.com/
    Fixes: 94cd1113aaa0 ("iio: tsl2772: add support for reading proximity led settings from
```

```
device tree")
    Signed-off-by: Brian Masney <bmasney@redhat.com>
    Link: https://lore.kernel.org/r/20230404011455.339454-1-bmasney@redhat.com
    Cc: <Stable@vger.kernel.org>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d6eb21f361deb835c4a943d7a69bd56e2230534
Author: Liang He <windhl@126.com>
Date:   Wed Mar 22 11:56:27 2023 +0800

    iio: dac: ad5755: Add missing fwnode_handle_put()

    commit ffef73791574b8da872cfbf881d8e3e9955fc130 upstream.

    In ad5755_parse_fw(), we should add fwnode_handle_put()
    when break out of the iteration device_for_each_child_node()
    as it will automatically increase and decrease the refcounter.

    Fixes: 3ac27afefd5d ("iio:dac:ad5755: Switch to generic firmware properties and drop
pdata")
    Signed-off-by: Liang He <windhl@126.com>
    Link: https://lore.kernel.org/r/20230322035627.1856421-1-windhl@126.com
    Cc: <Stable@vger.kernel.org>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fea7ae49cce10ee3eb64611780597f9517d2c22b
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Fri Apr 21 13:39:10 2023 -0700

    Revert "ACPICA: Events: Support fixed PCIe wake event"

    commit 8e41e0a575664d26bb87e012c39435c4c3914ed9 upstream.

    This reverts commit 5c62d5aab8752e5ee7bfbe75ed6060db1c787f98.

    This broke wake-on-lan for multiple people, and for much too long.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=217069
    Link: https://lore.kernel.org/all/754225a2-95a9-2c36-1886-7da1a78308c2@loongson.cn/
    Link: https://github.com/acpica/acpica/pull/866
    Cc: Rafael J. Wysocki <rafael@kernel.org>
    Cc: Jianmin Lv <lvjianmin@loongson.cn>
    Cc: Huacai Chen <chenhuacai@loongson.cn>
    Cc: Bob Moore <robert.moore@intel.com>
    Cc: stable@kernel.org # 6.2
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db468d5d3231988c68dfe7d2f904a10142b50d2a
Author: Peter Xu <peterx@redhat.com>
Date:   Wed Apr 12 12:38:52 2023 -0400

    Revert "userfaultfd: don't fail on unrecognized features"

    commit 2ff559f31a5d50c31a3f9d849f8af90dc36c7105 upstream.

    This is a proposal to revert commit 914eedcb9ba0ff53c33808.

    I found this when writing a simple UFFDIO_API test to be the first unit
    test in this set.  Two things breaks with the commit:

      - UFFDIO_API check was lost and missing.  According to man page, the
      kernel should reject ioctl(UFFDIO_API) if uffdio_api.api != 0xaa.  This
```

```
        check is needed if the api version will be extended in the future, or
        user app won't be able to identify which is a new kernel.

        - Feature flags checks were removed, which means UFFDIO_API with a
        feature that does not exist will also succeed.  According to the man
        page, we should (and it makes sense) to reject ioctl(UFFDIO_API) if
        unknown features passed in.

    Link: https://lore.kernel.org/r/20220722201513.1624158-1-axelrasmussen@google.com
    Link: https://lkml.kernel.org/r/20230412163922.327282-2-peterx@redhat.com
    Fixes: 914eedcb9ba0 ("userfaultfd: don't fail on unrecognized features")
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Acked-by: David Hildenbrand <david@redhat.com>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: Dmitry Safonov <0x7f454c46@gmail.com>
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Mike Rapoport (IBM) <rppt@kernel.org>
    Cc: Zach O'Keefe <zokeefe@google.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 747160941bdaa9935c12d9dc9e3ea81c701f1337
Author: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
Date:   Wed Mar 22 22:45:45 2023 +0100

    pwm: Zero-initialize the pwm_state passed to driver's .get_state()

    commit 1271a7b98e7989ba6bb978e14403fc84efe16e13 upstream.

    This is just to ensure that .usage_power is properly initialized and
    doesn't contain random stack data. The other members of struct pwm_state
    should get a value assigned in a successful call to .get_state(). So in
    the absence of bugs in driver implementations, this is only a safe-guard
    and no fix.

    Reported-by: Munehisa Kamata <kamatam@amazon.com>
    Link: https://lore.kernel.org/r/20230310214004.2619480-1-u.kleine-koenig@pengutronix.de
    Signed-off-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b1524d5e4bf09fb9a5807a9970627a1706fa19ef
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Feb 8 17:02:30 2023 +0100

    mtd: spi-nor: fix memory leak when using debugfs_lookup()

    [ Upstream commit ec738ca127d07ecac6afae36e2880341ec89150e ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To solve this, remove the
    lookup and create the directory on the first device found, and then
    remove it when the module is unloaded.

    Cc: Tudor Ambarus <tudor.ambarus@microchip.com>
    Cc: Pratyush Yadav <pratyush@kernel.org>
    Cc: Miquel Raynal <miquel.raynal@bootlin.com>
    Cc: Richard Weinberger <richard@nod.at>
    Cc: Vignesh Raghavendra <vigneshr@ti.com>
    Cc: linux-mtd@lists.infradead.org
    Reviewed-by: Michael Walle <michael@walle.cc>
    Link: https://lore.kernel.org/r/20230208160230.2179905-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit e48520be1bf46b3b6f57749a1028a4a5a35a8414
Author: weiliang1503 <weiliang1503@gmail.com>
Date:   Thu Mar 30 19:49:43 2023 +0800

    platform/x86: asus-nb-wmi: Add quirk_asus_tablet_mode to other ROG Flow X13 models

    [ Upstream commit e352d685fde427a8fc9beb2ba30888f5d6f2e5e6 ]

    Make quirk_asus_tablet_mode apply on other ROG Flow X13 devices,
    which only affects the GV301Q model before.

    Signed-off-by: weiliang1503 <weiliang1503@gmail.com>
    Link: https://lore.kernel.org/r/20230330114943.15057-1-weiliang1503@gmail.com
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e0bdd0d65790d84b18f2bd70bf07e5145b5e704c
Author: Hans de Goede <hdegoede@redhat.com>
Date:   Fri Mar 31 19:31:48 2023 +0200

    platform/x86: gigabyte-wmi: add support for X570S AORUS ELITE

    [ Upstream commit 52f91e51944808d83dfe2d5582601b5e84e472cc ]

    Add "X570S AORUS ELITE" to known working boards

    Reported-by: Brandon Nielsen <nielsenb@jetfuse.net>
    Link: https://lore.kernel.org/r/20230331014902.7864-1-nielsenb@jetfuse.net
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 537b809911500234af2728aaff40ffb25d062b55
Author: Juergen Gross <jgross@suse.com>
Date:   Wed Mar 29 10:02:59 2023 +0200

    xen/netback: use same error messages for same errors

    [ Upstream commit 2eca98e5b24d01c02b46c67be05a5f98cc9789b1 ]

    Issue the same error message in case an illegal page boundary crossing
    has been detected in both cases where this is tested.

    Suggested-by: Jan Beulich <jbeulich@suse.com>
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Reviewed-by: Jan Beulich <jbeulich@suse.com>
    Link: https://lore.kernel.org/r/20230329080259.14823-1-jgross@suse.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6220f6ece989f512f5544d46a42471c65ff19ca2
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Mon Mar 20 15:33:34 2023 +0200

    nvme-tcp: fix a possible UAF when failing to allocate an io queue

    [ Upstream commit 88eaba80328b31ef81813a1207b4056efd7006a6 ]

    When we allocate a nvme-tcp queue, we set the data_ready callback before
    we actually need to use it. This creates the potential that if a stray
    controller sends us data on the socket before we connect, we can trigger
    the io_work and start consuming the socket.

    In this case reported: we failed to allocate one of the io queues, and
```

```
    as we start releasing the queues that we already allocated, we get
    a UAF [1] from the io_work which is running before it should really.

    Fix this by setting the socket ops callbacks only before we start the
    queue, so that we can't accidentally schedule the io_work in the
    initialization phase before the queue started. While we are at it,
    rename nvme_tcp_restore_sock_calls to pair with nvme_tcp_setup_sock_ops.

    [1]:
    [16802.107284] nvme nvme4: starting error recovery
    [16802.109166] nvme nvme4: Reconnecting in 10 seconds...
    [16812.173535] nvme nvme4: failed to connect socket: -111
    [16812.173745] nvme nvme4: Failed reconnect attempt 1
    [16812.173747] nvme nvme4: Reconnecting in 10 seconds...
    [16822.413555] nvme nvme4: failed to connect socket: -111
    [16822.413762] nvme nvme4: Failed reconnect attempt 2
    [16822.413765] nvme nvme4: Reconnecting in 10 seconds...
    [16832.661274] nvme nvme4: creating 32 I/O queues.
    [16833.919887] BUG: kernel NULL pointer dereference, address: 0000000000000088
    [16833.920068] nvme nvme4: Failed reconnect attempt 3
    [16833.920094] #PF: supervisor write access in kernel mode
    [16833.920261] nvme nvme4: Reconnecting in 10 seconds...
    [16833.920368] #PF: error_code(0x0002) - not-present page
    [16833.921086] Workqueue: nvme_tcp_wq nvme_tcp_io_work [nvme_tcp]
    [16833.921191] RIP: 0010:_raw_spin_lock_bh+0x17/0x30
    ...
    [16833.923138] Call Trace:
    [16833.923271]  <TASK>
    [16833.923402]  lock_sock_nested+0x1e/0x50
    [16833.923545]  nvme_tcp_try_recv+0x40/0xa0 [nvme_tcp]
    [16833.923685]  nvme_tcp_io_work+0x68/0xa0 [nvme_tcp]
    [16833.923824]  process_one_work+0x1e8/0x390
    [16833.923969]  worker_thread+0x53/0x3d0
    [16833.924104]  ? process_one_work+0x390/0x390
    [16833.924240]  kthread+0x124/0x150
    [16833.924376]  ? set_kthread_struct+0x50/0x50
    [16833.924518]  ret_from_fork+0x1f/0x30
    [16833.924655]  </TASK>

    Reported-by: Yanjun Zhang <zhangyanjun@cestc.cn>
    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Tested-by: Yanjun Zhang <zhangyanjun@cestc.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4218f6383cb4a6106caeee3983c1cec62a1d13fe
Author: David Gow <davidgow@google.com>
Date:   Wed Mar 29 14:55:34 2023 +0800

    drm: test: Fix 32-bit issue in drm_buddy_test

    [ Upstream commit 25bbe844ef5c4fb4d7d8dcaa0080f922b7cd3a16 ]

    The drm_buddy_test KUnit tests verify that returned blocks have sizes
    which are powers of two using is_power_of_2(). However, is_power_of_2()
    operations on a 'long', but the block size is a u64. So on systems where
    long is 32-bit, this can sometimes fail even on correctly sized blocks.

    This only reproduces randomly, as the parameters passed to the buddy
    allocator in this test are random. The seed 0xb2e06022 reproduced it
    fine here.

    For now, just hardcode an is_power_of_2() implementation using
    x & (x - 1).
```

```
    Signed-off-by: David Gow <davidgow@google.com>
    Acked-by: Christian König <christian.koenig@amd.com>
    Reviewed-by: Maíra Canal <mcanal@igalia.com>
    Reviewed-by: Arunpravin Paneer Selvam <arunpravin.paneerselvam@amd.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230329065532.2122295-2-
davidgow@google.com
    Signed-off-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f9207ced710201e96920af54eb8be9f17767409d
Author: David Gow <davidgow@google.com>
Date:   Wed Mar 29 14:55:32 2023 +0800

    drm: buddy_allocator: Fix buddy allocator init on 32-bit systems

    [ Upstream commit 4453545b5b4c3eff941f69a5530f916d899db025 ]

    The drm buddy allocator tests were broken on 32-bit systems, as
    rounddown_pow_of_two() takes a long, and the buddy allocator handles
    64-bit sizes even on 32-bit systems.

    This can be reproduced with the drm_buddy_allocator KUnit tests on i386:
            ./tools/testing/kunit/kunit.py run --arch i386 \
            --kunitconfig ./drivers/gpu/drm/tests drm_buddy

    (It results in kernel BUG_ON() when too many blocks are created, due to
    the block size being too small.)

    This was independently uncovered (and fixed) by Luís Mendes, whose patch
    added a new u64 variant of rounddown_pow_of_two(). This version instead
    recalculates the size based on the order.

    Reported-by: Luís Mendes <luis.p.mendes@gmail.com>
    Link: https://lore.kernel.org/lkml/CAEzXK1oghXAB_KpKpm=-
CviDQbNaH0qfgYTSSjZgvvyj4U78AA@mail.gmail.com/T/
    Signed-off-by: David Gow <davidgow@google.com>
    Acked-by: Christian König <christian.koenig@amd.com>
    Reviewed-by: Arunpravin Paneer Selvam <arunpravin.paneerselvam@amd.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230329065532.2122295-1-
davidgow@google.com
    Signed-off-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7c5aec9cbf9fb9d6b33401cbff0703c0a54dce3a
Author: Heiko Carstens <hca@linux.ibm.com>
Date:   Mon Mar 6 12:31:30 2023 +0100

    s390/ptrace: fix PTRACE_GET_LAST_BREAK error handling

    [ Upstream commit f9bbf25e7b2b74b52b2f269216a92657774f239c ]

    Return -EFAULT if put_user() for the PTRACE_GET_LAST_BREAK
    request fails, instead of silently ignoring it.

    Reviewed-by: Sven Schnelle <svens@linux.ibm.com>
    Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b7498da05a024fb1081a8c9deea47ee106705f4c
Author: Thomas Weißschuh <linux@weissschuh.net>
Date:   Mon Mar 27 13:05:02 2023 +0000

    platform/x86: gigabyte-wmi: add support for B650 AORUS ELITE AX
```

```
    [ Upstream commit 441d901fbf669f6360566a4437b1e563b854de4a ]

    This has been reported as working.

    Suggested-by: got3nks <got3nks@users.noreply.github.com>
    Link: https://github.com/t-8ch/linux-gigabyte-wmi-driver/issues/15#issuecomment-
1483942966
    Signed-off-by: Thomas Weißschuh <linux@weissschuh.net>
    Link: https://lore.kernel.org/r/20230327-gigabyte-wmi-b650-elite-ax-v1-1-
d4d645c21d0b@weissschuh.net
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cda8141cf896ae8aecfea548f4714f20de79bccc
Author: Álvaro Fernández Rojas <noltari@gmail.com>
Date:   Thu Mar 23 20:48:41 2023 +0100

    net: dsa: b53: mmap: add phy ops

    [ Upstream commit 45977e58ce65ed0459edc9a0466d9dfea09463f5 ]

    Implement phy_read16() and phy_write16() ops for B53 MMAP to avoid accessing
    B53_PORT_MII_PAGE registers which hangs the device.
    This access should be done through the MDIO Mux bus controller.

    Signed-off-by: Álvaro Fernández Rojas <noltari@gmail.com>
    Acked-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f4916c60ba4b027cc8fe88abd149d96448338a7d
Author: Damien Le Moal <damien.lemoal@opensource.wdc.com>
Date:   Wed Mar 22 11:22:11 2023 +0900

    scsi: core: Improve scsi_vpd_inquiry() checks

    [ Upstream commit f0aa59a33d2ac2267d260fe21eaf92500df8e7b4 ]

    Some USB-SATA adapters have broken behavior when an unsupported VPD page is
    probed: Depending on the VPD page number, a 4-byte header with a valid VPD
    page number but with a 0 length is returned. Currently, scsi_vpd_inquiry()
    only checks that the page number is valid to determine if the page is
    valid, which results in receiving only the 4-byte header for the
    non-existent page. This error manifests itself very often with page 0xb9
    for the Concurrent Positioning Ranges detection done by sd_read_cpr(),
    resulting in the following error message:

    sd 0:0:0:0: [sda] Invalid Concurrent Positioning Ranges VPD page

    Prevent such misleading error message by adding a check in
    scsi_vpd_inquiry() to verify that the page length is not 0.

    Signed-off-by: Damien Le Moal <damien.lemoal@opensource.wdc.com>
    Link: https://lore.kernel.org/r/20230322022211.116327-1-damien.lemoal@opensource.wdc.com
    Reviewed-by: Benjamin Block <bblock@linux.ibm.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bbb3519a2fda4090d8af37320687591b17b946e8
Author: Tomas Henzl <thenzl@redhat.com>
Date:   Fri Mar 24 14:52:49 2023 +0100

    scsi: megaraid_sas: Fix fw_crash_buffer_show()
```

```
    [ Upstream commit 0808ed6ebbc292222ca069d339744870f6d801da ]

    If crash_dump_buf is not allocated then crash dump can't be available.
    Replace logical 'and' with 'or'.

    Signed-off-by: Tomas Henzl <thenzl@redhat.com>
    Link: https://lore.kernel.org/r/20230324135249.9733-1-thenzl@redhat.com
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5d0f53e6484e7aefcffbbfe6e309869331ea8076
Author: Nick Desaulniers <ndesaulniers@google.com>
Date:   Wed Mar 8 11:59:33 2023 -0800

    selftests: sigaltstack: fix -Wuninitialized

    [ Upstream commit 05107edc910135d27fe557267dc45be9630bf3dd ]

    Building sigaltstack with clang via:
    $ ARCH=x86 make LLVM=1 -C tools/testing/selftests/sigaltstack/

    produces the following warning:
      warning: variable 'sp' is uninitialized when used here [-Wuninitialized]
      if (sp < (unsigned long)sstack ||
          ^~

    Clang expects these to be declared at global scope; we've fixed this in
    the kernel proper by using the macro `current_stack_pointer`. This is
    defined in different headers for different target architectures, so just
    create a new header that defines the arch-specific register names for
    the stack pointer register, and define it for more targets (at least the
    ones that support current_stack_pointer/ARCH_HAS_CURRENT_STACK_POINTER).

    Reported-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Link:
https://lore.kernel.org/lkml/CA+G9fYsi3OOu7yCsMutpzKDnBMAzJBCPimBp86LhGBa0eCnEpA@mail.gmail.
com/
    Signed-off-by: Nick Desaulniers <ndesaulniers@google.com>
    Reviewed-by: Kees Cook <keescook@chromium.org>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Anders Roxell <anders.roxell@linaro.org>
    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f32b8513feffc92b4a04b30e0f83b1b3bd52c564
Author: Frank Crawford <frank@crawford.emu.id.au>
Date:   Sat Mar 18 20:14:41 2023 +1100

    platform/x86 (gigabyte-wmi): Add support for A320M-S2H V2

    [ Upstream commit b7c994f8c35e916e27c60803bb21457bc1373500 ]

    Add support for A320M-S2H V2.  Tested using module force_load option.

    Signed-off-by: Frank Crawford <frank@crawford.emu.id.au>
    Acked-by: Thomas Weißschuh <linux@weissschuh.net>
    Link: https://lore.kernel.org/r/20230318091441.1240921-1-frank@crawford.emu.id.au
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fbc0d7ab19128a10f2feb907e9e6fb6dd7bfc003
Author: Dongliang Mu <dzm91@hust.edu.cn>
Date:   Thu Mar 9 12:01:07 2023 +0800
```

```
    platform/x86/intel: vsec: Fix a memory leak in intel_vsec_add_aux

    [ Upstream commit da0ba0ccce54059d6c6b788a75099bfce95126da ]

    The first error handling code in intel_vsec_add_aux misses the
    deallocation of intel_vsec_dev->resource.

    Fix this by adding kfree(intel_vsec_dev->resource) in the error handling
    code.

    Reviewed-by: David E. Box <david.e.box@linux.intel.com>
    Signed-off-by: Dongliang Mu <dzm91@hust.edu.cn>
    Link: https://lore.kernel.org/r/20230309040107.534716-4-dzm91@hust.edu.cn
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f3e4b696b3b076ff560a9b3a215c52d40bad73c0
Author: Douglas Raillard <douglas.raillard@arm.com>
Date:   Mon Mar 6 12:25:49 2023 +0000

    f2fs: Fix f2fs_truncate_partial_nodes ftrace event

    [ Upstream commit 0b04d4c0542e8573a837b1d81b94209e48723b25 ]

    Fix the nid_t field so that its size is correctly reported in the text
    format embedded in trace.dat files. As it stands, it is reported as
    being of size 4:

            field:nid_t nid[3];     offset:24;      size:4; signed:0;

    Instead of 12:

            field:nid_t nid[3];     offset:24;      size:12;        signed:0;

    This also fixes the reported offset of subsequent fields so that they
    match with the actual struct layout.

    Signed-off-by: Douglas Raillard <douglas.raillard@arm.com>
    Reviewed-by: Mukesh Ojha <quic_mojha@quicinc.com>
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e06b8d11d5c9e08e41bdfa4a905ab883778591ea
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Tue Apr 18 18:59:02 2023 +0300

    net: bridge: switchdev: don't notify FDB entries with "master dynamic"

    [ Upstream commit 927cdea5d2095287ddd5246e5aa68eb5d68db2be ]

    There is a structural problem in switchdev, where the flag bits in
    struct switchdev_notifier_fdb_info (added_by_user, is_local etc) only
    represent a simplified / denatured view of what's in struct
    net_bridge_fdb_entry :: flags (BR_FDB_ADDED_BY_USER, BR_FDB_LOCAL etc).
    Each time we want to pass more information about struct
    net_bridge_fdb_entry :: flags to struct switchdev_notifier_fdb_info
    (here, BR_FDB_STATIC), we find that FDB entries were already notified to
    switchdev with no regard to this flag, and thus, switchdev drivers had
    no indication whether the notified entries were static or not.

    For example, this command:

    ip link add br0 type bridge && ip link set swp0 master br0
```

```
    bridge fdb add dev swp0 00:01:02:03:04:05 master dynamic

    has never worked as intended with switchdev. It causes a struct
    net_bridge_fdb_entry to be passed to br_switchdev_fdb_notify() which has
    a single flag set: BR_FDB_ADDED_BY_USER.

    This is further passed to the switchdev notifier chain, where interested
    drivers have no choice but to assume this is a static (does not age) and
    sticky (does not migrate) FDB entry. So currently, all drivers offload
    it to hardware as such, as can be seen below ("offload" is set).

    bridge fdb get 00:01:02:03:04:05 dev swp0 master
    00:01:02:03:04:05 dev swp0 offload master br0

    The software FDB entry expires $ageing_time centiseconds after the
    kernel last sees a packet with this MAC SA, and the bridge notifies its
    deletion as well, so it eventually disappears from hardware too.

    This is a problem, because it is actually desirable to start offloading
    "master dynamic" FDB entries correctly - they should expire $ageing_time
    centiseconds after the *hardware* port last sees a packet with this
    MAC SA - and this is how the current incorrect behavior was discovered.
    With an offloaded data plane, it can be expected that software only sees
    exception path packets, so an otherwise active dynamic FDB entry would
    be aged out by software sooner than it should.

    With the change in place, these FDB entries are no longer offloaded:

    bridge fdb get 00:01:02:03:04:05 dev swp0 master
    00:01:02:03:04:05 dev swp0 master br0

    and this also constitutes a better way (assuming a backport to stable
    kernels) for user space to determine whether the kernel has the
    capability of doing something sane with these or not.

    As opposed to "master dynamic" FDB entries, on the current behavior of
    which no one currently depends on (which can be deduced from the lack of
    kselftests), Ido Schimmel explains that entries with the "extern_learn"
    flag (BR_FDB_ADDED_BY_EXT_LEARN) should still be notified to switchdev,
    since the spectrum driver listens to them (and this is kind of okay,
    because although they are treated identically to "static", they are
    expected to not age, and to roam).

    Fixes: 6b26b51b1d13 ("net: bridge: Add support for notifying devices about FDB add/del")
    Link: https://lore.kernel.org/netdev/20230327115206.jk5q5l753aoelwus@skbuf/
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Reviewed-by: Jesse Brandeburg <jesse.brandeburg@intel.com>
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Tested-by: Ido Schimmel <idosch@nvidia.com>
    Link: https://lore.kernel.org/r/20230418155902.898627-1-vladimir.oltean@nxp.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eaabfa79e58e07c36964726eaed2b671dc4e8e1a
Author: Sebastian Basierski <sebastianx.basierski@intel.com>
Date:   Mon Apr 17 13:53:45 2023 -0700

    e1000e: Disable TSO on i219-LM card to increase speed

    [ Upstream commit 67d47b95119ad589b0a0b16b88b1dd9a04061ced ]

    While using i219-LM card currently it was only possible to achieve
    about 60% of maximum speed due to regression introduced in Linux 5.8.
    This was caused by TSO not being disabled by default despite commit
    f29801030ac6 ("e1000e: Disable TSO for buffer overrun workaround").
```

```
    Fix that by disabling TSO during driver probe.

    Fixes: f29801030ac6 ("e1000e: Disable TSO for buffer overrun workaround")
    Signed-off-by: Sebastian Basierski <sebastianx.basierski@intel.com>
    Signed-off-by: Mateusz Palczewski <mateusz.palczewski@intel.com>
    Tested-by: Naama Meir <naamax.meir@linux.intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Link: https://lore.kernel.org/r/20230417205345.1030801-1-anthony.l.nguyen@intel.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5809741596170763881f7f82bbb38a1dd6ad8f47
Author: Vadim Fedorenko <vadfed@meta.com>
Date:   Tue Apr 18 13:25:11 2023 -0700

    bnxt_en: fix free-runnig PHC mode

    [ Upstream commit 8c154d272c3e03b100baaf1df473f22a78fa403e ]

    The patch in fixes changed the way real-time mode is chosen for PHC on
    the NIC. Apparently there is one more use case of the check outside of
    ptp part of the driver which was not converted to the new macro and is
    making a lot of noise in free-running mode.

    Fixes: 131db4991622 ("bnxt_en: reset PHC frequency in free-running mode")
    Signed-off-by: Vadim Fedorenko <vadfed@meta.com>
    Reviewed-by: Michael Chan <michael.chan@broadcom.com>
    Reviewed-by: Pavan Chebbi <pavan.chebbi@broadcom.com>
    Link: https://lore.kernel.org/r/20230418202511.1544735-1-vadfed@meta.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c3d81540dff971b21b4bbe229deae926d6463f78
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Mon Apr 17 20:19:33 2023 +0200

    net: dsa: microchip: ksz8795: Correctly handle huge frame configuration

    [ Upstream commit 3d2f8f1f184c60508f7af3022536651d7ac2dd07 ]

    Because of the logic in place, SW_HUGE_PACKET can never be set.
    (If the first condition is true, then the 2nd one is also true, but is not
    executed)

    Change the logic and update each bit individually.

    Fixes: 29d1e85f45e0 ("net: dsa: microchip: ksz8: add MTU configuration support")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Reviewed-by: Oleksij Rempel <o.rempel@pengutronix.de>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Reviewed-by: Vladimir Oltean <olteanv@gmail.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Link:
https://lore.kernel.org/r/43107d9e8b5b8b05f0cbd4e1f47a2bb88c8747b2.1681755535.git.christophe
.jaillet@wanadoo.fr
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 71035a0508c04827b91a5bfeb2c9ef374f321e65
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Tue Apr 11 15:24:13 2023 +0000

    bpf: Fix incorrect verifier pruning due to missing register precision taints
```

```
    [ Upstream commit 71b547f561247897a0a14f3082730156c0533fed ]

    Juan Jose et al reported an issue found via fuzzing where the verifier's
    pruning logic prematurely marks a program path as safe.

    Consider the following program:

       0: (b7) r6 = 1024
       1: (b7) r7 = 0
       2: (b7) r8 = 0
       3: (b7) r9 = -2147483648
       4: (97) r6 %= 1025
       5: (05) goto pc+0
       6: (bd) if r6 <= r9 goto pc+2
       7: (97) r6 %= 1
       8: (b7) r9 = 0
       9: (bd) if r6 <= r9 goto pc+1
      10: (b7) r6 = 0
      11: (b7) r0 = 0
      12: (63) *(u32 *)(r10 -4) = r0
      13: (18) r4 = 0xffff888103693400 // map_ptr(ks=4,vs=48)
      15: (bf) r1 = r4
      16: (bf) r2 = r10
      17: (07) r2 += -4
      18: (85) call bpf_map_lookup_elem#1
      19: (55) if r0 != 0x0 goto pc+1
      20: (95) exit
      21: (77) r6 >>= 10
      22: (27) r6 *= 8192
      23: (bf) r1 = r0
      24: (0f) r0 += r6
      25: (79) r3 = *(u64 *)(r0 +0)
      26: (7b) *(u64 *)(r1 +0) = r3
      27: (95) exit

    The verifier treats this as safe, leading to oob read/write access due
    to an incorrect verifier conclusion:

      func#0 @0
      0: R1=ctx(off=0,imm=0) R10=fp0
      0: (b7) r6 = 1024                     ; R6_w=1024
      1: (b7) r7 = 0                        ; R7_w=0
      2: (b7) r8 = 0                        ; R8_w=0
      3: (b7) r9 = -2147483648              ; R9_w=-2147483648
      4: (97) r6 %= 1025                    ; R6_w=scalar()
      5: (05) goto pc+0
      6: (bd) if r6 <= r9 goto pc+2         ; R6_w=scalar(umin=18446744071562067969,var_off=
(0xffffffff00000000; 0xffffffff)) R9_w=-2147483648
      7: (97) r6 %= 1                       ; R6_w=scalar()
      8: (b7) r9 = 0                        ; R9=0
      9: (bd) if r6 <= r9 goto pc+1         ; R6=scalar(umin=1) R9=0
      10: (b7) r6 = 0                       ; R6_w=0
      11: (b7) r0 = 0                       ; R0_w=0
      12: (63) *(u32 *)(r10 -4) = r0
      last_idx 12 first_idx 9
      regs=1 stack=0 before 11: (b7) r0 = 0
      13: R0_w=0 R10=fp0 fp-8=0000????
      13: (18) r4 = 0xffff8ad3886c2a00      ; R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      15: (bf) r1 = r4                      ; R1_w=map_ptr(off=0,ks=4,vs=48,imm=0)
R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      16: (bf) r2 = r10                     ; R2_w=fp0 R10=fp0
      17: (07) r2 += -4                     ; R2_w=fp-4
      18: (85) call bpf_map_lookup_elem#1   ;
R0=map_value_or_null(id=1,off=0,ks=4,vs=48,imm=0)
      19: (55) if r0 != 0x0 goto pc+1       ; R0=0
```

```
       20: (95) exit

       from 19 to 21: R0=map_value(off=0,ks=4,vs=48,imm=0) R6=0 R7=0 R8=0 R9=0 R10=fp0 fp-
 8=mmmm????
       21: (77) r6 >>= 10                    ; R6_w=0
       22: (27) r6 *= 8192                   ; R6_w=0
       23: (bf) r1 = r0                      ; R0=map_value(off=0,ks=4,vs=48,imm=0)
 R1_w=map_value(off=0,ks=4,vs=48,imm=0)
       24: (0f) r0 += r6
       last_idx 24 first_idx 19
       regs=40 stack=0 before 23: (bf) r1 = r0
       regs=40 stack=0 before 22: (27) r6 *= 8192
       regs=40 stack=0 before 21: (77) r6 >>= 10
       regs=40 stack=0 before 19: (55) if r0 != 0x0 goto pc+1
       parent didn't have regs=40 stack=0 marks:
 R0_rw=map_value_or_null(id=1,off=0,ks=4,vs=48,imm=0) R6_rw=P0 R7=0 R8=0 R9=0 R10=fp0 fp-
 8=mmmm????
       last_idx 18 first_idx 9
       regs=40 stack=0 before 18: (85) call bpf_map_lookup_elem#1
       regs=40 stack=0 before 17: (07) r2 += -4
       regs=40 stack=0 before 16: (bf) r2 = r10
       regs=40 stack=0 before 15: (bf) r1 = r4
       regs=40 stack=0 before 13: (18) r4 = 0xffff8ad3886c2a00
       regs=40 stack=0 before 12: (63) *(u32 *)(r10 -4) = r0
       regs=40 stack=0 before 11: (b7) r0 = 0
       regs=40 stack=0 before 10: (b7) r6 = 0
       25: (79) r3 = *(u64 *)(r0 +0)         ; R0_w=map_value(off=0,ks=4,vs=48,imm=0)
 R3_w=scalar()
       26: (7b) *(u64 *)(r1 +0) = r3         ; R1_w=map_value(off=0,ks=4,vs=48,imm=0)
 R3_w=scalar()
       27: (95) exit

       from 9 to 11: R1=ctx(off=0,imm=0) R6=0 R7=0 R8=0 R9=0 R10=fp0
       11: (b7) r0 = 0                       ; R0_w=0
       12: (63) *(u32 *)(r10 -4) = r0
       last_idx 12 first_idx 11
       regs=1 stack=0 before 11: (b7) r0 = 0
       13: R0_w=0 R10=fp0 fp-8=0000????
       13: (18) r4 = 0xffff8ad3886c2a00      ; R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
       15: (bf) r1 = r4                      ; R1_w=map_ptr(off=0,ks=4,vs=48,imm=0)
 R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
       16: (bf) r2 = r10                     ; R2_w=fp0 R10=fp0
       17: (07) r2 += -4                     ; R2_w=fp-4
       18: (85) call bpf_map_lookup_elem#1
       frame 0: propagating r6
       last_idx 19 first_idx 11
       regs=40 stack=0 before 18: (85) call bpf_map_lookup_elem#1
       regs=40 stack=0 before 17: (07) r2 += -4
       regs=40 stack=0 before 16: (bf) r2 = r10
       regs=40 stack=0 before 15: (bf) r1 = r4
       regs=40 stack=0 before 13: (18) r4 = 0xffff8ad3886c2a00
       regs=40 stack=0 before 12: (63) *(u32 *)(r10 -4) = r0
       regs=40 stack=0 before 11: (b7) r0 = 0
       parent didn't have regs=40 stack=0 marks: R1=ctx(off=0,imm=0) R6_r=P0 R7=0 R8=0 R9=0
 R10=fp0
       last_idx 9 first_idx 9
       regs=40 stack=0 before 9: (bd) if r6 <= r9 goto pc+1
       parent didn't have regs=40 stack=0 marks: R1=ctx(off=0,imm=0) R6_rw=Pscalar() R7_w=0
 R8_w=0 R9_rw=0 R10=fp0
       last_idx 8 first_idx 0
       regs=40 stack=0 before 8: (b7) r9 = 0
       regs=40 stack=0 before 7: (97) r6 %= 1
       regs=40 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
       regs=40 stack=0 before 5: (05) goto pc+0
       regs=40 stack=0 before 4: (97) r6 %= 1025
```

```
       regs=40 stack=0 before 3: (b7) r9 = -2147483648
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024
       19: safe
       frame 0: propagating r6
       last_idx 9 first_idx 0
       regs=40 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
       regs=40 stack=0 before 5: (05) goto pc+0
       regs=40 stack=0 before 4: (97) r6 %= 1025
       regs=40 stack=0 before 3: (b7) r9 = -2147483648
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024

       from 6 to 9: safe
       verification time 110 usec
       stack depth 4
       processed 36 insns (limit 1000000) max_states_per_insn 0 total_states 3 peak_states 3
mark_read 2
```

The verifier considers this program as safe by mistakenly pruning unsafe code paths. In the above func#0, code lines 0-10 are of interest. In line 0-3 registers r6 to r9 are initialized with known scalar values. In line 4 the register r6 is reset to an unknown scalar given the verifier does not track modulo operations. Due to this, the verifier can also not determine precisely which branches in line 6 and 9 are taken, therefore it needs to explore them both.

As can be seen, the verifier starts with exploring the false/fall-through paths first. The 'from 19 to 21' path has both r6=0 and r9=0 and the pointer arithmetic on r0 += r6 is therefore considered safe. Given the arithmetic, r6 is correctly marked for precision tracking where backtracking kicks in where it walks back the current path all the way where r6 was set to 0 in the fall-through branch.

Next, the pruning logics pops the path 'from 9 to 11' from the stack. Also here, the state of the registers is the same, that is, r6=0 and r9=0, so that at line 19 the path can be pruned as it is considered safe. It is interesting to note that the conditional in line 9 turned r6 into a more precise state, that is, in the fall-through path at the beginning of line 10, it is R6=scalar(umin=1), and in the branch-taken path (which is analyzed here) at the beginning of line 11, r6 turned into a known const r6=0 as r9=0 prior to that and therefore (unsigned) r6 <= 0 concludes that r6 must be 0 (**):

```
  [...]                                ; R6_w=scalar()
  9: (bd) if r6 <= r9 goto pc+1        ; R6=scalar(umin=1) R9=0
  [...]

  from 9 to 11: R1=ctx(off=0,imm=0) R6=0 R7=0 R8=0 R9=0 R10=fp0
  [...]
```

The next path is 'from 6 to 9'. The verifier considers the old and current state equivalent, and therefore prunes the search incorrectly. Looking into the two states which are being compared by the pruning logic at line 9, the old state consists of R6_rwD=Pscalar() R9_rwD=0 R10=fp0 and the new state consists of R1=ctx(off=0,imm=0) R6_w=scalar(umax=18446744071562067968) R7_w=0 R8_w=0 R9_w=-2147483648 R10=fp0. While r6 had the reg->precise flag correctly set in the old state, r9 did not. Both r6'es are considered as equivalent given the old one is a superset of the current, more precise one, however, r9's actual values (0 vs 0x80000000) mismatch. Given the old r9 did not have reg->precise flag set, the verifier does not consider the register as contributing to the precision state of r6, and therefore it considered both r9 states as equivalent. However, for this specific pruned

```
    path (which is also the actual path taken at runtime), register r6 will be
    0x400 and r9 0x80000000 when reaching line 21, thus oob-accessing the map.

    The purpose of precision tracking is to initially mark registers (including
    spilled ones) as imprecise to help verifier's pruning logic finding equivalent
    states it can then prune if they don't contribute to the program's safety
    aspects. For example, if registers are used for pointer arithmetic or to pass
    constant length to a helper, then the verifier sets reg->precise flag and
    backtracks the BPF program instruction sequence and chain of verifier states
    to ensure that the given register or stack slot including their dependencies
    are marked as precisely tracked scalar. This also includes any other registers
    and slots that contribute to a tracked state of given registers/stack slot.
    This backtracking relies on recorded jmp_history and is able to traverse
    entire chain of parent states. This process ends only when all the necessary
    registers/slots and their transitive dependencies are marked as precise.

    The backtrack_insn() is called from the current instruction up to the first
    instruction, and its purpose is to compute a bitmask of registers and stack
    slots that need precision tracking in the parent's verifier state. For example,
    if a current instruction is r6 = r7, then r6 needs precision after this
    instruction and r7 needs precision before this instruction, that is, in the
    parent state. Hence for the latter r7 is marked and r6 unmarked.

    For the class of jmp/jmp32 instructions, backtrack_insn() today only looks
    at call and exit instructions and for all other conditionals the masks
    remain as-is. However, in the given situation register r6 has a dependency
    on r9 (as described above in **), so also that one needs to be marked for
    precision tracking. In other words, if an imprecise register influences a
    precise one, then the imprecise register should also be marked precise.
    Meaning, in the parent state both dest and src register need to be tracked
    for precision and therefore the marking must be more conservative by setting
    reg->precise flag for both. The precision propagation needs to cover both
    for the conditional: if the src reg was marked but not the dst reg and vice
    versa.

    After the fix the program is correctly rejected:

      func#0 @0
      0: R1=ctx(off=0,imm=0) R10=fp0
      0: (b7) r6 = 1024                     ; R6_w=1024
      1: (b7) r7 = 0                        ; R7_w=0
      2: (b7) r8 = 0                        ; R8_w=0
      3: (b7) r9 = -2147483648              ; R9_w=-2147483648
      4: (97) r6 %= 1025                    ; R6_w=scalar()
      5: (05) goto pc+0
      6: (bd) if r6 <= r9 goto pc+2         ; R6_w=scalar(umin=18446744071562067969,var_off=
(0xffffffff80000000; 0x7fffffff),u32_min=-2147483648) R9_w=-2147483648
      7: (97) r6 %= 1                       ; R6_w=scalar()
      8: (b7) r9 = 0                        ; R9=0
      9: (bd) if r6 <= r9 goto pc+1         ; R6=scalar(umin=1) R9=0
      10: (b7) r6 = 0                       ; R6_w=0
      11: (b7) r0 = 0                       ; R0_w=0
      12: (63) *(u32 *)(r10 -4) = r0
      last_idx 12 first_idx 9
      regs=1 stack=0 before 11: (b7) r0 = 0
      13: R0_w=0 R10=fp0 fp-8=0000????
      13: (18) r4 = 0xffff9290dc5bfe00      ; R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      15: (bf) r1 = r4                      ; R1_w=map_ptr(off=0,ks=4,vs=48,imm=0)
R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      16: (bf) r2 = r10                     ; R2_w=fp0 R10=fp0
      17: (07) r2 += -4                     ; R2_w=fp-4
      18: (85) call bpf_map_lookup_elem#1   ;
R0=map_value_or_null(id=1,off=0,ks=4,vs=48,imm=0)
      19: (55) if r0 != 0x0 goto pc+1       ; R0=0
      20: (95) exit
```

```
      from 19 to 21: R0=map_value(off=0,ks=4,vs=48,imm=0) R6=0 R7=0 R8=0 R9=0 R10=fp0 fp-
8=mmmm????
      21: (77) r6 >>= 10                    ; R6_w=0
      22: (27) r6 *= 8192                   ; R6_w=0
      23: (bf) r1 = r0                      ; R0=map_value(off=0,ks=4,vs=48,imm=0)
R1_w=map_value(off=0,ks=4,vs=48,imm=0)
      24: (0f) r0 += r6
      last_idx 24 first_idx 19
      regs=40 stack=0 before 23: (bf) r1 = r0
      regs=40 stack=0 before 22: (27) r6 *= 8192
      regs=40 stack=0 before 21: (77) r6 >>= 10
      regs=40 stack=0 before 19: (55) if r0 != 0x0 goto pc+1
      parent didn't have regs=40 stack=0 marks:
R0_rw=map_value_or_null(id=1,off=0,ks=4,vs=48,imm=0) R6_rw=P0 R7=0 R8=0 R9=0 R10=fp0 fp-
8=mmmm????
      last_idx 18 first_idx 9
      regs=40 stack=0 before 18: (85) call bpf_map_lookup_elem#1
      regs=40 stack=0 before 17: (07) r2 += -4
      regs=40 stack=0 before 16: (bf) r2 = r10
      regs=40 stack=0 before 15: (bf) r1 = r4
      regs=40 stack=0 before 13: (18) r4 = 0xffff9290dc5bfe00
      regs=40 stack=0 before 12: (63) *(u32 *)(r10 -4) = r0
      regs=40 stack=0 before 11: (b7) r0 = 0
      regs=40 stack=0 before 10: (b7) r6 = 0
      25: (79) r3 = *(u64 *)(r0 +0)         ; R0_w=map_value(off=0,ks=4,vs=48,imm=0)
R3_w=scalar()
      26: (7b) *(u64 *)(r1 +0) = r3         ; R1_w=map_value(off=0,ks=4,vs=48,imm=0)
R3_w=scalar()
      27: (95) exit

      from 9 to 11: R1=ctx(off=0,imm=0) R6=0 R7=0 R8=0 R9=0 R10=fp0
      11: (b7) r0 = 0                       ; R0_w=0
      12: (63) *(u32 *)(r10 -4) = r0
      last_idx 12 first_idx 11
      regs=1 stack=0 before 11: (b7) r0 = 0
      13: R0_w=0 R10=fp0 fp-8=0000????
      13: (18) r4 = 0xffff9290dc5bfe00      ; R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      15: (bf) r1 = r4                      ; R1_w=map_ptr(off=0,ks=4,vs=48,imm=0)
R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
      16: (bf) r2 = r10                     ; R2_w=fp0 R10=fp0
      17: (07) r2 += -4                     ; R2_w=fp-4
      18: (85) call bpf_map_lookup_elem#1
      frame 0: propagating r6
      last_idx 19 first_idx 11
      regs=40 stack=0 before 18: (85) call bpf_map_lookup_elem#1
      regs=40 stack=0 before 17: (07) r2 += -4
      regs=40 stack=0 before 16: (bf) r2 = r10
      regs=40 stack=0 before 15: (bf) r1 = r4
      regs=40 stack=0 before 13: (18) r4 = 0xffff9290dc5bfe00
      regs=40 stack=0 before 12: (63) *(u32 *)(r10 -4) = r0
      regs=40 stack=0 before 11: (b7) r0 = 0
      parent didn't have regs=40 stack=0 marks: R1=ctx(off=0,imm=0) R6_r=P0 R7=0 R8=0 R9=0
R10=fp0
      last_idx 9 first_idx 9
      regs=40 stack=0 before 9: (bd) if r6 <= r9 goto pc+1
      parent didn't have regs=240 stack=0 marks: R1=ctx(off=0,imm=0) R6_rw=Pscalar() R7_w=0
R8_w=0 R9_rw=P0 R10=fp0
      last_idx 8 first_idx 0
      regs=240 stack=0 before 8: (b7) r9 = 0
      regs=40 stack=0 before 7: (97) r6 %= 1
      regs=40 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
      regs=240 stack=0 before 5: (05) goto pc+0
      regs=240 stack=0 before 4: (97) r6 %= 1025
      regs=240 stack=0 before 3: (b7) r9 = -2147483648
```

```
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024
       19: safe

       from 6 to 9: R1=ctx(off=0,imm=0) R6_w=scalar(umax=18446744071562067968) R7_w=0 R8_w=0
 R9_w=-2147483648 R10=fp0
       9: (bd) if r6 <= r9 goto pc+1
       last_idx 9 first_idx 0
       regs=40 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
       regs=240 stack=0 before 5: (05) goto pc+0
       regs=240 stack=0 before 4: (97) r6 %= 1025
       regs=240 stack=0 before 3: (b7) r9 = -2147483648
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024
       last_idx 9 first_idx 0
       regs=200 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
       regs=240 stack=0 before 5: (05) goto pc+0
       regs=240 stack=0 before 4: (97) r6 %= 1025
       regs=240 stack=0 before 3: (b7) r9 = -2147483648
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024
       11: R6=scalar(umax=18446744071562067968) R9=-2147483648
       11: (b7) r0 = 0                       ; R0_w=0
       12: (63) *(u32 *)(r10 -4) = r0
       last_idx 12 first_idx 11
       regs=1 stack=0 before 11: (b7) r0 = 0
       13: R0_w=0 R10=fp0 fp-8=0000????
       13: (18) r4 = 0xffff9290dc5bfe00      ; R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
       15: (bf) r1 = r4                      ; R1_w=map_ptr(off=0,ks=4,vs=48,imm=0)
 R4_w=map_ptr(off=0,ks=4,vs=48,imm=0)
       16: (bf) r2 = r10                     ; R2_w=fp0 R10=fp0
       17: (07) r2 += -4                     ; R2_w=fp-4
       18: (85) call bpf_map_lookup_elem#1   ;
 R0_w=map_value_or_null(id=3,off=0,ks=4,vs=48,imm=0)
       19: (55) if r0 != 0x0 goto pc+1       ; R0_w=0
       20: (95) exit

       from 19 to 21: R0=map_value(off=0,ks=4,vs=48,imm=0)
 R6=scalar(umax=18446744071562067968) R7=0 R8=0 R9=-2147483648 R10=fp0 fp-8=mmmm????
       21: (77) r6 >>= 10                    ; R6_w=scalar(umax=18014398507384832,var_off=
 (0x0; 0x3fffffffffffff))
       22: (27) r6 *= 8192                   ;
 R6_w=scalar(smax=9223372036854767616,umax=18446744073709543424,var_off=(0x0;
 0xfffffffffffe000),s32_max=2147475456,u32_max=-8192)
       23: (bf) r1 = r0                      ; R0=map_value(off=0,ks=4,vs=48,imm=0)
 R1_w=map_value(off=0,ks=4,vs=48,imm=0)
       24: (0f) r0 += r6
       last_idx 24 first_idx 21
       regs=40 stack=0 before 23: (bf) r1 = r0
       regs=40 stack=0 before 22: (27) r6 *= 8192
       regs=40 stack=0 before 21: (77) r6 >>= 10
       parent didn't have regs=40 stack=0 marks: R0_rw=map_value(off=0,ks=4,vs=48,imm=0)
 R6_r=Pscalar(umax=18446744071562067968) R7=0 R8=0 R9=-2147483648 R10=fp0 fp-8=mmmm????
       last_idx 19 first_idx 11
       regs=40 stack=0 before 19: (55) if r0 != 0x0 goto pc+1
       regs=40 stack=0 before 18: (85) call bpf_map_lookup_elem#1
       regs=40 stack=0 before 17: (07) r2 += -4
       regs=40 stack=0 before 16: (bf) r2 = r10
       regs=40 stack=0 before 15: (bf) r1 = r4
       regs=40 stack=0 before 13: (18) r4 = 0xffff9290dc5bfe00
       regs=40 stack=0 before 12: (63) *(u32 *)(r10 -4) = r0
       regs=40 stack=0 before 11: (b7) r0 = 0
```

```
       parent didn't have regs=40 stack=0 marks: R1=ctx(off=0,imm=0)
  R6_rw=Pscalar(umax=18446744071562067968) R7_w=0 R8_w=0 R9_w=-2147483648 R10=fp0
       last_idx 9 first_idx 0
       regs=40 stack=0 before 9: (bd) if r6 <= r9 goto pc+1
       regs=240 stack=0 before 6: (bd) if r6 <= r9 goto pc+2
       regs=240 stack=0 before 5: (05) goto pc+0
       regs=240 stack=0 before 4: (97) r6 %= 1025
       regs=240 stack=0 before 3: (b7) r9 = -2147483648
       regs=40 stack=0 before 2: (b7) r8 = 0
       regs=40 stack=0 before 1: (b7) r7 = 0
       regs=40 stack=0 before 0: (b7) r6 = 1024
       math between map_value pointer and register with unbounded min value is not allowed
       verification time 886 usec
       stack depth 4
       processed 49 insns (limit 1000000) max_states_per_insn 1 total_states 5 peak_states 5
  mark_read 2

      Fixes: b5dc0163d8fd ("bpf: precise scalar_value tracking")
      Reported-by: Juan Jose Lopez Jaimez <jjlopezjaimez@google.com>
      Reported-by: Meador Inge <meadori@google.com>
      Reported-by: Simon Scannell <simonscannell@google.com>
      Reported-by: Nenad Stojanovski <thenenadx@google.com>
      Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
      Co-developed-by: Andrii Nakryiko <andrii@kernel.org>
      Signed-off-by: Andrii Nakryiko <andrii@kernel.org>
      Reviewed-by: John Fastabend <john.fastabend@gmail.com>
      Reviewed-by: Juan Jose Lopez Jaimez <jjlopezjaimez@google.com>
      Reviewed-by: Meador Inge <meadori@google.com>
      Reviewed-by: Simon Scannell <simonscannell@google.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit e5e31c9808da02614a08d6003e0fb090df16e495
  Author: Li Lanzhe <u202212060@hust.edu.cn>
  Date:   Wed Apr 19 07:50:29 2023 -0400

      spi: spi-rockchip: Fix missing unwind goto in rockchip_sfc_probe()

      [ Upstream commit 359f5b0d4e26b7a7bcc574d6148b31a17cefe47d ]

      If devm_request_irq() fails, then we are directly return 'ret' without
      clk_disable_unprepare(sfc->clk) and clk_disable_unprepare(sfc->hclk).

      Fix this by changing direct return to a goto 'err_irq'.

      Fixes: 0b89fc0a367e ("spi: rockchip-sfc: add rockchip serial flash controller")
      Signed-off-by: Li Lanzhe <u202212060@hust.edu.cn>
      Reviewed-by: Dongliang Mu <dzm91@hust.edu.cn>
      Link: https://lore.kernel.org/r/20230419115030.6029-1-u202212060@hust.edu.cn
      Signed-off-by: Mark Brown <broonie@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit d20f4daa38e940ee3ab2b0c20b1daedd8cb40ef9
  Author: Ido Schimmel <idosch@nvidia.com>
  Date:   Mon Apr 17 18:52:51 2023 +0200

      mlxsw: pci: Fix possible crash during initialization

      [ Upstream commit 1f64757ee2bb22a93ec89b4c71707297e8cca0ba ]

      During initialization the driver issues a reset command via its command
      interface in order to remove previous configuration from the device.

      After issuing the reset, the driver waits for 200ms before polling on
      the "system_status" register using memory-mapped IO until the device
      reaches a ready state (0x5E). The wait is necessary because the reset
```

```
    command only triggers the reset, but the reset itself happens
    asynchronously. If the driver starts polling too soon, the read of the
    "system_status" register will never return and the system will crash
    [1].

    The issue was discovered when the device was flashed with a development
    firmware version where the reset routine took longer to complete. The
    issue was fixed in the firmware, but it exposed the fact that the
    current wait time is borderline.

    Fix by increasing the wait time from 200ms to 400ms. With this patch and
    the buggy firmware version, the issue did not reproduce in 10 reboots
    whereas without the patch the issue is reproduced quite consistently.

    [1]
    mce: CPUs not responding to MCE broadcast (may include false positives): 0,4
    mce: CPUs not responding to MCE broadcast (may include false positives): 0,4
    Kernel panic - not syncing: Timeout: Not all CPUs entered broadcast exception handler
    Shutting down cpus with NMI
    Kernel Offset: 0x12000000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-
0xffffffffbfffffff)

    Fixes: ac004e84164e ("mlxsw: pci: Wait longer before accessing the device after reset")
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: Petr Machata <petrm@nvidia.com>
    Signed-off-by: Petr Machata <petrm@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 191642f5cfb38c0e44fb4783a37530bae15b8f8e
Author: Alexander Aring <aahringo@redhat.com>
Date:   Mon Apr 17 09:00:52 2023 -0400

    net: rpl: fix rpl header size calculation

    [ Upstream commit 4e006c7a6dac0ead4c1bf606000aa90a372fc253 ]

    This patch fixes a missing 8 byte for the header size calculation. The
    ipv6_rpl_srh_size() is used to check a skb_pull() on skb->data which
    points to skb_transport_header(). Currently we only check on the
    calculated addresses fields using CmprI and CmprE fields, see:

    https://www.rfc-editor.org/rfc/rfc6554#section-3

    there is however a missing 8 byte inside the calculation which stands
    for the fields before the addresses field. Those 8 bytes are represented
    by sizeof(struct ipv6_rpl_sr_hdr) expression.

    Fixes: 8610c7c6e3bd ("net: ipv6: add support for rpl sr exthdr")
    Signed-off-by: Alexander Aring <aahringo@redhat.com>
    Reported-by: maxpl0it <maxpl0it@protonmail.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea471045145923d164ce4a5ffacbefa1c8bd8550
Author: Ido Schimmel <idosch@nvidia.com>
Date:   Mon Apr 17 09:12:16 2023 +0300

    bonding: Fix memory leak when changing bond type to Ethernet

    [ Upstream commit c484fcc058bada604d7e4e5228d4affb646ddbc2 ]

    When a net device is put administratively up, its 'IFF_UP' flag is set
    (if not set already) and a 'NETDEV_UP' notification is emitted, which
```

```
    causes the 8021q driver to add VLAN ID 0 on the device. The reverse
    happens when a net device is put administratively down.

    When changing the type of a bond to Ethernet, its 'IFF_UP' flag is
    incorrectly cleared, resulting in the kernel skipping the above process
    and VLAN ID 0 being leaked [1].

    Fix by restoring the flag when changing the type to Ethernet, in a
    similar fashion to the restoration of the 'IFF_SLAVE' flag.

    The issue can be reproduced using the script in [2], with example out
    before and after the fix in [3].

    [1]
    unreferenced object 0xffff888103479900 (size 256):
      comm "ip", pid 329, jiffies 4294775225 (age 28.561s)
      hex dump (first 32 bytes):
        00 a0 0c 15 81 88 ff ff 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:
        [<ffffffff81a6051a>] kmalloc_trace+0x2a/0xe0
        [<ffffffff8406426c>] vlan_vid_add+0x30c/0x790
        [<ffffffff84068e21>] vlan_device_event+0x1491/0x21a0
        [<ffffffff81440c8e>] notifier_call_chain+0xbe/0x1f0
        [<ffffffff8372383a>] call_netdevice_notifiers_info+0xba/0x150
        [<ffffffff837590f2>] __dev_notify_flags+0x132/0x2e0
        [<ffffffff8375ad9f>] dev_change_flags+0x11f/0x180
        [<ffffffff8379af36>] do_setlink+0xb96/0x4060
        [<ffffffff837adf6a>] __rtnl_newlink+0xc0a/0x18a0
        [<ffffffff837aec6c>] rtnl_newlink+0x6c/0xa0
        [<ffffffff837ac64e>] rtnetlink_rcv_msg+0x43e/0xe00
        [<ffffffff839a99e0>] netlink_rcv_skb+0x170/0x440
        [<ffffffff839a738f>] netlink_unicast+0x53f/0x810
        [<ffffffff839a7fcb>] netlink_sendmsg+0x96b/0xe90
        [<ffffffff8369d12f>] ____sys_sendmsg+0x30f/0xa70
        [<ffffffff836a6d7a>] ___sys_sendmsg+0x13a/0x1e0
    unreferenced object 0xffff88810f6a83e0 (size 32):
      comm "ip", pid 329, jiffies 4294775225 (age 28.561s)
      hex dump (first 32 bytes):
        a0 99 47 03 81 88 ff ff a0 99 47 03 81 88 ff ff  ..G.......G.....
        81 00 00 00 01 00 00 00 cc cc cc cc cc cc cc cc  ................
      backtrace:
        [<ffffffff81a6051a>] kmalloc_trace+0x2a/0xe0
        [<ffffffff84064369>] vlan_vid_add+0x409/0x790
        [<ffffffff84068e21>] vlan_device_event+0x1491/0x21a0
        [<ffffffff81440c8e>] notifier_call_chain+0xbe/0x1f0
        [<ffffffff8372383a>] call_netdevice_notifiers_info+0xba/0x150
        [<ffffffff837590f2>] __dev_notify_flags+0x132/0x2e0
        [<ffffffff8375ad9f>] dev_change_flags+0x11f/0x180
        [<ffffffff8379af36>] do_setlink+0xb96/0x4060
        [<ffffffff837adf6a>] __rtnl_newlink+0xc0a/0x18a0
        [<ffffffff837aec6c>] rtnl_newlink+0x6c/0xa0
        [<ffffffff837ac64e>] rtnetlink_rcv_msg+0x43e/0xe00
        [<ffffffff839a99e0>] netlink_rcv_skb+0x170/0x440
        [<ffffffff839a738f>] netlink_unicast+0x53f/0x810
        [<ffffffff839a7fcb>] netlink_sendmsg+0x96b/0xe90
        [<ffffffff8369d12f>] ____sys_sendmsg+0x30f/0xa70
        [<ffffffff836a6d7a>] ___sys_sendmsg+0x13a/0x1e0

    [2]
    ip link add name t-nlmon type nlmon
    ip link add name t-dummy type dummy
    ip link add name t-bond type bond mode active-backup

    ip link set dev t-bond up
```

```
    ip link set dev t-nlmon master t-bond
    ip link set dev t-nlmon nomaster
    ip link show dev t-bond
    ip link set dev t-dummy master t-bond
    ip link show dev t-bond

    ip link del dev t-bond
    ip link del dev t-dummy
    ip link del dev t-nlmon

    [3]
    Before:

    12: t-bond: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN
 mode DEFAULT group default qlen 1000
        link/netlink
    12: t-bond: <BROADCAST,MULTICAST,MASTER,LOWER_UP> mtu 1500 qdisc noqueue state UP mode
 DEFAULT group default qlen 1000
        link/ether 46:57:39:a4:46:a2 brd ff:ff:ff:ff:ff:ff

    After:

    12: t-bond: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN
 mode DEFAULT group default qlen 1000
        link/netlink
    12: t-bond: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 mode DEFAULT group default qlen 1000
        link/ether 66:48:7b:74:b6:8a brd ff:ff:ff:ff:ff:ff

    Fixes: e36b9d16c6a6 ("bonding: clean muticast addresses when device changes type")
    Fixes: 75c78500ddad ("bonding: remap muticast addresses without using dev_close() and
 dev_open()")
    Fixes: 9ec7eb60dcbc ("bonding: restore IFF_MASTER/SLAVE flags on bond enslave ether type
 change")
    Reported-by: Mirsad Goran Todorovac <mirsad.todorovac@alu.unizg.hr>
    Link: https://lore.kernel.org/netdev/78a8a03b-6070-3e6b-5042-f848dab16fb8@alu.unizg.hr/
    Tested-by: Mirsad Goran Todorovac <mirsad.todorovac@alu.unizg.hr>
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Acked-by: Jay Vosburgh <jay.vosburgh@canonical.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 8b2482ba970ac99c70b13ed8c48e50b24b89a565
 Author: Huacai Chen <chenhuacai@kernel.org>
 Date:   Tue Apr 18 19:38:58 2023 +0800

    LoongArch: Fix build error if CONFIG_SUSPEND is not set

    [ Upstream commit 1cf62488f5e465b1cd814d19be238a4b7ad5be38 ]

    We can see the following build error on LoongArch if CONFIG_SUSPEND is
    not set:

      ld: drivers/acpi/sleep.o: in function 'acpi_pm_prepare':
      sleep.c:(.text+0x2b8): undefined reference to 'loongarch_wakeup_start'

    Here is the call trace:

      acpi_pm_prepare()
        __acpi_pm_prepare()
          acpi_sleep_prepare()
            acpi_get_wakeup_address()
              loongarch_wakeup_start()

    Root cause: loongarch_wakeup_start() is defined in arch/loongarch/power/
```

```
    suspend_asm.S which is only built under CONFIG_SUSPEND. In order to fix
    the build error, just let acpi_get_wakeup_address() return 0 if CONFIG_
    SUSPEND is not set.

    Fixes: 366bb35a8e48 ("LoongArch: Add suspend (ACPI S3) support")
    Reviewed-by: WANG Xuerui <git@xen0n.name>
    Reported-by: Randy Dunlap <rdunlap@infradead.org>
    Link: https://lore.kernel.org/all/11215033-fa3c-ecb1-2fc0-e9aeba47be9b@infradead.org/
    Signed-off-by: Tiezhu Yang <yangtiezhu@loongson.cn>
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 09ef093abda2c37d1c386eca0acf8d7165174cc7
Author: Nikita Zhandarovich <n.zhandarovich@fintech.ru>
Date:   Mon Apr 17 05:07:18 2023 -0700

    mlxfw: fix null-ptr-deref in mlxfw_mfa2_tlv_next()

    [ Upstream commit c0e73276f0fcbbd3d4736ba975d7dc7a48791b0c ]

    Function mlxfw_mfa2_tlv_multi_get() returns NULL if 'tlv' in
    question does not pass checks in mlxfw_mfa2_tlv_payload_get(). This
    behaviour may lead to NULL pointer dereference in 'multi->total_len'.
    Fix this issue by testing mlxfw_mfa2_tlv_multi_get()'s return value
    against NULL.

    Found by Linux Verification Center (linuxtesting.org) with static
    analysis tool SVACE.

    Fixes: 410ed13cae39 ("Add the mlxfw module for Mellanox firmware flash process")
    Co-developed-by: Natalia Petrova <n.petrova@fintech.ru>
    Signed-off-by: Nikita Zhandarovich <n.zhandarovich@fintech.ru>
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Link: https://lore.kernel.org/r/20230417120718.52325-1-n.zhandarovich@fintech.ru
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a1f76b8305235f7ec1f4484cd04d5144f4915d43
Author: Michael Chan <michael.chan@broadcom.com>
Date:   Sun Apr 16 23:58:18 2023 -0700

    bnxt_en: Do not initialize PTP on older P3/P4 chips

    [ Upstream commit e8b51a1a15d5a3cce231e0669f6a161dc5bb9b75 ]

    The driver does not support PTP on these older chips and it is assuming
    that firmware on these older chips will not return the
    PORT_MAC_PTP_QCFG_RESP_FLAGS_HWRM_ACCESS flag in __bnxt_hwrm_ptp_qcfg(),
    causing the function to abort quietly.

    But newer firmware now sets this flag and so __bnxt_hwrm_ptp_qcfg()
    will proceed further.  Eventually it will fail in bnxt_ptp_init() ->
    bnxt_map_ptp_regs() because there is no code to support the older chips.
    The driver will then complain:

    "PTP initialization failed.\n"

    Fix it so that we abort quietly earlier without going through the
    unnecessary steps and alarming the user with the warning log.

    Fixes: ae5c42f0b92c ("bnxt_en: Get PTP hardware capability from firmware")
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit e93b900030a3598d92ab11401484cd1a32ba101d
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Mon Apr 17 17:50:28 2023 +0200

    netfilter: nf_tables: tighten netlink attribute requirements for catch-all elements

    [ Upstream commit d4eb7e39929a3b1ff30fb751b4859fc2410702a0 ]

    If NFT_SET_ELEM_CATCHALL is set on, then userspace provides no set element
    key. Otherwise, bail out with -EINVAL.

    Fixes: aaa31047a6d2 ("netfilter: nftables: add catch-all set element support")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7977bb80817b552d97bcc8a1ddf2b3029ca5b6a8
Author: Duoming Zhou <duoming@zju.edu.cn>
Date:   Sat Apr 15 16:12:27 2023 +0800

    cxgb4: fix use after free bugs caused by circular dependency problem

    [ Upstream commit e50b9b9e8610d47b7c22529443e45a16b1ea3a15 ]

    The flower_stats_timer can schedule flower_stats_work and
    flower_stats_work can also arm the flower_stats_timer. The
    process is shown below:

    ----------- timer schedules work ------------
    ch_flower_stats_cb() //timer handler
      schedule_work(&adap->flower_stats_work);

    ----------- work arms timer ------------
    ch_flower_stats_handler() //workqueue callback function
      mod_timer(&adap->flower_stats_timer, ...);

    When the cxgb4 device is detaching, the timer and workqueue
    could still be rearmed. The process is shown below:

      (cleanup routine)           | (timer and workqueue routine)
    remove_one()                  |
      free_some_resources()       | ch_flower_stats_cb() //timer
        cxgb4_cleanup_tc_flower() |   schedule_work()
          del_timer_sync()        |
                                  | ch_flower_stats_handler() //workqueue
                                  |   mod_timer()
          cancel_work_sync()      |
      kfree(adapter) //FREE       | ch_flower_stats_cb() //timer
                                  |   adap->flower_stats_work //USE

    This patch changes del_timer_sync() to timer_shutdown_sync(),
    which could prevent rearming of the timer from the workqueue.

    Fixes: e0f911c81e93 ("cxgb4: fetch stats for offloaded tc flower flows")
    Signed-off-by: Duoming Zhou <duoming@zju.edu.cn>
    Link: https://lore.kernel.org/r/20230415081227.7463-1-duoming@zju.edu.cn
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 076574cd3276ea6114fbe1e830f442da7965b8dd
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Mon Apr 17 12:14:29 2023 +0200

    netfilter: nf_tables: validate catch-all set elements

    [ Upstream commit d46fc894147cf98dd6e8210aa99ed46854191840 ]
```

```
    catch-all set element might jump/goto to chain that uses expressions
    that require validation.

    Fixes: aaa31047a6d2 ("netfilter: nftables: add catch-all set element support")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ab3dde046508f5f9066c04fe4a1b77fe98bd02f3
Author: Aleksandr Loktionov <aleksandr.loktionov@intel.com>
Date:   Mon Apr 3 07:13:18 2023 +0200

    i40e: fix i40e_setup_misc_vector() error handling

    [ Upstream commit c86c00c6935505929cc9adb29ddb85e48c71f828 ]

    Add error handling of i40e_setup_misc_vector() in i40e_rebuild().
    In case interrupt vectors setup fails do not re-open vsi-s and
    do not bring up vf-s, we have no interrupts to serve a traffic
    anyway.

    Fixes: 41c445ff0f48 ("i40e: main driver core")
    Signed-off-by: Aleksandr Loktionov <aleksandr.loktionov@intel.com>
    Tested-by: Pucha Himasekhar Reddy <himasekharx.reddy.pucha@intel.com> (A Contingent
worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 382d0529c4820fbbcb516632d408f8d7fc6da37f
Author: Aleksandr Loktionov <aleksandr.loktionov@intel.com>
Date:   Fri Mar 24 18:16:38 2023 +0100

    i40e: fix accessing vsi->active_filters without holding lock

    [ Upstream commit 8485d093b076e59baff424552e8aecfc5bd2d261 ]

    Fix accessing vsi->active_filters without holding the mac_filter_hash_lock.
    Move vsi->active_filters = 0 inside critical section and
    move clear_bit(__I40E_VSI_OVERFLOW_PROMISC, vsi->state) after the critical
    section to ensure the new filters from other threads can be added only after
    filters cleaning in the critical section is finished.

    Fixes: 278e7d0b9d68 ("i40e: store MAC/VLAN filters in a hash with the MAC Address as
key")
    Signed-off-by: Aleksandr Loktionov <aleksandr.loktionov@intel.com>
    Tested-by: Pucha Himasekhar Reddy <himasekharx.reddy.pucha@intel.com> (A Contingent
worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4e5abb1c3a17222706dd9b8ce51ab21dd988f54e
Author: Florian Westphal <fw@strlen.de>
Date:   Mon Apr 17 10:21:36 2023 +0200

    netfilter: nf_tables: fix ifdef to also consider nf_tables=m

    [ Upstream commit c55c0e91c813589dc55bea6bf9a9fbfaa10ae41d ]

    nftables can be built as a module, so fix the preprocessor conditional
    accordingly.

    Fixes: 478b360a47b7 ("netfilter: nf_tables: fix nf_trace always-on with XT_TRACE=n")
    Reported-by: Florian Fainelli <f.fainelli@gmail.com>
    Reported-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Florian Westphal <fw@strlen.de>
```

```
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3588642fad67cdd054abc21953b739ac629080b
Author: Ding Hui <dinghui@sangfor.com.cn>
Date:   Fri Apr 14 23:23:06 2023 +0800

    sfc: Fix use-after-free due to selftest_work

    [ Upstream commit a80bb8e7233b2ad6ff119646b6e33fb3edcec37b ]

    There is a use-after-free scenario that is:

    When the NIC is down, user set mac address or vlan tag to VF,
    the xxx_set_vf_mac() or xxx_set_vf_vlan() will invoke efx_net_stop()
    and efx_net_open(), since netif_running() is false, the port will not
    start and keep port_enabled false, but selftest_work is scheduled
    in efx_net_open().

    If we remove the device before selftest_work run, the efx_stop_port()
    will not be called since the NIC is down, and then efx is freed,
    we will soon get a UAF in run_timer_softirq() like this:

    [ 1178.907941] ================================================================
    [ 1178.907948] BUG: KASAN: use-after-free in run_timer_softirq+0xdea/0xe90
    [ 1178.907950] Write of size 8 at addr ff11001f449cdc80 by task swapper/47/0
    [ 1178.907950]
    [ 1178.907953] CPU: 47 PID: 0 Comm: swapper/47 Kdump: loaded Tainted: G           O
--------- -t - 4.18.0 #1
    [ 1178.907954] Hardware name: SANGFOR X620G40/WI2HG-208T1061A, BIOS SPYH051032-U01
04/01/2022
    [ 1178.907955] Call Trace:
    [ 1178.907956]  <IRQ>
    [ 1178.907960]  dump_stack+0x71/0xab
    [ 1178.907963]  print_address_description+0x6b/0x290
    [ 1178.907965]  ? run_timer_softirq+0xdea/0xe90
    [ 1178.907967]  kasan_report+0x14a/0x2b0
    [ 1178.907968]  run_timer_softirq+0xdea/0xe90
    [ 1178.907971]  ? init_timer_key+0x170/0x170
    [ 1178.907973]  ? hrtimer_cancel+0x20/0x20
    [ 1178.907976]  ? sched_clock+0x5/0x10
    [ 1178.907978]  ? sched_clock_cpu+0x18/0x170
    [ 1178.907981]  __do_softirq+0x1c8/0x5fa
    [ 1178.907985]  irq_exit+0x213/0x240
    [ 1178.907987]  smp_apic_timer_interrupt+0xd0/0x330
    [ 1178.907989]  apic_timer_interrupt+0xf/0x20
    [ 1178.907990]  </IRQ>
    [ 1178.907991] RIP: 0010:mwait_idle+0xae/0x370

    If the NIC is not actually brought up, there is no need to schedule
    selftest_work, so let's move invoking efx_selftest_async_start()
    into efx_start_all(), and it will be canceled by broughting down.

    Fixes: dd40781e3a4e ("sfc: Run event/IRQ self-test asynchronously when interface is
brought up")
    Fixes: e340be923012 ("sfc: add ndo_set_vf_mac() function for EF10")
    Debugged-by: Huang Cun <huangcun@sangfor.com.cn>
    Cc: Donglin Peng <pengdonglin@sangfor.com.cn>
    Suggested-by: Martin Habets <habetsm.xilinx@gmail.com>
    Signed-off-by: Ding Hui <dinghui@sangfor.com.cn>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8c1c8155baf4616482f98767fee6b9fd2a83a77
Author: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
```

Date: Fri Apr 14 14:08:35 2023 +0800

    virtio_net: bugfix overflow inside xdp_linearize_page()

    [Upstream commit 853618d5886bf94812f31228091cd37d308230f7]

    Here we copy the data from the original buf to the new page. But we
    not check that it may be overflow.

    As long as the size received(including vnethdr) is greater than 3840
    (PAGE_SIZE -VIRTIO_XDP_HEADROOM). Then the memcpy will overflow.

    And this is completely possible, as long as the MTU is large, such
    as 4096. In our test environment, this will cause crash. Since crash is
    caused by the written memory, it is meaningless, so I do not include it.

    Fixes: 72979a6c3590 ("virtio_net: xdp, add slowpath case for non contiguous buffers")
    Signed-off-by: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
    Acked-by: Jason Wang <jasowang@redhat.com>
    Acked-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 420d014b19ff119e210ecc075ff611fe7844690c
Author: Gwangun Jung <exsociety@gmail.com>
Date: Thu Apr 13 19:35:54 2023 +0900

    net: sched: sch_qfq: prevent slab-out-of-bounds in qfq_activate_agg

    [Upstream commit 3037933448f60f9acb705997eae62013ecb81e0d]

    If the TCA_QFQ_LMAX value is not offered through nlattr, lmax is determined by the MTU
value of the network device.
    The MTU of the loopback device can be set up to 2^31-1.
    As a result, it is possible to have an lmax value that exceeds QFQ_MIN_LMAX.

    Due to the invalid lmax value, an index is generated that exceeds the QFQ_MAX_INDEX(=24)
value, causing out-of-bounds read/write errors.

    The following reports a oob access:

    [84.582666] BUG: KASAN: slab-out-of-bounds in qfq_activate_agg.constprop.0
(net/sched/sch_qfq.c:1027 net/sched/sch_qfq.c:1060 net/sched/sch_qfq.c:1313)
    [84.583267] Read of size 4 at addr ffff88810f676948 by task ping/301
    [84.583686]
    [84.583797] CPU: 3 PID: 301 Comm: ping Not tainted 6.3.0-rc5 #1
    [84.584164] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1
04/01/2014
    [84.584644] Call Trace:
    [84.584787] <TASK>
    [84.584906] dump_stack_lvl (lib/dump_stack.c:107 (discriminator 1))
    [84.585108] print_report (mm/kasan/report.c:320 mm/kasan/report.c:430)
    [84.585570] kasan_report (mm/kasan/report.c:538)
    [84.585988] qfq_activate_agg.constprop.0 (net/sched/sch_qfq.c:1027
net/sched/sch_qfq.c:1060 net/sched/sch_qfq.c:1313)
    [84.586599] qfq_enqueue (net/sched/sch_qfq.c:1255)
    [84.587607] dev_qdisc_enqueue (net/core/dev.c:3776)
    [84.587749] __dev_queue_xmit (./include/net/sch_generic.h:186 net/core/dev.c:3865
net/core/dev.c:4212)
    [84.588763] ip_finish_output2 (./include/net/neighbour.h:546
net/ipv4/ip_output.c:228)
    [84.589460] ip_output (net/ipv4/ip_output.c:430)
    [84.590132] ip_push_pending_frames (./include/net/dst.h:444 net/ipv4/ip_output.c:126
net/ipv4/ip_output.c:1586 net/ipv4/ip_output.c:1606)
    [84.590285] raw_sendmsg (net/ipv4/raw.c:649)

```
     [   84.591960] sock_sendmsg (net/socket.c:724 net/socket.c:747)
     [   84.592084] __sys_sendto (net/socket.c:2142)
     [   84.593306] __x64_sys_sendto (net/socket.c:2150)
     [   84.593779] do_syscall_64 (arch/x86/entry/common.c:50 arch/x86/entry/common.c:80)
     [   84.593902] entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120)
     [   84.594070] RIP: 0033:0x7fe568032066
     [   84.594192] Code: 0e 0d 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 41 89
  ca 64 8b 04 25 18 00 00 00 85 c09[ 84.594796] RSP: 002b:00007ffce388b4e8 EFLAGS: 00000246
  ORIG_RAX: 000000000000002c

     Code starting with the faulting instruction
     ===========================================
     [   84.595047] RAX: ffffffffffffffda RBX: 00007ffce388cc70 RCX: 00007fe568032066
     [   84.595281] RDX: 0000000000000040 RSI: 00005605fdad6d10 RDI: 0000000000000003
     [   84.595515] RBP: 00005605fdad6d10 R08: 00007ffce388eeec R09: 0000000000000010
     [   84.595749] R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000040
     [   84.595984] R13: 00007ffce388cc30 R14: 00007ffce388b4f0 R15: 0000001d00000001
     [   84.596218]  </TASK>
     [   84.596295]
     [   84.596351] Allocated by task 291:
     [   84.596467] kasan_save_stack (mm/kasan/common.c:46)
     [   84.596597] kasan_set_track (mm/kasan/common.c:52)
     [   84.596725] __kasan_kmalloc (mm/kasan/common.c:384)
     [   84.596852] __kmalloc_node (./include/linux/kasan.h:196 mm/slab_common.c:967
  mm/slab_common.c:974)
     [   84.596979] qdisc_alloc (./include/linux/slab.h:610 ./include/linux/slab.h:731
  net/sched/sch_generic.c:938)
     [   84.597100] qdisc_create (net/sched/sch_api.c:1244)
     [   84.597222] tc_modify_qdisc (net/sched/sch_api.c:1680)
     [   84.597357] rtnetlink_rcv_msg (net/core/rtnetlink.c:6174)
     [   84.597495] netlink_rcv_skb (net/netlink/af_netlink.c:2574)
     [   84.597627] netlink_unicast (net/netlink/af_netlink.c:1340
  net/netlink/af_netlink.c:1365)
     [   84.597759] netlink_sendmsg (net/netlink/af_netlink.c:1942)
     [   84.597891] sock_sendmsg (net/socket.c:724 net/socket.c:747)
     [   84.598016] ____sys_sendmsg (net/socket.c:2501)
     [   84.598147] ___sys_sendmsg (net/socket.c:2557)
     [   84.598275] __sys_sendmsg (./include/linux/file.h:31 net/socket.c:2586)
     [   84.598399] do_syscall_64 (arch/x86/entry/common.c:50 arch/x86/entry/common.c:80)
     [   84.598520] entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120)
     [   84.598688]
     [   84.598744] The buggy address belongs to the object at ffff88810f674000
     [   84.598744]  which belongs to the cache kmalloc-8k of size 8192
     [   84.599135] The buggy address is located 2664 bytes to the right of
     [   84.599135]  allocated 7904-byte region [ffff88810f674000, ffff88810f675ee0)
     [   84.599544]
     [   84.599598] The buggy address belongs to the physical page:
     [   84.599777] page:00000000e638567f refcount:1 mapcount:0 mapping:0000000000000000
  index:0x0 pfn:0x10f670
     [   84.600074] head:00000000e638567f order:3 entire_mapcount:0 nr_pages_mapped:0
  pincount:0
     [   84.600330] flags: 0x200000000010200(slab|head|node=0|zone=2)
     [   84.600517] raw: 0200000000010200 ffff888100043180 dead000000000122 0000000000000000
     [   84.600764] raw: 0000000000000000 0000000080020002 00000001ffffffff 0000000000000000
     [   84.601009] page dumped because: kasan: bad access detected
     [   84.601187]
     [   84.601241] Memory state around the buggy address:
     [   84.601396]  ffff88810f676800: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [   84.601620]  ffff88810f676880: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [   84.601845] >ffff88810f676900: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [   84.602069]                                                 ^
     [   84.602243]  ffff88810f676980: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [   84.602468]  ffff88810f676a00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [   84.602693] ==================================================================
     [   84.602924] Disabling lock debugging due to kernel taint
```

```
    Fixes: 3015f3d2a3cd ("pkt_sched: enable QFQ to support TSO/GSO")
    Reported-by: Gwangun Jung <exsociety@gmail.com>
    Signed-off-by: Gwangun Jung <exsociety@gmail.com>
    Acked-by: Jamal Hadi Salim<jhs@mojatatu.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bf5037b69cb1fc54b71177b3f41bb6bbe4efacce
Author: Cristian Ciocaltea <cristian.ciocaltea@collabora.com>
Date:   Thu Apr 6 20:18:01 2023 +0300

    regulator: fan53555: Fix wrong TCS_SLEW_MASK

    [ Upstream commit c5d5b55b3c1a314137a251efc1001dfd435c6242 ]

    The support for TCS4525 regulator has been introduced with a wrong
    ramp-rate mask, which has been defined as a logical expression instead
    of a bit shift operation.

    For clarity, fix it using GENMASK() macro.

    Fixes: 914df8faa7d6 ("regulator: fan53555: Add TCS4525 DCDC support")
    Signed-off-by: Cristian Ciocaltea <cristian.ciocaltea@collabora.com>
    Link: https://lore.kernel.org/r/20230406171806.948290-4-cristian.ciocaltea@collabora.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2e3947589eb3b9629bb1b4f0b982085273f7a3d1
Author: Cristian Ciocaltea <cristian.ciocaltea@collabora.com>
Date:   Thu Apr 6 20:18:00 2023 +0300

    regulator: fan53555: Explicitly include bits header

    [ Upstream commit 4fb9a5060f73627303bc531ceaab1b19d0a24aef ]

    Since commit f2a9eb975ab2 ("regulator: fan53555: Add support for
    FAN53526") the driver makes use of the BIT() macro, but relies on the
    bits header being implicitly included.

    Explicitly pull the header in to avoid potential build failures in some
    configurations.

    While here, reorder include directives alphabetically.

    Fixes: f2a9eb975ab2 ("regulator: fan53555: Add support for FAN53526")
    Signed-off-by: Cristian Ciocaltea <cristian.ciocaltea@collabora.com>
    Link: https://lore.kernel.org/r/20230406171806.948290-3-cristian.ciocaltea@collabora.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fe336a994900d63b9a96e49e8b73c278bd50fd0a
Author: Patrick Blass <patrickblass@mailbox.org>
Date:   Fri Mar 3 20:06:29 2023 +0100

    rust: str: fix requierments->requirements typo

    [ Upstream commit 88e8c2ec4ab84f9f05ed5af9693a3972baf386c4 ]

    Fix a trivial spelling error in the `rust/kernel/str.rs` file.

    Fixes: 247b365dc8dc ("rust: add `kernel` crate")
    Reported-by: Miguel Ojeda <ojeda@kernel.org>
    Link: https://github.com/Rust-for-Linux/linux/issues/978
    Signed-off-by: Patrick Blass <patrickblass@mailbox.org>
```

```
    Reviewed-by: Vincenzo Palazzo <vincenzopalazzodev@gmail.com>
    [Reworded slightly]
    Signed-off-by: Miguel Ojeda <ojeda@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 501d73e8ead774944d70ab382bf93909c482f05d
Author: Chen Aotian <chenaotian2@163.com>
Date:   Thu Apr 6 12:01:51 2023 +0800

    netfilter: nf_tables: Modify nla_memdup's flag to GFP_KERNEL_ACCOUNT

    [ Upstream commit af0acf22aea359e04412237d68787401f96bb583 ]

    For memory alloc that store user data from nla[NFTA_OBJ_USERDATA],
    use GFP_KERNEL_ACCOUNT is more suitable.

    Fixes: 33758c891479 ("memcg: enable accounting for nft objects")
    Signed-off-by: Chen Aotian <chenaotian2@163.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 22134b86de9c2afe28e1f406062cd93bdcac4149
Author: Florian Westphal <fw@strlen.de>
Date:   Mon Apr 3 13:54:37 2023 +0200

    netfilter: br_netfilter: fix recent physdev match breakage

    [ Upstream commit 94623f579ce338b5fa61b5acaa5beb8aa657fb9e ]

    Recent attempt to ensure PREROUTING hook is executed again when a
    decrypted ipsec packet received on a bridge passes through the network
    stack a second time broke the physdev match in INPUT hook.

    We can't discard the nf_bridge info strct from sabotage_in hook, as
    this is needed by the physdev match.

    Keep the struct around and handle this with another conditional instead.

    Fixes: 2b272bb558f1 ("netfilter: br_netfilter: disable sabotage_in hook after first
suppression")
    Reported-and-tested-by: Farid BENAMROUCHE <fariouche@yahoo.fr>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d59b498ae63f9788c3726fa9512c8d0d3ae342e4
Author: Peng Fan <peng.fan@nxp.com>
Date:   Tue Mar 28 14:19:05 2023 +0800

    arm64: dts: imx8mp-verdin: correct off-on-delay

    [ Upstream commit 02c447a0d79f0c966563e5095a017cbf9477ca6d ]

    The property should be off-on-delay-us, not off-on-delay

    Fixes: a39ed23bdf6e ("arm64: dts: freescale: add initial support for verdin imx8m plus")
    Signed-off-by: Peng Fan <peng.fan@nxp.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a8a51db3982ebb2b676553eea1e8dc155735a4ee
Author: Peng Fan <peng.fan@nxp.com>
Date:   Tue Mar 28 14:19:04 2023 +0800

    arm64: dts: imx8mm-verdin: correct off-on-delay
```

```
    [ Upstream commit 130c1f4306d56301216baaea68afdd909892c73f ]

    The property should be off-on-delay-us, not off-on-delay

    Fixes: 6a57f224f734 ("arm64: dts: freescale: add initial support for verdin imx8m mini")
    Signed-off-by: Peng Fan <peng.fan@nxp.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e0623941d3a1a6b3d154af6f4dae4cefded317f
Author: Peng Fan <peng.fan@nxp.com>
Date:   Mon Mar 27 18:03:21 2023 +0800

    arm64: dts: imx8mm-evk: correct pmic clock source

    [ Upstream commit 85af7ffd24da38e416a14bd6bf207154d94faa83 ]

    The osc_32k supports #clock-cells as 0, using an id is wrong, drop it.

    Fixes: a6a355ede574 ("arm64: dts: imx8mm-evk: Add 32.768 kHz clock to PMIC")
    Signed-off-by: Peng Fan <peng.fan@nxp.com>
    Reviewed-by: Marco Felsch <m.felsch@pengutronix.de>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8ab4d6642ad4cb5ea4dadb6e11ed38d6dbfb8575
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Mon Mar 27 14:29:48 2023 +0200

    arm64: dts: qcom: sc8280xp-pmics: fix pon compatible and registers

    [ Upstream commit ad8cd35c58ca3ec5e93f52a0124899627b98efb2 ]

    The pmk8280 PMIC PON peripheral is gen3 and uses two sets of registers;
    hlos and pbs.

    This specifically fixes the following error message during boot when the
    pbs registers are not defined:

            PON_PBS address missing, can't read HW debounce time

    Note that this also enables the spurious interrupt workaround introduced
    by commit 0b65118e6ba3 ("Input: pm8941-pwrkey - add software key press
    debouncing support") (which may or may not be needed).

    Fixes: ccd3517faf18 ("arm64: dts: qcom: sc8280xp: Add reference device")
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Reviewed-by: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Tested-by: Steev Klimaszewski <steev@kali.org> #Thinkpad X13s
    Reviewed-by: Konrad Dybcio <konrad.dybcio@linaro.org>
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Link: https://lore.kernel.org/r/20230327122948.4323-1-johan+linaro@kernel.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 52712ad57931e981d50df3717d84840f2d089576
Author: Marc Gonzalez <mgonzalez@freebox.fr>
Date:   Mon Mar 27 14:09:32 2023 +0200

    perf/amlogic: adjust register offsets

    [ Upstream commit f9d323e7c1724270d747657051099826744e91e7 ]

    Commit "perf/amlogic: resolve conflict between canvas & pmu"
    changed the base address.
```

```
    Fixes: 2016e2113d35 ("perf/amlogic: Add support for Amlogic meson G12 SoC DDR PMU
driver")
    Signed-off-by: Marc Gonzalez <mgonzalez@freebox.fr>
    Acked-by: Will Deacon <will@kernel.org>
    Reviewed-by: Neil Armstrong <neil.armstrong@linaro.org>
    Link: https://lore.kernel.org/r/20230327120932.2158389-4-mgonzalez@freebox.fr
    Signed-off-by: Neil Armstrong <neil.armstrong@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 69fb260403c680b9891487e6ea441933a47fc4b3
Author: Marc Gonzalez <mgonzalez@freebox.fr>
Date:   Mon Mar 27 14:09:31 2023 +0200

    arm64: dts: meson-g12-common: resolve conflict between canvas & pmu

    [ Upstream commit 33acea2049b5058b93d1dabb536b494f543f02a2 ]

    According to S905X2 Datasheet - Revision 07:

    DMC_MON area spans 0xff638080-0xff6380c0
    DDR_PLL area spans 0xff638c00-0xff638c34

    Round DDR_PLL area size up to 0x40

    Fixes: 90cf8e21016fa3 ("arm64: dts: meson: Add DDR PMU node")
    Signed-off-by: Marc Gonzalez <mgonzalez@freebox.fr>
    Reviewed-by: Neil Armstrong <neil.armstrong@linaro.org>
    Link: https://lore.kernel.org/r/20230327120932.2158389-3-mgonzalez@freebox.fr
    Signed-off-by: Neil Armstrong <neil.armstrong@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 225dc7aaf20dba66601b2c1432df6e415929cebc
Author: Marc Gonzalez <mgonzalez@freebox.fr>
Date:   Mon Mar 27 14:09:30 2023 +0200

    arm64: dts: meson-g12-common: specify full DMC range

    [ Upstream commit aec4353114a408b3a831a22ba34942d05943e462 ]

    According to S905X2 Datasheet - Revision 07:
    DRAM Memory Controller (DMC) register area spans ff638000-ff63a000.

    According to DeviceTree Specification - Release v0.4-rc1:
    simple-bus nodes do not require reg property.

    Fixes: 1499218c80c99a ("arm64: dts: move common G12A & G12B modes to meson-g12-
common.dtsi")
    Signed-off-by: Marc Gonzalez <mgonzalez@freebox.fr>
    Reviewed-by: Martin Blumenstingl <martin.blumenstingl@googlemail.com>
    Link: https://lore.kernel.org/r/20230327120932.2158389-2-mgonzalez@freebox.fr
    Signed-off-by: Neil Armstrong <neil.armstrong@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0c59d8ea7c21ae70feb4bb76f9170dcf8fe06a74
Author: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
Date:   Fri Mar 24 05:16:51 2023 +0300

    arm64: dts: qcom: ipq8074-hk10: enable QMP device, not the PHY node

    [ Upstream commit 1dc40551f206d20b7e46ea7dd538dcdd928451c6 ]

    Correct PCIe PHY enablement to refer the QMP device nodes rather than
    PHY device nodes. QMP nodes have 'status = "disabled"' property in the
    ipq8074.dtsi, while PHY nodes do not correspond to the actual device and
```

```
    do not have the status property.

    Fixes: 1ed34da63a37 ("arm64: dts: qcom: Add board support for HK10")
    Signed-off-by: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Link: https://lore.kernel.org/r/20230324021651.1799969-2-dmitry.baryshkov@linaro.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d5bd20fd2b730cafe7e68367eac07abeca1ae156
Author: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
Date:   Fri Mar 24 05:16:50 2023 +0300

    arm64: dts: qcom: ipq8074-hk01: enable QMP device, not the PHY node

    [ Upstream commit 72630ba422b70ea0874fc90d526353cf71c72488 ]

    Correct PCIe PHY enablement to refer the QMP device nodes rather than
    PHY device nodes. QMP nodes have 'status = "disabled"' property in the
    ipq8074.dtsi, while PHY nodes do not correspond to the actual device and
    do not have the status property.

    Fixes: e8a7fdc505bb ("arm64: dts: ipq8074: qcom: Re-arrange dts nodes based on address")
    Signed-off-by: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Link: https://lore.kernel.org/r/20230324021651.1799969-1-dmitry.baryshkov@linaro.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 14ab8b7829d6ff7aa1ba643e65eeb4f055f0098a
Author: Dan Johansen <strit@manjaro.org>
Date:   Sat Mar 4 17:41:35 2023 +0100

    arm64: dts: rockchip: Lower sd speed on rk3566-soquartz

    [ Upstream commit 5912b647bd0732ae8c78a6e5b259c82efd177d93 ]

    Just like the Quartz64 Model B the previously stated speed of sdr-104
    in soquartz is too high for the hardware to reliably communicate with
    some fast SD cards.
    Especially on some carrierboards.

    Lower this to sd-uhs-sdr50 to fix this.

    Fixes: 5859b5a9c3ac ("arm64: dts: rockchip: add SoQuartz CM4IO dts")
    Signed-off-by: Dan Johansen <strit@manjaro.org>
    Acked-by: Peter Geis <pgwipeout@gmail.com>
    Link: https://lore.kernel.org/r/20230304164135.28430-1-strit@manjaro.org
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5f52e3692bf329bff94e4cdd6bb798f9086c2254
Author: Jianqun Xu <jay.xu@rock-chips.com>
Date:   Wed Feb 8 17:14:11 2023 +0800

    ARM: dts: rockchip: fix a typo error for rk3288 spdif node

    [ Upstream commit 02c84f91adb9a64b75ec97d772675c02a3e65ed7 ]

    Fix the address in the spdif node name.

    Fixes: 874e568e500a ("ARM: dts: rockchip: Add SPDIF transceiver for RK3288")
    Signed-off-by: Jianqun Xu <jay.xu@rock-chips.com>
    Reviewed-by: Sjoerd Simons <sjoerd@collabora.com>
    Link: https://lore.kernel.org/r/20230208091411.1603142-1-jay.xu@rock-chips.com
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```