```
commit 1463281b5efd2a55223e4d39e78fbc2e2a5a192f
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Jun 23 14:44:11 2021 +0200

    Linux 5.12.13

    Link: https://lore.kernel.org/r/20210621154921.212599475@linuxfoundation.org
    Tested-by: Florian Fainelli <f.fainelli@gmail.com>
    Tested-by: Jason Self <jason@bluehome.net>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Rudi Heitbaum <rudi@heitbaum.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fa8c413e6b74ae5d12daf911c73238c5bdacd8e6
Author: Peter Chen <peter.chen@kernel.org>
Date:   Tue Jun 8 18:56:56 2021 +0800

    usb: dwc3: core: fix kernel panic when do reboot

    commit 4bf584a03eec674975ee9fe36c8583d9d470dab1 upstream.

    When do system reboot, it calls dwc3_shutdown and the whole debugfs
    for dwc3 has removed first, when the gadget tries to do deinit, and
    remove debugfs for its endpoints, it meets NULL pointer dereference
    issue when call debugfs_lookup. Fix it by removing the whole dwc3
    debugfs later than dwc3_drd_exit.

    [ 2924.958838] Unable to handle kernel NULL pointer dereference at virtual address
0000000000000002
    ....
    [ 2925.030994] pstate: 60000005 (nZCv daif -PAN -UAO -TCO BTYPE=--)
    [ 2925.037005] pc : inode_permission+0x2c/0x198
    [ 2925.041281] lr : lookup_one_len_common+0xb0/0xf8
    [ 2925.045903] sp : ffff80001276ba70
    [ 2925.049218] x29: ffff80001276ba70 x28: ffff0000c01f0000 x27: 0000000000000000
    [ 2925.056364] x26: ffff800011791e70 x25: 0000000000000008 x24: dead000000000100
    [ 2925.063510] x23: dead000000000122 x22: 0000000000000000 x21: 0000000000000001
    [ 2925.070652] x20: ffff8000122c6188 x19: 0000000000000000 x18: 0000000000000000
    [ 2925.077797] x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000004
    [ 2925.084943] x14: ffffffffffffffff x13: 0000000000000000 x12: 0000000000000030
    [ 2925.092087] x11: 0101010101010101 x10: 7f7f7f7f7f7f7f7f x9 : ffff8000102b2420
    [ 2925.099232] x8 : 7f7f7f7f7f7f7f7f x7 : feff73746e2f6f64 x6 : 0000000000008080
    [ 2925.106378] x5 : 61c8864680b583eb x4 : 209e6ec2d263dbb7 x3 : 000074756f307065
    [ 2925.113523] x2 : 0000000000000001 x1 : 0000000000000000 x0 : ffff8000122c6188
    [ 2925.120671] Call trace:
    [ 2925.123119]  inode_permission+0x2c/0x198
    [ 2925.127042]  lookup_one_len_common+0xb0/0xf8
    [ 2925.131315]  lookup_one_len_unlocked+0x34/0xb0
    [ 2925.135764]  lookup_positive_unlocked+0x14/0x50
    [ 2925.140296]  debugfs_lookup+0x68/0xa0
    [ 2925.143964]  dwc3_gadget_free_endpoints+0x84/0xb0
    [ 2925.148675]  dwc3_gadget_exit+0x28/0x78
    [ 2925.152518]  dwc3_drd_exit+0x100/0x1f8
    [ 2925.156267]  dwc3_remove+0x11c/0x120
    [ 2925.159851]  dwc3_shutdown+0x14/0x20
    [ 2925.163432]  platform_shutdown+0x28/0x38
    [ 2925.167360]  device_shutdown+0x15c/0x378
    [ 2925.171291]  kernel_restart_prepare+0x3c/0x48
    [ 2925.175650]  kernel_restart+0x1c/0x68
    [ 2925.179316]  __do_sys_reboot+0x218/0x240
    [ 2925.183247]  __arm64_sys_reboot+0x28/0x30
```

```
     [ 2925.187262]  invoke_syscall+0x48/0x100
     [ 2925.191017]  el0_svc_common.constprop.0+0x48/0xc8
     [ 2925.195726]  do_el0_svc+0x28/0x88
     [ 2925.199045]  el0_svc+0x20/0x30
     [ 2925.202104]  el0_sync_handler+0xa8/0xb0
     [ 2925.205942]  el0_sync+0x148/0x180
     [ 2925.209270] Code: a9025bf5 2a0203f5 121f0056 370802b5 (79400660)
     [ 2925.215372] ---[ end trace 124254d8e485a58b ]---
     [ 2925.220012] Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b
     [ 2925.227676] Kernel Offset: disabled
     [ 2925.231164] CPU features: 0x00001001,20000846
     [ 2925.235521] Memory Limit: none
     [ 2925.238580] ---[ end Kernel panic - not syncing: Attempted to kill init!
  exitcode=0x0000000b ]---

     Fixes: 8d396bb0a5b6 ("usb: dwc3: debugfs: Add and remove endpoint dirs dynamically")
     Cc: Jack Pham <jackp@codeaurora.org>
     Tested-by: Jack Pham <jackp@codeaurora.org>
     Signed-off-by: Peter Chen <peter.chen@kernel.org>
     Link: https://lore.kernel.org/r/20210608105656.10795-1-peter.chen@kernel.org
     (cherry picked from commit 2a042767814bd0edf2619f06fecd374e266ea068)
     Link: https://lore.kernel.org/r/20210615080847.GA10432@jackp-linux.qualcomm.com
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit afd8b0d091d5b4febe2d0ac3b7735c1826329302
  Author: Jack Pham <jackp@codeaurora.org>
  Date:   Sat May 29 12:29:32 2021 -0700

     usb: dwc3: debugfs: Add and remove endpoint dirs dynamically

     commit 8d396bb0a5b62b326f6be7594d8bd46b088296bd upstream.

     The DWC3 DebugFS directory and files are currently created once
     during probe.  This includes creation of subdirectories for each
     of the gadget's endpoints.  This works fine for peripheral-only
     controllers, as dwc3_core_init_mode() calls dwc3_gadget_init()
     just prior to calling dwc3_debugfs_init().

     However, for dual-role controllers, dwc3_core_init_mode() will
     instead call dwc3_drd_init() which is problematic in a few ways.
     First, the initial state must be determined, then dwc3_set_mode()
     will have to schedule drd_work and by then dwc3_debugfs_init()
     could have already been invoked.  Even if the initial mode is
     peripheral, dwc3_gadget_init() happens after the DebugFS files
     are created, and worse so if the initial state is host and the
     controller switches to peripheral much later.  And secondly,
     even if the gadget endpoints' debug entries were successfully
     created, if the controller exits peripheral mode, its dwc3_eps
     are freed so the debug files would now hold stale references.

     So it is best if the DebugFS endpoint entries are created and
     removed dynamically at the same time the underlying dwc3_eps are.
     Do this by calling dwc3_debugfs_create_endpoint_dir() as each
     endpoint is created, and conversely remove the DebugFS entry when
     the endpoint is freed.

     Fixes: 41ce1456e1db ("usb: dwc3: core: make dwc3_set_mode() work properly")
     Cc: stable <stable@vger.kernel.org>
     Reviewed-by: Peter Chen <peter.chen@kernel.org>
     Signed-off-by: Jack Pham <jackp@codeaurora.org>
     Link: https://lore.kernel.org/r/20210529192932.22912-1-jackp@codeaurora.org
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit c4aedcd7026b32565f700b026a9bafdeb1685083
  Author: Arnaldo Carvalho de Melo <acme@redhat.com>
```

```
Date:   Sat Jun 19 10:09:08 2021 -0300

    perf beauty: Update copy of linux/socket.h with the kernel sources

    commit ef83f9efe8461b8fd71eb60b53dbb6a5dd7b39e9 upstream.

    To pick the changes in:

      ea6932d70e223e02 ("net: make get_net_ns return error if NET_NS is disabled")

    That don't result in any changes in the tables generated from that
    header.

    This silences this perf build warning:

      Warning: Kernel ABI header at 'tools/perf/trace/beauty/include/linux/socket.h' differs
from latest version at 'include/linux/socket.h'
      diff -u tools/perf/trace/beauty/include/linux/socket.h include/linux/socket.h

    Cc: Changbin Du <changbin.du@intel.com>
    Cc: David S. Miller <davem@davemloft.net>
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 37699aef8dc6efcece81848a209846dad812cc38
Author: Arnaldo Carvalho de Melo <acme@redhat.com>
Date:   Sat Jun 19 10:15:22 2021 -0300

    tools headers UAPI: Sync linux/in.h copy with the kernel sources

    commit 1792a59eab9593de2eae36c40c5a22d70f52c026 upstream.

    To pick the changes in:

      321827477360934d ("icmp: don't send out ICMP messages with a source address of
0.0.0.0")

    That don't result in any change in tooling, as INADDR_ are not used to
    generate id->string tables used by 'perf trace'.

    This addresses this build warning:

      Warning: Kernel ABI header at 'tools/include/uapi/linux/in.h' differs from latest
version at 'include/uapi/linux/in.h'
      diff -u tools/include/uapi/linux/in.h include/uapi/linux/in.h

    Cc: David S. Miller <davem@davemloft.net>
    Cc: Toke Høiland-Jørgensen <toke@redhat.com>
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a5bbae600f013a70c3532b84e8f6b8b3b43c96df
Author: Fugang Duan <fugang.duan@nxp.com>
Date:   Wed Jun 16 17:14:25 2021 +0800

    net: fec_ptp: add clock rate zero check

    commit cb3cefe3f3f8af27c6076ef7d1f00350f502055d upstream.

    Add clock rate zero check to fix coverity issue of "divide by 0".

    Fixes: commit 85bd1798b24a ("net: fec: fix spin_lock dead lock")
    Signed-off-by: Fugang Duan <fugang.duan@nxp.com>
    Signed-off-by: Joakim Zhang <qiangqing.zhang@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
```

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

commit 97bf4dc3e1a3f154d08450292d059ee8baf097f3
Author: Joakim Zhang <qiangqing.zhang@nxp.com>
Date:   Wed Jun 16 17:10:24 2021 +0800

    net: stmmac: disable clocks in stmmac_remove_config_dt()

    commit 8f269102baf788aecfcbbc6313b6bceb54c9b990 upstream.

    Platform drivers may call stmmac_probe_config_dt() to parse dt, could
    call stmmac_remove_config_dt() in error handing after dt parsed, so need
    disable clocks in stmmac_remove_config_dt().

    Go through all platforms drivers which use stmmac_probe_config_dt(),
    none of them disable clocks manually, so it's safe to disable them in
    stmmac_remove_config_dt().

    Fixes: commit d2ed0a7755fe ("net: ethernet: stmmac: fix of-node and fixed-link-phydev
leaks")
    Signed-off-by: Joakim Zhang <qiangqing.zhang@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bcc0a8a25dc68c231d1ae75bb526061df7fc0ff6
Author: Andrew Morton <akpm@linux-foundation.org>
Date:   Tue Jun 15 18:23:39 2021 -0700

    mm/slub.c: include swab.h

    commit 1b3865d016815cbd69a1879ca1c8a8901fda1072 upstream.

    Fixes build with CONFIG_SLAB_FREELIST_HARDENED=y.

    Hopefully.  But it's the right thing to do anwyay.

    Fixes: 1ad53d9fa3f61 ("slub: improve bit diffusion for freelist ptr obfuscation")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=213417
    Reported-by: <vannguye@cisco.com>
    Acked-by: Kees Cook <keescook@chromium.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ce6e8bee7a3883e8008b30f5887dbb426aac6a35
Author: Kees Cook <keescook@chromium.org>
Date:   Tue Jun 15 18:23:26 2021 -0700

    mm/slub: actually fix freelist pointer vs redzoning

    commit e41a49fadbc80b60b48d3c095d9e2ee7ef7c9a8e upstream.

    It turns out that SLUB redzoning ("slub_debug=Z") checks from
    s->object_size rather than from s->inuse (which is normally bumped to
    make room for the freelist pointer), so a cache created with an object
    size less than 24 would have the freelist pointer written beyond
    s->object_size, causing the redzone to be corrupted by the freelist
    pointer.  This was very visible with "slub_debug=ZF":

      BUG test (Tainted: G    B            ): Right Redzone overwritten
      -----------------------------------------------------------------------------

      INFO: 0xffff957ead1c05de-0xffff957ead1c05df @offset=1502. First byte 0x1a instead of
0xbb
```

```
       INFO: Slab 0xffffef3950b47000 objects=170 used=170 fp=0x0000000000000000
  flags=0x8000000000000200
       INFO: Object 0xffff957ead1c05d8 @offset=1496 fp=0xffff957ead1c0620

       Redzone  (____ptrval____): bb bb bb bb bb bb bb bb               ........
       Object   (____ptrval____): 00 00 00 00 00 f6 f4 a5               ........
       Redzone  (____ptrval____): 40 1d e8 1a aa                        @....
       Padding  (____ptrval____): 00 00 00 00 00 00 00 00               ........
```

Adjust the offset to stay within s->object_size.

(Note that no caches of in this size range are known to exist in the kernel currently.)

Link: https://lkml.kernel.org/r/20210608183955.280836-4-keescook@chromium.org
Link: https://lore.kernel.org/linux-mm/20200807160627.GA1420741@elver.google.com/
Link: https://lore.kernel.org/lkml/0f7dd7b2-7496-5e2d-9488-2ec9f8e90441@suse.cz/Fixes: 89b83f282d8b (slub: avoid redzone when choosing freepointer location)
Link: https://lore.kernel.org/lkml/CANpmjNOwZ5VpKQn+SYWovTkFB4VsT-RPwyENBmaK0dLcpqStkA@mail.gmail.com
Signed-off-by: Kees Cook <keescook@chromium.org>
Reported-by: Marco Elver <elver@google.com>
Reported-by: "Lin, Zhenpeng" <zplin@psu.edu>
Tested-by: Marco Elver <elver@google.com>
Acked-by: Vlastimil Babka <vbabka@suse.cz>
Cc: Christoph Lameter <cl@linux.com>
Cc: David Rientjes <rientjes@google.com>
Cc: Joonsoo Kim <iamjoonsoo.kim@lge.com>
Cc: Pekka Enberg <penberg@kernel.org>
Cc: Roman Gushchin <guro@fb.com>
Cc: <stable@vger.kernel.org>
Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cf990cfae36bea4727aed2133087d059fd11310c
Author: Kees Cook <keescook@chromium.org>
Date:   Tue Jun 15 18:23:22 2021 -0700

mm/slub: fix redzoning for small allocations

commit 74c1d3e081533825f2611e46edea1fcdc0701985 upstream.

The redzone area for SLUB exists between s->object_size and s->inuse (which is at least the word-aligned object_size).  If a cache were created with an object_size smaller than sizeof(void *), the in-object stored freelist pointer would overwrite the redzone (e.g.  with boot param "slub_debug=ZF"):

```
       BUG test (Tainted: G    B            ): Right Redzone overwritten
       -----------------------------------------------------------------------------

       INFO: 0xffff957ead1c05de-0xffff957ead1c05df @offset=1502. First byte 0x1a instead of 0xbb
       INFO: Slab 0xffffef3950b47000 objects=170 used=170 fp=0x0000000000000000
  flags=0x8000000000000200
       INFO: Object 0xffff957ead1c05d8 @offset=1496 fp=0xffff957ead1c0620

       Redzone  (____ptrval____): bb bb bb bb bb bb bb bb    ........
       Object   (____ptrval____): f6 f4 a5 40 1d e8          ...@..
       Redzone  (____ptrval____): 1a aa                      ..
       Padding  (____ptrval____): 00 00 00 00 00 00 00 00    ........
```

Store the freelist pointer out of line when object_size is smaller than sizeof(void *) and redzoning is enabled.

```
    Additionally remove the "smaller than sizeof(void *)" check under
    CONFIG_DEBUG_VM in kmem_cache_sanity_check() as it is now redundant:
    SLAB and SLOB both handle small sizes.

    (Note that no caches within this size range are known to exist in the
    kernel currently.)

    Link: https://lkml.kernel.org/r/20210608183955.280836-3-keescook@chromium.org
    Fixes: 81819f0fc828 ("SLUB core")
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Acked-by: Vlastimil Babka <vbabka@suse.cz>
    Cc: Christoph Lameter <cl@linux.com>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Joonsoo Kim <iamjoonsoo.kim@lge.com>
    Cc: "Lin, Zhenpeng" <zplin@psu.edu>
    Cc: Marco Elver <elver@google.com>
    Cc: Pekka Enberg <penberg@kernel.org>
    Cc: Roman Gushchin <guro@fb.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ca28a428cf1a055d7d733d5b06fe054168ee6205
Author: Kees Cook <keescook@chromium.org>
Date:   Tue Jun 15 18:23:19 2021 -0700

    mm/slub: clarify verification reporting

    commit 8669dbab2ae56085c128894b181c2aa50f97e368 upstream.

    Patch series "Actually fix freelist pointer vs redzoning", v4.

    This fixes redzoning vs the freelist pointer (both for middle-position
    and very small caches).  Both are "theoretical" fixes, in that I see no
    evidence of such small-sized caches actually be used in the kernel, but
    that's no reason to let the bugs continue to exist, especially since
    people doing local development keep tripping over it.  :)

    This patch (of 3):

    Instead of repeating "Redzone" and "Poison", clarify which sides of
    those zones got tripped.  Additionally fix column alignment in the
    trailer.

    Before:

      BUG test (Tainted: G    B            ): Redzone overwritten
      ...
      Redzone (____ptrval____): bb bb bb bb bb bb bb bb      ........
      Object (____ptrval____): f6 f4 a5 40 1d e8            ...@..
      Redzone (____ptrval____): 1a aa                        ..
      Padding (____ptrval____): 00 00 00 00 00 00 00 00      ........

    After:

      BUG test (Tainted: G    B            ): Right Redzone overwritten
      ...
      Redzone  (____ptrval____): bb bb bb bb bb bb bb bb      ........
      Object   (____ptrval____): f6 f4 a5 40 1d e8            ...@..
      Redzone  (____ptrval____): 1a aa                        ..
      Padding  (____ptrval____): 00 00 00 00 00 00 00 00      ........

    The earlier commits that slowly resulted in the "Before" reporting were:
```

```
       d86bd1bece6f ("mm/slub: support left redzone")
       ffc79d288000 ("slub: use print_hex_dump")
       2492268472e7 ("SLUB: change error reporting format to follow lockdep loosely")

    Link: https://lkml.kernel.org/r/20210608183955.280836-1-keescook@chromium.org
    Link: https://lkml.kernel.org/r/20210608183955.280836-2-keescook@chromium.org
    Link: https://lore.kernel.org/lkml/cfdb11d7-fb8e-e578-c939-f7f5fb69a6bd@suse.cz/
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Acked-by: Vlastimil Babka <vbabka@suse.cz>
    Cc: Marco Elver <elver@google.com>
    Cc: "Lin, Zhenpeng" <zplin@psu.edu>
    Cc: Christoph Lameter <cl@linux.com>
    Cc: Pekka Enberg <penberg@kernel.org>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Joonsoo Kim <iamjoonsoo.kim@lge.com>
    Cc: Roman Gushchin <guro@fb.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3af098f31a82c8ac3dfbb895b3c32ad63888a6d9
Author: Mike Kravetz <mike.kravetz@oracle.com>
Date:   Tue Jun 15 18:23:29 2021 -0700

    mm/hugetlb: expand restore_reserve_on_error functionality

    commit 846be08578edb81f02bc8534577e6c367ef34f41 upstream.

    The routine restore_reserve_on_error is called to restore reservation
    information when an error occurs after page allocation.  The routine
    alloc_huge_page modifies the mapping reserve map and potentially the
    reserve count during allocation.  If code calling alloc_huge_page
    encounters an error after allocation and needs to free the page, the
    reservation information needs to be adjusted.

    Currently, restore_reserve_on_error only takes action on pages for which
    the reserve count was adjusted(HPageRestoreReserve flag).  There is
    nothing wrong with these adjustments.  However, alloc_huge_page ALWAYS
    modifies the reserve map during allocation even if the reserve count is
    not adjusted.  This can cause issues as observed during development of
    this patch [1].

    One specific series of operations causing an issue is:

     - Create a shared hugetlb mapping
       Reservations for all pages created by default

     - Fault in a page in the mapping
       Reservation exists so reservation count is decremented

     - Punch a hole in the file/mapping at index previously faulted
       Reservation and any associated pages will be removed

     - Allocate a page to fill the hole
       No reservation entry, so reserve count unmodified
       Reservation entry added to map by alloc_huge_page

     - Error after allocation and before instantiating the page
       Reservation entry remains in map

     - Allocate a page to fill the hole
       Reservation entry exists, so decrement reservation count
```

```
    This will cause a reservation count underflow as the reservation count
    was decremented twice for the same index.

    A user would observe a very large number for HugePages_Rsvd in
    /proc/meminfo.  This would also likely cause subsequent allocations of
    hugetlb pages to fail as it would 'appear' that all pages are reserved.

    This sequence of operations is unlikely to happen, however they were
    easily reproduced and observed using hacked up code as described in [1].

    Address the issue by having the routine restore_reserve_on_error take
    action on pages where HPageRestoreReserve is not set.  In this case, we
    need to remove any reserve map entry created by alloc_huge_page.  A new
    helper routine vma_del_reservation assists with this operation.

    There are three callers of alloc_huge_page which do not currently call
    restore_reserve_on error before freeing a page on error paths.  Add
    those missing calls.

    [1] https://lore.kernel.org/linux-mm/20210528005029.88088-1-almasrymina@google.com/

    Link: https://lkml.kernel.org/r/20210607204510.22617-1-mike.kravetz@oracle.com
    Fixes: 96b96a96ddee ("mm/hugetlb: fix huge page reservation leak in private mapping
error paths"
    Signed-off-by: Mike Kravetz <mike.kravetz@oracle.com>
    Reviewed-by: Mina Almasry <almasrymina@google.com>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: Muchun Song <songmuchun@bytedance.com>
    Cc: Michal Hocko <mhocko@suse.com>
    Cc: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d3369218f92274997583af5a7aaa8acfc4f79ad6
Author: Peter Xu <peterx@redhat.com>
Date:   Tue Jun 15 18:23:16 2021 -0700

    mm/swap: fix pte_same_as_swp() not removing uffd-wp bit when compare

    commit 099dd6878b9b12d6bbfa6bf29ce0c8ddd38f6901 upstream.

    I found it by pure code review, that pte_same_as_swp() of unuse_vma()
    didn't take uffd-wp bit into account when comparing ptes.
    pte_same_as_swp() returning false negative could cause failure to
    swapoff swap ptes that was wr-protected by userfaultfd.

    Link: https://lkml.kernel.org/r/20210603180546.9083-1-peterx@redhat.com
    Fixes: f45ec5ff16a7 ("userfaultfd: wp: support swap and page migration")
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Acked-by: Hugh Dickins <hughd@google.com>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: <stable@vger.kernel.org>    [5.7+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7db3a9e6e465e1715b2318b2cfe3c0be07c48f91
Author: Naoya Horiguchi <naoya.horiguchi@nec.com>
Date:   Tue Jun 15 18:23:13 2021 -0700

    mm,hwpoison: fix race with hugetlb page allocation
```

```
    commit 25182f05ffed0b45602438693e4eed5d7f3ebadd upstream.

    When hugetlb page fault (under overcommitting situation) and
    memory_failure() race, VM_BUG_ON_PAGE() is triggered by the following
    race:

        CPU0:                           CPU1:

                                        gather_surplus_pages()
                                          page = alloc_surplus_huge_page()
        memory_failure_hugetlb()
          get_hwpoison_page(page)
            __get_hwpoison_page(page)
              get_page_unless_zero(page)
                                          zero = put_page_testzero(page)
                                          VM_BUG_ON_PAGE(!zero, page)
                                          enqueue_huge_page(h, page)
          put_page(page)

    __get_hwpoison_page() only checks the page refcount before taking an
    additional one for memory error handling, which is not enough because
    there's a time window where compound pages have non-zero refcount during
    hugetlb page initialization.

    So make __get_hwpoison_page() check page status a bit more for hugetlb
    pages with get_hwpoison_huge_page().  Checking hugetlb-specific flags
    under hugetlb_lock makes sure that the hugetlb page is not transitive.
    It's notable that another new function, HWPoisonHandlable(), is helpful
    to prevent a race against other transitive page states (like a generic
    compound page just before PageHuge becomes true).

    Link: https://lkml.kernel.org/r/20210603233632.2964832-2-nao.horiguchi@gmail.com
    Fixes: ead07f6a867b ("mm/memory-failure: introduce get_hwpoison_page() for consistent
refcount handling")
    Signed-off-by: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Reported-by: Muchun Song <songmuchun@bytedance.com>
    Acked-by: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Oscar Salvador <osalvador@suse.de>
    Cc: Michal Hocko <mhocko@suse.com>
    Cc: Tony Luck <tony.luck@intel.com>
    Cc: <stable@vger.kernel.org>    [5.12+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 25053a8404ba17ca48f5553d487afc1882e9f56c
Author: Nikolay Aleksandrov <nikolay@nvidia.com>
Date:   Thu Jun 10 15:04:11 2021 +0300

    net: bridge: fix vlan tunnel dst refcnt when egressing

    commit cfc579f9d89af4ada58c69b03bcaa4887840f3b3 upstream.

    The egress tunnel code uses dst_clone() and directly sets the result
    which is wrong because the entry might have 0 refcnt or be already deleted,
    causing number of problems. It also triggers the WARN_ON() in dst_hold()[1]
    when a refcnt couldn't be taken. Fix it by using dst_hold_safe() and
    checking if a reference was actually taken before setting the dst.

    [1] dmesg WARN_ON log and following refcnt errors
     WARNING: CPU: 5 PID: 38 at include/net/dst.h:230
br_handle_egress_vlan_tunnel+0x10b/0x134 [bridge]
     Modules linked in: 8021q garp mrp bridge stp llc bonding ipv6 virtio_net
     CPU: 5 PID: 38 Comm: ksoftirqd/5 Kdump: loaded Tainted: G        W         5.13.0-rc3+
#360
```

```
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-1.fc33 04/01/2014
      RIP: 0010:br_handle_egress_vlan_tunnel+0x10b/0x134 [bridge]
      Code: e8 85 bc 01 e1 45 84 f6 74 90 45 31 f6 85 db 48 c7 c7 a0 02 19 a0 41 0f 94 c6 31
 c9 31 d2 44 89 f6 e8 64 bc 01 e1 85 db 75 02 <0f> 0b 31 c9 31 d2 44 89 f6 48 c7 c7 70 02 19
 a0 e8 4b bc 01 e1 49
      RSP: 0018:ffff8881003d39e8 EFLAGS: 00010246
      RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000
      RDX: 0000000000000000 RSI: 0000000000000001 RDI: ffffffffa01902a0
      RBP: ffff8881040c6700 R08: 0000000000000000 R09: 0000000000000001
      R10: 2ce93d0054fe0d00 R11: 54fe0d00000e0000 R12: ffff888109515000
      R13: 0000000000000000 R14: 0000000000000001 R15: 0000000000000401
      FS:  0000000000000000(0000) GS:ffff88822bf40000(0000) knlGS:0000000000000000
      CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      CR2: 00007f42ba70f030 CR3: 0000000109926000 CR4: 00000000000006e0
      Call Trace:
       br_handle_vlan+0xbc/0xca [bridge]
       __br_forward+0x23/0x164 [bridge]
       deliver_clone+0x41/0x48 [bridge]
       br_handle_frame_finish+0x36f/0x3aa [bridge]
       ? skb_dst+0x2e/0x38 [bridge]
       ? br_handle_ingress_vlan_tunnel+0x3e/0x1c8 [bridge]
       ? br_handle_frame_finish+0x3aa/0x3aa [bridge]
       br_handle_frame+0x2c3/0x377 [bridge]
       ? __skb_pull+0x33/0x51
       ? vlan_do_receive+0x4f/0x36a
       ? br_handle_frame_finish+0x3aa/0x3aa [bridge]
       __netif_receive_skb_core+0x539/0x7c6
       ? __list_del_entry_valid+0x16e/0x1c2
       __netif_receive_skb_list_core+0x6d/0xd6
       netif_receive_skb_list_internal+0x1d9/0x1fa
       gro_normal_list+0x22/0x3e
       dev_gro_receive+0x55b/0x600
       ? detach_buf_split+0x58/0x140
       napi_gro_receive+0x94/0x12e
       virtnet_poll+0x15d/0x315 [virtio_net]
       __napi_poll+0x2c/0x1c9
       net_rx_action+0xe6/0x1fb
       __do_softirq+0x115/0x2d8
       run_ksoftirqd+0x18/0x20
       smpboot_thread_fn+0x183/0x19c
       ? smpboot_unregister_percpu_thread+0x66/0x66
       kthread+0x10a/0x10f
       ? kthread_mod_delayed_work+0xb6/0xb6
       ret_from_fork+0x22/0x30
      ---[ end trace 49f61b07f775fd2b ]---
      dst_release: dst:00000000c02d677a refcnt:-1
      dst_release underflow

    Cc: stable@vger.kernel.org
    Fixes: 11538d039ac6 ("bridge: vlan dst_metadata hooks in ingress and egress paths")
    Signed-off-by: Nikolay Aleksandrov <nikolay@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit abb02e05cb1c0a30dd873a29f33bc092067dc35d
 Author: Nikolay Aleksandrov <nikolay@nvidia.com>
 Date:   Thu Jun 10 15:04:10 2021 +0300

    net: bridge: fix vlan tunnel dst null pointer dereference

    commit 58e2071742e38f29f051b709a5cca014ba51166f upstream.

    This patch fixes a tunnel_dst null pointer dereference due to lockless
    access in the tunnel egress path. When deleting a vlan tunnel the
    tunnel_dst pointer is set to NULL without waiting a grace period (i.e.
```

```
    while it's still usable) and packets egressing are dereferencing it
    without checking. Use READ/WRITE_ONCE to annotate the lockless use of
    tunnel_id, use RCU for accessing tunnel_dst and make sure it is read
    only once and checked in the egress path. The dst is already properly RCU
    protected so we don't need to do anything fancy than to make sure
    tunnel_id and tunnel_dst are read only once and checked in the egress path.

    Cc: stable@vger.kernel.org
    Fixes: 11538d039ac6 ("bridge: vlan dst_metadata hooks in ingress and egress paths")
    Signed-off-by: Nikolay Aleksandrov <nikolay@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b6982493ed2dee412ccae062ccf7cf50fbe5a6a8
Author: Esben Haabendal <esben@geanix.com>
Date:   Fri Jun 18 12:52:33 2021 +0200

    net: ll_temac: Fix TX BD buffer overwrite

    commit c364df2489b8ef2f5e3159b1dff1ff1fdb16040d upstream.

    Just as the initial check, we need to ensure num_frag+1 buffers available,
    as that is the number of buffers we are going to use.

    This fixes a buffer overflow, which might be seen during heavy network
    load. Complete lockup of TEMAC was reproducible within about 10 minutes of
    a particular load.

    Fixes: 84823ff80f74 ("net: ll_temac: Fix race condition causing TX hang")
    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Esben Haabendal <esben@geanix.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e8afe05bd359ebe12a61dbdc94c06c00ea3e8d4b
Author: Esben Haabendal <esben@geanix.com>
Date:   Fri Jun 18 12:52:23 2021 +0200

    net: ll_temac: Make sure to free skb when it is completely used

    commit 6aa32217a9a446275440ee8724b1ecaf1838df47 upstream.

    With the skb pointer piggy-backed on the TX BD, we have a simple and
    efficient way to free the skb buffer when the frame has been transmitted.
    But in order to avoid freeing the skb while there are still fragments from
    the skb in use, we need to piggy-back on the TX BD of the skb, not the
    first.

    Without this, we are doing use-after-free on the DMA side, when the first
    BD of a multi TX BD packet is seen as completed in xmit_done, and the
    remaining BDs are still being processed.

    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Esben Haabendal <esben@geanix.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ee85fdbcea82317c74bd5257de3946abb1fd1c2f
Author: Yifan Zhang <yifan1.zhang@amd.com>
Date:   Thu Jun 10 09:55:01 2021 +0800

    drm/amdgpu/gfx9: fix the doorbell missing when in CGPG issue.

    commit 4cbbe34807938e6e494e535a68d5ff64edac3f20 upstream.
```

```
    If GC has entered CGPG, ringing doorbell > first page doesn't wakeup GC.
    Enlarge CP_MEC_DOORBELL_RANGE_UPPER to workaround this issue.

    Signed-off-by: Yifan Zhang <yifan1.zhang@amd.com>
    Reviewed-by: Felix Kuehling <Felix.Kuehling@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@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 df6cd610bbe52fc78bd77fec67850f0f3497679d
Author: Yifan Zhang <yifan1.zhang@amd.com>
Date:   Thu Jun 10 10:10:07 2021 +0800

    drm/amdgpu/gfx10: enlarge CP_MEC_DOORBELL_RANGE_UPPER to cover full doorbell.

    commit 1c0b0efd148d5b24c4932ddb3fa03c8edd6097b3 upstream.

    If GC has entered CGPG, ringing doorbell > first page doesn't wakeup GC.
    Enlarge CP_MEC_DOORBELL_RANGE_UPPER to workaround this issue.

    Signed-off-by: Yifan Zhang <yifan1.zhang@amd.com>
    Reviewed-by: Felix Kuehling <Felix.Kuehling@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@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 6fd67a68154bc65149157750149dfe91ca225bf6
Author: Avraham Stern <avraham.stern@intel.com>
Date:   Fri Jun 18 13:41:31 2021 +0300

    cfg80211: avoid double free of PMSR request

    commit 0288e5e16a2e18f0b7e61a2b70d9037fc6e4abeb upstream.

    If cfg80211_pmsr_process_abort() moves all the PMSR requests that
    need to be freed into a local list before aborting and freeing them.
    As a result, it is possible that cfg80211_pmsr_complete() will run in
    parallel and free the same PMSR request.

    Fix it by freeing the request in cfg80211_pmsr_complete() only if it
    is still in the original pmsr list.

    Cc: stable@vger.kernel.org
    Fixes: 9bb7e0f24e7e ("cfg80211: add peer measurement with FTM initiator API")
    Signed-off-by: Avraham Stern <avraham.stern@intel.com>
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Link:
https://lore.kernel.org/r/iwlwifi.20210618133832.1fbef57e269a.I00294bebdb0680b892f8d1d5c871f
d9dbe785a5e@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 34e2e11e2282ad2e619b3aede5196affc94cc537
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Fri Jun 18 13:41:29 2021 +0300

    cfg80211: make certificate generation more robust

    commit b5642479b0f7168fe16d156913533fe65ab4f8d5 upstream.

    If all net/wireless/certs/*.hex files are deleted, the build
    will hang at this point since the 'cat' command will have no
    arguments. Do "echo | cat - ..." so that even if the "..."
```

```
    part is empty, the whole thing won't hang.

    Cc: stable@vger.kernel.org
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Link:
https://lore.kernel.org/r/iwlwifi.20210618133832.c989056c3664.Ic3b77531d00b30b26dcd69c64e55a
e2f60c3f31e@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2ffac7f3c81a3a659a3b852852982db3e994f7a0
Author: Felix Fietkau <nbd@nbd.name>
Date:   Thu Jun 17 12:38:54 2021 +0200

    mac80211: minstrel_ht: fix sample time check

    commit 1236af327af476731aa548dfcbbefb1a3ec6726a upstream.

    We need to skip sampling if the next sample time is after jiffies, not before.
    This patch fixes an issue where in some cases only very little sampling (or none
    at all) is performed, leading to really bad data rates

    Fixes: 80d55154b2f8 ("mac80211: minstrel_ht: significantly redesign the rate probing
strategy")
    Cc: stable@vger.kernel.org
    Signed-off-by: Felix Fietkau <nbd@nbd.name>
    Link: https://lore.kernel.org/r/20210617103854.61875-1-nbd@nbd.name
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 077ad15b8b72a7461bef8783108130a8bea3f146
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Tue Jun 8 11:32:30 2021 +0200

    mac80211: move interface shutdown out of wiphy lock

    commit f5baf287f5da5641099ad5c809b3b4ebfc08506d upstream.

    When reconfiguration fails, we shut down everything, but we
    cannot call cfg80211_shutdown_all_interfaces() with the wiphy
    mutex held. Since cfg80211 now calls it on resume errors, we
    only need to do likewise for where we call reconfig (whether
    directly or indirectly), but not under the wiphy lock.

    Cc: stable@vger.kernel.org
    Fixes: 2fe8ef106238 ("cfg80211: change netdev registration/unregistration semantics")
    Link:
https://lore.kernel.org/r/20210608113226.78233c80f548.Iecc104aceb89f0568f50e9670a9cb191a1c88
87b@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db40ccfec26d9d21ee83076a7035d90cf88707a8
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Tue Jun 8 11:32:29 2021 +0200

    cfg80211: shut down interfaces on failed resume

    commit 65bec836da8394b1d56bdec2c478dcac21cf12a4 upstream.

    If resume fails, we should shut down all interfaces as the
    hardware is probably dead. This was/is already done now in
    mac80211, but we need to change that due to locking issues,
    so move it here and do it without the wiphy lock held.
```

```
    Cc: stable@vger.kernel.org
    Fixes: 2fe8ef106238 ("cfg80211: change netdev registration/unregistration semantics")
    Link:
https://lore.kernel.org/r/20210608113226.d564ca69de7c.I2e3c3e5d410b72a4f63bade4fb075df041b3d
92f@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 721b9c56b271a615f7351b9ec5d387a0cba24c3d
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Tue Jun 8 11:32:28 2021 +0200

    cfg80211: fix phy80211 symlink creation

    commit 43076c1e074359f11c85d7d1b85ede1bbb8ee6b9 upstream.

    When I moved around the code here, I neglected that we could still
    call register_netdev() or similar without the wiphy mutex held,
    which then calls cfg80211_register_wdev() - that's also done from
    cfg80211_register_netdevice(), but the phy80211 symlink creation
    was only there. Now, the symlink isn't needed for a *pure* wdev,
    but a netdev not registered via cfg80211_register_wdev() should
    still have the symlink, so move the creation to the right place.

    Cc: stable@vger.kernel.org
    Fixes: 2fe8ef106238 ("cfg80211: change netdev registration/unregistration semantics")
    Link:
https://lore.kernel.org/r/20210608113226.a5dc4c1e488c.Ia42fe663cefe47b0883af78c98f284c5555bb
e5d@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5ea9123f46313e4793785193a5230726d7f946f7
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Tue Jun 8 11:32:27 2021 +0200

    mac80211: fix 'reset' debugfs locking

    commit adaed1b9daf5a045be71e923e04b5069d2bee664 upstream.

    cfg80211 now calls suspend/resume with the wiphy lock
    held, and while there's a problem with that needing
    to be fixed, we should do the same in debugfs.

    Cc: stable@vger.kernel.org
    Fixes: a05829a7222e ("cfg80211: avoid holding the RTNL when calling the driver")
    Link:
https://lore.kernel.org/r/20210608113226.14020430e449.I78e19db0a55a8295a376e15ac4cf77dbb4c6f
b51@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7b1b88232e40396ee1d8b19d42a050cda0a5aebb
Author: Mathy Vanhoef <Mathy.Vanhoef@kuleuven.be>
Date:   Sun May 30 15:32:26 2021 +0200

    mac80211: Fix NULL ptr deref for injected rate info

    commit bddc0c411a45d3718ac535a070f349be8eca8d48 upstream.

    The commit cb17ed29a7a5 ("mac80211: parse radiotap header when selecting Tx
    queue") moved the code to validate the radiotap header from
    ieee80211_monitor_start_xmit to ieee80211_parse_tx_radiotap. This made is
    possible to share more code with the new Tx queue selection code for
```

```
    injected frames. But at the same time, it now required the call of
    ieee80211_parse_tx_radiotap at the beginning of functions which wanted to
    handle the radiotap header. And this broke the rate parser for radiotap
    header parser.

    The radiotap parser for rates is operating most of the time only on the
    data in the actual radiotap header. But for the 802.11a/b/g rates, it must
    also know the selected band from the chandef information. But this
    information is only written to the ieee80211_tx_info at the end of the
    ieee80211_monitor_start_xmit - long after ieee80211_parse_tx_radiotap was
    already called. The info->band information was therefore always 0
    (NL80211_BAND_2GHZ) when the parser code tried to access it.

    For a 5GHz only device, injecting a frame with 802.11a rates would cause a
    NULL pointer dereference because local->hw.wiphy->bands[NL80211_BAND_2GHZ]
    would most likely have been NULL when the radiotap parser searched for the
    correct rate index of the driver.

    Cc: stable@vger.kernel.org
    Reported-by: Ben Greear <greearb@candelatech.com>
    Fixes: cb17ed29a7a5 ("mac80211: parse radiotap header when selecting Tx queue")
    Signed-off-by: Mathy Vanhoef <Mathy.Vanhoef@kuleuven.be>
    [sven@narfation.org: added commit message]
    Signed-off-by: Sven Eckelmann <sven@narfation.org>
    Link: https://lore.kernel.org/r/20210530133226.40587-1-sven@narfation.org
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8043903fcb72f545c52e3ec74d6fd82ef79ce7c5
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Mon May 17 16:03:23 2021 +0200

    mac80211: fix deadlock in AP/VLAN handling

    commit d5befb224edbe53056c2c18999d630dafb4a08b9 upstream.

    Syzbot reports that when you have AP_VLAN interfaces that are up
    and close the AP interface they belong to, we get a deadlock. No
    surprise - since we dev_close() them with the wiphy mutex held,
    which goes back into the netdev notifier in cfg80211 and tries to
    acquire the wiphy mutex there.

    To fix this, we need to do two things:
     1) prevent changing iftype while AP_VLANs are up, we can't
        easily fix this case since cfg80211 already calls us with
        the wiphy mutex held, but change_interface() is relatively
        rare in drivers anyway, so changing iftype isn't used much
        (and userspace has to fall back to down/change/up anyway)
     2) pull the dev_close() loop over VLANs out of the wiphy mutex
        section in the normal stop case

    Cc: stable@vger.kernel.org
    Reported-by: syzbot+452ea4fbbef700ff0a56@syzkaller.appspotmail.com
    Fixes: a05829a7222e ("cfg80211: avoid holding the RTNL when calling the driver")
    Link:
https://lore.kernel.org/r/20210517160322.9b8f356c0222.I392cb0e2fa5a1a94cf2e637555d702c7e512c
1ff@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 789a43de7876db145107b30cd13efc5c45b862df
Author: Bumyong Lee <bumyong.lee@samsung.com>
Date:   Fri May 7 15:36:47 2021 +0900

    dmaengine: pl330: fix wrong usage of spinlock flags in dma_cyclc
```

```
    commit 4ad5dd2d7876d79507a20f026507d1a93b8fff10 upstream.

    flags varible which is the input parameter of pl330_prep_dma_cyclic()
    should not be used by spinlock_irq[save/restore] function.

    Signed-off-by: Jongho Park <jongho7.park@samsung.com>
    Signed-off-by: Bumyong Lee <bumyong.lee@samsung.com>
    Signed-off-by: Chanho Park <chanho61.park@samsung.com>
    Link: https://lore.kernel.org/r/20210507063647.111209-1-chanho61.park@samsung.com
    Fixes: f6f2421c0a1c ("dmaengine: pl330: Merge dma_pl330_dmac and pl330_dmac structs")
    Cc: stable@vger.kernel.org
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 70fd2a63fc1cce3ceaa1f19c42010e8290807c73
Author: Pingfan Liu <kernelfans@gmail.com>
Date:   Tue Jun 15 18:23:36 2021 -0700

    crash_core, vmcoreinfo: append 'SECTION_SIZE_BITS' to vmcoreinfo

    commit 4f5aecdff25f59fb5ea456d5152a913906ecf287 upstream.

    As mentioned in kernel commit 1d50e5d0c505 ("crash_core, vmcoreinfo:
    Append 'MAX_PHYSMEM_BITS' to vmcoreinfo"), SECTION_SIZE_BITS in the
    formula:

        #define SECTIONS_SHIFT    (MAX_PHYSMEM_BITS - SECTION_SIZE_BITS)

    Besides SECTIONS_SHIFT, SECTION_SIZE_BITS is also used to calculate
    PAGES_PER_SECTION in makedumpfile just like kernel.

    Unfortunately, this arch-dependent macro SECTION_SIZE_BITS changes, e.g.
    recently in kernel commit f0b13ee23241 ("arm64/sparsemem: reduce
    SECTION_SIZE_BITS").  But user space wants a stable interface to get
    this info.  Such info is impossible to be deduced from a crashdump
    vmcore.  Hence append SECTION_SIZE_BITS to vmcoreinfo.

    Link: https://lkml.kernel.org/r/20210608103359.84907-1-kernelfans@gmail.com
    Link: http://lists.infradead.org/pipermail/kexec/2021-June/022676.html
    Signed-off-by: Pingfan Liu <kernelfans@gmail.com>
    Acked-by: Baoquan He <bhe@redhat.com>
    Cc: Bhupesh Sharma <bhupesh.sharma@linaro.org>
    Cc: Kazuhito Hagio <k-hagio@ab.jp.nec.com>
    Cc: Dave Young <dyoung@redhat.com>
    Cc: Boris Petkov <bp@alien8.de>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: James Morse <james.morse@arm.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Will Deacon <will@kernel.org>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Michael Ellerman <mpe@ellerman.id.au>
    Cc: Paul Mackerras <paulus@samba.org>
    Cc: Benjamin Herrenschmidt <benh@kernel.crashing.org>
    Cc: Dave Anderson <anderson@redhat.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f5644a1cf61712d11424cacea01ed5a69d3e8e20
Author: Tor Vic <torvic9@mailbox.org>
Date:   Sun Jun 13 13:07:49 2021 +0000
```

```
    Makefile: lto: Pass -warn-stack-size only on LLD < 13.0.0

    commit 0236526d76b87c1dc2cbe3eb31ae29be5b0ca151 upstream.

    Since LLVM commit fc018eb, the '-warn-stack-size' flag has been dropped
    [1], leading to the following error message when building with Clang-13
    and LLD-13:

        ld.lld: error: -plugin-opt=-: ld.lld: Unknown command line argument
        '-warn-stack-size=2048'.  Try: 'ld.lld --help'
        ld.lld: Did you mean '--asan-stack=2048'?

    In the same way as with commit 2398ce80152a ("x86, lto: Pass
    -stack-alignment only on LLD < 13.0.0") , make '-warn-stack-size'
    conditional on LLD < 13.0.0.

    [1] https://reviews.llvm.org/D103928

    Fixes: 24845dcb170e ("Makefile: LTO: have linker check -Wframe-larger-than")
    Cc: stable@vger.kernel.org
    Link: https://github.com/ClangBuiltLinux/linux/issues/1377
    Signed-off-by: Tor Vic <torvic9@mailbox.org>
    Reviewed-by: Nathan Chancellor <nathan@kernel.org>
    Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/7631bab7-a8ab-f884-ab54-f4198976125c@mailbox.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 74c3c34a04bc226f77b9e515aa067072cff44e52
Author: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
Date:   Thu Jun 17 13:55:06 2021 -0400

    powerpc/perf: Fix crash in perf_instruction_pointer() when ppmu is not set

    commit 60b7ed54a41b550d50caf7f2418db4a7e75b5bdc upstream.

    On systems without any specific PMU driver support registered, running
    perf record causes Oops.

    The relevant portion from call trace:

      BUG: Kernel NULL pointer dereference on read at 0x00000040
      Faulting instruction address: 0xc0021f0c
      Oops: Kernel access of bad area, sig: 11 [#1]
      BE PAGE_SIZE=4K PREEMPT CMPCPRO
      SAF3000 DIE NOTIFICATION
      CPU: 0 PID: 442 Comm: null_syscall Not tainted 5.13.0-rc6-s3k-dev-01645-g7649ee3d2957
#5164
      NIP:  c0021f0c LR: c00e8ad8 CTR: c00d8a5c
      NIP perf_instruction_pointer+0x10/0x60
      LR  perf_prepare_sample+0x344/0x674
      Call Trace:
        perf_prepare_sample+0x7c/0x674 (unreliable)
        perf_event_output_forward+0x3c/0x94
        __perf_event_overflow+0x74/0x14c
        perf_swevent_hrtimer+0xf8/0x170
        __hrtimer_run_queues.constprop.0+0x160/0x318
        hrtimer_interrupt+0x148/0x3b0
        timer_interrupt+0xc4/0x22c
        Decrementer_virt+0xb8/0xbc

    During perf record session, perf_instruction_pointer() is called to
    capture the sample IP. This function in core-book3s accesses
    ppmu->flags. If a platform specific PMU driver is not registered, ppmu
    is set to NULL and accessing its members results in a crash. Fix this
```

```
    crash by checking if ppmu is set.

    Fixes: 2ca13a4cc56c ("powerpc/perf: Use regs->nip when SIAR is zero")
    Cc: stable@vger.kernel.org # v5.11+
    Reported-by: Christophe Leroy <christophe.leroy@csgroup.eu>
    Signed-off-by: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Tested-by: Christophe Leroy <christophe.leroy@csgroup.eu>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/1623952506-1431-1-git-send-email-
atrajeev@linux.vnet.ibm.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6d427e1730dad6087a61ce53111bcd842bef1921
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Wed Jun 9 21:18:00 2021 +0200

    x86/fpu: Reset state for all signal restore failures

    commit efa165504943f2128d50f63de0c02faf6dcceb0d upstream.

    If access_ok() or fpregs_soft_set() fails in __fpu__restore_sig() then the
    function just returns but does not clear the FPU state as it does for all
    other fatal failures.

    Clear the FPU state for these failures as well.

    Fixes: 72a671ced66d ("x86, fpu: Unify signal handling code paths for x86 and x86_64
kernels")
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/87mtryyhhz.ffs@nanos.tec.linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 002665dcba4bbec8c82f0aeb4bd3f44334ed2c14
Author: Andy Lutomirski <luto@kernel.org>
Date:   Tue Jun 8 16:36:19 2021 +0200

    x86/fpu: Invalidate FPU state after a failed XRSTOR from a user buffer

    commit d8778e393afa421f1f117471144f8ce6deb6953a upstream.

    Both Intel and AMD consider it to be architecturally valid for XRSTOR to
    fail with #PF but nonetheless change the register state.  The actual
    conditions under which this might occur are unclear [1], but it seems
    plausible that this might be triggered if one sibling thread unmaps a page
    and invalidates the shared TLB while another sibling thread is executing
    XRSTOR on the page in question.

    __fpu__restore_sig() can execute XRSTOR while the hardware registers
    are preserved on behalf of a different victim task (using the
    fpu_fpregs_owner_ctx mechanism), and, in theory, XRSTOR could fail but
    modify the registers.

    If this happens, then there is a window in which __fpu__restore_sig()
    could schedule out and the victim task could schedule back in without
    reloading its own FPU registers. This would result in part of the FPU
    state that __fpu__restore_sig() was attempting to load leaking into the
    victim task's user-visible state.

    Invalidate preserved FPU registers on XRSTOR failure to prevent this
    situation from corrupting any state.

    [1] Frequent readers of the errata lists might imagine "complex
        microarchitectural conditions".
```

```
    Fixes: 1d731e731c4c ("x86/fpu: Add a fastpath to __fpu__restore_sig()")
    Signed-off-by: Andy Lutomirski <luto@kernel.org>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Acked-by: Dave Hansen <dave.hansen@linux.intel.com>
    Acked-by: Rik van Riel <riel@surriel.com>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20210608144345.758116583@linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ec25ea1f3f05d6f8ee51d1277efea986eafd4f2a
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Tue Jun 8 16:36:18 2021 +0200

    x86/fpu: Prevent state corruption in __fpu__restore_sig()

    commit 484cea4f362e1eeb5c869abbfb5f90eae6421b38 upstream.

    The non-compacted slowpath uses __copy_from_user() and copies the entire
    user buffer into the kernel buffer, verbatim.  This means that the kernel
    buffer may now contain entirely invalid state on which XRSTOR will #GP.
    validate_user_xstate_header() can detect some of that corruption, but that
    leaves the onus on callers to clear the buffer.

    Prior to XSAVES support, it was possible just to reinitialize the buffer,
    completely, but with supervisor states that is not longer possible as the
    buffer clearing code split got it backwards. Fixing that is possible but
    not corrupting the state in the first place is more robust.

    Avoid corruption of the kernel XSAVE buffer by using copy_user_to_xstate()
    which validates the XSAVE header contents before copying the actual states
    to the kernel. copy_user_to_xstate() was previously only called for
    compacted-format kernel buffers, but it works for both compacted and
    non-compacted forms.

    Using it for the non-compacted form is slower because of multiple
    __copy_from_user() operations, but that cost is less important than robust
    code in an already slow path.

    [ Changelog polished by Dave Hansen ]

    Fixes: b860eb8dce59 ("x86/fpu/xstate: Define new functions for clearing fpregs and
xstates")
    Reported-by: syzbot+2067e764dbcd10721e2e@syzkaller.appspotmail.com
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Reviewed-by: Borislav Petkov <bp@suse.de>
    Acked-by: Dave Hansen <dave.hansen@linux.intel.com>
    Acked-by: Rik van Riel <riel@surriel.com>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20210608144345.611833074@linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 811148810edab2f35a5bf233787b951d9d2e5f30
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Tue Jun 8 16:36:21 2021 +0200

    x86/pkru: Write hardware init value to PKRU when xstate is init

    commit 510b80a6a0f1a0d114c6e33bcea64747d127973c upstream.

    When user space brings PKRU into init state, then the kernel handling is
    broken:
```

```
       T1 user space
          xsave(state)
          state.header.xfeatures &= ~XFEATURE_MASK_PKRU;
          xrstor(state)

       T1 -> kernel
          schedule()
            XSAVE(S) -> T1->xsave.header.xfeatures[PKRU] == 0
            T1->flags |= TIF_NEED_FPU_LOAD;

            wrpkru();

          schedule()
            ...
            pk = get_xsave_addr(&T1->fpu->state.xsave, XFEATURE_PKRU);
            if (pk)
              wrpkru(pk->pkru);
            else
              wrpkru(DEFAULT_PKRU);

     Because the xfeatures bit is 0 and therefore the value in the xsave
     storage is not valid, get_xsave_addr() returns NULL and switch_to()
     writes the default PKRU. -> FAIL #1!

     So that wrecks any copy_to/from_user() on the way back to user space
     which hits memory which is protected by the default PKRU value.

     Assumed that this does not fail (pure luck) then T1 goes back to user
     space and because TIF_NEED_FPU_LOAD is set it ends up in

       switch_fpu_return()
           __fpregs_load_activate()
             if (!fpregs_state_valid()) {
              load_XSTATE_from_task();
             }

     But if nothing touched the FPU between T1 scheduling out and back in,
     then the fpregs_state is still valid which means switch_fpu_return()
     does nothing and just clears TIF_NEED_FPU_LOAD. Back to user space with
     DEFAULT_PKRU loaded. -> FAIL #2!

     The fix is simple: if get_xsave_addr() returns NULL then set the
     PKRU value to 0 instead of the restrictive default PKRU value in
     init_pkru_value.

      [ bp: Massage in minor nitpicks from folks. ]

     Fixes: 0cecca9d03c9 ("x86/fpu: Eager switch PKRU state")
     Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
     Signed-off-by: Borislav Petkov <bp@suse.de>
     Acked-by: Dave Hansen <dave.hansen@linux.intel.com>
     Acked-by: Rik van Riel <riel@surriel.com>
     Tested-by: Babu Moger <babu.moger@amd.com>
     Cc: stable@vger.kernel.org
     Link: https://lkml.kernel.org/r/20210608144346.045616965@linutronix.de
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit b7a05aba39f733ec337c5b952e112dd2dc4fc404
  Author: Tom Lendacky <thomas.lendacky@amd.com>
  Date:   Tue Jun 8 11:54:33 2021 +0200

     x86/ioremap: Map EFI-reserved memory as encrypted for SEV

     commit 8d651ee9c71bb12fc0c8eb2786b66cbe5aa3e43b upstream.
```

```
    Some drivers require memory that is marked as EFI boot services
    data. In order for this memory to not be re-used by the kernel
    after ExitBootServices(), efi_mem_reserve() is used to preserve it
    by inserting a new EFI memory descriptor and marking it with the
    EFI_MEMORY_RUNTIME attribute.

    Under SEV, memory marked with the EFI_MEMORY_RUNTIME attribute needs to
    be mapped encrypted by Linux, otherwise the kernel might crash at boot
    like below:

      EFI Variables Facility v0.08 2004-May-17
      general protection fault, probably for non-canonical address 0x3597688770a868b2: 0000
[#1] SMP NOPTI
      CPU: 13 PID: 1 Comm: swapper/0 Not tainted 5.12.4-2-default #1 openSUSE Tumbleweed
      Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 0.0.0 02/06/2015
      RIP: 0010:efi_mokvar_entry_next
      [...]
      Call Trace:
       efi_mokvar_sysfs_init
       ? efi_mokvar_table_init
       do_one_initcall
       ? __kmalloc
       kernel_init_freeable
       ? rest_init
       kernel_init
       ret_from_fork

    Expand the __ioremap_check_other() function to additionally check for
    this other type of boot data reserved at runtime and indicate that it
    should be mapped encrypted for an SEV guest.

     [ bp: Massage commit message. ]

    Fixes: 58c909022a5a ("efi: Support for MOK variable config table")
    Reported-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Tom Lendacky <thomas.lendacky@amd.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Tested-by: Joerg Roedel <jroedel@suse.de>
    Cc: <stable@vger.kernel.org> # 5.10+
    Link: https://lkml.kernel.org/r/20210608095439.12668-2-joro@8bytes.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e85c3112ddb4fc20a473bb4ac3b41684805c8c77
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Tue Jun 8 16:36:20 2021 +0200

    x86/process: Check PF_KTHREAD and not current->mm for kernel threads

    commit 12f7764ac61200e32c916f038bdc08f884b0b604 upstream.

    switch_fpu_finish() checks current->mm as indicator for kernel threads.
    That's wrong because kernel threads can temporarily use a mm of a user
    process via kthread_use_mm().

    Check the task flags for PF_KTHREAD instead.

    Fixes: 0cecca9d03c9 ("x86/fpu: Eager switch PKRU state")
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Acked-by: Dave Hansen <dave.hansen@linux.intel.com>
    Acked-by: Rik van Riel <riel@surriel.com>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20210608144345.912645927@linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 14225dfbbada065c806829ae246faf3c3302b057
Author: Fan Du <fan.du@intel.com>
Date:   Thu Jun 17 12:46:57 2021 -0700

    x86/mm: Avoid truncating memblocks for SGX memory

    commit 28e5e44aa3f4e0e0370864ed008fb5e2d85f4dc8 upstream.

    tl;dr:

    Several SGX users reported seeing the following message on NUMA systems:

      sgx: [Firmware Bug]: Unable to map EPC section to online node. Fallback to the NUMA
node 0.

    This turned out to be the memblock code mistakenly throwing away SGX
    memory.

    === Full Changelog ===

    The 'max_pfn' variable represents the highest known RAM address.  It can
    be used, for instance, to quickly determine for which physical addresses
    there is mem_map[] space allocated.  The numa_meminfo code makes an
    effort to throw out ("trim") all memory blocks which are above 'max_pfn'.

    SGX memory is not considered RAM (it is marked as "Reserved" in the
    e820) and is not taken into account by max_pfn. Despite this, SGX memory
    areas have NUMA affinity and are enumerated in the ACPI SRAT table. The
    existing SGX code uses the numa_meminfo mechanism to look up the NUMA
    affinity for its memory areas.

    In cases where SGX memory was above max_pfn (usually just the one EPC
    section in the last highest NUMA node), the numa_memblock is truncated
    at 'max_pfn', which is below the SGX memory.  When the SGX code tries to
    look up the affinity of this memory, it fails and produces an error message:

      sgx: [Firmware Bug]: Unable to map EPC section to online node. Fallback to the NUMA
node 0.

    and assigns the memory to NUMA node 0.

    Instead of silently truncating the memory block at 'max_pfn' and
    dropping the SGX memory, add the truncated portion to
    'numa_reserved_meminfo'.  This allows the SGX code to later determine
    the NUMA affinity of its 'Reserved' area.

    Before, numa_meminfo looked like this (from 'crash'):

      blk = { start =          0x0, end = 0x2080000000, nid = 0x0 }
            { start = 0x2080000000, end = 0x4000000000, nid = 0x1 }

    numa_reserved_meminfo is empty.

    With this, numa_meminfo looks like this:

      blk = { start =          0x0, end = 0x2080000000, nid = 0x0 }
            { start = 0x2080000000, end = 0x4000000000, nid = 0x1 }

    and numa_reserved_meminfo has an entry for node 1's SGX memory:

      blk =  { start = 0x4000000000, end = 0x4080000000, nid = 0x1 }

     [ daveh: completely rewrote/reworked changelog ]
```

```
    Fixes: 5d30f92e7631 ("x86/NUMA: Provide a range-to-target_node lookup facility")
    Reported-by: Reinette Chatre <reinette.chatre@intel.com>
    Signed-off-by: Fan Du <fan.du@intel.com>
    Signed-off-by: Dave Hansen <dave.hansen@intel.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Reviewed-by: Jarkko Sakkinen <jarkko@kernel.org>
    Reviewed-by: Dan Williams <dan.j.williams@intel.com>
    Reviewed-by: Dave Hansen <dave.hansen@intel.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lkml.kernel.org/r/20210617194657.0A99CB22@viggo.jf.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f99607667fd1d37f7a0d9285d5e2d99f2b91c5fb
Author: Vineet Gupta <vgupta@synopsys.com>
Date:   Tue Jun 8 19:39:25 2021 -0700

    ARCv2: save ABI registers across signal handling

    commit 96f1b00138cb8f04c742c82d0a7c460b2202e887 upstream.

    ARCv2 has some configuration dependent registers (r30, r58, r59) which
    could be targetted by the compiler. To keep the ABI stable, these were
    unconditionally part of the glibc ABI
    (sysdeps/unix/sysv/linux/arc/sys/ucontext.h:mcontext_t) however we
    missed populating them (by saving/restoring them across signal
    handling).

    This patch fixes the issue by
     - adding arcv2 ABI regs to kernel struct sigcontext
     - populating them during signal handling

    Change to struct sigcontext might seem like a glibc ABI change (although
    it primarily uses ucontext_t:mcontext_t) but the fact is
     - it has only been extended (existing fields are not touched)
     - the old sigcontext was ABI incomplete to begin with anyways

    Fixes: https://github.com/foss-for-synopsys-dwc-arc-processors/linux/issues/53
    Cc: <stable@vger.kernel.org>
    Tested-by: kernel test robot <lkp@intel.com>
    Reported-by: Vladimir Isaev <isaev@synopsys.com>
    Signed-off-by: Vineet Gupta <vgupta@synopsys.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6c800b5a60aff90a25b96f14f8f65ff37db6dad3
Author: Harald Freudenberger <freude@linux.ibm.com>
Date:   Tue Jun 1 08:27:29 2021 +0200

    s390/ap: Fix hanging ioctl caused by wrong msg counter

    commit e73a99f3287a740a07d6618e9470f4d6cb217da8 upstream.

    When a AP queue is switched to soft offline, all pending
    requests are purged out of the pending requests list and
    'received' by the upper layer like zcrypt device drivers.
    This is also done for requests which are already enqueued
    into the firmware queue. A request in a firmware queue
    may eventually produce an response message, but there is
    no waiting process any more. However, the response was
    counted with the queue_counter and as this counter was
    reset to 0 with the offline switch, the pending response
    caused the queue_counter to get negative. The next request
    increased this counter to 0 (instead of 1) which caused
    the ap code to assume there is nothing to receive and so
    the response for this valid request was never tried to
    fetch from the firmware queue.
```

```
    This all caused a queue to not work properly after a
    switch offline/online and in the end processes to hang
    forever when trying to send a crypto request after an
    queue offline/online switch cicle.

    Fixed by a) making sure the counter does not drop below 0
    and b) on a successful enqueue of a message has at least
    a value of 1.

    Additionally a warning is emitted, when a reply can't get
    assigned to a waiting process. This may be normal operation
    (process had timeout or has been killed) but may give a
    hint that something unexpected happened (like this odd
    behavior described above).

    Signed-off-by: Harald Freudenberger <freude@linux.ibm.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 99de738e5b16760982b9d746015d5d6c8ca28feb
Author: Alexander Gordeev <agordeev@linux.ibm.com>
Date:   Mon May 17 08:18:11 2021 +0200

    s390/mcck: fix calculation of SIE critical section size

    commit 5bcbe3285fb614c49db6b238253f7daff7e66312 upstream.

    The size of SIE critical section is calculated wrongly
    as result of a missed subtraction in commit 0b0ed657fe00
    ("s390: remove critical section cleanup from entry.S")

    Fixes: 0b0ed657fe00 ("s390: remove critical section cleanup from entry.S")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Alexander Gordeev <agordeev@linux.ibm.com>
    Reviewed-by: Christian Borntraeger <borntraeger@de.ibm.com>
    Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a4edc506abd5ad0f46d1aa9f7cdf20efaee0618c
Author: Wanpeng Li <wanpengli@tencent.com>
Date:   Thu Jun 10 21:59:33 2021 -0700

    KVM: X86: Fix x86_emulator slab cache leak

    commit dfdc0a714d241bfbf951886c373cd1ae463fcc25 upstream.

    Commit c9b8b07cded58 (KVM: x86: Dynamically allocate per-vCPU emulation context)
    tries to allocate per-vCPU emulation context dynamically, however, the
    x86_emulator slab cache is still exiting after the kvm module is unload
    as below after destroying the VM and unloading the kvm module.

    grep x86_emulator /proc/slabinfo
    x86_emulator          36     36   2672   12    8 : tunables    0    0    0 : slabdata
3      3      0

    This patch fixes this slab cache leak by destroying the x86_emulator slab cache
    when the kvm module is unloaded.

    Fixes: c9b8b07cded58 (KVM: x86: Dynamically allocate per-vCPU emulation context)
    Cc: stable@vger.kernel.org
    Signed-off-by: Wanpeng Li <wanpengli@tencent.com>
    Message-Id: <1623387573-5969-1-git-send-email-wanpengli@tencent.com>
```

```
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c87dc2e491d44cbb9d8d54089305405bce37fd7f
Author: Sean Christopherson <seanjc@google.com>
Date:   Thu Jun 10 15:00:26 2021 -0700

    KVM: x86/mmu: Calculate and check "full" mmu_role for nested MMU

    commit 654430efde27248be563df9a88631204b5fe2df2 upstream.

    Calculate and check the full mmu_role when initializing the MMU context
    for the nested MMU, where "full" means the bits and pieces of the role
    that aren't handled by kvm_calc_mmu_role_common().  While the nested MMU
    isn't used for shadow paging, things like the number of levels in the
    guest's page tables are surprisingly important when walking the guest
    page tables.  Failure to reinitialize the nested MMU context if L2's
    paging mode changes can result in unexpected and/or missed page faults,
    and likely other explosions.

    E.g. if an L1 vCPU is running both a 32-bit PAE L2 and a 64-bit L2, the
    "common" role calculation will yield the same role for both L2s.  If the
    64-bit L2 is run after the 32-bit PAE L2, L0 will fail to reinitialize
    the nested MMU context, ultimately resulting in a bad walk of L2's page
    tables as the MMU will still have a guest root_level of PT32E_ROOT_LEVEL.

      WARNING: CPU: 4 PID: 167334 at arch/x86/kvm/vmx/vmx.c:3075 ept_save_pdptrs+0x15/0xe0
[kvm_intel]
      Modules linked in: kvm_intel]
      CPU: 4 PID: 167334 Comm: CPU 3/KVM Not tainted 5.13.0-rc1-d849817d5673-reqs #185
      Hardware name: ASUS Q87M-E/Q87M-E, BIOS 1102 03/03/2014
      RIP: 0010:ept_save_pdptrs+0x15/0xe0 [kvm_intel]
      Code: <0f> 0b c3 f6 87 d8 02 00f
      RSP: 0018:ffffbba702dbba00 EFLAGS: 00010202
      RAX: 0000000000000011 RBX: 0000000000000002 RCX: ffffffff810a2c08
      RDX: ffff91d7bc30acc0 RSI: 0000000000000011 RDI: ffff91d7bc30a600
      RBP: ffff91d7bc30a600 R08: 0000000000000010 R09: 0000000000000007
      R10: 0000000000000000 R11: 0000000000000000 R12: ffff91d7bc30a600
      R13: ffff91d7bc30acc0 R14: ffff91d67c123460 R15: 0000000115d7e005
      FS:  00007fe8e9ffb700(0000) GS:ffff91d90fb00000(0000) knlGS:0000000000000000
      CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      CR2: 0000000000000000 CR3: 000000029f15a001 CR4: 0000000001726e0
      Call Trace:
       kvm_pdptr_read+0x3a/0x40 [kvm]
       paging64_walk_addr_generic+0x327/0x6a0 [kvm]
       paging64_gva_to_gpa_nested+0x3f/0xb0 [kvm]
       kvm_fetch_guest_virt+0x4c/0xb0 [kvm]
       __do_insn_fetch_bytes+0x11a/0x1f0 [kvm]
       x86_decode_insn+0x787/0x1490 [kvm]
       x86_decode_emulated_instruction+0x58/0x1e0 [kvm]
       x86_emulate_instruction+0x122/0x4f0 [kvm]
       vmx_handle_exit+0x120/0x660 [kvm_intel]
       kvm_arch_vcpu_ioctl_run+0xe25/0x1cb0 [kvm]
       kvm_vcpu_ioctl+0x211/0x5a0 [kvm]
       __x64_sys_ioctl+0x83/0xb0
       do_syscall_64+0x40/0xb0
       entry_SYSCALL_64_after_hwframe+0x44/0xae

    Cc: Vitaly Kuznetsov <vkuznets@redhat.com>
    Cc: stable@vger.kernel.org
    Fixes: bf627a928837 ("x86/kvm/mmu: check if MMU reconfiguration is needed in
init_kvm_nested_mmu()")
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Message-Id: <20210610220026.1364486-1-seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
```

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

commit df9a40cfb3be2cbeb1c17bb67c59251ba16630f3
Author: Sean Christopherson <seanjc@google.com>
Date:   Wed Jun 9 11:56:11 2021 -0700

    KVM: x86: Immediately reset the MMU context when the SMM flag is cleared

    commit 78fcb2c91adfec8ce3a2ba6b4d0dda89f2f4a7c6 upstream.

    Immediately reset the MMU context when the vCPU's SMM flag is cleared so
    that the SMM flag in the MMU role is always synchronized with the vCPU's
    flag.  If RSM fails (which isn't correctly emulated), KVM will bail
    without calling post_leave_smm() and leave the MMU in a bad state.

    The bad MMU role can lead to a NULL pointer dereference when grabbing a
    shadow page's rmap for a page fault as the initial lookups for the gfn
    will happen with the vCPU's SMM flag (=0), whereas the rmap lookup will
    use the shadow page's SMM flag, which comes from the MMU (=1).  SMM has
    an entirely different set of memslots, and so the initial lookup can find
    a memslot (SMM=0) and then explode on the rmap memslot lookup (SMM=1).

      general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000
[#1] PREEMPT SMP KASAN
      KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007]
      CPU: 1 PID: 8410 Comm: syz-executor382 Not tainted 5.13.0-rc5-syzkaller #0
      Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
01/01/2011
      RIP: 0010:__gfn_to_rmap arch/x86/kvm/mmu/mmu.c:935 [inline]
      RIP: 0010:gfn_to_rmap+0x2b0/0x4d0 arch/x86/kvm/mmu/mmu.c:947
      Code: <42> 80 3c 20 00 74 08 4c 89 ff e8 f1 79 a9 00 4c 89 fb 4d 8b 37 44
      RSP: 0018:ffffc90000ffef98 EFLAGS: 00010246
      RAX: 0000000000000000 RBX: ffff888015b9f414 RCX: ffff888019669c40
      RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000001
      RBP: 0000000000000001 R08: ffffffff811d9cdb R09: ffffed10065a6002
      R10: ffffed10065a6002 R11: 0000000000000000 R12: dfffffc0000000000
      R13: 0000000000000003 R14: 0000000000000001 R15: 0000000000000000
      FS:  000000000124b300(0000) GS:ffff8880b9b00000(0000) knlGS:0000000000000000
      CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      CR2: 0000000000000000 CR3: 0000000028e31000 CR4: 00000000001526e0
      DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
      DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
      Call Trace:
       rmap_add arch/x86/kvm/mmu/mmu.c:965 [inline]
       mmu_set_spte+0x862/0xe60 arch/x86/kvm/mmu/mmu.c:2604
       __direct_map arch/x86/kvm/mmu/mmu.c:2862 [inline]
       direct_page_fault+0x1f74/0x2b70 arch/x86/kvm/mmu/mmu.c:3769
       kvm_mmu_do_page_fault arch/x86/kvm/mmu.h:124 [inline]
       kvm_mmu_page_fault+0x199/0x1440 arch/x86/kvm/mmu/mmu.c:5065
       vmx_handle_exit+0x26/0x160 arch/x86/kvm/vmx/vmx.c:6122
       vcpu_enter_guest+0x3bdd/0x9630 arch/x86/kvm/x86.c:9428
       vcpu_run+0x416/0xc20 arch/x86/kvm/x86.c:9494
       kvm_arch_vcpu_ioctl_run+0x4e8/0xa40 arch/x86/kvm/x86.c:9722
       kvm_vcpu_ioctl+0x70f/0xbb0 arch/x86/kvm/../../../virt/kvm/kvm_main.c:3460
       vfs_ioctl fs/ioctl.c:51 [inline]
       __do_sys_ioctl fs/ioctl.c:1069 [inline]
       __se_sys_ioctl+0xfb/0x170 fs/ioctl.c:1055
       do_syscall_64+0x3f/0xb0 arch/x86/entry/common.c:47
       entry_SYSCALL_64_after_hwframe+0x44/0xae
      RIP: 0033:0x440ce9

    Cc: stable@vger.kernel.org
    Reported-by: syzbot+fb0b6a7e8713aeb0319c@syzkaller.appspotmail.com
    Fixes: 9ec19493fb86 ("KVM: x86: clear SMM flags before loading state while leaving SMM")
    Signed-off-by: Sean Christopherson <seanjc@google.com>
```

```
    Message-Id: <20210609185619.992058-2-seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 54bab5cfa8c1f64c684126c0c773c3a233198999
Author: Alexander Gordeev <agordeev@linux.ibm.com>
Date:   Mon May 17 08:18:12 2021 +0200

    s390/mcck: fix invalid KVM guest condition check

    commit 1874cb13d5d7cafa61ce93a760093ebc5485b6ab upstream.

    Wrong condition check is used to decide if a machine check hit
    while in KVM guest. As result of this check the instruction
    following the SIE critical section might be considered as still
    in KVM guest and _CIF_MCCK_GUEST CPU flag mistakenly set as
    result.

    Fixes: c929500d7a5a ("s390/nmi: s390: New low level handling for machine check happening
in guest")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Alexander Gordeev <agordeev@linux.ibm.com>
    Reviewed-by: Christian Borntraeger <borntraeger@de.ibm.com>
    Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 47dbe49b19534febf967ea729689ce6f36ef831b
Author: Naohiro Aota <naohiro.aota@wdc.com>
Date:   Thu Jun 17 13:56:18 2021 +0900

    btrfs: zoned: fix negative space_info->bytes_readonly

    commit f9f28e5bd0baee9708c9011897196f06ae3a2733 upstream.

    Consider we have a using block group on zoned btrfs.

    |<- ZU ->|<- used ->|<---free--->|
                        `- Alloc offset
    ZU: Zone unusable

    Marking the block group read-only will migrate the zone unusable bytes
    to the read-only bytes. So, we will have this.

    |<- RO ->|<- used ->|<--- RO --->|

    RO: Read only

    When marking it back to read-write, btrfs_dec_block_group_ro()
    subtracts the above "RO" bytes from the
    space_info->bytes_readonly. And, it moves the zone unusable bytes back
    and again subtracts those bytes from the space_info->bytes_readonly,
    leading to negative bytes_readonly.

    This can be observed in the output as eg.:

      Data, single: total=512.00MiB, used=165.21MiB, zone_unusable=16.00EiB
      Data, single: total=536870912, used=173256704, zone_unusable=18446744073603186688

    This commit fixes the issue by reordering the operations.

    Link: https://github.com/naota/linux/issues/37
    Reported-by: David Sterba <dsterba@suse.com>
    Fixes: 169e0da91a21 ("btrfs: zoned: track unusable bytes for zones")
    CC: stable@vger.kernel.org # 5.12+
```

```
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Signed-off-by: Naohiro Aota <naohiro.aota@wdc.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fb4af05cc6220b69632d4a969613bb0e93856d51
Author: Chiqijun <chiqijun@huawei.com>
Date:   Mon May 24 17:44:07 2021 -0500

    PCI: Work around Huawei Intelligent NIC VF FLR erratum

    commit ce00322c2365e1f7b0312f2f493539c833465d97 upstream.

    pcie_flr() starts a Function Level Reset (FLR), waits 100ms (the maximum
    time allowed for FLR completion by PCIe r5.0, sec 6.6.2), and waits for the
    FLR to complete.  It assumes the FLR is complete when a config read returns
    valid data.

    When we do an FLR on several Huawei Intelligent NIC VFs at the same time,
    firmware on the NIC processes them serially.  The VF may respond to config
    reads before the firmware has completed its reset processing.  If we bind a
    driver to the VF (e.g., by assigning the VF to a virtual machine) in the
    interval between the successful config read and completion of the firmware
    reset processing, the NIC VF driver may fail to load.

    Prevent this driver failure by waiting for the NIC firmware to complete its
    reset processing.  Not all NIC firmware supports this feature.

    [bhelgaas: commit log]
    Link: https://support.huawei.com/enterprise/en/doc/EDOC1100063073/87950645/vm-oss-
occasionally-fail-to-load-the-in200-driver-when-the-vf-performs-flr
    Link: https://lore.kernel.org/r/20210414132301.1793-1-chiqijun@huawei.com
    Signed-off-by: Chiqijun <chiqijun@huawei.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Reviewed-by: Alex Williamson <alex.williamson@redhat.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4879d99a1af034d0dbd11f46e33894756b9f300b
Author: Sriharsha Basavapatna <sriharsha.basavapatna@broadcom.com>
Date:   Fri May 21 21:13:17 2021 -0400

    PCI: Add ACS quirk for Broadcom BCM57414 NIC

    commit db2f77e2bd99dbd2fb23ddde58f0fae392fe3338 upstream.

    The Broadcom BCM57414 NIC may be a multi-function device.  While it does
    not advertise an ACS capability, peer-to-peer transactions are not possible
    between the individual functions, so it is safe to treat them as fully
    isolated.

    Add an ACS quirk for this device so the functions can be in independent
    IOMMU groups and attached individually to userspace applications using
    VFIO.

    [bhelgaas: commit log]
    Link: https://lore.kernel.org/r/1621645997-16251-1-git-send-email-
michael.chan@broadcom.com
    Signed-off-by: Sriharsha Basavapatna <sriharsha.basavapatna@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3d213a4ddf49a860be6e795482c17f87e0c82b2a
```

```
Author: Pali Rohár <pali@kernel.org>
Date:   Tue Jun 8 22:36:55 2021 +0200

    PCI: aardvark: Fix kernel panic during PIO transfer

    commit f18139966d072dab8e4398c95ce955a9742e04f7 upstream.

    Trying to start a new PIO transfer by writing value 0 in PIO_START register
    when previous transfer has not yet completed (which is indicated by value 1
    in PIO_START) causes an External Abort on CPU, which results in kernel
    panic:

        SError Interrupt on CPU0, code 0xbf000002 -- SError
        Kernel panic - not syncing: Asynchronous SError Interrupt

    To prevent kernel panic, it is required to reject a new PIO transfer when
    previous one has not finished yet.

    If previous PIO transfer is not finished yet, the kernel may issue a new
    PIO request only if the previous PIO transfer timed out.

    In the past the root cause of this issue was incorrectly identified (as it
    often happens during link retraining or after link down event) and special
    hack was implemented in Trusted Firmware to catch all SError events in EL3,
    to ignore errors with code 0xbf000002 and not forwarding any other errors
    to kernel and instead throw panic from EL3 Trusted Firmware handler.

    Links to discussion and patches about this issue:
    https://git.trustedfirmware.org/TF-A/trusted-firmware-a.git/commit/?id=3c7dcdac5c50
    https://lore.kernel.org/linux-pci/20190316161243.29517-1-repk@triplefau.lt/
    https://lore.kernel.org/linux-pci/971be151d24312cc533989a64bd454b4@www.loen.fr/
    https://review.trustedfirmware.org/c/TF-A/trusted-firmware-a/+/1541

    But the real cause was the fact that during link retraining or after link
    down event the PIO transfer may take longer time, up to the 1.44s until it
    times out. This increased probability that a new PIO transfer would be
    issued by kernel while previous one has not finished yet.

    After applying this change into the kernel, it is possible to revert the
    mentioned TF-A hack and SError events do not have to be caught in TF-A EL3.

    Link: https://lore.kernel.org/r/20210608203655.31228-1-pali@kernel.org
    Signed-off-by: Pali Rohár <pali@kernel.org>
    Signed-off-by: Lorenzo Pieralisi <lorenzo.pieralisi@arm.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Reviewed-by: Marek Behún <kabel@kernel.org>
    Cc: stable@vger.kernel.org # 7fbcb5da811b ("PCI: aardvark: Don't rely on jiffies while
holding spinlock")
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 74c1ea1b1b82a2ed906374d603c9a1a01728d1ae
Author: Evan Quan <evan.quan@amd.com>
Date:   Wed Jun 2 10:12:55 2021 +0800

    PCI: Mark AMD Navi14 GPU ATS as broken

    commit e8946a53e2a698c148b3b3ed732f43c7747fbeb6 upstream.

    Observed unexpected GPU hang during runpm stress test on 0x7341 rev 0x00.
    Further debugging shows broken ATS is related.

    Disable ATS on this part.  Similar issues on other devices:

      a2da5d8cc0b0 ("PCI: Mark AMD Raven iGPU ATS as broken in some platforms")
      45beb31d3afb ("PCI: Mark AMD Navi10 GPU rev 0x00 ATS as broken")
```

```
       5e89cd303e3a ("PCI: Mark AMD Navi14 GPU rev 0xc5 ATS as broken")

     Suggested-by: Alex Deucher <alexander.deucher@amd.com>
     Link: https://lore.kernel.org/r/20210602021255.939090-1-evan.quan@amd.com
     Signed-off-by: Evan Quan <evan.quan@amd.com>
     Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
     Reviewed-by: Krzysztof Wilczyński <kw@linux.com>
     Cc: stable@vger.kernel.org
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 02bbc04c263067f9ec17d600b9f4c2559f4385ea
 Author: Shanker Donthineni <sdonthineni@nvidia.com>
 Date:   Tue Jun 8 11:18:56 2021 +0530

     PCI: Mark some NVIDIA GPUs to avoid bus reset

     commit 4c207e7121fa92b66bf1896bf8ccb9edfb0f9731 upstream.

     Some NVIDIA GPU devices do not work with SBR.  Triggering SBR leaves the
     device inoperable for the current system boot. It requires a system
     hard-reboot to get the GPU device back to normal operating condition
     post-SBR. For the affected devices, enable NO_BUS_RESET quirk to avoid the
     issue.

     This issue will be fixed in the next generation of hardware.

     Link: https://lore.kernel.org/r/20210608054857.18963-8-ameynarkhede03@gmail.com
     Signed-off-by: Shanker Donthineni <sdonthineni@nvidia.com>
     Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
     Reviewed-by: Sinan Kaya <okaya@kernel.org>
     Cc: stable@vger.kernel.org
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 39fc44156f3a2e042a3eb305de562f27e51b3697
 Author: Antti Järvinen <antti.jarvinen@gmail.com>
 Date:   Mon Mar 15 10:26:06 2021 +0000

     PCI: Mark TI C667X to avoid bus reset

     commit b5cf198e74a91073d12839a3e2db99994a39995d upstream.

     Some TI KeyStone C667X devices do not support bus/hot reset.  The PCIESS
     automatically disables LTSSM when Secondary Bus Reset is received and
     device stops working.  Prevent bus reset for these devices.  With this
     change, the device can be assigned to VMs with VFIO, but it will leak state
     between VMs.

     Reference: https://e2e.ti.com/support/processors/f/791/t/954382
     Link: https://lore.kernel.org/r/20210315102606.17153-1-antti.jarvinen@gmail.com
     Signed-off-by: Antti Järvinen <antti.jarvinen@gmail.com>
     Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
     Reviewed-by: Kishon Vijay Abraham I <kishon@ti.com>
     Cc: stable@vger.kernel.org
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit c89a2fda651e549ec93c4a7d512515881360b716
 Author: Steven Rostedt (VMware) <rostedt@goodmis.org>
 Date:   Thu Jun 17 17:12:35 2021 -0400

     tracing: Do no increment trace_clock_global() by one

     commit 89529d8b8f8daf92d9979382b8d2eb39966846ea upstream.

     The trace_clock_global() tries to make sure the events between CPUs is
     somewhat in order. A global value is used and updated by the latest read
```

```
    of a clock. If one CPU is ahead by a little, and is read by another CPU, a
    lock is taken, and if the timestamp of the other CPU is behind, it will
    simply use the other CPUs timestamp.

    The lock is also only taken with a "trylock" due to tracing, and strange
    recursions can happen. The lock is not taken at all in NMI context.

    In the case where the lock is not able to be taken, the non synced
    timestamp is returned. But it will not be less than the saved global
    timestamp.

    The problem arises because when the time goes "backwards" the time
    returned is the saved timestamp plus 1. If the lock is not taken, and the
    plus one to the timestamp is returned, there's a small race that can cause
    the time to go backwards!

           CPU0                            CPU1
           ----                            ----
                                  trace_clock_global() {
                                      ts = clock() [ 1000 ]
                                      trylock(clock_lock) [ success ]
                                      global_ts = ts; [ 1000 ]

                                      <interrupted by NMI>
     trace_clock_global() {
        ts = clock() [ 999 ]
        if (ts < global_ts)
            ts = global_ts + 1 [ 1001 ]

        trylock(clock_lock) [ fail ]

        return ts [ 1001]
     }
                                      unlock(clock_lock);
                                      return ts; [ 1000 ]
                                  }

     trace_clock_global() {
        ts = clock() [ 1000 ]
        if (ts < global_ts) [ false 1000 == 1000 ]

        trylock(clock_lock) [ success ]
        global_ts = ts; [ 1000 ]
        unlock(clock_lock)

        return ts; [ 1000 ]
     }

    The above case shows to reads of trace_clock_global() on the same CPU, but
    the second read returns one less than the first read. That is, time when
    backwards, and this is not what is allowed by trace_clock_global().

    This was triggered by heavy tracing and the ring buffer checker that tests
    for the clock going backwards:

     Ring buffer clock went backwards: 20613921464 -> 20613921463
     ------------[ cut here ]------------
     WARNING: CPU: 2 PID: 0 at kernel/trace/ring_buffer.c:3412 check_buffer+0x1b9/0x1c0
     Modules linked in:
     [..]
     [CPU: 2]TIME DOES NOT MATCH expected:20620711698 actual:20620711697 delta:6790234
 before:20613921463 after:20613921463
       [20613915818] PAGE TIME STAMP
       [20613915818] delta:0
       [20613915819] delta:1
```

```
        [20613916035] delta:216
        [20613916465] delta:430
        [20613916575] delta:110
        [20613916749] delta:174
        [20613917248] delta:499
        [20613917333] delta:85
        [20613917775] delta:442
        [20613917921] delta:146
        [20613918321] delta:400
        [20613918568] delta:247
        [20613918768] delta:200
        [20613919306] delta:538
        [20613919353] delta:47
        [20613919980] delta:627
        [20613920296] delta:316
        [20613920571] delta:275
        [20613920862] delta:291
        [20613921152] delta:290
        [20613921464] delta:312
        [20613921464] delta:0 TIME EXTEND
        [20613921464] delta:0
```

    This happened more than once, and always for an off by one result. It also
    started happening after commit aafe104aa9096 was added.

    Cc: stable@vger.kernel.org
    Fixes: aafe104aa9096 ("tracing: Restructure trace_clock_global() to never block")
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5b5b8be020c75db649f66c2b7e84d8e29ed2da45
Author: Steven Rostedt (VMware) <rostedt@goodmis.org>
Date:   Thu Jun 17 14:32:34 2021 -0400

    tracing: Do not stop recording comms if the trace file is being read

    commit 4fdd595e4f9a1ff6d93ec702eaecae451cfc6591 upstream.

    A while ago, when the "trace" file was opened, tracing was stopped, and
    code was added to stop recording the comms to saved_cmdlines, for mapping
    of the pids to the task name.

    Code has been added that only records the comm if a trace event occurred,
    and there's no reason to not trace it if the trace file is opened.

    Cc: stable@vger.kernel.org
    Fixes: 7ffbd48d5cab2 ("tracing: Cache comms only after an event occurred")
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0eee3ce5249f5da5110b2a5406a3763cf8c8b297
Author: Steven Rostedt (VMware) <rostedt@goodmis.org>
Date:   Thu Jun 17 13:47:25 2021 -0400

    tracing: Do not stop recording cmdlines when tracing is off

    commit 85550c83da421fb12dc1816c45012e1e638d2b38 upstream.

    The saved_cmdlines is used to map pids to the task name, such that the
    output of the tracing does not just show pids, but also gives a human
    readable name for the task.

    If the name is not mapped, the output looks like this:

```
        <...>-1316          [005] ...2   132.044039: ...
```

```
    Instead of this:

        gnome-shell-1316    [005] ...2   132.044039: ...

    The names are updated when tracing is running, but are skipped if tracing
    is stopped. Unfortunately, this stops the recording of the names if the
    top level tracer is stopped, and not if there's other tracers active.

    The recording of a name only happens when a new event is written into a
    ring buffer, so there is no need to test if tracing is on or not. If
    tracing is off, then no event is written and no need to test if tracing is
    off or not.

    Remove the check, as it hides the names of tasks for events in the
    instance buffers.

    Cc: stable@vger.kernel.org
    Fixes: 7ffbd48d5cab2 ("tracing: Cache comms only after an event occurred")
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c12f71e86f2272bd61a543e35e0f4152f23b3807
Author: Breno Lima <breno.lima@nxp.com>
Date:   Mon Jun 14 13:50:13 2021 -0400

    usb: chipidea: imx: Fix Battery Charger 1.2 CDP detection

    commit c6d580d96f140596d69220f60ce0cfbea4ee5c0f upstream.

    i.MX8MM cannot detect certain CDP USB HUBs. usbmisc_imx.c driver is not
    following CDP timing requirements defined by USB BC 1.2 specification
    and section 3.2.4 Detection Timing CDP.

    During Primary Detection the i.MX device should turn on VDP_SRC and
    IDM_SINK for a minimum of 40ms (TVDPSRC_ON). After a time of TVDPSRC_ON,
    the i.MX is allowed to check the status of the D- line. Current
    implementation is waiting between 1ms and 2ms, and certain BC 1.2
    complaint USB HUBs cannot be detected. Increase delay to 40ms allowing
    enough time for primary detection.

    During secondary detection the i.MX is required to disable VDP_SRC and
    IDM_SNK, and enable VDM_SRC and IDP_SINK for at least 40ms (TVDMSRC_ON).

    Current implementation is not disabling VDP_SRC and IDM_SNK, introduce
    disable sequence in imx7d_charger_secondary_detection() function.

    VDM_SRC and IDP_SINK should be enabled for at least 40ms (TVDMSRC_ON).
    Increase delay allowing enough time for detection.

    Cc: <stable@vger.kernel.org>
    Fixes: 746f316b753a ("usb: chipidea: introduce imx7d USB charger detection")
    Signed-off-by: Breno Lima <breno.lima@nxp.com>
    Signed-off-by: Jun Li <jun.li@nxp.com>
    Link: https://lore.kernel.org/r/20210614175013.495808-1-breno.lima@nxp.com
    Signed-off-by: Peter Chen <peter.chen@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 955b2bd83c3870bd46b2aa73a4ba38b71e88bb65
Author: Andrew Lunn <andrew@lunn.ch>
Date:   Mon Jun 14 17:55:23 2021 +0200

    usb: core: hub: Disable autosuspend for Cypress CY7C65632

    commit a7d8d1c7a7f73e780aa9ae74926ae5985b2f895f upstream.
```

```
    The Cypress CY7C65632 appears to have an issue with auto suspend and
    detecting devices, not too dissimilar to the SMSC 5534B hub. It is
    easiest to reproduce by connecting multiple mass storage devices to
    the hub at the same time. On a Lenovo Yoga, around 1 in 3 attempts
    result in the devices not being detected. It is however possible to
    make them appear using lsusb -v.

    Disabling autosuspend for this hub resolves the issue.

    Fixes: 1208f9e1d758 ("USB: hub: Fix the broken detection of USB3 device in SMSC hub")
    Cc: stable@vger.kernel.org
    Signed-off-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20210614155524.2228800-1-andrew@lunn.ch
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d0760a4ef85697bc756d06eae17ae27f3f055401
Author: Pavel Skripkin <paskripkin@gmail.com>
Date:   Thu Jun 10 00:58:33 2021 +0300

    can: mcba_usb: fix memory leak in mcba_usb

    commit 91c02557174be7f72e46ed7311e3bea1939840b0 upstream.

    Syzbot reported memory leak in SocketCAN driver for Microchip CAN BUS
    Analyzer Tool. The problem was in unfreed usb_coherent.

    In mcba_usb_start() 20 coherent buffers are allocated and there is
    nothing, that frees them:

    1) In callback function the urb is resubmitted and that's all
    2) In disconnect function urbs are simply killed, but URB_FREE_BUFFER
       is not set (see mcba_usb_start) and this flag cannot be used with
       coherent buffers.

    Fail log:
    | [ 1354.053291][ T8413] mcba_usb 1-1:0.0 can0: device disconnected
    | [ 1367.059384][ T8420] kmemleak: 20 new suspected memory leaks (see
/sys/kernel/debug/kmem)

    So, all allocated buffers should be freed with usb_free_coherent()
    explicitly

    NOTE:
    The same pattern for allocating and freeing coherent buffers
    is used in drivers/net/can/usb/kvaser_usb/kvaser_usb_core.c

    Fixes: 51f3baad7de9 ("can: mcba_usb: Add support for Microchip CAN BUS Analyzer")
    Link: https://lore.kernel.org/r/20210609215833.30393-1-paskripkin@gmail.com
    Cc: linux-stable <stable@vger.kernel.org>
    Reported-and-tested-by: syzbot+57281c762a3922e14dfe@syzkaller.appspotmail.com
    Signed-off-by: Pavel Skripkin <paskripkin@gmail.com>
    Signed-off-by: Marc Kleine-Budde <mkl@pengutronix.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1071065eeb33d32b7d98c2ce7591881ae7381705
Author: Oleksij Rempel <linux@rempel-privat.de>
Date:   Fri May 21 13:57:20 2021 +0200

    can: j1939: fix Use-after-Free, hold skb ref while in use

    commit 2030043e616cab40f510299f09b636285e0a3678 upstream.

    This patch fixes a Use-after-Free found by the syzbot.
```

```
    The problem is that a skb is taken from the per-session skb queue,
    without incrementing the ref count. This leads to a Use-after-Free if
    the skb is taken concurrently from the session queue due to a CTS.

    Fixes: 9d71dd0c7009 ("can: add support of SAE J1939 protocol")
    Link: https://lore.kernel.org/r/20210521115720.7533-1-o.rempel@pengutronix.de
    Cc: Hillf Danton <hdanton@sina.com>
    Cc: linux-stable <stable@vger.kernel.org>
    Reported-by: syzbot+220c1a29987a9a490903@syzkaller.appspotmail.com
    Reported-by: syzbot+45199c1b73b4013525cf@syzkaller.appspotmail.com
    Signed-off-by: Oleksij Rempel <o.rempel@pengutronix.de>
    Signed-off-by: Marc Kleine-Budde <mkl@pengutronix.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e89912962fa0321cb71448b2843942cdfb33e78d
Author: Tetsuo Handa <penguin-kernel@i-love.sakura.ne.jp>
Date:   Sat Jun 5 19:26:35 2021 +0900

    can: bcm/raw/isotp: use per module netdevice notifier

    commit 8d0caedb759683041d9db82069937525999ada53 upstream.

    syzbot is reporting hung task at register_netdevice_notifier() [1] and
    unregister_netdevice_notifier() [2], for cleanup_net() might perform
    time consuming operations while CAN driver's raw/bcm/isotp modules are
    calling {register,unregister}_netdevice_notifier() on each socket.

    Change raw/bcm/isotp modules to call register_netdevice_notifier() from
    module's __init function and call unregister_netdevice_notifier() from
    module's __exit function, as with gw/j1939 modules are doing.

    Link: https://syzkaller.appspot.com/bug?id=391b9498827788b3cc6830226d4ff5be87107c30 [1]
    Link: https://syzkaller.appspot.com/bug?id=1724d278c83ca6e6df100a2e320c10d991cf2bce [2]
    Link: https://lore.kernel.org/r/54a5f451-05ed-f977-8534-79e7aa2bcc8f@i-love.sakura.ne.jp
    Cc: linux-stable <stable@vger.kernel.org>
    Reported-by: syzbot <syzbot+355f8edb2ff45d5f95fa@syzkaller.appspotmail.com>
    Reported-by: syzbot <syzbot+0f1827363a305f74996f@syzkaller.appspotmail.com>
    Reviewed-by: Kirill Tkhai <ktkhai@virtuozzo.com>
    Tested-by: syzbot <syzbot+355f8edb2ff45d5f95fa@syzkaller.appspotmail.com>
    Tested-by: Oliver Hartkopp <socketcan@hartkopp.net>
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Signed-off-by: Marc Kleine-Budde <mkl@pengutronix.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dc6415cb5cf8ebc8b334b7d0be916a0bf4353779
Author: Norbert Slusarek <nslusarek@gmx.net>
Date:   Sat Jun 12 22:18:54 2021 +0200

    can: bcm: fix infoleak in struct bcm_msg_head

    commit 5e87ddbe3942e27e939bdc02deb8579b0cbd8ecc upstream.

    On 64-bit systems, struct bcm_msg_head has an added padding of 4 bytes between
    struct members count and ival1. Even though all struct members are initialized,
    the 4-byte hole will contain data from the kernel stack. This patch zeroes out
    struct bcm_msg_head before usage, preventing infoleaks to userspace.

    Fixes: ffd980f976e7 ("[CAN]: Add broadcast manager (bcm) protocol")
    Link: https://lore.kernel.org/r/trinity-7c1b2e82-e34f-4885-8060-2cd7a13769ce-
1623532166177@3c-app-gmx-bs52
    Cc: linux-stable <stable@vger.kernel.org>
    Signed-off-by: Norbert Slusarek <nslusarek@gmx.net>
    Acked-by: Oliver Hartkopp <socketcan@hartkopp.net>
    Signed-off-by: Marc Kleine-Budde <mkl@pengutronix.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 68a1936e1812653b68c5b68e698d88fb35018835
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Fri May 28 13:47:27 2021 +0000

    bpf: Do not mark insn as seen under speculative path verification

    [ Upstream commit fe9a5ca7e370e613a9a75a13008a3845ea759d6e ]

    ... in such circumstances, we do not want to mark the instruction as seen given
    the goal is still to jmp-1 rewrite/sanitize dead code, if it is not reachable
    from the non-speculative path verification. We do however want to verify it for
    safety regardless.

    With the patch as-is all the insns that have been marked as seen before the
    patch will also be marked as seen after the patch (just with a potentially
    different non-zero count). An upcoming patch will also verify paths that are
    unreachable in the non-speculative domain, hence this extension is needed.

    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Reviewed-by: John Fastabend <john.fastabend@gmail.com>
    Reviewed-by: Benedict Schlueter <benedict.schlueter@rub.de>
    Reviewed-by: Piotr Krysiuk <piotras@gmail.com>
    Acked-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 408a4956acde24413f3c684912b1d3e404bed8e2
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Fri May 28 13:03:30 2021 +0000

    bpf: Inherit expanded/patched seen count from old aux data

    [ Upstream commit d203b0fd863a2261e5d00b97f3d060c4c2a6db71 ]

    Instead of relying on current env->pass_cnt, use the seen count from the
    old aux data in adjust_insn_aux_data(), and expand it to the new range of
    patched instructions. This change is valid given we always expand 1:n
    with n>=1, so what applies to the old/original instruction needs to apply
    for the replacement as well.

    Not relying on env->pass_cnt is a prerequisite for a later change where we
    want to avoid marking an instruction seen when verified under speculative
    execution path.

    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Reviewed-by: John Fastabend <john.fastabend@gmail.com>
    Reviewed-by: Benedict Schlueter <benedict.schlueter@rub.de>
    Reviewed-by: Piotr Krysiuk <piotras@gmail.com>
    Acked-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 99c028fb3a99f2f2d7748f793cb701daa7c6fcc9
Author: John Garry <john.garry@huawei.com>
Date:   Thu Jun 10 22:33:00 2021 +0800

    perf metricgroup: Return error code from metricgroup__add_metric_sys_event_iter()

    [ Upstream commit fe7a98b9d9b36e5c8a22d76b67d29721f153f66e ]

    The error code is not set at all in the sys event iter function.

    This may lead to an uninitialized value of "ret" in
    metricgroup__add_metric() when no CPU metric is added.

    Fix by properly setting the error code.
```

```
    It is not necessary to init "ret" to 0 in metricgroup__add_metric(), as
    if we have no CPU or sys event metric matching, then "has_match" should
    be 0 and "ret" is set to -EINVAL.

    However gcc cannot detect that it may not have been set after the
    map_for_each_metric() loop for CPU metrics, which is strange.

    Fixes: be335ec28efa8 ("perf metricgroup: Support adding metrics for system PMUs")
    Signed-off-by: John Garry <john.garry@huawei.com>
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link: http://lore.kernel.org/lkml/1623335580-187317-3-git-send-email-
john.garry@huawei.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b390fbe6581603e99c36b0c760611fdbd0a67e94
Author: John Garry <john.garry@huawei.com>
Date:   Thu Jun 10 22:32:59 2021 +0800

    perf metricgroup: Fix find_evsel_group() event selector

    [ Upstream commit fc96ec4d5d4155c61cbafd49fb2dd403c899a9f4 ]

    The following command segfaults on my x86 broadwell:

      $ ./perf stat  -M frontend_bound,retiring,backend_bound,bad_speculation sleep 1
      WARNING: grouped events cpus do not match, disabling group:
        anon group { raw 0x10e }
        anon group { raw 0x10e }
      perf: util/evsel.c:1596: get_group_fd: Assertion `!(!leader->core.fd)' failed.
      Aborted (core dumped)

    The issue shows itself as a use-after-free in evlist__check_cpu_maps(),
    whereby the leader of an event selector (evsel) has been deleted (yet we
    still attempt to verify for an evsel).

    Fundamentally the problem comes from metricgroup__setup_events() ->
    find_evsel_group(), and has developed from the previous fix attempt in
    commit 9c880c24cb0d ("perf metricgroup: Fix for metrics containing
    duration_time").

    The problem now is that the logic in checking if an evsel is in the same
    group is subtly broken for the "cycles" event. For the "cycles" event,
    the pmu_name is NULL; however the logic in find_evsel_group() may set an
    event matched against "cycles" as used, when it should not be.

    This leads to a condition where an evsel is set, yet its leader is not.

    Fix the check for evsel pmu_name by not matching evsels when either has a
    NULL pmu_name.

    There is still a pre-existing metric issue whereby the ordering of the
    metrics may break the 'stat' function, as discussed at:
    https://lore.kernel.org/lkml/49c6fccb-b716-1bf0-18a6-cace1cdb66b9@huawei.com/

    Fixes: 9c880c24cb0d ("perf metricgroup: Fix for metrics containing duration_time")
    Signed-off-by: John Garry <john.garry@huawei.com>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com> # On a Thinkpad T450S
```

```
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link: http://lore.kernel.org/lkml/1623335580-187317-2-git-send-email-
john.garry@huawei.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8a484eebcc93ebfd598749f7a4abed3d2ecb3e27
Author: Marc Zyngier <maz@kernel.org>
Date:   Thu Jun 10 15:13:46 2021 +0100

    irqchip/gic-v3: Workaround inconsistent PMR setting on NMI entry

    [ Upstream commit 382e6e177bc1c02473e56591fe5083ae1e4904f6 ]

    The arm64 entry code suffers from an annoying issue on taking
    a NMI, as it sets PMR to a value that actually allows IRQs
    to be acknowledged. This is done for consistency with other parts
    of the code, and is in the process of being fixed. This shouldn't
    be a problem, as we are not enabling interrupts whilst in NMI
    context.

    However, in the infortunate scenario that we took a spurious NMI
    (retired before the read of IAR) *and* that there is an IRQ pending
    at the same time, we'll ack the IRQ in NMI context. Too bad.

    In order to avoid deadlocks while running something like perf,
    teach the GICv3 driver about this situation: if we were in
    a context where no interrupt should have fired, transiently
    set PMR to a value that only allows NMIs before acking the pending
    interrupt, and restore the original value after that.

    This papers over the core issue for the time being, and makes
    NMIs great again. Sort of.

    Fixes: 4d6a38da8e79e94c ("arm64: entry: always set GIC_PRIO_PSR_I_SET during entry")
    Co-developed-by: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Reviewed-by: Mark Rutland <mark.rutland@arm.com>
    Link: https://lore.kernel.org/lkml/20210610145731.1350460-1-maz@kernel.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c71845655436e16267e20f23cb70f2c6b905957d
Author: Feng Tang <feng.tang@intel.com>
Date:   Fri Jun 11 09:54:42 2021 +0800

    mm: relocate 'write_protect_seq' in struct mm_struct

    [ Upstream commit 2e3025434a6ba090c85871a1d4080ff784109e1f ]

    0day robot reported a 9.2% regression for will-it-scale mmap1 test
    case[1], caused by commit 57efa1fe5957 ("mm/gup: prevent gup_fast from
    racing with COW during fork").

    Further debug shows the regression is due to that commit changes the
    offset of hot fields 'mmap_lock' inside structure 'mm_struct', thus some
    cache alignment changes.

    From the perf data, the contention for 'mmap_lock' is very severe and
```

```
    takes around 95% cpu cycles, and it is a rw_semaphore

            struct rw_semaphore {
                    atomic_long_t count;    /* 8 bytes */
                    atomic_long_t owner;    /* 8 bytes */
                    struct optimistic_spin_queue osq; /* spinner MCS lock */
                    ...

    Before commit 57efa1fe5957 adds the 'write_protect_seq', it happens to
    have a very optimal cache alignment layout, as Linus explained:

     "and before the addition of the 'write_protect_seq' field, the
      mmap_sem was at offset 120 in 'struct mm_struct'.

      Which meant that count and owner were in two different cachelines,
      and then when you have contention and spend time in
      rwsem_down_write_slowpath(), this is probably *exactly* the kind
      of layout you want.

      Because first the rwsem_write_trylock() will do a cmpxchg on the
      first cacheline (for the optimistic fast-path), and then in the
      case of contention, rwsem_down_write_slowpath() will just access
      the second cacheline.

      Which is probably just optimal for a load that spends a lot of
      time contended - new waiters touch that first cacheline, and then
      they queue themselves up on the second cacheline."

    After the commit, the rw_semaphore is at offset 128, which means the
    'count' and 'owner' fields are now in the same cacheline, and causes
    more cache bouncing.

    Currently there are 3 "#ifdef CONFIG_XXX" before 'mmap_lock' which will
    affect its offset:

      CONFIG_MMU
      CONFIG_MEMBARRIER
      CONFIG_HAVE_ARCH_COMPAT_MMAP_BASES

    The layout above is on 64 bits system with 0day's default kernel config
    (similar to RHEL-8.3's config), in which all these 3 options are 'y'.
    And the layout can vary with different kernel configs.

    Relayouting a structure is usually a double-edged sword, as sometimes it
    can helps one case, but hurt other cases.  For this case, one solution
    is, as the newly added 'write_protect_seq' is a 4 bytes long seqcount_t
    (when CONFIG_DEBUG_LOCK_ALLOC=n), placing it into an existing 4 bytes
    hole in 'mm_struct' will not change other fields' alignment, while
    restoring the regression.

    Link: https://lore.kernel.org/lkml/20210525031636.GB7744@xsang-OptiPlex-9020/ [1]
    Reported-by: kernel test robot <oliver.sang@intel.com>
    Signed-off-by: Feng Tang <feng.tang@intel.com>
    Reviewed-by: John Hubbard <jhubbard@nvidia.com>
    Reviewed-by: Jason Gunthorpe <jgg@nvidia.com>
    Cc: Peter Xu <peterx@redhat.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 68c5ac88abccd6daaee5d9ce909e1efba09956c4
Author: Jisheng Zhang <jszhang@kernel.org>
Date:   Tue May 11 00:28:38 2021 +0800

    riscv: code patching only works on !XIP_KERNEL
```

```
    [ Upstream commit 42e0e0b453bc6ead49c573ed512502069627546b ]

    Some features which need code patching such as KPROBES, DYNAMIC_FTRACE
    KGDB can only work on !XIP_KERNEL. Add dependencies for these features
    that rely on code patching.

    Signed-off-by: Jisheng Zhang <jszhang@kernel.org>
    Signed-off-by: Palmer Dabbelt <palmerdabbelt@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4a737ccdb6512b32ef619a9873774e735c353b55
Author: Riwen Lu <luriwen@kylinos.cn>
Date:   Fri Jun 4 11:09:59 2021 +0800

    hwmon: (scpi-hwmon) shows the negative temperature properly

    [ Upstream commit 78d13552346289bad4a9bf8eabb5eec5e5a321a5 ]

    The scpi hwmon shows the sub-zero temperature in an unsigned integer,
    which would confuse the users when the machine works in low temperature
    environment. This shows the sub-zero temperature in an signed value and
    users can get it properly from sensors.

    Signed-off-by: Riwen Lu <luriwen@kylinos.cn>
    Tested-by: Xin Chen <chenxin@kylinos.cn>
    Link: https://lore.kernel.org/r/20210604030959.736379-1-luriwen@kylinos.cn
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7a01fdd060eb51aaa68d0097b94aea7605200830
Author: Chen Li <chenli@uniontech.com>
Date:   Fri Jun 4 16:43:02 2021 +0800

    radeon: use memcpy_to/fromio for UVD fw upload

    [ Upstream commit ab8363d3875a83f4901eb1cc00ce8afd24de6c85 ]

    I met a gpu addr bug recently and the kernel log
    tells me the pc is memcpy/memset and link register is
    radeon_uvd_resume.

    As we know, in some architectures, optimized memcpy/memset
    may not work well on device memory. Trival memcpy_toio/memset_io
    can fix this problem.

    BTW, amdgpu has already done it in:
    commit ba0b2275a678 ("drm/amdgpu: use memcpy_to/fromio for UVD fw upload"),
    that's why it has no this issue on the same gpu and platform.

    Signed-off-by: Chen Li <chenli@uniontech.com>
    Reviewed-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit becfe762bf3657dfa44320b347cd7c02eee444e2
Author: Srinivasa Rao Mandadapu <srivasam@codeaurora.org>
Date:   Fri Jun 4 23:45:45 2021 +0800

    ASoC: qcom: lpass-cpu: Fix pop noise during audio capture begin

    [ Upstream commit c8a4556d98510ca05bad8d02265a4918b03a8c0b ]

    This patch fixes PoP noise of around 15ms observed during audio
    capture begin.
    Enables BCLK and LRCLK in snd_soc_dai_ops prepare call for
```

```
    introducing some delay before capture start.

    (am from https://patchwork.kernel.org/patch/12276369/)
    (also found at https://lore.kernel.org/r/20210524142114.18676-1-srivasam@codeaurora.org)

    Co-developed-by: Judy Hsiao <judyhsiao@chromium.org>
    Signed-off-by: Judy Hsiao <judyhsiao@chromium.org>
    Signed-off-by: Srinivasa Rao Mandadapu <srivasam@codeaurora.org>
    Reviewed-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Link: https://lore.kernel.org/r/20210604154545.1198337-1-judyhsiao@chromium.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6b935731cdc9ddc0f45520363e263f46c92a97b8
Author: Saravana Kannan <saravanak@google.com>
Date:   Mon Jun 7 10:58:36 2021 +0200

    drm/sun4i: dw-hdmi: Make HDMI PHY into a platform device

    [ Upstream commit 9bf3797796f570b34438235a6a537df85832bdad ]

    On sunxi boards that use HDMI output, HDMI device probe keeps being
    avoided indefinitely with these repeated messages in dmesg:

      platform 1ee0000.hdmi: probe deferral - supplier 1ef0000.hdmi-phy
        not ready

    There's a fwnode_link being created with fw_devlink=on between hdmi
    and hdmi-phy nodes, because both nodes have 'compatible' property set.

    Fw_devlink code assumes that nodes that have compatible property
    set will also have a device associated with them by some driver
    eventually. This is not the case with the current sun8i-hdmi
    driver.

    This commit makes sun8i-hdmi-phy into a proper platform device
    and fixes the display pipeline probe on sunxi boards that use HDMI.

    More context: https://lkml.org/lkml/2021/5/16/203

    Signed-off-by: Saravana Kannan <saravanak@google.com>
    Signed-off-by: Ondrej Jirman <megous@megous.com>
    Tested-by: Andre Przywara <andre.przywara@arm.com>
    Signed-off-by: Maxime Ripard <maxime@cerno.tech>
    Link: https://patchwork.freedesktop.org/patch/msgid/20210607085836.2827429-1-
megous@megous.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7e7d112f7a2c9cdd4eaccc7b3365921ada09c63d
Author: Sergio Paracuellos <sergio.paracuellos@gmail.com>
Date:   Fri Jun 4 07:53:37 2021 +0200

    pinctrl: ralink: rt2880: avoid to error in calls is pin is already enabled

    [ Upstream commit eb367d875f94a228c17c8538e3f2efcf2eb07ead ]

    In 'rt2880_pmx_group_enable' driver is printing an error and returning
    -EBUSY if a pin has been already enabled. This begets anoying messages
    in the caller when this happens like the following:

    rt2880-pinmux pinctrl: pcie is already enabled
    mt7621-pci 1e140000.pcie: Error applying setting, reverse things back

    To avoid this just print the already enabled message in the pinctrl
    driver and return 0 instead to not confuse the user with a real
```

```
    bad problem.

    Signed-off-by: Sergio Paracuellos <sergio.paracuellos@gmail.com>
    Link: https://lore.kernel.org/r/20210604055337.20407-1-sergio.paracuellos@gmail.com
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f9ae1750ac6ca8d1c08847f4b478359509a2f394
Author: Oder Chiou <oder_chiou@realtek.com>
Date:   Fri Jun 4 14:31:50 2021 +0800

    ASoC: rt5682: Fix the fast discharge for headset unplugging in soundwire mode

    [ Upstream commit 49783c6f4a4f49836b5a109ae0daf2f90b0d7713 ]

    Based on ("5a15cd7fce20b1fd4aece6a0240e2b58cd6a225d"), the setting also
    should be set in soundwire mode.

    Signed-off-by: Oder Chiou <oder_chiou@realtek.com>
    Link: https://lore.kernel.org/r/20210604063150.29925-1-oder_chiou@realtek.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc68f0c9e4a001e02376fe87f4bdcacadb27e8a1
Author: Axel Lin <axel.lin@ingics.com>
Date:   Thu Jun 3 17:49:44 2021 +0800

    regulator: rt4801: Fix NULL pointer dereference if priv->enable_gpios is NULL

    [ Upstream commit cb2381cbecb81a8893b2d1e1af29bc2e5531df27 ]

    devm_gpiod_get_array_optional may return NULL if no GPIO was assigned.

    Signed-off-by: Axel Lin <axel.lin@ingics.com>
    Link: https://lore.kernel.org/r/20210603094944.1114156-1-axel.lin@ingics.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 600831a22047f91a59289cee63fad09854b987c3
Author: Patrice Chotard <patrice.chotard@foss.st.com>
Date:   Thu Jun 3 09:34:21 2021 +0200

    spi: stm32-qspi: Always wait BUSY bit to be cleared in stm32_qspi_wait_cmd()

    [ Upstream commit d38fa9a155b2829b7e2cfcf8a4171b6dd3672808 ]

    In U-boot side, an issue has been encountered when QSPI source clock is
    running at low frequency (24 MHz for example), waiting for TCF bit to be
    set didn't ensure that all data has been send out the FIFO, we should also
    wait that BUSY bit is cleared.

    To prevent similar issue in kernel driver, we implement similar behavior
    by always waiting BUSY bit to be cleared.

    Signed-off-by: Patrice Chotard <patrice.chotard@foss.st.com>
    Link: https://lore.kernel.org/r/20210603073421.8441-1-patrice.chotard@foss.st.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b4c0a756d88e629d82f9c82db453bba9e85c5634
Author: Axel Lin <axel.lin@ingics.com>
Date:   Sat May 29 09:32:36 2021 +0800

    regulator: hi6421v600: Fix .vsel_mask setting
```

```
    [ Upstream commit 50bec7fb4cb1bcf9d387046b6dec7186590791ec ]

    Take ldo3_voltages as example, the ARRAY_SIZE(ldo3_voltages) is 16.
    i.e. the valid selector is 0 ~ 0xF.
    But in current code the vsel_mask is "(1 << 15) - 1", i.e. 0x7FFF. Fix it.

    Signed-off-by: Axel Lin <axel.lin@ingics.com>
    Link: https://lore.kernel.org/r/20210529013236.373847-1-axel.lin@ingics.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa2b159f3839f1a1a92c109e86bc3d492e0fd956
Author: Richard Weinberger <richard@nod.at>
Date:   Sun May 30 22:34:46 2021 +0200

    ASoC: tas2562: Fix TDM_CFG0_SAMPRATE values

    [ Upstream commit 8bef925e37bdc9b6554b85eda16ced9a8e3c135f ]

    TAS2562_TDM_CFG0_SAMPRATE_MASK starts at bit 1, not 0.
    So all values need to be left shifted by 1.

    Signed-off-by: Richard Weinberger <richard@nod.at>
    Link: https://lore.kernel.org/r/20210530203446.19022-1-richard@nod.at
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f292028099b13dd9b3c0ed4e8f28e63ef7cffbb4
Author: Vincent Guittot <vincent.guittot@linaro.org>
Date:   Tue Jun 1 10:58:32 2021 +0200

    sched/pelt: Ensure that *_sum is always synced with *_avg

    [ Upstream commit fcf6631f3736985ec89bdd76392d3c7bfb60119f ]

    Rounding in PELT calculation happening when entities are attached/detached
    of a cfs_rq can result into situations where util/runnable_avg is not null
    but util/runnable_sum is. This is normally not possible so we need to
    ensure that util/runnable_sum stays synced with util/runnable_avg.

    detach_entity_load_avg() is the last place where we don't sync
    util/runnable_sum with util/runnbale_avg when moving some sched_entities

    Signed-off-by: Vincent Guittot <vincent.guittot@linaro.org>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Link: https://lkml.kernel.org/r/20210601085832.12626-1-vincent.guittot@linaro.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6d655c27bab2e0991b559a37eaec61cfa6d2e5f8
Author: zpershuai <zpershuai@gmail.com>
Date:   Thu May 27 18:20:57 2021 +0800

    spi: spi-zynq-qspi: Fix some wrong goto jumps & missing error code

    [ Upstream commit f131767eefc47de2f8afb7950cdea78397997d66 ]

    In zynq_qspi_probe function, when enable the device clock is done,
    the return of all the functions should goto the clk_dis_all label.

    If num_cs is not right then this should return a negative error
    code but currently it returns success.

    Signed-off-by: zpershuai <zpershuai@gmail.com>
    Link: https://lore.kernel.org/r/1622110857-21812-1-git-send-email-zpershuai@gmail.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
```

```
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6e47a8167e5d60bac14ec36e17b16f743b80775e
Author: ChiYuan Huang <cy_huang@richtek.com>
Date:   Tue Jun 1 18:09:15 2021 +0800

    regulator: rtmv20: Fix to make regcache value first reading back from HW

    [ Upstream commit 46639a5e684edd0b80ae9dff220f193feb356277 ]

    - Fix to make regcache value first reading back from HW.

    Signed-off-by: ChiYuan Huang <cy_huang@richtek.com>
    Link: https://lore.kernel.org/r/1622542155-6373-1-git-send-email-u0084500@gmail.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fccd7c3574c90adbe9808f18069eda767666bb95
Author: Axel Lin <axel.lin@ingics.com>
Date:   Sun May 30 10:21:09 2021 +0800

    regulator: mt6315: Fix function prototype for mt6315_map_mode

    [ Upstream commit 89082179ec5028bcd58c87171e08ada035689542 ]

    The .of_map_mode should has below function prototype:
            unsigned int (*of_map_mode)(unsigned int mode);

    Signed-off-by: Axel Lin <axel.lin@ingics.com>
    Link: https://lore.kernel.org/r/20210530022109.425054-1-axel.lin@ingics.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e1ffb123e96df255190d429d9b4b01a3f2a3aab1
Author: Nicolas Cavallari <nicolas.cavallari@green-communications.fr>
Date:   Thu May 27 18:34:09 2021 +0200

    ASoC: fsl-asoc-card: Set .owner attribute when registering card.

    [ Upstream commit a8437f05384cb472518ec21bf4fffbe8f0a47378 ]

    Otherwise, when compiled as module, a WARN_ON is triggered:

    WARNING: CPU: 0 PID: 5 at sound/core/init.c:208 snd_card_new+0x310/0x39c [snd]
    [...]
    CPU: 0 PID: 5 Comm: kworker/0:0 Not tainted 5.10.39 #1
    Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
    Workqueue: events deferred_probe_work_func
    [<c0111988>] (unwind_backtrace) from [<c010c8ac>] (show_stack+0x10/0x14)
    [<c010c8ac>] (show_stack) from [<c092784c>] (dump_stack+0xdc/0x104)
    [<c092784c>] (dump_stack) from [<c0129710>] (__warn+0xd8/0x114)
    [<c0129710>] (__warn) from [<c0922a48>] (warn_slowpath_fmt+0x5c/0xc4)
    [<c0922a48>] (warn_slowpath_fmt) from [<bf0496f8>] (snd_card_new+0x310/0x39c [snd])
    [<bf0496f8>] (snd_card_new [snd]) from [<bf1d7df8>] (snd_soc_bind_card+0x334/0x9c4
[snd_soc_core])
    [<bf1d7df8>] (snd_soc_bind_card [snd_soc_core]) from [<bf1e9cd8>]
(devm_snd_soc_register_card+0x30/0x6c [snd_soc_core])
    [<bf1e9cd8>] (devm_snd_soc_register_card [snd_soc_core]) from [<bf22d964>]
(fsl_asoc_card_probe+0x550/0xcc8 [snd_soc_fsl_asoc_card])
    [<bf22d964>] (fsl_asoc_card_probe [snd_soc_fsl_asoc_card]) from [<c060c930>]
(platform_drv_probe+0x48/0x98)
    [...]

    Signed-off-by: Nicolas Cavallari <nicolas.cavallari@green-communications.fr>
    Acked-by: Shengjiu Wang <shengjiu.wang@gmail.com>
```

```
    Link: https://lore.kernel.org/r/20210527163409.22049-1-nicolas.cavallari@green-
communications.fr
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6472955af5e88b5489b6d78316082ad56ea3e489
Author: Tiezhu Yang <yangtiezhu@loongson.cn>
Date:   Wed May 19 18:37:39 2021 +0800

    phy: phy-mtk-tphy: Fix some resource leaks in mtk_phy_init()

    [ Upstream commit aaac9a1bd370338ce372669eb9a6059d16b929aa ]

    Use clk_disable_unprepare() in the error path of mtk_phy_init() to fix
    some resource leaks.

    Reported-by: kernel test robot <lkp@intel.com>
    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Tiezhu Yang <yangtiezhu@loongson.cn>
    Reviewed-by: Chunfeng Yun <chunfeng.yun@mediatek.com>
    Link: https://lore.kernel.org/r/1621420659-15858-1-git-send-email-yangtiezhu@loongson.cn
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b437e028276483c9998fb797fb741e1f080f92f2
Author: Jack Yu <jack.yu@realtek.com>
Date:   Thu May 27 01:06:51 2021 +0000

    ASoC: rt5659: Fix the lost powers for the HDA header

    [ Upstream commit 6308c44ed6eeadf65c0a7ba68d609773ed860fbb ]

    The power of "LDO2", "MICBIAS1" and "Mic Det Power" were powered off after
    the DAPM widgets were added, and these powers were set by the JD settings
    "RT5659_JD_HDA_HEADER" in the probe function. In the codec probe function,
    these powers were ignored to prevent them controlled by DAPM.

    Signed-off-by: Oder Chiou <oder_chiou@realtek.com>
    Signed-off-by: Jack Yu <jack.yu@realtek.com>
    Message-Id: <15fced51977b458798ca4eebf03dafb9@realtek.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f3a4ed2f8168980ba33aa3f07ce3c71c66df04c0
Author: Til Jasper Ullrich <tju@tju.me>
Date:   Tue May 25 17:09:52 2021 +0200

    platform/x86: thinkpad_acpi: Add X1 Carbon Gen 9 second fan support

    [ Upstream commit c0e0436cb4f6627146acdae8c77828f18db01151 ]

    The X1 Carbon Gen 9 uses two fans instead of one like the previous
    generation. This adds support for the second fan. It has been tested
    on my X1 Carbon Gen 9 (20XXS00100) and works fine.

    Signed-off-by: Til Jasper Ullrich <tju@tju.me>
    Link: https://lore.kernel.org/r/20210525150950.14805-1-tju@tju.me
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 83581c57152094eb47bce0f36669cb3f014e6f06
Author: Axel Lin <axel.lin@ingics.com>
Date:   Sun May 23 15:10:44 2021 +0800

    regulator: bd70528: Fix off-by-one for buck123 .n_voltages setting
```

```
    [ Upstream commit 0514582a1a5b4ac1a3fd64792826d392d7ae9ddc ]

    The valid selectors for bd70528 bucks are 0 ~ 0xf, so the .n_voltages
    should be 16 (0x10). Use 0x10 to make it consistent with BD70528_LDO_VOLTS.
    Also remove redundant defines for BD70528_BUCK_VOLTS.

    Signed-off-by: Axel Lin <axel.lin@ingics.com>
    Acked-by: Matti Vaittinen <matti.vaittinen@fi.rohmeurope.com>
    Link: https://lore.kernel.org/r/20210523071045.2168904-1-axel.lin@ingics.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 76f0004671b0539e543bad19aa314df2dcc5ad5b
Author: Axel Lin <axel.lin@ingics.com>
Date:   Wed May 12 15:58:24 2021 +0800

    regulator: cros-ec: Fix error code in dev_err message

    [ Upstream commit 3d681804efcb6e5d8089a433402e19179347d7ae ]

    Show proper error code instead of 0.

    Signed-off-by: Axel Lin <axel.lin@ingics.com>
    Link: https://lore.kernel.org/r/20210512075824.620580-1-axel.lin@ingics.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d11d79e52ba080ee567cb7d7eb42a5ade60a8130
Author: Pavel Skripkin <paskripkin@gmail.com>
Date:   Fri Jun 18 16:49:02 2021 +0300

    net: ethernet: fix potential use-after-free in ec_bhf_remove

    [ Upstream commit 9cca0c2d70149160407bda9a9446ce0c29b6e6c6 ]

    static void ec_bhf_remove(struct pci_dev *dev)
    {
    ...
            struct ec_bhf_priv *priv = netdev_priv(net_dev);

            unregister_netdev(net_dev);
            free_netdev(net_dev);

            pci_iounmap(dev, priv->dma_io);
            pci_iounmap(dev, priv->io);
    ...
    }

    priv is netdev private data, but it is used
    after free_netdev(). It can cause use-after-free when accessing priv
    pointer. So, fix it by moving free_netdev() after pci_iounmap()
    calls.

    Fixes: 6af55ff52b02 ("Driver for Beckhoff CX5020 EtherCAT master module.")
    Signed-off-by: Pavel Skripkin <paskripkin@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9069a7e0dd596c9294131c4537776da2d89a565e
Author: Toke Høiland-Jørgensen <toke@redhat.com>
Date:   Fri Jun 18 13:04:35 2021 +0200

    icmp: don't send out ICMP messages with a source address of 0.0.0.0
```

```
    [ Upstream commit 321827477360934dc040e9d3c626bf1de6c3ab3c ]

    When constructing ICMP response messages, the kernel will try to pick a
    suitable source address for the outgoing packet. However, if no IPv4
    addresses are configured on the system at all, this will fail and we end up
    producing an ICMP message with a source address of 0.0.0.0. This can happen
    on a box routing IPv4 traffic via v6 nexthops, for instance.

    Since 0.0.0.0 is not generally routable on the internet, there's a good
    chance that such ICMP messages will never make it back to the sender of the
    original packet that the ICMP message was sent in response to. This, in
    turn, can create connectivity and PMTUd problems for senders. Fortunately,
    RFC7600 reserves a dummy address to be used as a source for ICMP
    messages (192.0.0.8/32), so let's teach the kernel to substitute that
    address as a last resort if the regular source address selection procedure
    fails.

    Below is a quick example reproducing this issue with network namespaces:

    ip netns add ns0
    ip l add type veth peer netns ns0
    ip l set dev veth0 up
    ip a add 10.0.0.1/24 dev veth0
    ip a add fc00:dead:cafe:42::1/64 dev veth0
    ip r add 10.1.0.0/24 via inet6 fc00:dead:cafe:42::2
    ip -n ns0 l set dev veth0 up
    ip -n ns0 a add fc00:dead:cafe:42::2/64 dev veth0
    ip -n ns0 r add 10.0.0.0/24 via inet6 fc00:dead:cafe:42::1
    ip netns exec ns0 sysctl -w net.ipv4.icmp_ratelimit=0
    ip netns exec ns0 sysctl -w net.ipv4.ip_forward=1
    tcpdump -tpni veth0 -c 2 icmp &
    ping -w 1 10.1.0.1 > /dev/null
    tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
    listening on veth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
    IP 10.0.0.1 > 10.1.0.1: ICMP echo request, id 29, seq 1, length 64
    IP 0.0.0.0 > 10.0.0.1: ICMP net 10.1.0.1 unreachable, length 92
    2 packets captured
    2 packets received by filter
    0 packets dropped by kernel

    With this patch the above capture changes to:
    IP 10.0.0.1 > 10.1.0.1: ICMP echo request, id 31127, seq 1, length 64
    IP 192.0.0.8 > 10.0.0.1: ICMP net 10.1.0.1 unreachable, length 92

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Reported-by: Juliusz Chroboczek <jch@irif.fr>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: Toke Høiland-Jørgensen <toke@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bddd2091e1c4a636cc7243d03e5bd0aaabcb2544
Author: Somnath Kotur <somnath.kotur@broadcom.com>
Date:   Fri Jun 18 02:07:27 2021 -0400

    bnxt_en: Call bnxt_ethtool_free() in bnxt_init_one() error path

    [ Upstream commit 03400aaa69f916a376e11526cf591901a96a3a5c ]

    bnxt_ethtool_init() may have allocated some memory and we need to
    call bnxt_ethtool_free() to properly unwind if bnxt_init_one()
    fails.

    Fixes: 7c3809181468 ("bnxt_en: Refactor bnxt_init_one() and turn on TPA support on 57500
chips.")
```

```
    Signed-off-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0490cea41ab15131e51736d8619aa63841a8ce6f
Author: Rukhsana Ansari <rukhsana.ansari@broadcom.com>
Date:   Fri Jun 18 02:07:26 2021 -0400

    bnxt_en: Fix TQM fastpath ring backing store computation

    [ Upstream commit c12e1643d2738bcd4e26252ce531878841dd3f38 ]

    TQM fastpath ring needs to be sized to store both the requester
    and responder side of RoCE QPs in TQM for supporting bi-directional
    tests.  Fix bnxt_alloc_ctx_mem() to multiply the RoCE QPs by a factor of
    2 when computing the number of entries for TQM fastpath ring.  This
    fixes an RX pipeline stall issue when running bi-directional max
    RoCE QP tests.

    Fixes: c7dd7ab4b204 ("bnxt_en: Improve TQM ring context memory sizing formulas.")
    Signed-off-by: Rukhsana Ansari <rukhsana.ansari@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc5ebaf83af9d880aabbbc37cb9bdaf49efead30
Author: Michael Chan <michael.chan@broadcom.com>
Date:   Fri Jun 18 02:07:25 2021 -0400

    bnxt_en: Rediscover PHY capabilities after firmware reset

    [ Upstream commit 0afd6a4e8028cc487c240b6cfe04094e45a306e4 ]

    There is a missing bnxt_probe_phy() call in bnxt_fw_init_one() to
    rediscover the PHY capabilities after a firmware reset.  This can cause
    some PHY related functionalities to fail after a firmware reset.  For
    example, in multi-host, the ability for any host to configure the PHY
    settings may be lost after a firmware reset.

    Fixes: ec5d31e3c15d ("bnxt_en: Handle firmware reset status during IF_UP.")
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6b3496e07913363151896083a28e228194ed9974
Author: Pavel Machek <pavel@denx.de>
Date:   Fri Jun 18 11:29:48 2021 +0200

    cxgb4: fix wrong shift.

    [ Upstream commit 39eb028183bc7378bb6187067e20bf6d8c836407 ]

    While fixing coverity warning, commit dd2c79677375 introduced typo in
    shift value. Fix that.

    Signed-off-by: Pavel Machek (CIP) <pavel@denx.de>
    Fixes: dd2c79677375 ("cxgb4: Fix unintentional sign extension issues")
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b4f7a9fc9d094c0c4a66f2ad7c37b1dbe9e78f88
Author: Linyu Yuan <linyyuan@codeaurora.org>
Date:   Thu Jun 17 07:32:32 2021 +0800
```

```
    net: cdc_eem: fix tx fixup skb leak

    [ Upstream commit c3b26fdf1b32f91c7a3bc743384b4a298ab53ad7 ]

    when usbnet transmit a skb, eem fixup it in eem_tx_fixup(),
    if skb_copy_expand() failed, it return NULL,
    usbnet_start_xmit() will have no chance to free original skb.

    fix it by free orginal skb in eem_tx_fixup() first,
    then check skb clone status, if failed, return NULL to usbnet.

    Fixes: 9f722c0978b0 ("usbnet: CDC EEM support (v5)")
    Signed-off-by: Linyu Yuan <linyyuan@codeaurora.org>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f4de2b43d13b7cf3ced9310e371b90c836dbd7cd
Author: Pavel Skripkin <paskripkin@gmail.com>
Date:   Wed Jun 16 22:09:06 2021 +0300

    net: hamradio: fix memory leak in mkiss_close

    [ Upstream commit 7edcc682301492380fbdd604b4516af5ae667a13 ]

    My local syzbot instance hit memory leak in
    mkiss_open()[1]. The problem was in missing
    free_netdev() in mkiss_close().

    In mkiss_open() netdevice is allocated and then
    registered, but in mkiss_close() netdevice was
    only unregistered, but not freed.

    Fail log:

    BUG: memory leak
    unreferenced object 0xffff8880281ba000 (size 4096):
      comm "syz-executor.1", pid 11443, jiffies 4295046091 (age 17.660s)
      hex dump (first 32 bytes):
        61 78 30 00 00 00 00 00 00 00 00 00 00 00 00 00  ax0.............
        00 27 fa 2a 80 88 ff ff 00 00 00 00 00 00 00 00  .'.*............
      backtrace:
        [<ffffffff81a27201>] kvmalloc_node+0x61/0xf0
        [<ffffffff8706e7e8>] alloc_netdev_mqs+0x98/0xe80
        [<ffffffff84e64192>] mkiss_open+0xb2/0x6f0 [1]
        [<ffffffff842355db>] tty_ldisc_open+0x9b/0x110
        [<ffffffff84236488>] tty_set_ldisc+0x2e8/0x670
        [<ffffffff8421f7f3>] tty_ioctl+0xda3/0x1440
        [<ffffffff81c9f273>] __x64_sys_ioctl+0x193/0x200
        [<ffffffff8911263a>] do_syscall_64+0x3a/0xb0
        [<ffffffff89200068>] entry_SYSCALL_64_after_hwframe+0x44/0xae

    BUG: memory leak
    unreferenced object 0xffff8880141a9a00 (size 96):
      comm "syz-executor.1", pid 11443, jiffies 4295046091 (age 17.660s)
      hex dump (first 32 bytes):
        e8 a2 1b 28 80 88 ff ff e8 a2 1b 28 80 88 ff ff  ...(.......(....
        98 92 9c aa b0 40 02 00 00 00 00 00 00 00 00 00  .....@..........
      backtrace:
        [<ffffffff8709f68b>] __hw_addr_create_ex+0x5b/0x310
        [<ffffffff8709fb38>] __hw_addr_add_ex+0x1f8/0x2b0
        [<ffffffff870a0c7b>] dev_addr_init+0x10b/0x1f0
        [<ffffffff8706e88b>] alloc_netdev_mqs+0x13b/0xe80
        [<ffffffff84e64192>] mkiss_open+0xb2/0x6f0 [1]
        [<ffffffff842355db>] tty_ldisc_open+0x9b/0x110
```

```
          [<ffffffff84236488>] tty_set_ldisc+0x2e8/0x670
          [<ffffffff8421f7f3>] tty_ioctl+0xda3/0x1440
          [<ffffffff81c9f273>] __x64_sys_ioctl+0x193/0x200
          [<ffffffff8911263a>] do_syscall_64+0x3a/0xb0
          [<ffffffff89200068>] entry_SYSCALL_64_after_hwframe+0x44/0xae

      BUG: memory leak
      unreferenced object 0xffff8880219bfc00 (size 512):
        comm "syz-executor.1", pid 11443, jiffies 4295046091 (age 17.660s)
        hex dump (first 32 bytes):
          00 a0 1b 28 80 88 ff ff 80 8f b1 8d ff ff ff ff  ...(............
          80 8f b1 8d ff ff ff ff 00 00 00 00 00 00 00 00  ................
        backtrace:
          [<ffffffff81a27201>] kvmalloc_node+0x61/0xf0
          [<ffffffff8706eec7>] alloc_netdev_mqs+0x777/0xe80
          [<ffffffff84e64192>] mkiss_open+0xb2/0x6f0 [1]
          [<ffffffff842355db>] tty_ldisc_open+0x9b/0x110
          [<ffffffff84236488>] tty_set_ldisc+0x2e8/0x670
          [<ffffffff8421f7f3>] tty_ioctl+0xda3/0x1440
          [<ffffffff81c9f273>] __x64_sys_ioctl+0x193/0x200
          [<ffffffff8911263a>] do_syscall_64+0x3a/0xb0
          [<ffffffff89200068>] entry_SYSCALL_64_after_hwframe+0x44/0xae

      BUG: memory leak
      unreferenced object 0xffff888029b2b200 (size 256):
        comm "syz-executor.1", pid 11443, jiffies 4295046091 (age 17.660s)
        hex dump (first 32 bytes):
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
        backtrace:
          [<ffffffff81a27201>] kvmalloc_node+0x61/0xf0
          [<ffffffff8706f062>] alloc_netdev_mqs+0x912/0xe80
          [<ffffffff84e64192>] mkiss_open+0xb2/0x6f0 [1]
          [<ffffffff842355db>] tty_ldisc_open+0x9b/0x110
          [<ffffffff84236488>] tty_set_ldisc+0x2e8/0x670
          [<ffffffff8421f7f3>] tty_ioctl+0xda3/0x1440
          [<ffffffff81c9f273>] __x64_sys_ioctl+0x193/0x200
          [<ffffffff8911263a>] do_syscall_64+0x3a/0xb0
          [<ffffffff89200068>] entry_SYSCALL_64_after_hwframe+0x44/0xae

      Fixes: 815f62bf7427 ("[PATCH] SMP rewrite of mkiss")
      Signed-off-by: Pavel Skripkin <paskripkin@gmail.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit fc2fd420b053347669f5507e386bbd89c1ed8b8b
  Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
  Date:   Wed Jun 16 20:43:37 2021 +0200

      be2net: Fix an error handling path in 'be_probe()'

      [ Upstream commit c19c8c0e666f9259e2fc4d2fa4b9ff8e3b40ee5d ]

      If an error occurs after a 'pci_enable_pcie_error_reporting()' call, it
      must be undone by a corresponding 'pci_disable_pcie_error_reporting()'
      call, as already done in the remove function.

      Fixes: d6b6d9877878 ("be2net: use PCIe AER capability")
      Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
      Acked-by: Somnath Kotur <somnath.kotur@broadcom.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit d7aeb00dc5a0790314aa134e85c7164116f9d2c4
  Author: Aya Levin <ayal@nvidia.com>
```

```
Date:   Thu Jun 10 14:20:28 2021 +0300

    net/mlx5: Reset mkey index on creation

    [ Upstream commit 0232fc2ddcf4ffe01069fd1aa07922652120f44a ]

    Reset only the index part of the mkey and keep the variant part. On
    devlink reload, driver recreates mkeys, so the mkey index may change.
    Trying to preserve the variant part of the mkey, driver mistakenly
    merged the mkey index with current value. In case of a devlink reload,
    current value of index part is dirty, so the index may be corrupted.

    Fixes: 54c62e13ad76 ("{IB,net}/mlx5: Setup mkey variant before mr create command
invocation")
    Signed-off-by: Aya Levin <ayal@nvidia.com>
    Signed-off-by: Amir Tzin <amirtz@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a537892fa85ed518ab907d20dcac9e29c54246e1
Author: Dmytro Linkin <dlinkin@nvidia.com>
Date:   Fri May 14 11:14:19 2021 +0300

    net/mlx5e: Don't create devices during unload flow

    [ Upstream commit a5ae8fc9058e37437c8c1f82b3d412b4abd1b9e6 ]

    Running devlink reload command for port in switchdev mode cause
    resources to corrupt: driver can't release allocated EQ and reclaim
    memory pages, because "rdma" auxiliary device had add CQs which blocks
    EQ from deletion.
    Erroneous sequence happens during reload-down phase, and is following:

    1. detach device - suspends auxiliary devices which support it, destroys
       others. During this step "eth-rep" and "rdma-rep" are destroyed,
       "eth" - suspended.
    2. disable SRIOV - moves device to legacy mode; as part of disablement -
       rescans drivers. This step adds "rdma" auxiliary device.
    3. destroy EQ table - <failure>.

    Driver shouldn't create any device during unload flows. To handle that
    implement MLX5_PRIV_FLAGS_DETACH flag, set it on device detach and unset
    on device attach. If flag is set do no-op on drivers rescan.

    Fixes: a925b5e309c9 ("net/mlx5: Register mlx5 devices to auxiliary virtual bus")
    Signed-off-by: Dmytro Linkin <dlinkin@nvidia.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Reviewed-by: Roi Dayan <roid@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7fac9dc2dc09e573b3d1276b6d3ed4ef8ccd7ef3
Author: Alex Vesker <valex@nvidia.com>
Date:   Tue Jun 1 18:10:06 2021 +0300

    net/mlx5: DR, Fix STEv1 incorrect L3 decapsulation padding

    [ Upstream commit 65fb7d109abe3a1a9f1c2d3ba7e1249bc978d5f0 ]

    Decapsulation L3 on small inner packets which are less than
    64 Bytes was done incorrectly. In small packets there is an
    extra padding added in L2 which should not be included in L3
    length. The issue was that after decapL3 the extra L2 padding
    caused an update on the L3 length.
```

```
    To avoid this issue the new header is pushed to the beginning
    of the packet (offset 0) which should not cause a HW reparse
    and update the L3 length.

    Fixes: c349b4137cfd ("net/mlx5: DR, Add STEv1 modify header logic")
    Reviewed-by: Erez Shitrit <erezsh@nvidia.com>
    Reviewed-by: Yevgeny Kliteynik <kliteyn@nvidia.com>
    Signed-off-by: Alex Vesker <valex@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0069be27bf644cc8cc618a2eb5a590c0b4c580c6
Author: Parav Pandit <parav@nvidia.com>
Date:   Thu Jun 10 18:39:53 2021 +0300

    net/mlx5: SF_DEV, remove SF device on invalid state

    [ Upstream commit c7d6c19b3bde66d7aebbe93e0f9e6d9ff57fc3fa ]

    When auxiliary bus autoprobe is disabled and SF is in ACTIVE state,
    on SF port deletion it transitions from ACTIVE->ALLOCATED->INVALID.

    When VHCA event handler queries the state, it is already transition
    to INVALID state.

    In this scenario, event handler missed to delete the SF device.

    Fix it by deleting the SF when SF state is INVALID.

    Fixes: 90d010b8634b ("net/mlx5: SF, Add auxiliary device support")
    Signed-off-by: Parav Pandit <parav@nvidia.com>
    Reviewed-by: Vu Pham <vuhuong@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c08fd2ddb41872336cc86b485675cf67e78e0d45
Author: Parav Pandit <parav@nvidia.com>
Date:   Tue Jun 8 19:03:24 2021 +0300

    net/mlx5: E-Switch, Allow setting GUID for host PF vport

    [ Upstream commit ca36fc4d77b35b8d142cf1ed0eae5ec2e071dc3c ]

    E-switch should be able to set the GUID of host PF vport.
    Currently it returns an error. This results in below error
    when user attempts to configure MAC address of the PF of an
    external controller.

    $ devlink port function set pci/0000:03:00.0/196608 \
       hw_addr 00:00:00:11:22:33

    mlx5_core 0000:03:00.0: mlx5_esw_set_vport_mac_locked:1876:(pid 6715):\
    "Failed to set vport 0 node guid, err = -22.
    RDMA_CM will not function properly for this VF."

    Check for zero vport is no longer needed.

    Fixes: 330077d14de1 ("net/mlx5: E-switch, Supporting setting devlink port function mac
address")
    Signed-off-by: Yuval Avnery <yuvalav@nvidia.com>
    Signed-off-by: Parav Pandit <parav@nvidia.com>
    Reviewed-by: Bodong Wang <bodong@nvidia.com>
    Reviewed-by: Alaa Hleihel <alaa@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
```

```
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 648a07c4d5de07170e9a0196509dd7bddef1dec8
Author: Parav Pandit <parav@nvidia.com>
Date:   Tue Jun 8 19:14:08 2021 +0300

    net/mlx5: E-Switch, Read PF mac address

    [ Upstream commit bbc8222dc49db8d49add0f27bcac33f4b92193dc ]

    External controller PF's MAC address is not read from the device during
    vport setup. Fail to read this results in showing all zeros to user
    while the factory programmed MAC is a valid value.

    $ devlink port show eth1 -jp
    {
        "port": {
            "pci/0000:03:00.0/196608": {
                "type": "eth",
                "netdev": "eth1",
                "flavour": "pcipf",
                "controller": 1,
                "pfnum": 0,
                "splittable": false,
                "function": {
                    "hw_addr": "00:00:00:00:00:00"
                }
            }
        }
    }

    Hence, read it when enabling a vport.

    After the fix,

    $ devlink port show eth1 -jp
    {
        "port": {
            "pci/0000:03:00.0/196608": {
                "type": "eth",
                "netdev": "eth1",
                "flavour": "pcipf",
                "controller": 1,
                "pfnum": 0,
                "splittable": false,
                "function": {
                    "hw_addr": "98:03:9b:a0:60:11"
                }
            }
        }
    }

    Fixes: f099fde16db3 ("net/mlx5: E-switch, Support querying port function mac address")
    Signed-off-by: Bodong Wang <bodong@nvidia.com>
    Signed-off-by: Parav Pandit <parav@nvidia.com>
    Reviewed-by: Alaa Hleihel <alaa@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1666c186fe8a952a8cb1c22964394913005f1ff9
Author: Leon Romanovsky <leon@kernel.org>
Date:   Sun Mar 21 19:57:14 2021 +0200

    net/mlx5: Check that driver was probed prior attaching the device
```

```
    [ Upstream commit 2058cc9c8041fde9c0bdd8e868c72b137cff8563 ]

    The device can be requested to be attached despite being not probed.
    This situation is possible if devlink reload races with module removal,
    and the following kernel panic is an outcome of such race.

     mlx5_core 0000:00:09.0: firmware version: 4.7.9999
     mlx5_core 0000:00:09.0: 0.000 Gb/s available PCIe bandwidth (8.0 GT/s PCIe x255 link)
     BUG: unable to handle page fault for address: fffffffffffffff0
     #PF: supervisor read access in kernel mode
     #PF: error_code(0x0000) - not-present page
     PGD 3218067 P4D 3218067 PUD 321a067 PMD 0
     Oops: 0000 [#1] SMP KASAN NOPTI
     CPU: 7 PID: 250 Comm: devlink Not tainted 5.12.0-rc2+ #2836
     Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-
prebuilt.qemu.org 04/01/2014
     RIP: 0010:mlx5_attach_device+0x80/0x280 [mlx5_core]
     Code: f8 48 c1 e8 03 42 80 3c 38 00 0f 85 80 01 00 00 48 8b 45 68 48 8d 78 f0 48 89 fe
48 c1 ee 03 42 80 3c 3e 00 0f 85 70 01 00 00 <48> 8b 40 f0 48 85 c0 74 0d 48 89 ef ff d0 85
c0 0f 85 84 05 0e 00
     RSP: 0018:ffff8880129675f0 EFLAGS: 00010246
     RAX: 0000000000000000 RBX: 0000000000000001 RCX: ffffffff827407f1
     RDX: 1ffff110011336cf RSI: 1fffffffffffffe RDI: fffffffffffffff0
     RBP: ffff888008e0c000 R08: 0000000000000008 R09: ffffffffa0662ee7
     R10: fffffbfff40cc5dc R11: 0000000000000000 R12: ffff88800ea002e0
     R13: ffffed1001d459f7 R14: ffffffffa05ef4f8 R15: dffffc0000000000
     FS:  00007f51dfeaf740(0000) GS:ffff88806d5c0000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     CR2: fffffffffffffff0 CR3: 000000000bc82006 CR4: 0000000000370ea0
     DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     Call Trace:
      mlx5_load_one+0x117/0x1d0 [mlx5_core]
      devlink_reload+0x2d5/0x520
      ? devlink_remote_reload_actions_performed+0x30/0x30
      ? mutex_trylock+0x24b/0x2d0
      ? devlink_nl_cmd_reload+0x62b/0x1070
      devlink_nl_cmd_reload+0x66d/0x1070
      ? devlink_reload+0x520/0x520
      ? devlink_nl_pre_doit+0x64/0x4d0
      genl_family_rcv_msg_doit+0x1e9/0x2f0
      ? mutex_lock_io_nested+0x1130/0x1130
      ? genl_family_rcv_msg_attrs_parse.constprop.0+0x240/0x240
      ? security_capable+0x51/0x90
      genl_rcv_msg+0x27f/0x4a0
      ? genl_get_cmd+0x3c0/0x3c0
      ? lock_acquire+0x1a9/0x6d0
      ? devlink_reload+0x520/0x520
      ? lock_release+0x6c0/0x6c0
      netlink_rcv_skb+0x11d/0x340
      ? genl_get_cmd+0x3c0/0x3c0
      ? netlink_ack+0x9f0/0x9f0
      ? lock_release+0x1f9/0x6c0
      genl_rcv+0x24/0x40
      netlink_unicast+0x433/0x700
      ? netlink_attachskb+0x730/0x730
      ? _copy_from_iter_full+0x178/0x650
      ? __alloc_skb+0x113/0x2b0
      netlink_sendmsg+0x6f1/0xbd0
      ? netlink_unicast+0x700/0x700
      ? netlink_unicast+0x700/0x700
      sock_sendmsg+0xb0/0xe0
      __sys_sendto+0x193/0x240
      ? __x64_sys_getpeername+0xb0/0xb0
      ? copy_page_range+0x2300/0x2300
```

```
       ? __up_read+0x1a1/0x7b0
       ? do_user_addr_fault+0x219/0xdc0
       __x64_sys_sendto+0xdd/0x1b0
       ? syscall_enter_from_user_mode+0x1d/0x50
       do_syscall_64+0x2d/0x40
       entry_SYSCALL_64_after_hwframe+0x44/0xae
      RIP: 0033:0x7f51dffb514a
      Code: d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 f3 0f 1e fa 41 89 ca 64 8b 04 25
 18 00 00 00 85 c0 75 15 b8 2c 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 76 c3 0f 1f 44 00 00 55
 48 83 ec 30 44 89 4c
      RSP: 002b:00007ffcaef22e78 EFLAGS: 00000246 ORIG_RAX: 000000000000002c
      RAX: ffffffffffffffda RBX: 0000000000000003 RCX: 00007f51dffb514a
      RDX: 0000000000000030 RSI: 000055750daf2440 RDI: 0000000000000003
      RBP: 000055750daf2410 R08: 00007f51e0081200 R09: 000000000000000c
      R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000
      R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
      Modules linked in: mlx5_core(-) ptp pps_core ib_ipoib rdma_ucm rdma_cm iw_cm ib_cm
 ib_umad ib_uverbs ib_core [last unloaded: mlx5_ib]
      CR2: fffffffffffffff0
      ---[ end trace 7789831bfe74fa42 ]---

    Fixes: a925b5e309c9 ("net/mlx5: Register mlx5 devices to auxiliary virtual bus")
    Signed-off-by: Leon Romanovsky <leonro@nvidia.com>
    Reviewed-by: Parav Pandit <parav@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit e384aeadab1b61d3e47cc1a3991b0006cb707e8a
 Author: Leon Romanovsky <leon@kernel.org>
 Date:   Mon Mar 8 15:41:55 2021 +0200

    net/mlx5: Fix error path for set HCA defaults

    [ Upstream commit 94a4b8414d3e91104873007b659252f855ee344a ]

    In the case of the failure to execute mlx5_core_set_hca_defaults(),
    we used wrong goto label to execute error unwind flow.

    Fixes: 5bef709d76a2 ("net/mlx5: Enable host PF HCA after eswitch is initialized")
    Reviewed-by: Saeed Mahameed <saeedm@nvidia.com>
    Reviewed-by: Moshe Shemesh <moshe@nvidia.com>
    Signed-off-by: Leon Romanovsky <leonro@nvidia.com>
    Reviewed-by: Parav Pandit <parav@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 3cbfeea44b8d0444a643d2817cb1109cb1765eb2
 Author: Eric Dumazet <edumazet@google.com>
 Date:   Wed Jun 16 07:47:15 2021 -0700

    net/af_unix: fix a data-race in unix_dgram_sendmsg / unix_release_sock

    [ Upstream commit a494bd642d9120648b06bb7d28ce6d05f55a7819 ]

    While unix_may_send(sk, osk) is called while osk is locked, it appears
    unix_release_sock() can overwrite unix_peer() after this lock has been
    released, making KCSAN unhappy.

    Changing unix_release_sock() to access/change unix_peer()
    before lock is released should fix this issue.

    BUG: KCSAN: data-race in unix_dgram_sendmsg / unix_release_sock

    write to 0xffff88810465a338 of 8 bytes by task 20852 on cpu 1:
     unix_release_sock+0x4ed/0x6e0 net/unix/af_unix.c:558
```

```
       unix_release+0x2f/0x50 net/unix/af_unix.c:859
       __sock_release net/socket.c:599 [inline]
       sock_close+0x6c/0x150 net/socket.c:1258
       __fput+0x25b/0x4e0 fs/file_table.c:280
       ____fput+0x11/0x20 fs/file_table.c:313
       task_work_run+0xae/0x130 kernel/task_work.c:164
       tracehook_notify_resume include/linux/tracehook.h:189 [inline]
       exit_to_user_mode_loop kernel/entry/common.c:175 [inline]
       exit_to_user_mode_prepare+0x156/0x190 kernel/entry/common.c:209
       __syscall_exit_to_user_mode_work kernel/entry/common.c:291 [inline]
       syscall_exit_to_user_mode+0x20/0x40 kernel/entry/common.c:302
       do_syscall_64+0x56/0x90 arch/x86/entry/common.c:57
       entry_SYSCALL_64_after_hwframe+0x44/0xae

      read to 0xffff88810465a338 of 8 bytes by task 20888 on cpu 0:
       unix_may_send net/unix/af_unix.c:189 [inline]
       unix_dgram_sendmsg+0x923/0x1610 net/unix/af_unix.c:1712
       sock_sendmsg_nosec net/socket.c:654 [inline]
       sock_sendmsg net/socket.c:674 [inline]
       ____sys_sendmsg+0x360/0x4d0 net/socket.c:2350
       ___sys_sendmsg net/socket.c:2404 [inline]
       __sys_sendmmsg+0x315/0x4b0 net/socket.c:2490
       __do_sys_sendmmsg net/socket.c:2519 [inline]
       __se_sys_sendmmsg net/socket.c:2516 [inline]
       __x64_sys_sendmmsg+0x53/0x60 net/socket.c:2516
       do_syscall_64+0x4a/0x90 arch/x86/entry/common.c:47
       entry_SYSCALL_64_after_hwframe+0x44/0xae

      value changed: 0xffff888167905400 -> 0x0000000000000000

      Reported by Kernel Concurrency Sanitizer on:
      CPU: 0 PID: 20888 Comm: syz-executor.0 Not tainted 5.13.0-rc5-syzkaller #0
      Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
  01/01/2011

      Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
      Signed-off-by: Eric Dumazet <edumazet@google.com>
      Reported-by: syzbot <syzkaller@googlegroups.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 77de6ee73f54a9a89c0afa0bf4c53b239aa9953a
  Author: Chengyang Fan <cy.fan@huawei.com>
  Date:   Wed Jun 16 17:59:25 2021 +0800

      net: ipv4: fix memory leak in ip_mc_add1_src

      [ Upstream commit d8e2973029b8b2ce477b564824431f3385c77083 ]

      BUG: memory leak
      unreferenced object 0xffff888101bc4c00 (size 32):
        comm "syz-executor527", pid 360, jiffies 4294807421 (age 19.329s)
        hex dump (first 32 bytes):
          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
          01 00 00 00 00 00 00 00 ac 14 14 bb 00 00 02 00 ................
        backtrace:
          [<00000000f17c5244>] kmalloc include/linux/slab.h:558 [inline]
          [<00000000f17c5244>] kzalloc include/linux/slab.h:688 [inline]
          [<00000000f17c5244>] ip_mc_add1_src net/ipv4/igmp.c:1971 [inline]
          [<00000000f17c5244>] ip_mc_add_src+0x95f/0xdb0 net/ipv4/igmp.c:2095
          [<000000001cb99709>] ip_mc_source+0x84c/0xea0 net/ipv4/igmp.c:2416
          [<0000000052cf19ed>] do_ip_setsockopt net/ipv4/ip_sockglue.c:1294 [inline]
          [<0000000052cf19ed>] ip_setsockopt+0x114b/0x30c0 net/ipv4/ip_sockglue.c:1423
          [<00000000477edfbc>] raw_setsockopt+0x13d/0x170 net/ipv4/raw.c:857
          [<00000000e75ca9bb>] __sys_setsockopt+0x158/0x270 net/socket.c:2117
```

```
         [<00000000bdb993a8>] __do_sys_setsockopt net/socket.c:2128 [inline]
         [<00000000bdb993a8>] __se_sys_setsockopt net/socket.c:2125 [inline]
         [<00000000bdb993a8>] __x64_sys_setsockopt+0xba/0x150 net/socket.c:2125
         [<000000006a1ffdbd>] do_syscall_64+0x40/0x80 arch/x86/entry/common.c:47
         [<00000000b11467c4>] entry_SYSCALL_64_after_hwframe+0x44/0xae

    In commit 24803f38a5c0 ("igmp: do not remove igmp souce list info when set
    link down"), the ip_mc_clear_src() in ip_mc_destroy_dev() was removed,
    because it was also called in igmpv3_clear_delrec().

    Rough callgraph:

    inetdev_destroy
    -> ip_mc_destroy_dev
         -> igmpv3_clear_delrec
            -> ip_mc_clear_src
    -> RCU_INIT_POINTER(dev->ip_ptr, NULL)

    However, ip_mc_clear_src() called in igmpv3_clear_delrec() doesn't
    release in_dev->mc_list->sources. And RCU_INIT_POINTER() assigns the
    NULL to dev->ip_ptr. As a result, in_dev cannot be obtained through
    inetdev_by_index() and then in_dev->mc_list->sources cannot be released
    by ip_mc_del1_src() in the sock_close. Rough call sequence goes like:

    sock_close
    -> __sock_release
       -> inet_release
          -> ip_mc_drop_socket
             -> inetdev_by_index
             -> ip_mc_leave_src
                -> ip_mc_del_src
                   -> ip_mc_del1_src

    So we still need to call ip_mc_clear_src() in ip_mc_destroy_dev() to free
    in_dev->mc_list->sources.

    Fixes: 24803f38a5c0 ("igmp: do not remove igmp souce list info ...")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: Chengyang Fan <cy.fan@huawei.com>
    Acked-by: Hangbin Liu <liuhangbin@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a993bca5ba69c63213c81bc3e81f61cd733b007
Author: Joakim Zhang <qiangqing.zhang@nxp.com>
Date:   Wed Jun 16 17:14:26 2021 +0800

    net: fec_ptp: fix issue caused by refactor the fec_devtype

    [ Upstream commit d23765646e71b43ed2b809930411ba5c0aadee7b ]

    Commit da722186f654 ("net: fec: set GPR bit on suspend by DT configuration.")
    refactor the fec_devtype, need adjust ptp driver accordingly.

    Fixes: da722186f654 ("net: fec: set GPR bit on suspend by DT configuration.")
    Signed-off-by: Joakim Zhang <qiangqing.zhang@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 14616c372a7be01a2fb8c56c9d8debd232b9e43d
Author: Dongliang Mu <mudongliangabcd@gmail.com>
Date:   Wed Jun 16 10:48:33 2021 +0800

    net: usb: fix possible use-after-free in smsc75xx_bind
```

```
    [ Upstream commit 56b786d86694e079d8aad9b314e015cd4ac02a3d ]

    The commit 46a8b29c6306 ("net: usb: fix memory leak in smsc75xx_bind")
    fails to clean up the work scheduled in smsc75xx_reset->
    smsc75xx_set_multicast, which leads to use-after-free if the work is
    scheduled to start after the deallocation. In addition, this patch
    also removes a dangling pointer - dev->data[0].

    This patch calls cancel_work_sync to cancel the scheduled work and set
    the dangling pointer to NULL.

    Fixes: 46a8b29c6306 ("net: usb: fix memory leak in smsc75xx_bind")
    Signed-off-by: Dongliang Mu <mudongliangabcd@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5e006cdb9b759f604c4fc69b410aab37cf45f5b4
Author: Aleksander Jan Bajkowski <olek2@wp.pl>
Date:   Tue Jun 15 22:42:57 2021 +0200

    lantiq: net: fix duplicated skb in rx descriptor ring

    [ Upstream commit 7ea6cd16f1599c1eac6018751eadbc5fc736b99a ]

    The previous commit didn't fix the bug properly. By mistake, it replaces
    the pointer of the next skb in the descriptor ring instead of the current
    one. As a result, the two descriptors are assigned the same SKB. The error
    is seen during the iperf test when skb_put tries to insert a second packet
    and exceeds the available buffer.

    Fixes: c7718ee96dbc ("net: lantiq: fix memory corruption in RX ring ")
    Signed-off-by: Aleksander Jan Bajkowski <olek2@wp.pl>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 62e2f20e2e99662ffa1c87762f7241b93446d221
Author: Maciej Żenczykowski <maze@google.com>
Date:   Tue Jun 15 01:05:49 2021 -0700

    net: cdc_ncm: switch to eth%d interface naming

    [ Upstream commit c1a3d4067309451e68c33dbd356032549cc0bd8e ]

    This is meant to make the host side cdc_ncm interface consistently
    named just like the older CDC protocols: cdc_ether & cdc_ecm
    (and even rndis_host), which all use 'FLAG_ETHER | FLAG_POINTTOPOINT'.

    include/linux/usb/usbnet.h:
      #define FLAG_ETHER    0x0020          /* maybe use "eth%d" names */
      #define FLAG_WLAN     0x0080          /* use "wlan%d" names */
      #define FLAG_WWAN     0x0400          /* use "wwan%d" names */
      #define FLAG_POINTTOPOINT 0x1000      /* possibly use "usb%d" names */

    drivers/net/usb/usbnet.c @ line 1711:
      strcpy (net->name, "usb%d");
      ...
      // heuristic:  "usb%d" for links we know are two-host,
      // else "eth%d" when there's reasonable doubt.  userspace
      // can rename the link if it knows better.
      if ((dev->driver_info->flags & FLAG_ETHER) != 0 &&
          ((dev->driver_info->flags & FLAG_POINTTOPOINT) == 0 ||
           (net->dev_addr [0] & 0x02) == 0))
              strcpy (net->name, "eth%d");
      /* WLAN devices should always be named "wlan%d" */
      if ((dev->driver_info->flags & FLAG_WLAN) != 0)
```

```
                strcpy(net->name, "wlan%d");
        /* WWAN devices should always be named "wwan%d" */
        if ((dev->driver_info->flags & FLAG_WWAN) != 0)
                strcpy(net->name, "wwan%d");
```

So by using ETHER | POINTTOPOINT the interface naming is
either usb%d or eth%d based on the global uniqueness of the
mac address of the device.

Without this 2.5gbps ethernet dongles which all seem to use the cdc_ncm
driver end up being called usb%d instead of eth%d even though they're
definitely not two-host.  (All 1gbps & 5gbps ethernet usb dongles I've
tested don't hit this problem due to use of different drivers, primarily
r8152 and aqc111)

Fixes tag is based purely on git blame, and is really just here to make
sure this hits LTS branches newer than v4.5.

Cc: Lorenzo Colitti <lorenzo@google.com>
Fixes: 4d06dd537f95 ("cdc_ncm: do not call usbnet_link_change from cdc_ncm_bind")
Signed-off-by: Maciej Żenczykowski <maze@google.com>
Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 220c3c36b1f846dab94842e2827bf685d817ac17
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Mon Jun 14 15:24:05 2021 -0700

ptp: improve max_adj check against unreasonable values

[ Upstream commit 475b92f932168a78da8109acd10bfb7578b8f2bb ]

Scaled PPM conversion to PPB may (on 64bit systems) result
in a value larger than s32 can hold (freq/scaled_ppm is a long).
This means the kernel will not correctly reject unreasonably
high ->freq values (e.g. > 4294967295ppb, 281474976645 scaled PPM).

The conversion is equivalent to a division by ~66 (65.536),
so the value of ppb is always smaller than ppm, but not small
enough to assume narrowing the type from long -> s32 is okay.

Note that reasonable user space (e.g. ptp4l) will not use such
high values, anyway, 4289046510ppb ~= 4.3x, so the fix is
somewhat pedantic.

Fixes: d39a743511cd ("ptp: validate the requested frequency adjustment.")
Fixes: d94ba80ebbea ("ptp: Added a brand new class driver for ptp clocks.")
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Acked-by: Richard Cochran <richardcochran@gmail.com>
Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7d14c66f906c7f4c4abd91c112e6cde06092f2ae
Author: Subash Abhinov Kasiviswanathan <subashab@codeaurora.org>
Date:   Mon Jun 14 15:03:25 2021 -0600

net: mhi_net: Update the transmit handler prototype

[ Upstream commit 2214fb53006e6cfa6371b706070cb99794c68c3b ]

Update the function prototype of mhi_ndo_xmit to match
ndo_start_xmit. This otherwise leads to run time failures when
CFI is enabled in kernel.

Fixes: 3ffec6a14f24 ("net: Add mhi-net driver")

```
    Signed-off-by: Subash Abhinov Kasiviswanathan <subashab@codeaurora.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4a99047ed51c98a09a537fe2c12420d815dfe296
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Fri May 28 15:47:32 2021 +0000

    bpf: Fix leakage under speculation on mispredicted branches

    [ Upstream commit 9183671af6dbf60a1219371d4ed73e23f43b49db ]

    The verifier only enumerates valid control-flow paths and skips paths that
    are unreachable in the non-speculative domain. And so it can miss issues
    under speculative execution on mispredicted branches.

    For example, a type confusion has been demonstrated with the following
    crafted program:

      // r0 = pointer to a map array entry
      // r6 = pointer to readable stack slot
      // r9 = scalar controlled by attacker
      1: r0 = *(u64 *)(r0) // cache miss
      2: if r0 != 0x0 goto line 4
      3: r6 = r9
      4: if r0 != 0x1 goto line 6
      5: r9 = *(u8 *)(r6)
      6: // leak r9

    Since line 3 runs iff r0 == 0 and line 5 runs iff r0 == 1, the verifier
    concludes that the pointer dereference on line 5 is safe. But: if the
    attacker trains both the branches to fall-through, such that the following
    is speculatively executed ...

      r6 = r9
      r9 = *(u8 *)(r6)
      // leak r9

    ... then the program will dereference an attacker-controlled value and could
    leak its content under speculative execution via side-channel. This requires
    to mistrain the branch predictor, which can be rather tricky, because the
    branches are mutually exclusive. However such training can be done at
    congruent addresses in user space using different branches that are not
    mutually exclusive. That is, by training branches in user space ...

      A:  if r0 != 0x0 goto line C
      B:  ...
      C:  if r0 != 0x0 goto line D
      D:  ...

    ... such that addresses A and C collide to the same CPU branch prediction
    entries in the PHT (pattern history table) as those of the BPF program's
    lines 2 and 4, respectively. A non-privileged attacker could simply brute
    force such collisions in the PHT until observing the attack succeeding.

    Alternative methods to mistrain the branch predictor are also possible that
    avoid brute forcing the collisions in the PHT. A reliable attack has been
    demonstrated, for example, using the following crafted program:

      // r0 = pointer to a [control] map array entry
      // r7 = *(u64 *)(r0 + 0), training/attack phase
      // r8 = *(u64 *)(r0 + 8), oob address
      // [...]
      // r0 = pointer to a [data] map array entry
      1: if r7 == 0x3 goto line 3
```

```
  2: r8 = r0
  // crafted sequence of conditional jumps to separate the conditional
  // branch in line 193 from the current execution flow
  3: if r0 != 0x0 goto line 5
  4: if r0 == 0x0 goto exit
  5: if r0 != 0x0 goto line 7
  6: if r0 == 0x0 goto exit
  [...]
  187: if r0 != 0x0 goto line 189
  188: if r0 == 0x0 goto exit
  // load any slowly-loaded value (due to cache miss in phase 3) ...
  189: r3 = *(u64 *)(r0 + 0x1200)
  // ... and turn it into known zero for verifier, while preserving slowly-
  // loaded dependency when executing:
  190: r3 &= 1
  191: r3 &= 2
  // speculatively bypassed phase dependency
  192: r7 += r3
  193: if r7 == 0x3 goto exit
  194: r4 = *(u8 *)(r8 + 0)
  // leak r4
```

As can be seen, in training phase (phase != 0x3), the condition in line 1 turns into false and therefore r8 with the oob address is overridden with the valid map value address, which in line 194 we can read out without issues. However, in attack phase, line 2 is skipped, and due to the cache miss in line 189 where the map value is (zeroed and later) added to the phase register, the condition in line 193 takes the fall-through path due to prior branch predictor training, where under speculation, it'll load the byte at oob address r8 (unknown scalar type at that point) which could then be leaked via side-channel.

One way to mitigate these is to 'branch off' an unreachable path, meaning, the current verification path keeps following the is_branch_taken() path and we push the other branch to the verification stack. Given this is unreachable from the non-speculative domain, this branch's vstate is explicitly marked as speculative. This is needed for two reasons: i) if this path is solely seen from speculative execution, then we later on still want the dead code elimination to kick in in order to sanitize these instructions with jmp-1s, and ii) to ensure that paths walked in the non-speculative domain are not pruned from earlier walks of paths walked in the speculative domain. Additionally, for robustness, we mark the registers which have been part of the conditional as unknown in the speculative path given there should be no assumptions made on their content.

The fix in here mitigates type confusion attacks described earlier due to i) all code paths in the BPF program being explored and ii) existing verifier logic already ensuring that given memory access instruction references one specific data structure.

An alternative to this fix that has also been looked at in this scope was to mark aux->alu_state at the jump instruction with a BPF_JMP_TAKEN state as well as direction encoding (always-goto, always-fallthrough, unknown), such that mixing of different always-* directions themselves as well as mixing of always-* with unknown directions would cause a program rejection by the verifier, e.g. programs with constructs like 'if ([...]) { x = 0; } else { x = 1; }' with subsequent 'if (x == 1) { [...] }'. For unprivileged, this would result in only single direction always-* taken paths, and unknown taken paths being allowed, such that the former could be patched from a conditional jump to an unconditional jump (ja). Compared to this approach here, it would have two downsides: i) valid programs that otherwise are not performing any pointer arithmetic, etc, would potentially be rejected/broken, and ii) we are required to turn off path pruning for unprivileged, where both can be avoided in this work through pushing the invalid branch to the verification stack.

```
    The issue was originally discovered by Adam and Ofek, and later independently
    discovered and reported as a result of Benedict and Piotr's research work.

    Fixes: b2157399cc98 ("bpf: prevent out-of-bounds speculation")
    Reported-by: Adam Morrison <mad@cs.tau.ac.il>
    Reported-by: Ofek Kirzner <ofekkir@gmail.com>
    Reported-by: Benedict Schlueter <benedict.schlueter@rub.de>
    Reported-by: Piotr Krysiuk <piotras@gmail.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Reviewed-by: John Fastabend <john.fastabend@gmail.com>
    Reviewed-by: Benedict Schlueter <benedict.schlueter@rub.de>
    Reviewed-by: Piotr Krysiuk <piotras@gmail.com>
    Acked-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 19892ab9c9d838e2e5a7744d36e4bb8b7c3292fe
Author: Pavel Skripkin <paskripkin@gmail.com>
Date:   Mon Jun 14 15:06:50 2021 +0300

    net: qrtr: fix OOB Read in qrtr_endpoint_post

    [ Upstream commit ad9d24c9429e2159d1e279dc3a83191ccb4daf1d ]

    Syzbot reported slab-out-of-bounds Read in
    qrtr_endpoint_post. The problem was in wrong
    _size_ type:

            if (len != ALIGN(size, 4) + hdrlen)
                    goto err;

    If size from qrtr_hdr is 4294967293 (0xfffffffd), the result of
    ALIGN(size, 4) will be 0. In case of len == hdrlen and size == 4294967293
    in header this check won't fail and

            skb_put_data(skb, data + hdrlen, size);

    will read out of bound from data, which is hdrlen allocated block.

    Fixes: 194ccc88297a ("net: qrtr: Support decoding incoming v2 packets")
    Reported-and-tested-by: syzbot+1917d778024161609247@syzkaller.appspotmail.com
    Signed-off-by: Pavel Skripkin <paskripkin@gmail.com>
    Reviewed-by: Bjorn Andersson <bjorn.andersson@linaro.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 55c6d93e0b380a7c55dde39e18756d93f3295996
Author: David Ahern <dsahern@kernel.org>
Date:   Sat Jun 12 18:24:59 2021 -0600

    ipv4: Fix device used for dst_alloc with local routes

    [ Upstream commit b87b04f5019e821c8c6c7761f258402e43500a1f ]

    Oliver reported a use case where deleting a VRF device can hang
    waiting for the refcnt to drop to 0. The root cause is that the dst
    is allocated against the VRF device but cached on the loopback
    device.

    The use case (added to the selftests) has an implicit VRF crossing
    due to the ordering of the FIB rules (lookup local is before the
    l3mdev rule, but the problem occurs even if the FIB rules are
    re-ordered with local after l3mdev because the VRF table does not
    have a default route to terminate the lookup). The end result is
    is that the FIB lookup returns the loopback device as the nexthop,
    but the ingress device is in a VRF. The mismatch causes the dst
```

```
    alloc against the VRF device but then cached on the loopback.

    The fix is to bring the trick used for IPv6 (see ip6_rt_get_dev_rcu):
    pick the dst alloc device based the fib lookup result but with checks
    that the result has a nexthop device (e.g., not an unreachable or
    prohibit entry).

    Fixes: f5a0aab84b74 ("net: ipv4: dst for local input routes should use l3mdev if
relevant")
    Reported-by: Oliver Herms <oliver.peter.herms@gmail.com>
    Signed-off-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 490e879c3848f38881f1f04cb76b14435210b291
Author: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
Date:   Sat Jun 12 19:20:44 2021 +0530

    cxgb4: fix wrong ethtool n-tuple rule lookup

    [ Upstream commit 09427c1915f754ebe7d3d8e54e79bbee48afe916 ]

    The TID returned during successful filter creation is relative to
    the region in which the filter is created. Using it directly always
    returns Hi Prio/Normal filter region's entry for the first couple of
    entries, even though the rule is actually inserted in Hash region.
    Fix by analyzing in which region the filter has been inserted and
    save the absolute TID to be used for lookup later.

    Fixes: db43b30cd89c ("cxgb4: add ethtool n-tuple filter deletion")
    Signed-off-by: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 45988cab933e9b59fd15217be8b544f61dc9392a
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Sat Jun 12 14:53:12 2021 +0200

    netxen_nic: Fix an error handling path in 'netxen_nic_probe()'

    [ Upstream commit 49a10c7b176295f8fafb338911cf028e97f65f4d ]

    If an error occurs after a 'pci_enable_pcie_error_reporting()' call, it
    must be undone by a corresponding 'pci_disable_pcie_error_reporting()'
    call, as already done in the remove function.

    Fixes: e87ad5539343 ("netxen: support pci error handlers")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ca9c08db4a8ab36e9d2cf038ce1c3645524ebdde
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Sat Jun 12 14:37:46 2021 +0200

    qlcnic: Fix an error handling path in 'qlcnic_probe()'

    [ Upstream commit cb3376604a676e0302258b01893911bdd7aa5278 ]

    If an error occurs after a 'pci_enable_pcie_error_reporting()' call, it
    must be undone by a corresponding 'pci_disable_pcie_error_reporting()'
    call, as already done in the remove function.

    Fixes: 451724c821c1 ("qlcnic: aer support")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
```

```
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cfc7f0e70d649e6d2233fba0d9390b525677d971
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Fri Jun 11 18:49:48 2021 -0700

    ethtool: strset: fix message length calculation

    [ Upstream commit e175aef902697826d344ce3a12189329848fe898 ]

    Outer nest for ETHTOOL_A_STRSET_STRINGSETS is not accounted for.
    This may result in ETHTOOL_MSG_STRSET_GET producing a warning like:

        calculated message payload length (684) not sufficient
        WARNING: CPU: 0 PID: 30967 at net/ethtool/netlink.c:369
ethnl_default_doit+0x87a/0xa20

    and a splat.

    As usually with such warnings three conditions must be met for the warning
    to trigger:
     - there must be no skb size rounding up (e.g. reply_size of 684);
     - string set must be per-device (so that the header gets populated);
     - the device name must be at least 12 characters long.

    all in all with current user space it looks like reading priv flags
    is the only place this could potentially happen. Or with syzbot :)

    Reported-by: syzbot+59aa77b92d06cd5a54f2@syzkaller.appspotmail.com
    Fixes: 71921690f974 ("ethtool: provide string sets with STRSET_GET request")
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4556e8ed3a82fd5d24eda17368ff758089c7b527
Author: Alex Elder <elder@linaro.org>
Date:   Fri Jun 11 13:26:00 2021 -0500

    net: qualcomm: rmnet: don't over-count statistics

    [ Upstream commit 994c393bb6886d6d94d628475b274a8cb3fc67a4 ]

    The purpose of the loop using u64_stats_fetch_*_irq() is to ensure
    statistics on a given CPU are collected atomically. If one of the
    statistics values gets updated within the begin/retry window, the
    loop will run again.

    Currently the statistics totals are updated inside that window.
    This means that if the loop ever retries, the statistics for the
    CPU will be counted more than once.

    Fix this by taking a snapshot of a CPU's statistics inside the
    protected window, and then updating the counters with the snapshot
    values after exiting the loop.

    (Also add a newline at the end of this file...)

    Fixes: 192c4b5d48f2a ("net: qualcomm: rmnet: Add support for 64 bit stats")
    Signed-off-by: Alex Elder <elder@linaro.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5816edd370a7d67879259922bb573ee4af164808
Author: Changbin Du <changbin.du@intel.com>
```

```
Date:   Fri Jun 11 22:29:59 2021 +0800

    net: make get_net_ns return error if NET_NS is disabled

    [ Upstream commit ea6932d70e223e02fea3ae20a4feff05d7c1ea9a ]

    There is a panic in socket ioctl cmd SIOCGSKNS when NET_NS is not enabled.
    The reason is that nsfs tries to access ns->ops but the proc_ns_operations
    is not implemented in this case.

    [7.670023] Unable to handle kernel NULL pointer dereference at virtual address 00000010
    [7.670268] pgd = 32b54000
    [7.670544] [00000010] *pgd=00000000
    [7.671861] Internal error: Oops: 5 [#1] SMP ARM
    [7.672315] Modules linked in:
    [7.672918] CPU: 0 PID: 1 Comm: systemd Not tainted 5.13.0-rc3-00375-g6799d4f2da49 #16
    [7.673309] Hardware name: Generic DT based system
    [7.673642] PC is at nsfs_evict+0x24/0x30
    [7.674486] LR is at clear_inode+0x20/0x9c

    The same to tun SIOCGSKNS command.

    To fix this problem, we make get_net_ns() return -EINVAL when NET_NS is
    disabled. Meanwhile move it to right place net/core/net_namespace.c.

    Signed-off-by: Changbin Du <changbin.du@gmail.com>
    Fixes: c62cce2caee5 ("net: add an ioctl to get a socket network namespace")
    Cc: Cong Wang <xiyou.wangcong@gmail.com>
    Cc: Jakub Kicinski <kuba@kernel.org>
    Cc: David Laight <David.Laight@ACULAB.COM>
    Cc: Christian Brauner <christian.brauner@ubuntu.com>
    Suggested-by: Jakub Kicinski <kuba@kernel.org>
    Acked-by: Christian Brauner <christian.brauner@ubuntu.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fd99cacdc70f20daf81353782d35f65f35ff5b31
Author: Jisheng Zhang <Jisheng.Zhang@synaptics.com>
Date:   Fri Jun 11 15:16:11 2021 +0800

    net: stmmac: dwmac1000: Fix extended MAC address registers definition

    [ Upstream commit 1adb20f0d496b2c61e9aa1f4761b8d71f93d258e ]

    The register starts from 0x800 is the 16th MAC address register rather
    than the first one.

    Fixes: cffb13f4d6fb ("stmmac: extend mac addr reg and fix perfect filering")
    Signed-off-by: Jisheng Zhang <Jisheng.Zhang@synaptics.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit decb9c3ba468679286c75cb763ec608a7d6a1cfd
Author: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
Date:   Fri Jun 11 12:17:47 2021 +0530

    cxgb4: halt chip before flashing PHY firmware image

    [ Upstream commit 6d297540f75d759489054e8b07932208fc4db2cb ]

    When using firmware-assisted PHY firmware image write to flash,
    halt the chip before beginning the flash write operation to allow
    the running firmware to store the image persistently. Otherwise,
    the running firmware will only store the PHY image in local on-chip
    RAM, which will be lost after next reset.
```

```
    Fixes: 4ee339e1e92a ("cxgb4: add support to flash PHY image")
    Signed-off-by: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2e4829cae14861822ffacc2126fb6a5dcd8cd589
Author: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
Date:   Fri Jun 11 12:17:46 2021 +0530

    cxgb4: fix sleep in atomic when flashing PHY firmware

    [ Upstream commit f046bd0ae15d8a0bbe57d4647da182420f720c3d ]

    Before writing new PHY firmware to on-chip memory, driver queries
    firmware for current running PHY firmware version, which can result
    in sleep waiting for reply. So, move spinlock closer to the actual
    on-chip memory write operation, instead of taking it at the callers.

    Fixes: 5fff701c838e ("cxgb4: always sync access when flashing PHY firmware")
    Signed-off-by: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 291c5e3b8ca0c00af07bea25f6fc69a56aae7d9c
Author: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
Date:   Fri Jun 11 12:17:45 2021 +0530

    cxgb4: fix endianness when flashing boot image

    [ Upstream commit 42a2039753a7f758ba5c85cb199fcf10dc2111eb ]

    Boot images are copied to memory and updated with current underlying
    device ID before flashing them to adapter. Ensure the updated images
    are always flashed in Big Endian to allow the firmware to read the
    new images during boot properly.

    Fixes: 550883558f17 ("cxgb4: add support to flash boot image")
    Signed-off-by: Rahul Lakkireddy <rahul.lakkireddy@chelsio.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e029da0a957de40b99f5be311ef108c46c4f60d
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Fri Jun 11 08:13:39 2021 +0200

    alx: Fix an error handling path in 'alx_probe()'

    [ Upstream commit 33e381448cf7a05d76ac0b47d4a6531ecd0e5c53 ]

    If an error occurs after a 'pci_enable_pcie_error_reporting()' call, it
    must be undone by a corresponding 'pci_disable_pcie_error_reporting()'
    call, as already done in the remove function.

    Fixes: ab69bde6b2e9 ("alx: add a simple AR816x/AR817x device driver")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 27ef25c72373222aaa5fe7b5cd890ae9cfb89a8d
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Thu Jun 10 15:59:44 2021 -0700

    mptcp: fix soft lookup in subflow_error_report()
```

```
    [ Upstream commit 499ada5073361c631f2a3c4a8aed44d53b6f82ec ]

    Maxim reported a soft lookup in subflow_error_report():

     watchdog: BUG: soft lockup - CPU#0 stuck for 22s! [swapper/0:0]
     RIP: 0010:native_queued_spin_lock_slowpath
     RSP: 0018:ffffa859c0003bc0 EFLAGS: 00000202
     RAX: 0000000000000101 RBX: 0000000000000001 RCX: 0000000000000000
     RDX: ffff9195c2772d88 RSI: 0000000000000000 RDI: ffff9195c2772d88
     RBP: ffff9195c2772d00 R08: 00000000000067b0 R09: c6e31da9eb1e44f4
     R10: ffff9195ef379700 R11: ffff9195edb50710 R12: ffff9195c2772d88
     R13: ffff9195f500e3d0 R14: ffff9195ef379700 R15: ffff9195ef379700
     FS:  0000000000000000(0000) GS:ffff91961f400000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     CR2: 000000c000407000 CR3: 0000000002988000 CR4: 00000000000006f0
     Call Trace:
      <IRQ>
     _raw_spin_lock_bh
     subflow_error_report
     mptcp_subflow_data_available
     __mptcp_move_skbs_from_subflow
     mptcp_data_ready
     tcp_data_queue
     tcp_rcv_established
     tcp_v4_do_rcv
     tcp_v4_rcv
     ip_protocol_deliver_rcu
     ip_local_deliver_finish
     __netif_receive_skb_one_core
     netif_receive_skb
     rtl8139_poll 8139too
     __napi_poll
     net_rx_action
     __do_softirq
     __irq_exit_rcu
     common_interrupt
      </IRQ>

    The calling function - mptcp_subflow_data_available() - can be invoked
    from different contexts:
    - plain ssk socket lock
    - ssk socket lock + mptcp_data_lock
    - ssk socket lock + mptcp_data_lock + msk socket lock.

    Since subflow_error_report() tries to acquire the mptcp_data_lock, the
    latter two call chains will cause soft lookup.

    This change addresses the issue moving the error reporting call to
    outer functions, where the held locks list is known and the we can
    acquire only the needed one.

    Reported-by: Maxim Galaganov <max@internet.ru>
    Fixes: 15cc10453398 ("mptcp: deliver ssk errors to msk")
    Closes: https://github.com/multipath-tcp/mptcp_net-next/issues/199
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4dd7ed31e66bf933a511b8c3269573e1dbff3749
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Thu Jun 10 15:59:43 2021 -0700

    selftests: mptcp: enable syncookie only in absence of reorders
```

```
    [ Upstream commit 2395da0e17935ce9158cdfae433962bdb6cbfa67 ]

    Syncookie validation may fail for OoO packets, causing spurious
    resets and self-tests failures, so let's force syncookie only
    for tests iteration with no OoO.

    Fixes: fed61c4b584c ("selftests: mptcp: make 2nd net namespace use tcp syn cookies
unconditionally")
    Closes: https://github.com/multipath-tcp/mptcp_net-next/issues/198
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 083e54e4c761353d2e86b4dd8ec3b7de41098224
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Thu Jun 10 15:59:42 2021 -0700

    mptcp: do not warn on bad input from the network

    [ Upstream commit 61e710227e97172355d5f150d5c78c64175d9fb2 ]

    warn_bad_map() produces a kernel WARN on bad input coming
    from the network. Use pr_debug() to avoid spamming the system
    log.

    Additionally, when the right bound check fails, warn_bad_map() reports
    the wrong ssn value, let's fix it.

    Fixes: 648ef4b88673 ("mptcp: Implement MPTCP receive path")
    Closes: https://github.com/multipath-tcp/mptcp_net-next/issues/107
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 59f4b11b9a7a23c0492a801e320d717a6fdbbbeb
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Thu Jun 10 15:59:41 2021 -0700

    mptcp: wake-up readers only for in sequence data

    [ Upstream commit 99d1055ce2469dca3dd14be0991ff8133e25e3d0 ]

    Currently we rely on the subflow->data_avail field, which is subject to
    races:

            ssk1
                    skb len = 500 DSS(seq=1, len=1000, off=0)
                    # data_avail == MPTCP_SUBFLOW_DATA_AVAIL

            ssk2
                    skb len = 500 DSS(seq = 501, len=1000)
                    # data_avail == MPTCP_SUBFLOW_DATA_AVAIL

            ssk1
                    skb len = 500 DSS(seq = 1, len=1000, off =500)
                    # still data_avail == MPTCP_SUBFLOW_DATA_AVAIL,
                    # as the skb is covered by a pre-existing map,
                    # which was in-sequence at reception time.

    Instead we can explicitly check if some has been received in-sequence,
    propagating the info from __mptcp_move_skbs_from_subflow().

    Additionally add the 'ONCE' annotation to the 'data_avail' memory
```

```
    access, as msk will read it outside the subflow socket lock.

    Fixes: 648ef4b88673 ("mptcp: Implement MPTCP receive path")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6fb5ea5dd0a64a30aabc0da549585db63a6a9bc2
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Thu Jun 10 15:59:40 2021 -0700

    mptcp: try harder to borrow memory from subflow under pressure

    [ Upstream commit 72f961320d5d15bfcb26dbe3edaa3f7d25fd2c8a ]

    If the host is under sever memory pressure, and RX forward
    memory allocation for the msk fails, we try to borrow the
    required memory from the ingress subflow.

    The current attempt is a bit flaky: if skb->truesize is less
    than SK_MEM_QUANTUM, the ssk will not release any memory, and
    the next schedule will fail again.

    Instead, directly move the required amount of pages from the
    ssk to the msk, if available

    Fixes: 9c3f94e1681b ("mptcp: add missing memory scheduling in the rx path")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3371392c60e2685af30bd4547badd880f5df2b3f
Author: Maxim Mikityanskiy <maximmi@nvidia.com>
Date:   Thu Jun 10 19:40:31 2021 +0300

    sch_cake: Fix out of bounds when parsing TCP options and header

    [ Upstream commit ba91c49dedbde758ba0b72f57ac90b06ddf8e548 ]

    The TCP option parser in cake qdisc (cake_get_tcpopt and
    cake_tcph_may_drop) could read one byte out of bounds. When the length
    is 1, the execution flow gets into the loop, reads one byte of the
    opcode, and if the opcode is neither TCPOPT_EOL nor TCPOPT_NOP, it reads
    one more byte, which exceeds the length of 1.

    This fix is inspired by commit 9609dad263f8 ("ipv4: tcp_input: fix stack
    out of bounds when parsing TCP options.").

    v2 changes:

    Added doff validation in cake_get_tcphdr to avoid parsing garbage as TCP
    header. Although it wasn't strictly an out-of-bounds access (memory was
    allocated), garbage values could be read where CAKE expected the TCP
    header if doff was smaller than 5.

    Cc: Young Xiao <92siuyang@gmail.com>
    Fixes: 8b7138814f29 ("sch_cake: Add optional ACK filter")
    Signed-off-by: Maxim Mikityanskiy <maximmi@nvidia.com>
    Acked-by: Toke Høiland-Jørgensen <toke@toke.dk>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 76e02b8905d0691e89e104a882f3bba7dd0f6037
```

```
Author: Maxim Mikityanskiy <maximmi@nvidia.com>
Date:   Thu Jun 10 19:40:30 2021 +0300

    mptcp: Fix out of bounds when parsing TCP options

    [ Upstream commit 07718be265680dcf496347d475ce1a5442f55ad7 ]

    The TCP option parser in mptcp (mptcp_get_options) could read one byte
    out of bounds. When the length is 1, the execution flow gets into the
    loop, reads one byte of the opcode, and if the opcode is neither
    TCPOPT_EOL nor TCPOPT_NOP, it reads one more byte, which exceeds the
    length of 1.

    This fix is inspired by commit 9609dad263f8 ("ipv4: tcp_input: fix stack
    out of bounds when parsing TCP options.").

    Cc: Young Xiao <92siuyang@gmail.com>
    Fixes: cec37a6e41aa ("mptcp: Handle MP_CAPABLE options for outgoing connections")
    Signed-off-by: Maxim Mikityanskiy <maximmi@nvidia.com>
    Reviewed-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f648089337cb8ed40b2bb96e244f72b9d97dc96b
Author: Maxim Mikityanskiy <maximmi@nvidia.com>
Date:   Thu Jun 10 19:40:29 2021 +0300

    netfilter: synproxy: Fix out of bounds when parsing TCP options

    [ Upstream commit 5fc177ab759418c9537433e63301096e733fb915 ]

    The TCP option parser in synproxy (synproxy_parse_options) could read
    one byte out of bounds. When the length is 1, the execution flow gets
    into the loop, reads one byte of the opcode, and if the opcode is
    neither TCPOPT_EOL nor TCPOPT_NOP, it reads one more byte, which exceeds
    the length of 1.

    This fix is inspired by commit 9609dad263f8 ("ipv4: tcp_input: fix stack
    out of bounds when parsing TCP options.").

    v2 changes:

    Added an early return when length < 0 to avoid calling
    skb_header_pointer with negative length.

    Cc: Young Xiao <92siuyang@gmail.com>
    Fixes: 48b1de4c110a ("netfilter: add SYNPROXY core/target")
    Signed-off-by: Maxim Mikityanskiy <maximmi@nvidia.com>
    Reviewed-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 56c8b8333eb18513fef821591b18082162e92ebc
Author: Willem de Bruijn <willemb@google.com>
Date:   Wed Jun 9 18:41:57 2021 -0400

    skbuff: fix incorrect msg_zerocopy copy notifications

    [ Upstream commit 3bdd5ee0ec8c14131d560da492e6df452c6fdd75 ]

    msg_zerocopy signals if a send operation required copying with a flag
    in serr->ee.ee_code.

    This field can be incorrect as of the below commit, as a result of
    both structs uarg and serr pointing into the same skb->cb[].
```

```
    uarg->zerocopy must be read before skb->cb[] is reinitialized to hold
    serr. Similar to other fields len, hi and lo, use a local variable to
    temporarily hold the value.

    This was not a problem before, when the value was passed as a function
    argument.

    Fixes: 75518851a2a0 ("skbuff: Push status and refcounts into sock_zerocopy_callback")
    Reported-by: Talal Ahmad <talalahmad@google.com>
    Signed-off-by: Willem de Bruijn <willemb@google.com>
    Acked-by: Soheil Hassas Yeganeh <soheil@google.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ed22996926808701f5dd204918d1ab59a97ee3e2
Author: Aya Levin <ayal@nvidia.com>
Date:   Wed May 26 10:40:36 2021 +0300

    net/mlx5e: Block offload of outer header csum for GRE tunnel

    [ Upstream commit 54e1217b90486c94b26f24dcee1ee5ef5372f832 ]

    The device is able to offload either the outer header csum or inner
    header csum. The driver utilizes the inner csum offload. So, prohibit
    setting of tx-gre-csum-segmentation and let it be: off[fixed].

    Fixes: 2729984149e6 ("net/mlx5e: Support TSO and TX checksum offloads for GRE tunnels")
    Signed-off-by: Aya Levin <ayal@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b38c57a01a327ef9d38425980244bee8fa184845
Author: Aya Levin <ayal@nvidia.com>
Date:   Mon May 10 14:34:58 2021 +0300

    net/mlx5e: Block offload of outer header csum for UDP tunnels

    [ Upstream commit 6d6727dddc7f93fcc155cb8d0c49c29ae0e71122 ]

    The device is able to offload either the outer header csum or inner
    header csum. The driver utilizes the inner csum offload. Hence, block
    setting of tx-udp_tnl-csum-segmentation and set it to off[fixed].

    Fixes: b49663c8fb49 ("net/mlx5e: Add support for UDP tunnel segmentation with outer
checksum offload")
    Signed-off-by: Aya Levin <ayal@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit af452c9d5f78d56bd5e9f0ff37a10a0c4d35b4d0
Author: Shay Drory <shayd@nvidia.com>
Date:   Thu Feb 25 12:27:53 2021 +0200

    Revert "net/mlx5: Arm only EQs with EQEs"

    [ Upstream commit 7a545077cb6701957e84c7f158630bb5c984e648 ]

    In the scenario described below, an EQ can remain in FIRED state which
    can result in missing an interrupt generation.

    The scenario:
```

```
    device                        mlx5_core driver
    ------                        ----------------
    EQ1.eqe generated
    EQ1.MSI-X sent
    EQ1.state = FIRED
    EQ2.eqe generated
                                  mlx5_irq()
                                    polls - eq1_eqes()
                                    arm eq1
                                    polls - eq2_eqes()
                                    arm eq2
    EQ2.MSI-X sent
    EQ2.state = FIRED
                                   mlx5_irq()
                                   polls - eq2_eqes() -- no eqes found
                                   driver skips EQ arming;

    ->EQ2 remains fired, misses generating interrupt.

    Hence, always arm the EQ by reverting the cited commit in fixes tag.

    Fixes: d894892dda25 ("net/mlx5: Arm only EQs with EQEs")
    Signed-off-by: Shay Drory <shayd@nvidia.com>
    Reviewed-by: Parav Pandit <parav@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 574a9f20f1c0d18e4c48fa9e261d1460f69b6aaa
Author: Maor Gottlieb <maorg@nvidia.com>
Date:   Sun Jun 6 11:23:41 2021 +0300

    net/mlx5: DR, Don't use SW steering when RoCE is not supported

    [ Upstream commit 4aaf96ac8b45d8e2e019b6b53cce65a73c4ace2c ]

    SW steering uses RC QP to write/read to/from ICM, hence it's not
    supported when RoCE is not supported as well.

    Fixes: 70605ea545e8 ("net/mlx5: DR, Expose APIs for direct rule managing")
    Signed-off-by: Maor Gottlieb <maorg@nvidia.com>
    Reviewed-by: Alex Vesker <valex@nvidia.com>
    Reviewed-by: Yevgeny Kliteynik <kliteyn@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a84c6df0eb57b332574b12930de5cf6579e7b45
Author: Maor Gottlieb <maorg@nvidia.com>
Date:   Sun Jun 6 11:20:46 2021 +0300

    net/mlx5: Consider RoCE cap before init RDMA resources

    [ Upstream commit c189716b2a7c1d2d8658e269735273caa1c38b54 ]

    Check if RoCE is supported by the device before enable it in
    the vport context and create all the RDMA steering objects.

    Fixes: 80f09dfc237f ("net/mlx5: Eswitch, enable RoCE loopback traffic")
    Signed-off-by: Maor Gottlieb <maorg@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b374c1304f6d3d4752ad1412427b7bf02bb1fd61
Author: Dima Chumak <dchumak@nvidia.com>
Date:   Wed May 26 13:45:10 2021 +0300
```

```
    net/mlx5e: Fix page reclaim for dead peer hairpin

    [ Upstream commit a3e5fd9314dfc4314a9567cde96e1aef83a7458a ]

    When adding a hairpin flow, a firmware-side send queue is created for
    the peer net device, which claims some host memory pages for its
    internal ring buffer. If the peer net device is removed/unbound before
    the hairpin flow is deleted, then the send queue is not destroyed which
    leads to a stack trace on pci device remove:

    [ 748.005230] mlx5_core 0000:08:00.2: wait_func:1094:(pid 12985): MANAGE_PAGES(0x108)
 timeout. Will cause a leak of a command resource
    [ 748.005231] mlx5_core 0000:08:00.2: reclaim_pages:514:(pid 12985): failed reclaiming
 pages: err -110
    [ 748.001835] mlx5_core 0000:08:00.2: mlx5_reclaim_root_pages:653:(pid 12985): failed
 reclaiming pages (-110) for func id 0x0
    [ 748.002171] ------------[ cut here ]------------
    [ 748.001177] FW pages counter is 4 after reclaiming all pages
    [ 748.001186] WARNING: CPU: 1 PID: 12985 at
 drivers/net/ethernet/mellanox/mlx5/core/pagealloc.c:685
 mlx5_reclaim_startup_pages+0x34b/0x460 [mlx5_core]                      [  +0.002771]
 Modules linked in: cls_flower mlx5_ib mlx5_core ptp pps_core act_mirred sch_ingress
 openvswitch nsh xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype
 iptable_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 br_netfilter rpcrdma rdma_ucm
 ib_iser libiscsi scsi_transport_iscsi rdma_cm ib_umad ib_ipoib iw_cm ib_cm ib_uverbs ib_core
 overlay fuse [last unloaded: pps_core]
    [ 748.007225] CPU: 1 PID: 12985 Comm: tee Not tainted 5.12.0+ #1
    [ 748.001376] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-
 gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014
    [ 748.002315] RIP: 0010:mlx5_reclaim_startup_pages+0x34b/0x460 [mlx5_core]
    [ 748.001679] Code: 28 00 00 00 0f 85 22 01 00 00 48 81 c4 b0 00 00 00 31 c0 5b 5d 41 5c
 41 5d 41 5e 41 5f c3 48 c7 c7 40 cc 19 a1 e8 9f 71 0e e2 <0f> 0b e9 30 ff ff ff 48 c7 c7 a0
 cc 19 a1 e8 8c 71 0e e2 0f 0b e9
    [ 748.003781] RSP: 0018:ffff88815220faf8 EFLAGS: 00010286
    [ 748.001149] RAX: 0000000000000000 RBX: ffff8881b4900280 RCX: 0000000000000000
    [ 748.001445] RDX: 0000000000000027 RSI: 0000000000000004 RDI: ffffed102a441f51
    [ 748.001614] RBP: 00000000000032b9 R08: 0000000000000001 R09: ffffed1054a15ee8
    [ 748.001446] R10: ffff8882a50af73b R11: ffffed1054a15ee7 R12: fffffbfff07c1e30
    [ 748.001447] R13: dfffc0000000000 R14: ffff8881b492cba8 R15: 0000000000000000
    [ 748.001429] FS:  00007f58bd08b580(0000) GS:ffff8882a5080000(0000)
 knlGS:0000000000000000
    [ 748.001695] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [ 748.001309] CR2: 000055a026351740 CR3: 00000001d3b48006 CR4: 0000000000370ea0
    [ 748.001506] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [ 748.001483] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [ 748.001654] Call Trace:
    [ 748.000576]  ? mlx5_satisfy_startup_pages+0x290/0x290 [mlx5_core]
    [ 748.001416]  ? mlx5_cmd_teardown_hca+0xa2/0xd0 [mlx5_core]
    [ 748.001354]  ? mlx5_cmd_init_hca+0x280/0x280 [mlx5_core]
    [ 748.001203]  mlx5_function_teardown+0x30/0x60 [mlx5_core]
    [ 748.001275]  mlx5_uninit_one+0xa7/0xc0 [mlx5_core]
    [ 748.001200]  remove_one+0x5f/0xc0 [mlx5_core]
    [ 748.001075]  pci_device_remove+0x9f/0x1d0
    [ 748.000833]  device_release_driver_internal+0x1e0/0x490
    [ 748.001207]  unbind_store+0x19f/0x200
    [ 748.000942]  ? sysfs_file_ops+0x170/0x170
    [ 748.001000]  kernfs_fop_write_iter+0x2bc/0x450
    [ 748.000970]  new_sync_write+0x373/0x610
    [ 748.001124]  ? new_sync_read+0x600/0x600
    [ 748.001057]  ? lock_acquire+0x4d6/0x700
    [ 748.000908]  ? lockdep_hardirqs_on_prepare+0x400/0x400
    [ 748.001126]  ? fd_install+0x1c9/0x4d0
    [ 748.000951]  vfs_write+0x4d0/0x800
    [ 748.000804]  ksys_write+0xf9/0x1d0
```

```
     [ 748.000868]  ? __x64_sys_read+0xb0/0xb0
     [ 748.000811]  ? filp_open+0x50/0x50
     [ 748.000919]  ? syscall_enter_from_user_mode+0x1d/0x50
     [ 748.001223]  do_syscall_64+0x3f/0x80
     [ 748.000892]  entry_SYSCALL_64_after_hwframe+0x44/0xae
     [ 748.001026] RIP: 0033:0x7f58bcfb22f7
     [ 748.000944] Code: 0d 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b7 0f 1f 00 f3 0f 1e fa
 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 51 c3 48 83
 ec 28 48 89 54 24 18 48 89 74 24
     [ 748.003925] RSP: 002b:00007fffd7f2aaa8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
     [ 748.001732] RAX: ffffffffffffffda RBX: 000000000000000d RCX: 00007f58bcfb22f7
     [ 748.001426] RDX: 000000000000000d RSI: 00007fffd7f2abc0 RDI: 0000000000000003
     [ 748.001746] RBP: 00007fffd7f2abc0 R08: 0000000000000000 R09: 0000000000000001
     [ 748.001631] R10: 00000000000001b6 R11: 0000000000000246 R12: 000000000000000d
     [ 748.001537] R13: 00005597ac2c24a0 R14: 000000000000000d R15: 00007f58bd084700
     [ 748.001564] irq event stamp: 0
     [ 748.000787] hardirqs last  enabled at (0): [<0000000000000000>] 0x0
     [ 748.001399] hardirqs last disabled at (0): [<ffffffff813132cf>]
 copy_process+0x146f/0x5eb0
     [ 748.001854] softirqs last  enabled at (0): [<ffffffff8131330e>]
 copy_process+0x14ae/0x5eb0
     [ 748.013431] softirqs last disabled at (0): [<0000000000000000>] 0x0
     [ 748.001492] ---[ end trace a6fabd773d1c51ae ]---

     Fix by destroying the send queue of a hairpin peer net device that is
     being removed/unbound, which returns the allocated ring buffer pages to
     the host.

     Fixes: 4d8fcf216c90 ("net/mlx5e: Avoid unbounded peer devices when unpairing TC hairpin
 rules")
     Signed-off-by: Dima Chumak <dchumak@nvidia.com>
     Reviewed-by: Roi Dayan <roid@nvidia.com>
     Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 462abaee88f508577345fdc2493e59cb4adfa17e
 Author: Huy Nguyen <huyn@nvidia.com>
 Date:   Fri May 28 13:20:32 2021 -0500

     net/mlx5e: Remove dependency in IPsec initialization flows

     [ Upstream commit 8ad893e516a77209a1818a2072d2027d87db809f ]

     Currently, IPsec feature is disabled because mlx5e_build_nic_netdev
     is required to be called after mlx5e_ipsec_init. This requirement is
     invalid as mlx5e_build_nic_netdev and mlx5e_ipsec_init initialize
     independent resources.

     Remove ipsec pointer check in mlx5e_build_nic_netdev so that the
     two functions can be called at any order.

     Fixes: 547eede070eb ("net/mlx5e: IPSec, Innova IPSec offload infrastructure")
     Signed-off-by: Huy Nguyen <huyn@nvidia.com>
     Reviewed-by: Raed Salem <raeds@nvidia.com>
     Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit b6447b72aca571632e71bb73a797118d5ce46a93
 Author: Vlad Buslov <vladbu@nvidia.com>
 Date:   Mon May 31 16:28:39 2021 +0300

     net/mlx5e: Fix use-after-free of encap entry in neigh update handler

     [ Upstream commit fb1a3132ee1ac968316e45d21a48703a6db0b6c3 ]
```

```
    Function mlx5e_rep_neigh_update() wasn't updated to accommodate rtnl lock
    removal from TC filter update path and properly handle concurrent encap
    entry insertion/deletion which can lead to following use-after-free:

     [23827.464923] ==================================================================
     [23827.469446] BUG: KASAN: use-after-free in mlx5e_encap_take+0x72/0x140 [mlx5_core]
     [23827.470971] Read of size 4 at addr ffff8881d132228c by task kworker/u20:6/21635
     [23827.472251]
     [23827.472615] CPU: 9 PID: 21635 Comm: kworker/u20:6 Not tainted 5.13.0-rc3+ #5
     [23827.473788] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-
gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014
     [23827.475639] Workqueue: mlx5e mlx5e_rep_neigh_update [mlx5_core]
     [23827.476731] Call Trace:
     [23827.477260]  dump_stack+0xbb/0x107
     [23827.477906]  print_address_description.constprop.0+0x18/0x140
     [23827.478896]  ? mlx5e_encap_take+0x72/0x140 [mlx5_core]
     [23827.479879]  ? mlx5e_encap_take+0x72/0x140 [mlx5_core]
     [23827.480905]  kasan_report.cold+0x7c/0xd8
     [23827.481701]  ? mlx5e_encap_take+0x72/0x140 [mlx5_core]
     [23827.482744]  kasan_check_range+0x145/0x1a0
     [23827.493112]  mlx5e_encap_take+0x72/0x140 [mlx5_core]
     [23827.494054]  ? mlx5e_tc_tun_encap_info_equal_generic+0x140/0x140 [mlx5_core]
     [23827.495296]  mlx5e_rep_neigh_update+0x41e/0x5e0 [mlx5_core]
     [23827.496338]  ? mlx5e_rep_neigh_entry_release+0xb80/0xb80 [mlx5_core]
     [23827.497486]  ? read_word_at_a_time+0xe/0x20
     [23827.498250]  ? strscpy+0xa0/0x2a0
     [23827.498889]  process_one_work+0x8ac/0x14e0
     [23827.499638]  ? lockdep_hardirqs_on_prepare+0x400/0x400
     [23827.500537]  ? pwq_dec_nr_in_flight+0x2c0/0x2c0
     [23827.501359]  ? rwlock_bug.part.0+0x90/0x90
     [23827.502116]  worker_thread+0x53b/0x1220
     [23827.502831]  ? process_one_work+0x14e0/0x14e0
     [23827.503627]  kthread+0x328/0x3f0
     [23827.504254]  ? _raw_spin_unlock_irq+0x24/0x40
     [23827.505065]  ? __kthread_bind_mask+0x90/0x90
     [23827.505912]  ret_from_fork+0x1f/0x30
     [23827.506621]
     [23827.506987] Allocated by task 28248:
     [23827.507694]  kasan_save_stack+0x1b/0x40
     [23827.508476]  __kasan_kmalloc+0x7c/0x90
     [23827.509197]  mlx5e_attach_encap+0xde1/0x1d40 [mlx5_core]
     [23827.510194]  mlx5e_tc_add_fdb_flow+0x397/0xc40 [mlx5_core]
     [23827.511218]  __mlx5e_add_fdb_flow+0x519/0xb30 [mlx5_core]
     [23827.512234]  mlx5e_configure_flower+0x191c/0x4870 [mlx5_core]
     [23827.513298]  tc_setup_cb_add+0x1d5/0x420
     [23827.514023]  fl_hw_replace_filter+0x382/0x6a0 [cls_flower]
     [23827.514975]  fl_change+0x2ceb/0x4a51 [cls_flower]
     [23827.515821]  tc_new_tfilter+0x89a/0x2070
     [23827.516548]  rtnetlink_rcv_msg+0x644/0x8c0
     [23827.517300]  netlink_rcv_skb+0x11d/0x340
     [23827.518021]  netlink_unicast+0x42b/0x700
     [23827.518742]  netlink_sendmsg+0x743/0xc20
     [23827.519467]  sock_sendmsg+0xb2/0xe0
     [23827.520131]  ____sys_sendmsg+0x590/0x770
     [23827.520851]  ___sys_sendmsg+0xd8/0x160
     [23827.521552]  __sys_sendmsg+0xb7/0x140
     [23827.522238]  do_syscall_64+0x3a/0x70
     [23827.522907]  entry_SYSCALL_64_after_hwframe+0x44/0xae
     [23827.523797]
     [23827.524163] Freed by task 25948:
     [23827.524780]  kasan_save_stack+0x1b/0x40
     [23827.525488]  kasan_set_track+0x1c/0x30
     [23827.526187]  kasan_set_free_info+0x20/0x30
     [23827.526968]  __kasan_slab_free+0xed/0x130
     [23827.527709]  slab_free_freelist_hook+0xcf/0x1d0
```

```
      [23827.528528]  kmem_cache_free_bulk+0x33a/0x6e0
      [23827.529317]  kfree_rcu_work+0x55f/0xb70
      [23827.530024]  process_one_work+0x8ac/0x14e0
      [23827.530770]  worker_thread+0x53b/0x1220
      [23827.531480]  kthread+0x328/0x3f0
      [23827.532114]  ret_from_fork+0x1f/0x30
      [23827.532785]
      [23827.533147] Last potentially related work creation:
      [23827.534007]  kasan_save_stack+0x1b/0x40
      [23827.534710]  kasan_record_aux_stack+0xab/0xc0
      [23827.535492]  kvfree_call_rcu+0x31/0x7b0
      [23827.536206]  mlx5e_tc_del_fdb_flow+0x577/0xef0 [mlx5_core]
      [23827.537305]  mlx5e_flow_put+0x49/0x80 [mlx5_core]
      [23827.538290]  mlx5e_delete_flower+0x6d1/0xe60 [mlx5_core]
      [23827.539300]  tc_setup_cb_destroy+0x18e/0x2f0
      [23827.540144]  fl_hw_destroy_filter+0x1d2/0x310 [cls_flower]
      [23827.541148]  __fl_delete+0x4dc/0x660 [cls_flower]
      [23827.541985]  fl_delete+0x97/0x160 [cls_flower]
      [23827.542782]  tc_del_tfilter+0x7ab/0x13d0
      [23827.543503]  rtnetlink_rcv_msg+0x644/0x8c0
      [23827.544257]  netlink_rcv_skb+0x11d/0x340
      [23827.544981]  netlink_unicast+0x42b/0x700
      [23827.545700]  netlink_sendmsg+0x743/0xc20
      [23827.546424]  sock_sendmsg+0xb2/0xe0
      [23827.547084]  ____sys_sendmsg+0x590/0x770
      [23827.547850]  ___sys_sendmsg+0xd8/0x160
      [23827.548606]  __sys_sendmsg+0xb7/0x140
      [23827.549303]  do_syscall_64+0x3a/0x70
      [23827.549969]  entry_SYSCALL_64_after_hwframe+0x44/0xae
      [23827.550853]
      [23827.551217] The buggy address belongs to the object at ffff8881d1322200
      [23827.551217]  which belongs to the cache kmalloc-256 of size 256
      [23827.553341] The buggy address is located 140 bytes inside of
      [23827.553341]  256-byte region [ffff8881d1322200, ffff8881d1322300)
      [23827.555747] The buggy address belongs to the page:
      [23827.556847] page:00000000898762aa refcount:1 mapcount:0 mapping:0000000000000000
index:0x0 pfn:0x1d1320
      [23827.558651] head:00000000898762aa order:2 compound_mapcount:0 compound_pincount:0
      [23827.559961] flags: 0x2ffff800010200(slab|head|node=0|zone=2|lastcpupid=0x1ffff)
      [23827.561243] raw: 002ffff800010200 dead000000000100 dead000000000122 ffff888100042b40
      [23827.562653] raw: 0000000000000000 0000000000200020 00000001ffffffff 0000000000000000
      [23827.564112] page dumped because: kasan: bad access detected
      [23827.565439]
      [23827.565932] Memory state around the buggy address:
      [23827.566917]  ffff8881d1322180: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
      [23827.568485]  ffff8881d1322200: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
      [23827.569818] >ffff8881d1322280: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
      [23827.571143]                    ^
      [23827.571879]  ffff8881d1322300: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
      [23827.573283]  ffff8881d1322380: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
      [23827.574654] ==================================================================
```

    Most of the necessary logic is already correctly implemented by
    mlx5e_get_next_valid_encap() helper that is used in neigh stats update
    handler. Make the handler generic by renaming it to
    mlx5e_get_next_matching_encap() and use callback to test whether flow is
    matching instead of hardcoded check for 'valid' flag value. Implement
    mlx5e_get_next_valid_encap() by calling mlx5e_get_next_matching_encap()
    with callback that tests encap MLX5_ENCAP_ENTRY_VALID flag. Implement new
    mlx5e_get_next_init_encap() helper by calling
    mlx5e_get_next_matching_encap() with callback that tests encap completion
    result to be non-error and use it in mlx5e_rep_neigh_update() to safely
    iterate over nhe->encap_list.

    Remove encap completion logic from mlx5e_rep_update_flows() since the encap

```
    entries passed to this function are already guaranteed to be properly
    initialized by similar code in mlx5e_get_next_init_encap().

    Fixes: 2a1f1768fa17 ("net/mlx5e: Refactor neigh update for concurrent execution")
    Signed-off-by: Vlad Buslov <vladbu@nvidia.com>
    Reviewed-by: Roi Dayan <roid@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7c7dd4e03be90fba24519508932d0d024ab6d849
Author: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
Date:   Wed Jun 9 11:23:56 2021 -0300

    net/sched: act_ct: handle DNAT tuple collision

    [ Upstream commit 13c62f5371e3eb4fc3400cfa26e64ca75f888008 ]

    This this the counterpart of 8aa7b526dc0b ("openvswitch: handle DNAT
    tuple collision") for act_ct. From that commit changelog:

    """
    With multiple DNAT rules it's possible that after destination
    translation the resulting tuples collide.

    ...

    Netfilter handles this case by allocating a null binding for SNAT at
    egress by default.  Perform the same operation in openvswitch for DNAT
    if no explicit SNAT is requested by the user and allocate a null binding
    for SNAT for packets in the "original" direction.
    """

    Fixes: 95219afbb980 ("act_ct: support asymmetric conntrack")
    Signed-off-by: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 69a54b4899dd2328d0fec89582c5722469f995e5
Author: Ido Schimmel <idosch@nvidia.com>
Date:   Wed Jun 9 14:17:53 2021 +0300

    rtnetlink: Fix regression in bridge VLAN configuration

    [ Upstream commit d2e381c4963663bca6f30c3b996fa4dbafe8fcb5 ]

    Cited commit started returning errors when notification info is not
    filled by the bridge driver, resulting in the following regression:

     # ip link add name br1 type bridge vlan_filtering 1
     # bridge vlan add dev br1 vid 555 self pvid untagged
     RTNETLINK answers: Invalid argument

    As long as the bridge driver does not fill notification info for the
    bridge device itself, an empty notification should not be considered as
    an error. This is explained in commit 59ccaaaa49b5 ("bridge: dont send
    notification when skb->len == 0 in rtnl_bridge_notify").

    Fix by removing the error and add a comment to avoid future bugs.

    Fixes: a8db57c1d285 ("rtnetlink: Fix missing error code in rtnl_bridge_notify()")
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: Nikolay Aleksandrov <nikolay@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 65310b0aff86980a011c7c7bfa487a333d4ca241
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Wed Jun 9 11:49:01 2021 +0200

    udp: fix race between close() and udp_abort()

    [ Upstream commit a8b897c7bcd47f4147d066e22cc01d1026d7640e ]

    Kaustubh reported and diagnosed a panic in udp_lib_lookup().
    The root cause is udp_abort() racing with close(). Both
    racing functions acquire the socket lock, but udp{v6}_destroy_sock()
    release it before performing destructive actions.

    We can't easily extend the socket lock scope to avoid the race,
    instead use the SOCK_DEAD flag to prevent udp_abort from doing
    any action when the critical race happens.

    Diagnosed-and-tested-by: Kaustubh Pandey <kapandey@codeaurora.org>
    Fixes: 5d77dca82839 ("net: diag: support SOCK_DESTROY for UDP sockets")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c4c9de226916d5eaf0371467efed83a983965604
Author: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
Date:   Thu May 20 08:35:00 2021 +0200

    ice: parameterize functions responsible for Tx ring management

    [ Upstream commit 2e84f6b3773f43263124c76499c0c4ec3f40aa9b ]

    Commit ae15e0ba1b33 ("ice: Change number of XDP Tx queues to match
    number of Rx queues") tried to address the incorrect setting of XDP
    queue count that was based on the Tx queue count, whereas in theory we
    should provide the XDP queue per Rx queue. However, the routines that
    setup and destroy the set of Tx resources are still based on the
    vsi->num_txq.

    Ice supports the asynchronous Tx/Rx queue count, so for a setup where
    vsi->num_txq > vsi->num_rxq, ice_vsi_stop_tx_rings and ice_vsi_cfg_txqs
    will be accessing the vsi->xdp_rings out of the bounds.

    Parameterize two mentioned functions so they get the size of Tx resources
    array as the input.

    Fixes: ae15e0ba1b33 ("ice: Change number of XDP Tx queues to match number of Rx queues")
    Signed-off-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Tested-by: Kiran Bhandare <kiranx.bhandare@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 57b2b26fa6569b319312dcf15140f3c52feacb34
Author: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
Date:   Thu May 20 08:34:59 2021 +0200

    ice: add ndo_bpf callback for safe mode netdev ops

    [ Upstream commit ebc5399ea1dfcddac31974091086a3379141899b ]

    ice driver requires a programmable pipeline firmware package in order to
    have a support for advanced features. Otherwise, driver falls back to so
    called 'safe mode'. For that mode, ndo_bpf callback is not exposed and
    when user tries to load XDP program, the following happens:

    $ sudo ./xdp1 enp179s0f1
```

```
    libbpf: Kernel error message: Underlying driver does not support XDP in native mode
    link set xdp fd failed

    which is sort of confusing, as there is a native XDP support, but not in
    the current mode. Improve the user experience by providing the specific
    ndo_bpf callback dedicated for safe mode which will make use of extack
    to explicitly let the user know that the DDP package is missing and
    that's the reason that the XDP can't be loaded onto interface currently.

    Cc: Jamal Hadi Salim <jhs@mojatatu.com>
    Fixes: efc2214b6047 ("ice: Add support for XDP")
    Signed-off-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Tested-by: Kiran Bhandare <kiranx.bhandare@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b499e673dc471828fea0fe945843e498d692211f
Author: Florian Westphal <fw@strlen.de>
Date:   Tue Jun 8 13:48:18 2021 +0200

    netfilter: nft_fib_ipv6: skip ipv6 packets from any to link-local

    [ Upstream commit 12f36e9bf678a81d030ca1b693dcda62b55af7c5 ]

    The ip6tables rpfilter match has an extra check to skip packets with
    "::" source address.

    Extend this to ipv6 fib expression.  Else ipv6 duplicate address detection
    packets will fail rpf route check -- lookup returns -ENETUNREACH.

    While at it, extend the prerouting check to also cover the ingress hook.

    Closes: https://bugzilla.netfilter.org/show_bug.cgi?id=1543
    Fixes: f6d0cbcf09c5 ("netfilter: nf_tables: add fib expression")
    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 a1f6740fa3938baa2a45c3dfc53402b78122a8a8
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Fri Jun 4 03:07:28 2021 +0200

    netfilter: nf_tables: initialize set before expression setup

    [ Upstream commit ad9f151e560b016b6ad3280b48e42fa11e1a5440 ]

    nft_set_elem_expr_alloc() needs an initialized set if expression sets on
    the NFT_EXPR_GC flag. Move set fields initialization before expression
    setup.

    [4512935.019450] ================================================================
    [4512935.019456] BUG: KASAN: null-ptr-deref in nft_set_elem_expr_alloc+0x84/0xd0
[nf_tables]
    [4512935.019487] Read of size 8 at addr 0000000000000070 by task nft/23532
    [4512935.019494] CPU: 1 PID: 23532 Comm: nft Not tainted 5.12.0-rc4+ #48
    [...]
    [4512935.019502] Call Trace:
    [4512935.019505]  dump_stack+0x89/0xb4
    [4512935.019512]  ? nft_set_elem_expr_alloc+0x84/0xd0 [nf_tables]
    [4512935.019536]  ? nft_set_elem_expr_alloc+0x84/0xd0 [nf_tables]
    [4512935.019560]  kasan_report.cold.12+0x5f/0xd8
    [4512935.019566]  ? nft_set_elem_expr_alloc+0x84/0xd0 [nf_tables]
    [4512935.019590]  nft_set_elem_expr_alloc+0x84/0xd0 [nf_tables]
    [4512935.019615]  nf_tables_newset+0xc7f/0x1460 [nf_tables]
```

```
    Reported-by: syzbot+ce96ca2b1d0b37c6422d@syzkaller.appspotmail.com
    Fixes: 65038428b2c6 ("netfilter: nf_tables: allow to specify stateful expression in set
definition")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2e44117758bffaa3d1eed9eef473b130c66ae8d0
Author: Aleksander Jan Bajkowski <olek2@wp.pl>
Date:   Tue Jun 8 23:21:07 2021 +0200

    net: lantiq: disable interrupt before sheduling NAPI

    [ Upstream commit f2386cf7c5f4ff5d7b584f5d92014edd7df6c676 ]

    This patch fixes TX hangs with threaded NAPI enabled. The scheduled
    NAPI seems to be executed in parallel with the interrupt on second
    thread. Sometimes it happens that ltq_dma_disable_irq() is executed
    after xrx200_tx_housekeeping(). The symptom is that TX interrupts
    are disabled in the DMA controller. As a result, the TX hangs after
    a few seconds of the iperf test. Scheduling NAPI after disabling
    interrupts fixes this issue.

    Tested on Lantiq xRX200 (BT Home Hub 5A).

    Fixes: 9423361da523 ("net: lantiq: Disable IRQs only if NAPI gets scheduled ")
    Signed-off-by: Aleksander Jan Bajkowski <olek2@wp.pl>
    Acked-by: Hauke Mehrtens <hauke@hauke-m.de>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2b66c0119c87c8dce75e2ac8dd9c518309619e6d
Author: Shay Agroskin <shayagr@amazon.com>
Date:   Tue Jun 8 19:42:54 2021 +0300

    net: ena: fix DMA mapping function issues in XDP

    [ Upstream commit 504fd6a5390c30b1b7670768e314dd5d473da06a ]

    This patch fixes several bugs found when (DMA/LLQ) mapping a packet for
    transmission. The mapping procedure makes the transmitted packet
    accessible by the device.
    When using LLQ, this requires copying the packet's header to push header
    (which would be passed to LLQ) and creating DMA mapping for the payload
    (if the packet doesn't fit the maximum push length).
    When not using LLQ, we map the whole packet with DMA.

    The following bugs are fixed in the code:
        1. Add support for non-LLQ machines:
           The ena_xdp_tx_map_frame() function assumed that LLQ is
           supported, and never mapped the whole packet using DMA. On some
           instances, which don't support LLQ, this causes loss of traffic.

        2. Wrong DMA buffer length passed to device:
           When using LLQ, the first 'tx_max_header_size' bytes of the
           packet would be copied to push header. The rest of the packet
           would be copied to a DMA'd buffer.

        3. Freeing the XDP buffer twice in case of a mapping error:
           In case a buffer DMA mapping fails, the function uses
           xdp_return_frame_rx_napi() to free the RX buffer and returns from
           the function with an error. XDP frames that fail to xmit get
           freed by the kernel and so there is no need for this call.

    Fixes: 548c4940b9f1 ("net: ena: Implement XDP_TX action")
    Signed-off-by: Shay Agroskin <shayagr@amazon.com>
```

```
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b0a744f7ac5af582e4a8f620f5e6686829bc455a
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Tue Jun 8 14:15:35 2021 +0300

    net: dsa: felix: re-enable TX flow control in ocelot_port_flush()

    [ Upstream commit 1650bdb1c516c248fb06f6d076559ff6437a5853 ]

    Because flow control is set up statically in ocelot_init_port(), and not
    in phylink_mac_link_up(), what happens is that after the blamed commit,
    the flow control remains disabled after the port flushing procedure.

    Fixes: eb4733d7cffc ("net: dsa: felix: implement port flushing on
.phylink_mac_link_down")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b25b60d076164edb3025e85aabd2cf50a5215b91
Author: Pavel Skripkin <paskripkin@gmail.com>
Date:   Tue Jun 8 11:06:41 2021 +0300

    net: rds: fix memory leak in rds_recvmsg

    [ Upstream commit 49bfcbfd989a8f1f23e705759a6bb099de2cff9f ]

    Syzbot reported memory leak in rds. The problem
    was in unputted refcount in case of error.

    int rds_recvmsg(struct socket *sock, struct msghdr *msg, size_t size,
                    int msg_flags)
    {
    ...

            if (!rds_next_incoming(rs, &inc)) {
                    ...
            }

    After this "if" inc refcount incremented and

            if (rds_cmsg_recv(inc, msg, rs)) {
                    ret = -EFAULT;
                    goto out;
            }
    ...
    out:
            return ret;
    }

    in case of rds_cmsg_recv() fail the refcount won't be
    decremented. And it's easy to see from ftrace log, that
    rds_inc_addref() don't have rds_inc_put() pair in
    rds_recvmsg() after rds_cmsg_recv()

     1)               |  rds_recvmsg() {
     1)   3.721 us    |    rds_inc_addref();
     1)   3.853 us    |    rds_message_inc_copy_to_user();
     1) + 10.395 us   |    rds_cmsg_recv();
     1) + 34.260 us   |  }

    Fixes: bdbe6fbc6a2f ("RDS: recv.c")
    Reported-and-tested-by: syzbot+5134cdf021c4ed5aaa5f@syzkaller.appspotmail.com
```

```
    Signed-off-by: Pavel Skripkin <paskripkin@gmail.com>
    Reviewed-by: Håkon Bugge <haakon.bugge@oracle.com>
    Acked-by: Santosh Shilimkar <santosh.shilimkar@oracle.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2f032ebff9b1fca4e4888a8d50a98134e0c407f2
Author: Nicolas Dichtel <nicolas.dichtel@6wind.com>
Date:   Tue Jun 8 16:59:51 2021 +0200

    vrf: fix maximum MTU

    [ Upstream commit 9bb392f62447d73cc7dd7562413a2cd9104c82f8 ]

    My initial goal was to fix the default MTU, which is set to 65536, ie above
    the maximum defined in the driver: 65535 (ETH_MAX_MTU).

    In fact, it's seems more consistent, wrt min_mtu, to set the max_mtu to
    IP6_MAX_MTU (65535 + sizeof(struct ipv6hdr)) and use it by default.

    Let's also, for consistency, set the mtu in vrf_setup(). This function
    calls ether_setup(), which set the mtu to 1500. Thus, the whole mtu config
    is done in the same function.

    Before the patch:
    $ ip link add blue type vrf table 1234
    $ ip link list blue
    9: blue: <NOARP,MASTER> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen
1000
        link/ether fa:f5:27:70:24:2a brd ff:ff:ff:ff:ff:ff
    $ ip link set dev blue mtu 65535
    $ ip link set dev blue mtu 65536
    Error: mtu greater than device maximum.

    Fixes: 5055376a3b44 ("net: vrf: Fix ping failed when vrf mtu is set to 0")
    CC: Miaohe Lin <linmiaohe@huawei.com>
    Signed-off-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Reviewed-by: David Ahern <dsahern@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0ffb460be3abac86f884a8c548bb02724ec370f4
Author: Nanyong Sun <sunnanyong@huawei.com>
Date:   Tue Jun 8 09:51:58 2021 +0800

    net: ipv4: fix memory leak in netlbl_cipsov4_add_std

    [ Upstream commit d612c3f3fae221e7ea736d196581c2217304bbbc ]

    Reported by syzkaller:
    BUG: memory leak
    unreferenced object 0xffff888105df7000 (size 64):
    comm "syz-executor842", pid 360, jiffies 4294824824 (age 22.546s)
    hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
    backtrace:
    [<00000000e67ed558>] kmalloc include/linux/slab.h:590 [inline]
    [<00000000e67ed558>] kzalloc include/linux/slab.h:720 [inline]
    [<00000000e67ed558>] netlbl_cipsov4_add_std net/netlabel/netlabel_cipso_v4.c:145
[inline]
    [<00000000e67ed558>] netlbl_cipsov4_add+0x390/0x2340
net/netlabel/netlabel_cipso_v4.c:416
    [<0000000006040154>] genl_family_rcv_msg_doit.isra.0+0x20e/0x320
net/netlink/genetlink.c:739
```

```
     [<00000000204d7a1c>] genl_family_rcv_msg net/netlink/genetlink.c:783 [inline]
     [<00000000204d7a1c>] genl_rcv_msg+0x2bf/0x4f0 net/netlink/genetlink.c:800
     [<00000000c0d6a995>] netlink_rcv_skb+0x134/0x3d0 net/netlink/af_netlink.c:2504
     [<00000000d78b9d2c>] genl_rcv+0x24/0x40 net/netlink/genetlink.c:811
     [<000000009733081b>] netlink_unicast_kernel net/netlink/af_netlink.c:1314 [inline]
     [<000000009733081b>] netlink_unicast+0x4a0/0x6a0 net/netlink/af_netlink.c:1340
     [<00000000d5fd43b8>] netlink_sendmsg+0x789/0xc70 net/netlink/af_netlink.c:1929
     [<000000000a2d1e40>] sock_sendmsg_nosec net/socket.c:654 [inline]
     [<000000000a2d1e40>] sock_sendmsg+0x139/0x170 net/socket.c:674
     [<00000000321d1969>] ____sys_sendmsg+0x658/0x7d0 net/socket.c:2350
     [<00000000964e16bc>] ___sys_sendmsg+0xf8/0x170 net/socket.c:2404
     [<000000001615e288>] __sys_sendmsg+0xd3/0x190 net/socket.c:2433
     [<000000004ee8b6a5>] do_syscall_64+0x37/0x90 arch/x86/entry/common.c:47
     [<00000000171c7cee>] entry_SYSCALL_64_after_hwframe+0x44/0xae

     The memory of doi_def->map.std pointing is allocated in
     netlbl_cipsov4_add_std, but no place has freed it. It should be
     freed in cipso_v4_doi_free which frees the cipso DOI resource.

     Fixes: 96cb8e3313c7a ("[NetLabel]: CIPSOv4 and Unlabeled packet integration")
     Reported-by: Hulk Robot <hulkci@huawei.com>
     Signed-off-by: Nanyong Sun <sunnanyong@huawei.com>
     Acked-by: Paul Moore <paul@paul-moore.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit c54a64e7c0accd096e0b2ec7b61379a19ba0106e
  Author: Kev Jackson <foamdino@gmail.com>
  Date:   Mon Jun 7 14:08:35 2021 +0100

     libbpf: Fixes incorrect rx_ring_setup_done

     [ Upstream commit 11fc79fc9f2e395aa39fa5baccae62767c5d8280 ]

     When calling xsk_socket__create_shared(), the logic at line 1097 marks a
     boolean flag true within the xsk_umem structure to track setup progress
     in order to support multiple calls to the function.  However, instead of
     marking umem->tx_ring_setup_done, the code incorrectly sets
     umem->rx_ring_setup_done.  This leads to improper behaviour when
     creating and destroying xsk and umem structures.

     Multiple calls to this function is documented as supported.

     Fixes: ca7a83e2487a ("libbpf: Only create rx and tx XDP rings when necessary")
     Signed-off-by: Kev Jackson <foamdino@gmail.com>
     Signed-off-by: Andrii Nakryiko <andrii@kernel.org>
     Acked-by: Yonghong Song <yhs@fb.com>
     Link: https://lore.kernel.org/bpf/YL4aU4f3Aaik7CN0@linux-dev
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit ffc6be4cb86133ddb1c398bc3b3ca5207026d1c0
  Author: Mykola Kostenok <c_mykolak@nvidia.com>
  Date:   Sun Jun 6 11:24:32 2021 +0300

     mlxsw: core: Set thermal zone polling delay argument to real value at init

     [ Upstream commit 2fd8d84ce3095e8a7b5fe96532c91b1b9e07339c ]

     Thermal polling delay argument for modules and gearboxes thermal zones
     used to be initialized with zero value, while actual delay was used to
     be set by mlxsw_thermal_set_mode() by thermal operation callback
     set_mode(). After operations set_mode()/get_mode() have been removed by
     cited commits, modules and gearboxes thermal zones always have polling
     time set to zero and do not perform temperature monitoring.
```

```
    Set non-zero "polling_delay" in thermal_zone_device_register() routine,
    thus, the relevant thermal zones will perform thermal monitoring.

    Cc: Andrzej Pietrasiewicz <andrzej.p@collabora.com>
    Fixes: 5d7bd8aa7c35 ("thermal: Simplify or eliminate unnecessary set_mode() methods")
    Fixes: 1ee14820fd8e ("thermal: remove get_mode() operation of drivers")
    Signed-off-by: Mykola Kostenok <c_mykolak@nvidia.com>
    Acked-by: Vadim Pasternak <vadimp@nvidia.com>
    Reviewed-by: Jiri Pirko <jiri@nvidia.com>
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f313da6d46f4a800263547a0040d7c593641bd7d
Author: Petr Machata <petrm@nvidia.com>
Date:   Sun Jun 6 11:24:30 2021 +0300

    mlxsw: reg: Spectrum-3: Enforce lowest max-shaper burst size of 11

    [ Upstream commit 306b9228c097b4101c150ccd262372ded8348644 ]

    A max-shaper is the HW component responsible for delaying egress traffic
    above a configured transmission rate. Burst size is the amount of traffic
    that is allowed to pass without accounting. The burst size value needs to
    be such that it can be expressed as 2^BS * 512 bits, where BS lies in a
    certain ASIC-dependent range. mlxsw enforces that this holds before
    attempting to configure the shaper.

    The assumption for Spectrum-3 was that the lower limit of BS would be 5,
    like for Spectrum-1. But as of now, the limit is still 11. Therefore fix
    the driver accordingly, so that incorrect values are rejected early with a
    proper message.

    Fixes: 23effa2479ba ("mlxsw: reg: Add max_shaper_bs to QoS ETS Element Configuration")
    Reported-by: Maksym Yaremchuk <maksymy@nvidia.com>
    Signed-off-by: Petr Machata <petrm@nvidia.com>
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1b949c70206178b12027f66edc088d40375b5cb
Author: Du Cheng <ducheng2@gmail.com>
Date:   Mon May 10 12:16:49 2021 +0800

    mac80211: fix skb length check in ieee80211_scan_rx()

    [ Upstream commit e298aa358f0ca658406d524b6639fe389cb6e11e ]

    Replace hard-coded compile-time constants for header length check
    with dynamic determination based on the frame type. Otherwise, we
    hit a validation WARN_ON in cfg80211 later.

    Fixes: cd418ba63f0c ("mac80211: convert S1G beacon to scan results")
    Reported-by: syzbot+405843667e93b9790fc1@syzkaller.appspotmail.com
    Signed-off-by: Du Cheng <ducheng2@gmail.com>
    Link: https://lore.kernel.org/r/20210510041649.589754-1-ducheng2@gmail.com
    [style fixes, reword commit message]
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit faca4702ab228af9a02e8d365552de4499600668
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Mon Apr 26 21:28:02 2021 +0200

    staging: rtl8723bs: fix monitor netdev register/unregister
```

```
    [ Upstream commit b90f51e8e1f5014c01c82a7bf4c611643d0a8bcb ]

    Due to the locking changes and callbacks happening inside
    cfg80211, we need to use cfg80211 versions of the register
    and unregister functions if called within cfg80211 methods,
    otherwise deadlocks occur.

    Fixes: a05829a7222e ("cfg80211: avoid holding the RTNL when calling the driver")
    Acked-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Link:
https://lore.kernel.org/r/20210426212801.3d902cc9e6f4.Ie0b1e0c545920c61400a4b7d0f384ea61feb6
45a@changeid
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2eb4e0b3631832a4291c8bf4c9db873f60b128c8
Author: Sven Eckelmann <sven@narfation.org>
Date:   Tue May 18 21:00:27 2021 +0200

    batman-adv: Avoid WARN_ON timing related checks

    [ Upstream commit 9f460ae31c4435fd022c443a6029352217a16ac1 ]

    The soft/batadv interface for a queued OGM can be changed during the time
    the OGM was queued for transmission and when the OGM is actually
    transmitted by the worker.

    But WARN_ON must be used to denote kernel bugs and not to print simple
    warnings. A warning can simply be printed using pr_warn.

    Reported-by: Tetsuo Handa <penguin-kernel@i-love.sakura.ne.jp>
    Reported-by: syzbot+c0b807de416427ff3dd1@syzkaller.appspotmail.com
    Fixes: ef0a937f7a14 ("batman-adv: consider outgoing interface in OGM sending")
    Signed-off-by: Sven Eckelmann <sven@narfation.org>
    Signed-off-by: Simon Wunderlich <sw@simonwunderlich.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 476de3f94ef44b77fb781572cd1e0a612d1c46ab
Author: Matthew Bobrowski <repnop@google.com>
Date:   Fri Jun 11 13:32:06 2021 +1000

    fanotify: fix copy_event_to_user() fid error clean up

    [ Upstream commit f644bc449b37cc32d3ce7b36a88073873aa21bd5 ]

    Ensure that clean up is performed on the allocated file descriptor and
    struct file object in the event that an error is encountered while copying
    fid info objects. Currently, we return directly to the caller when an error
    is experienced in the fid info copying helper, which isn't ideal given that
    the listener process could be left with a dangling file descriptor in their
    fdtable.

    Fixes: 5e469c830fdb ("fanotify: copy event fid info to user")
    Fixes: 44d705b0370b ("fanotify: report name info for FAN_DIR_MODIFY event")
    Link: https://lore.kernel.org/linux-
fsdevel/YMKv1U7tNPK955ho@google.com/T/#m15361cd6399dad4396aad650de25dbf6b312288e
    Link:
https://lore.kernel.org/r/1ef8ae9100101eb1a91763c516c2e9a3a3b112bd.1623376346.git.repnop@goo
gle.com
    Signed-off-by: Matthew Bobrowski <repnop@google.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a2aff09807fbe4018c269d3773a629949058b210
```

```
Author: Jim Mattson <jmattson@google.com>
Date:   Wed Jun 2 13:52:24 2021 -0700

    kvm: LAPIC: Restore guard to prevent illegal APIC register access

    [ Upstream commit 218bf772bddd221489c38dde6ef8e917131161f6 ]

    Per the SDM, "any access that touches bytes 4 through 15 of an APIC
    register may cause undefined behavior and must not be executed."
    Worse, such an access in kvm_lapic_reg_read can result in a leak of
    kernel stack contents. Prior to commit 01402cf81051 ("kvm: LAPIC:
    write down valid APIC registers"), such an access was explicitly
    disallowed. Restore the guard that was removed in that commit.

    Fixes: 01402cf81051 ("kvm: LAPIC: write down valid APIC registers")
    Signed-off-by: Jim Mattson <jmattson@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Message-Id: <20210602205224.3189316-1-jmattson@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 28788dc5c70597395b6b451dae4549bbaa8e2c56
Author: yangerkun <yangerkun@huawei.com>
Date:   Tue Jun 15 18:23:32 2021 -0700

    mm/memory-failure: make sure wait for page writeback in memory_failure

    [ Upstream commit e8675d291ac007e1c636870db880f837a9ea112a ]

    Our syzkaller trigger the "BUG_ON(!list_empty(&inode->i_wb_list))" in
    clear_inode:

      kernel BUG at fs/inode.c:519!
      Internal error: Oops - BUG: 0 [#1] SMP
      Modules linked in:
      Process syz-executor.0 (pid: 249, stack limit = 0x00000000a12409d7)
      CPU: 1 PID: 249 Comm: syz-executor.0 Not tainted 4.19.95
      Hardware name: linux,dummy-virt (DT)
      pstate: 80000005 (Nzcv daif -PAN -UAO)
      pc : clear_inode+0x280/0x2a8
      lr : clear_inode+0x280/0x2a8
      Call trace:
        clear_inode+0x280/0x2a8
        ext4_clear_inode+0x38/0xe8
        ext4_free_inode+0x130/0xc68
        ext4_evict_inode+0xb20/0xcb8
        evict+0x1a8/0x3c0
        iput+0x344/0x460
        do_unlinkat+0x260/0x410
        __arm64_sys_unlinkat+0x6c/0xc0
        el0_svc_common+0xdc/0x3b0
        el0_svc_handler+0xf8/0x160
        el0_svc+0x10/0x218
      Kernel panic - not syncing: Fatal exception

    A crash dump of this problem show that someone called __munlock_pagevec
    to clear page LRU without lock_page: do_mmap -> mmap_region -> do_munmap
    -> munlock_vma_pages_range -> __munlock_pagevec.

    As a result memory_failure will call identify_page_state without
    wait_on_page_writeback.  And after truncate_error_page clear the mapping
    of this page.  end_page_writeback won't call sb_clear_inode_writeback to
    clear inode->i_wb_list.  That will trigger BUG_ON in clear_inode!

    Fix it by checking PageWriteback too to help determine should we skip
```

```
    wait_on_page_writeback.

    Link: https://lkml.kernel.org/r/20210604084705.3729204-1-yangerkun@huawei.com
    Fixes: 0bc1f8b0682c ("hwpoison: fix the handling path of the victimized page frame that
belong to non-LRU")
    Signed-off-by: yangerkun <yangerkun@huawei.com>
    Acked-by: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Cc: Jan Kara <jack@suse.cz>
    Cc: Theodore Ts'o <tytso@mit.edu>
    Cc: Oscar Salvador <osalvador@suse.de>
    Cc: Yu Kuai <yukuai3@huawei.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 43ea6532ea877eb23ce52eb60dd7eebd86d5376f
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Tue Jun 15 08:39:52 2021 +0100

    afs: Fix an IS_ERR() vs NULL check

    [ Upstream commit a33d62662d275cee22888fa7760fe09d5b9cd1f9 ]

    The proc_symlink() function returns NULL on error, it doesn't return
    error pointers.

    Fixes: 5b86d4ff5dce ("afs: Implement network namespacing")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: David Howells <dhowells@redhat.com>
    cc: linux-afs@lists.infradead.org
    Link: https://lore.kernel.org/r/YLjMRKX40pTrJvgf@mwanda/
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a5fadcb0ac108c6d15cb41527fb4be7a47ff041
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Tue May 18 22:11:08 2021 +0800

    dmaengine: stedma40: add missing iounmap() on error in d40_probe()

    [ Upstream commit fffdaba402cea79b8d219355487d342ec23f91c6 ]

    Add the missing iounmap() before return from d40_probe()
    in the error handling case.

    Fixes: 8d318a50b3d7 ("DMAENGINE: Support for ST-Ericssons DMA40 block v3")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Reviewed-by: Linus Walleij <linus.walleij@linaro.org>
    Link: https://lore.kernel.org/r/20210518141108.1324127-1-yangyingliang@huawei.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c8e0794226f4f1eb1988e64d950b416f4dfb456a
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Fri May 21 19:13:12 2021 -0700

    dmaengine: SF_PDMA depends on HAS_IOMEM

    [ Upstream commit 8e2e4f3c58528c6040b5762b666734f8cceba568 ]

    When CONFIG_HAS_IOMEM is not set/enabled, certain iomap() family
    functions [including ioremap(), devm_ioremap(), etc.] are not
    available.
    Drivers that use these functions should depend on HAS_IOMEM so that
```

```
    they do not cause build errors.

    Mends this build error:
    s390-linux-ld: drivers/dma/sf-pdma/sf-pdma.o: in function `sf_pdma_probe':
    sf-pdma.c:(.text+0x1668): undefined reference to `devm_ioremap_resource'

    Fixes: 6973886ad58e ("dmaengine: sf-pdma: add platform DMA support for HiFive Unleashed
A00")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Reported-by: kernel test robot <lkp@intel.com>
    Cc: Green Wan <green.wan@sifive.com>
    Cc: Vinod Koul <vkoul@kernel.org>
    Cc: dmaengine@vger.kernel.org
    Link: https://lore.kernel.org/r/20210522021313.16405-4-rdunlap@infradead.org
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 55b1c329a15770a886b384166dd825551e084b24
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Fri May 21 19:13:11 2021 -0700

    dmaengine: QCOM_HIDMA_MGMT depends on HAS_IOMEM

    [ Upstream commit 0cfbb589d67f16fa55b26ae02b69c31b52e344b1 ]

    When CONFIG_HAS_IOMEM is not set/enabled, certain iomap() family
    functions [including ioremap(), devm_ioremap(), etc.] are not
    available.
    Drivers that use these functions should depend on HAS_IOMEM so that
    they do not cause build errors.

    Rectifies these build errors:
    s390-linux-ld: drivers/dma/qcom/hidma_mgmt.o: in function `hidma_mgmt_probe':
    hidma_mgmt.c:(.text+0x780): undefined reference to `devm_ioremap_resource'
    s390-linux-ld: drivers/dma/qcom/hidma_mgmt.o: in function `hidma_mgmt_init':
    hidma_mgmt.c:(.init.text+0x126): undefined reference to `of_address_to_resource'
    s390-linux-ld: hidma_mgmt.c:(.init.text+0x16e): undefined reference to
`of_address_to_resource'

    Fixes: 67a2003e0607 ("dmaengine: add Qualcomm Technologies HIDMA channel driver")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Reported-by: kernel test robot <lkp@intel.com>
    Cc: Sinan Kaya <okaya@codeaurora.org>
    Cc: Vinod Koul <vkoul@kernel.org>
    Cc: dmaengine@vger.kernel.org
    Link: https://lore.kernel.org/r/20210522021313.16405-3-rdunlap@infradead.org
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a215987f731bacd4302c7a9e8663890f84ea4082
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Fri May 21 19:13:10 2021 -0700

    dmaengine: ALTERA_MSGDMA depends on HAS_IOMEM

    [ Upstream commit 253697b93c2a1c237d34d3ae326e394aeb0ca7b3 ]

    When CONFIG_HAS_IOMEM is not set/enabled, certain iomap() family
    functions [including ioremap(), devm_ioremap(), etc.] are not
    available.
    Drivers that use these functions should depend on HAS_IOMEM so that
    they do not cause build errors.

    Repairs this build error:
    s390-linux-ld: drivers/dma/altera-msgdma.o: in function `request_and_map':
```

```
    altera-msgdma.c:(.text+0x14b0): undefined reference to `devm_ioremap'

    Fixes: a85c6f1b2921 ("dmaengine: Add driver for Altera / Intel mSGDMA IP core")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Reported-by: kernel test robot <lkp@intel.com>
    Cc: Stefan Roese <sr@denx.de>
    Cc: Vinod Koul <vkoul@kernel.org>
    Cc: dmaengine@vger.kernel.org
    Reviewed-by: Stefan Roese <sr@denx.de>
    Phone: (+49)-8142-66989-51 Fax: (+49)-8142-66989-80 Email: sr@denx.de
    Link: https://lore.kernel.org/r/20210522021313.16405-2-rdunlap@infradead.org
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b476c74c1ff0a44f45bd2f1e45c288d25ec43d77
Author: Quanyang Wang <quanyang.wang@windriver.com>
Date:   Fri Apr 30 14:40:41 2021 +0800

    dmaengine: xilinx: dpdma: initialize registers before request_irq

    [ Upstream commit 538ea65a9fd1194352a41313bff876b74b5d90c5 ]

    In some scenarios (kdump), dpdma hardware irqs has been enabled when
    calling request_irq in probe function, and then the dpdma irq handler
    xilinx_dpdma_irq_handler is invoked to access xdev->chan[i]. But at
    this moment xdev->chan[i] hasn't been initialized.

    We should ensure the dpdma controller to be in a consistent and
    clean state before further initialization. So add dpdma_hw_init()
    to do this.

    Furthermore, in xilinx_dpdma_disable_irq, disable all interrupts
    instead of error interrupts.

    This patch is to fix the kdump kernel crash as below:

    [    3.696128] Unable to handle kernel NULL pointer dereference at virtual address
000000000000012c
    [    3.696710] xilinx-zynqmp-dpdma fd4c0000.dma-controller: Xilinx DPDMA engine is
probed
    [    3.704900] Mem abort info:
    [    3.704902]   ESR = 0x96000005
    [    3.704905]   EC = 0x25: DABT (current EL), IL = 32 bits
    [    3.704907]   SET = 0, FnV = 0
    [    3.704912]   EA = 0, S1PTW = 0
    [    3.713800] ahci-ceva fd0c0000.ahci: supply ahci not found, using dummy regulator
    [    3.715585] Data abort info:
    [    3.715587]   ISV = 0, ISS = 0x00000005
    [    3.715589]   CM = 0, WnR = 0
    [    3.715592] [000000000000012c] user address but active_mm is swapper
    [    3.715596] Internal error: Oops: 96000005 [#1] SMP
    [    3.715599] Modules linked in:
    [    3.715608] CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.10.0-12170-g60894882155f-
dirty #77
    [    3.723937] Hardware name: ZynqMP ZCU102 Rev1.0 (DT)
    [    3.723942] pstate: 80000085 (Nzcv daIf -PAN -UAO -TCO BTYPE=--)
    [    3.723956] pc : xilinx_dpdma_irq_handler+0x418/0x560
    [    3.793049] lr : xilinx_dpdma_irq_handler+0x3d8/0x560
    [    3.798089] sp : ffffffc01186bdf0
    [    3.801388] x29: ffffffc01186bdf0 x28: ffffffc011836f28
    [    3.806692] x27: ffffff8023e0ac80 x26: 0000000000000080
    [    3.811996] x25: 0000000008000408 x24: 0000000000000003
    [    3.817300] x23: ffffffc01186be70 x22: ffffffc011291740
    [    3.822604] x21: 0000000000000000 x20: 0000000008000408
    [    3.827908] x19: 0000000000000000 x18: 0000000000000010
```

```
[    3.833212] x17: 0000000000000000 x16: 0000000000000000
[    3.838516] x15: 0000000000000000 x14: ffffffc011291740
[    3.843820] x13: ffffffc02eb4d000 x12: 0000000034d4d91d
[    3.849124] x11: 0000000000000040 x10: ffffffc0112d2d48
[    3.854428] x9 : ffffffc0112d2d40 x8 : ffffff8021c00268
[    3.859732] x7 : 0000000000000000 x6 : ffffffc011836000
[    3.865036] x5 : 0000000000000003 x4 : 0000000000000000
[    3.870340] x3 : 0000000000000001 x2 : 0000000000000000
[    3.875644] x1 : 0000000000000000 x0 : 000000000000012c
[    3.880948] Call trace:
[    3.883382]  xilinx_dpdma_irq_handler+0x418/0x560
[    3.888079]  __handle_irq_event_percpu+0x5c/0x178
[    3.892774]  handle_irq_event_percpu+0x34/0x98
[    3.897210]  handle_irq_event+0x44/0xb8
[    3.901030]  handle_fasteoi_irq+0xd0/0x190
[    3.905117]  generic_handle_irq+0x30/0x48
[    3.909111]  __handle_domain_irq+0x64/0xc0
[    3.913192]  gic_handle_irq+0x78/0xa0
[    3.916846]  el1_irq+0xc4/0x180
[    3.919982]  cpuidle_enter_state+0x134/0x2f8
[    3.924243]  cpuidle_enter+0x38/0x50
[    3.927810]  call_cpuidle+0x1c/0x40
[    3.931290]  do_idle+0x20c/0x270
[    3.934502]  cpu_startup_entry+0x28/0x58
[    3.938410]  rest_init+0xbc/0xcc
[    3.941631]  arch_call_rest_init+0x10/0x1c
[    3.945718]  start_kernel+0x51c/0x558
```

    Fixes: 7cbb0c63de3f ("dmaengine: xilinx: dpdma: Add the Xilinx DisplayPort DMA engine driver")
    Signed-off-by: Quanyang Wang <quanyang.wang@windriver.com>
    Link: https://lore.kernel.org/r/20210430064041.4058180-1-quanyang.wang@windriver.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit becd2ff7ebf657a8a1397daa20ef05e5b0ba6720
Author: Zhen Lei <thunder.leizhen@huawei.com>
Date:   Sat May 8 11:00:56 2021 +0800

    dmaengine: fsl-dpaa2-qdma: Fix error return code in two functions

    [ Upstream commit 17866bc6b2ae1c3075c9fe7bcbeb8ea50eb4c3fc ]

    Fix to return a negative error code from the error handling case instead
    of 0, as done elsewhere in the function where it is.

    Fixes: 7fdf9b05c73b ("dmaengine: fsl-dpaa2-qdma: Add NXP dpaa2 qDMA controller driver for Layerscape SoCs")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: Zhen Lei <thunder.leizhen@huawei.com>
    Link: https://lore.kernel.org/r/20210508030056.2027-1-thunder.leizhen@huawei.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6b82f6921a36e839b4c526294c6226a1ea265e75
Author: Dave Jiang <dave.jiang@intel.com>
Date:   Mon Apr 26 16:32:24 2021 -0700

    dmaengine: idxd: add missing dsa driver unregister

    [ Upstream commit 077cdb355b3d8ee0f258856962e6dac06e744401 ]

    The idxd_unregister_driver() has never been called for the idxd driver upon
    removal. Add fix to call unregister driver on module removal.

```
    Fixes: c52ca478233c ("dmaengine: idxd: add configuration component of driver")
    Signed-off-by: Dave Jiang <dave.jiang@intel.com>
    Link: https://lore.kernel.org/r/161947994449.1053102.13189942817915448216.stgit@djiang5-
desk3.ch.intel.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c1ec6d46b63d366e2e6dcaf288a99423d09f8df8
Author: Dave Jiang <dave.jiang@intel.com>
Date:   Mon Apr 26 16:09:19 2021 -0700

    dmaengine: idxd: add engine 'struct device' missing bus type assignment

    [ Upstream commit 1c4841ccbd2b185587010d6178aac11953f61d4c ]

    engine 'struct device' setup is missing assigning the bus type. Add it to
    dsa_bus_type.

    Fixes: 75b911309060 ("dmaengine: idxd: fix engine conf_dev lifetime")
    Signed-off-by: Dave Jiang <dave.jiang@intel.com>
    Link: https://lore.kernel.org/r/161947841562.984844.17505646725993659651.stgit@djiang5-
desk3.ch.intel.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```