```
commit 7da3d9e6cd75cca48bfd99ba55c874900c0135d9
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Sat Jun 22 08:09:16 2019 +0200

    Linux 5.1.13

commit d7dcfd25208d0d17f407cd09fc8aa4571cf50228
Author: Andrea Arcangeli <aarcange@redhat.com>
Date:   Thu Jun 13 15:56:11 2019 -0700

    coredump: fix race condition between collapse_huge_page() and core dumping

    commit 59ea6d06cfa9247b586a695c21f94afa7183af74 upstream.

    When fixing the race conditions between the coredump and the mmap_sem
    holders outside the context of the process, we focused on
    mmget_not_zero()/get_task_mm() callers in 04f5866e41fb70 ("coredump: fix
    race condition between mmget_not_zero()/get_task_mm() and core
    dumping"), but those aren't the only cases where the mmap_sem can be
    taken outside of the context of the process as Michal Hocko noticed
    while backporting that commit to older -stable kernels.

    If mmgrab() is called in the context of the process, but then the
    mm_count reference is transferred outside the context of the process,
    that can also be a problem if the mmap_sem has to be taken for writing
    through that mm_count reference.

    khugepaged registration calls mmgrab() in the context of the process,
    but the mmap_sem for writing is taken later in the context of the
    khugepaged kernel thread.

    collapse_huge_page() after taking the mmap_sem for writing doesn't
    modify any vma, so it's not obvious that it could cause a problem to the
    coredump, but it happens to modify the pmd in a way that breaks an
    invariant that pmd_trans_huge_lock() relies upon.  collapse_huge_page()
    needs the mmap_sem for writing just to block concurrent page faults that
    call pmd_trans_huge_lock().

    Specifically the invariant that "!pmd_trans_huge()" cannot become a
    "pmd_trans_huge()" doesn't hold while collapse_huge_page() runs.

    The coredump will call __get_user_pages() without mmap_sem for reading,
    which eventually can invoke a lockless page fault which will need a
    functional pmd_trans_huge_lock().

    So collapse_huge_page() needs to use mmget_still_valid() to check it's
    not running concurrently with the coredump...  as long as the coredump
    can invoke page faults without holding the mmap_sem for reading.

    This has "Fixes: khugepaged" to facilitate backporting, but in my view
    it's more a bug in the coredump code that will eventually have to be
    rewritten to stop invoking page faults without the mmap_sem for reading.
    So the long term plan is still to drop all mmget_still_valid().

    Link: http://lkml.kernel.org/r/20190607161558.32104-1-aarcange@redhat.com
    Fixes: ba76149f47d8 ("thp: khugepaged")
    Signed-off-by: Andrea Arcangeli <aarcange@redhat.com>
    Reported-by: Michal Hocko <mhocko@suse.com>
    Acked-by: Michal Hocko <mhocko@suse.com>
    Acked-by: Kirill A. Shutemov <kirill.shutemov@linux.intel.com>
    Cc: Oleg Nesterov <oleg@redhat.com>
    Cc: Jann Horn <jannh@google.com>
    Cc: Hugh Dickins <hughd@google.com>
    Cc: Mike Rapoport <rppt@linux.vnet.ibm.com>
```

```
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: Jason Gunthorpe <jgg@mellanox.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 3111e132f46e64440878d7b855fc1709d20ff7d8
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Tue May 28 22:49:05 2019 -0700

    nvme-tcp: fix queue mapping when queue count is limited

    commit 6486199378a505c58fddc47459631235c9fb7638 upstream.

    When the controller supports less queues than requested, we
    should make sure that queue mapping does the right thing and
    not assume that all queues are available. This fixes a crash
    when the controller supports less queues than requested.

    The rules are:
    1. if no write queues are requested, we assign the available queues
       to the default queue map. The default and read queue maps share the
       existing queues.
    2. if write queues are requested:
      - first make sure that read queue map gets the requested
        nr_io_queues count
      - then grant the default queue map the minimum between the requested
        nr_write_queues and the remaining queues. If there are no available
        queues to dedicate to the default queue map, fallback to (1) and
        share all the queues in the existing queue map.

    Also, provide a log indication on how we constructed the different
    queue maps.

    Reported-by: Harris, James R <james.r.harris@intel.com>
    Tested-by: Jim Harris <james.r.harris@intel.com>
    Cc: <stable@vger.kernel.org> # v5.0+
    Suggested-by: Roy Shterman <roys@lightbitslabs.com>
    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d28c4813569c2a870978c39eb6769b19b9bc8a63
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Mon Apr 29 16:25:48 2019 -0700

    nvme-tcp: fix possible null deref on a timed out io queue connect

    commit f34e25898a608380a60135288019c4cb6013bec8 upstream.

    If I/O queue connect times out, we might have freed the queue socket
    already, so check for that on the error path in nvme_tcp_start_queue.

    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bd445693bef262bb9e02d44479027b3a0fc14ada
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Wed Apr 24 11:53:19 2019 -0700

    nvme-tcp: rename function to have nvme_tcp prefix

    commit efb973b19b88642bb7e08b8ce8e03b0bbd2a7e2a upstream.
```

```
    usually nvme_ prefix is for core functions.
    While we're cleaning up, remove redundant empty lines

    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Reviewed-by: Minwoo Im <minwoo.im@samsung.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b936bac388e05f30794cc3b4e50fdf62203edcef
Author: Yang Shi <yang.shi@linux.alibaba.com>
Date:   Thu Jun 13 15:56:05 2019 -0700

    mm: mmu_gather: remove __tlb_reset_range() for force flush

    commit 7a30df49f63ad92318ddf1f7498d1129a77dd4bd upstream.

    A few new fields were added to mmu_gather to make TLB flush smarter for
    huge page by telling what level of page table is changed.

    __tlb_reset_range() is used to reset all these page table state to
    unchanged, which is called by TLB flush for parallel mapping changes for
    the same range under non-exclusive lock (i.e.  read mmap_sem).

    Before commit dd2283f2605e ("mm: mmap: zap pages with read mmap_sem in
    munmap"), the syscalls (e.g.  MADV_DONTNEED, MADV_FREE) which may update
    PTEs in parallel don't remove page tables.  But, the forementioned
    commit may do munmap() under read mmap_sem and free page tables.  This
    may result in program hang on aarch64 reported by Jan Stancek.  The
    problem could be reproduced by his test program with slightly modified
    below.

    ---8<---

    static int map_size = 4096;
    static int num_iter = 500;
    static long threads_total;

    static void *distant_area;

    void *map_write_unmap(void *ptr)
    {
            int *fd = ptr;
            unsigned char *map_address;
            int i, j = 0;

            for (i = 0; i < num_iter; i++) {
                    map_address = mmap(distant_area, (size_t) map_size, PROT_WRITE |
PROT_READ,
                            MAP_SHARED | MAP_ANONYMOUS, -1, 0);
                    if (map_address == MAP_FAILED) {
                            perror("mmap");
                            exit(1);
                    }

                    for (j = 0; j < map_size; j++)
                            map_address[j] = 'b';

                    if (munmap(map_address, map_size) == -1) {
                            perror("munmap");
                            exit(1);
                    }
            }

            return NULL;
```

```
}

void *dummy(void *ptr)
{
        return NULL;
}

int main(void)
{
        pthread_t thid[2];

        /* hint for mmap in map_write_unmap() */
        distant_area = mmap(0, DISTANT_MMAP_SIZE, PROT_WRITE | PROT_READ,
                        MAP_ANONYMOUS | MAP_PRIVATE, -1, 0);
        munmap(distant_area, (size_t)DISTANT_MMAP_SIZE);
        distant_area += DISTANT_MMAP_SIZE / 2;

        while (1) {
                pthread_create(&thid[0], NULL, map_write_unmap, NULL);
                pthread_create(&thid[1], NULL, dummy, NULL);

                pthread_join(thid[0], NULL);
                pthread_join(thid[1], NULL);
        }
}
---8<---

The program may bring in parallel execution like below:

        t1                                              t2
munmap(map_address)
  downgrade_write(&mm->mmap_sem);
  unmap_region()
  tlb_gather_mmu()
    inc_tlb_flush_pending(tlb->mm);
  free_pgtables()
    tlb->freed_tables = 1
    tlb->cleared_pmds = 1

                                            pthread_exit()
                                            madvise(thread_stack, 8M, MADV_DONTNEED)
                                              zap_page_range()
                                                tlb_gather_mmu()
                                                  inc_tlb_flush_pending(tlb->mm);

  tlb_finish_mmu()
    if (mm_tlb_flush_nested(tlb->mm))
      __tlb_reset_range()

__tlb_reset_range() would reset freed_tables and cleared_* bits, but this
may cause inconsistency for munmap() which do free page tables.  Then it
may result in some architectures, e.g.  aarch64, may not flush TLB
completely as expected to have stale TLB entries remained.

Use fullmm flush since it yields much better performance on aarch64 and
non-fullmm doesn't yields significant difference on x86.

The original proposed fix came from Jan Stancek who mainly debugged this
issue, I just wrapped up everything together.

Jan's testing results:

v5.2-rc2-24-gbec7550cca10
---------------------------
         mean     stddev
```

```
    real    37.382   2.780
    user     1.420   0.078
    sys     54.658   1.855

    v5.2-rc2-24-gbec7550cca10 + "mm: mmu_gather: remove __tlb_reset_range() for force flush"
    ----------------------------------------------------------------------------------------_
             mean     stddev
    real    37.119   2.105
    user     1.548   0.087
    sys     55.698   1.357

    [akpm@linux-foundation.org: coding-style fixes]
    Link: http://lkml.kernel.org/r/1558322252-113575-1-git-send-email-
yang.shi@linux.alibaba.com
    Fixes: dd2283f2605e ("mm: mmap: zap pages with read mmap_sem in munmap")
    Signed-off-by: Yang Shi <yang.shi@linux.alibaba.com>
    Signed-off-by: Jan Stancek <jstancek@redhat.com>
    Reported-by: Jan Stancek <jstancek@redhat.com>
    Tested-by: Jan Stancek <jstancek@redhat.com>
    Suggested-by: Will Deacon <will.deacon@arm.com>
    Tested-by: Will Deacon <will.deacon@arm.com>
    Acked-by: Will Deacon <will.deacon@arm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Nick Piggin <npiggin@gmail.com>
    Cc: "Aneesh Kumar K.V" <aneesh.kumar@linux.ibm.com>
    Cc: Nadav Amit <namit@vmware.com>
    Cc: Minchan Kim <minchan@kernel.org>
    Cc: Mel Gorman <mgorman@suse.de>
    Cc: <stable@vger.kernel.org>    [4.20+]
    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 bb1313f8e3e009b327cb1a289c517a6291d44ca3
Author: Tobin C. Harding <tobin@kernel.org>
Date:   Fri May 31 22:30:29 2019 -0700

    ocfs2: fix error path kobject memory leak

    [ Upstream commit b9fba67b3806e21b98bd5a98dc3921a8e9b42d61 ]

    If a call to kobject_init_and_add() fails we should call kobject_put()
    otherwise we leak memory.

    Add call to kobject_put() in the error path of call to
    kobject_init_and_add().  Please note, this has the side effect that the
    release method is called if kobject_init_and_add() fails.

    Link: http://lkml.kernel.org/r/20190513033458.2824-1-tobin@kernel.org
    Signed-off-by: Tobin C. Harding <tobin@kernel.org>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Reviewed-by: Joseph Qi <joseph.qi@linux.alibaba.com>
    Cc: Mark Fasheh <mark@fasheh.com>
    Cc: Joel Becker <jlbec@evilplan.org>
    Cc: Junxiao Bi <junxiao.bi@oracle.com>
    Cc: Changwei Ge <gechangwei@live.cn>
    Cc: Gang He <ghe@suse.com>
    Cc: Jun Piao <piaojun@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 f8d7891817dd2fd3ed983074f9631bf422ab00d1
Author: Amit Cohen <amitc@mellanox.com>
Date:   Wed May 29 10:59:45 2019 +0300
```

```
    mlxsw: spectrum: Prevent force of 56G

    [ Upstream commit 275e928f19117d22f6d26dee94548baf4041b773 ]

    Force of 56G is not supported by hardware in Ethernet devices. This
    configuration fails with a bad parameter error from firmware.

    Add check of this case. Instead of trying to set 56G with autoneg off,
    return a meaningful error.

    Fixes: 56ade8fe3fe1 ("mlxsw: spectrum: Add initial support for Spectrum ASIC")
    Signed-off-by: Amit Cohen <amitc@mellanox.com>
    Acked-by: Jiri Pirko <jiri@mellanox.com>
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 16044c98e4d73f749e4afaa071618664d24b5da1
Author: Jason Yan <yanaijie@huawei.com>
Date:   Tue May 14 10:42:39 2019 +0800

    scsi: libsas: delete sas port if expander discover failed

    [ Upstream commit 3b0541791453fbe7f42867e310e0c9eb6295364d ]

    The sas_port(phy->port) allocated in sas_ex_discover_expander() will not be
    deleted when the expander failed to discover. This will cause resource leak
    and a further issue of kernel BUG like below:

    [159785.843156]  port-2:17:29: trying to add phy phy-2:17:29 fails: it's
    already part of another port
    [159785.852144] ------------[ cut here  ]------------
    [159785.856833] kernel BUG at drivers/scsi/scsi_transport_sas.c:1086!
    [159785.863000] Internal error: Oops - BUG: 0 [#1] SMP
    [159785.867866] CPU: 39 PID: 16993 Comm: kworker/u96:2 Tainted: G
    W  OE     4.19.25-vhulk1901.1.0.h111.aarch64 #1
    [159785.878458] Hardware name: Huawei Technologies Co., Ltd.
    Hi1620EVBCS/Hi1620EVBCS, BIOS Hi1620 CS B070 1P TA 03/21/2019
    [159785.889231] Workqueue: 0000:74:02.0_disco_q sas_discover_domain
    [159785.895224] pstate: 40c00009 (nZcv daif +PAN +UAO)
    [159785.900094] pc : sas_port_add_phy+0x188/0x1b8
    [159785.904524] lr : sas_port_add_phy+0x188/0x1b8
    [159785.908952] sp : ffff0001120e3b80
    [159785.912341] x29: ffff0001120e3b80 x28: 0000000000000000
    [159785.917727] x27: ffff802ade8f5400 x26: ffff0000681b7560
    [159785.923111] x25: ffff802adf11a800 x24: ffff0000680e8000
    [159785.928496] x23: ffff802ade8f5728 x22: ffff802ade8f5708
    [159785.933880] x21: ffff802adea2db40 x20: ffff802ade8f5400
    [159785.939264] x19: ffff802adea2d800 x18: 0000000000000010
    [159785.944649] x17: 00000000821bf734 x16: ffff00006714faa0
    [159785.950033] x15: ffff0000e8ab4ecf x14: 7261702079646165
    [159785.955417] x13: 726c612073277469 x12: ffff00006887b830
    [159785.960802] x11: ffff00006773eaa0 x10: 7968702079687020
    [159785.966186] x9 : 0000000000002453 x8 : 726f702072656874
    [159785.971570] x7 : 6f6e6120666f2074 x6 : ffff802bcfb21290
    [159785.976955] x5 : ffff802bcfb21290 x4 : 0000000000000000
    [159785.982339] x3 : ffff802bcfb298c8 x2 : 337752b234c2ab00
    [159785.987723] x1 : 337752b234c2ab00 x0 : 0000000000000000
    [159785.993108] Process kworker/u96:2 (pid: 16993, stack limit =
    0x0000000072dae094)
    [159786.000576] Call trace:
    [159786.003097]  sas_port_add_phy+0x188/0x1b8
    [159786.007179]  sas_ex_get_linkrate.isra.5+0x134/0x140
    [159786.012130]  sas_ex_discover_expander+0x128/0x408
```

```
      [159786.016906]  sas_ex_discover_dev+0x218/0x4c8
      [159786.021249]  sas_ex_discover_devices+0x9c/0x1a8
      [159786.025852]  sas_discover_root_expander+0x134/0x160
      [159786.030802]  sas_discover_domain+0x1b8/0x1e8
      [159786.035148]  process_one_work+0x1b4/0x3f8
      [159786.039230]  worker_thread+0x54/0x470
      [159786.042967]  kthread+0x134/0x138
      [159786.046269]  ret_from_fork+0x10/0x18
      [159786.049918] Code: 91322300 f0004402 91178042 97fe4c9b (d4210000)
      [159786.056083] Modules linked in: hns3_enet_ut(OE) hclge(OE) hnae3(OE)
      hisi_sas_test_hw(OE) hisi_sas_test_main(OE) serdes(OE)
      [159786.067202] ---[ end trace 03622b9e2d99e196  ]---
      [159786.071893] Kernel panic - not syncing: Fatal exception
      [159786.077190] SMP: stopping secondary CPUs
      [159786.081192] Kernel Offset: disabled
      [159786.084753] CPU features: 0x2,a2a00a38

      Fixes: 2908d778ab3e ("[SCSI] aic94xx: new driver")
      Reported-by: Jian Luo <luojian5@huawei.com>
      Signed-off-by: Jason Yan <yanaijie@huawei.com>
      CC: John Garry <john.garry@huawei.com>
      Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 5a52460378085e943af6412118f60d7d89041cfd
  Author: YueHaibing <yuehaibing@huawei.com>
  Date:   Mon May 27 22:22:09 2019 +0800

      scsi: scsi_dh_alua: Fix possible null-ptr-deref

      [ Upstream commit 12e750bc62044de096ab9a95201213fd912b9994 ]

      If alloc_workqueue fails in alua_init, it should return -ENOMEM, otherwise
      it will trigger null-ptr-deref while unloading module which calls
      destroy_workqueue dereference
      wq->lock like this:

      BUG: KASAN: null-ptr-deref in __lock_acquire+0x6b4/0x1ee0
      Read of size 8 at addr 0000000000000080 by task syz-executor.0/7045

      CPU: 0 PID: 7045 Comm: syz-executor.0 Tainted: G         C        5.1.0+ #28
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1
      Call Trace:
       dump_stack+0xa9/0x10e
       __kasan_report+0x171/0x18d
       ? __lock_acquire+0x6b4/0x1ee0
       kasan_report+0xe/0x20
       __lock_acquire+0x6b4/0x1ee0
       lock_acquire+0xb4/0x1b0
       __mutex_lock+0xd8/0xb90
       drain_workqueue+0x25/0x290
       destroy_workqueue+0x1f/0x3f0
       __x64_sys_delete_module+0x244/0x330
       do_syscall_64+0x72/0x2a0
       entry_SYSCALL_64_after_hwframe+0x49/0xbe

      Reported-by: Hulk Robot <hulkci@huawei.com>
      Fixes: 03197b61c5ec ("scsi_dh_alua: Use workqueue for RTPG")
      Signed-off-by: YueHaibing <yuehaibing@huawei.com>
      Reviewed-by: Bart Van Assche <bvanassche@acm.org>
      Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit cc58a0b1b4fbc926dbf416cb2d3f3e40f17f49c6
  Author: Lianbo Jiang <lijiang@redhat.com>
```

```
Date:   Mon May 27 08:59:34 2019 +0800

    scsi: smartpqi: properly set both the DMA mask and the coherent DMA mask

    [ Upstream commit 1d94f06e7f5df4064ef336b7b710f50143b64a53 ]

    When SME is enabled, the smartpqi driver won't work on the HP DL385 G10
    machine, which causes the failure of kernel boot because it fails to
    allocate pqi error buffer. Please refer to the kernel log:
    ....
    [    9.431749] usbcore: registered new interface driver uas
    [    9.441524] Microsemi PQI Driver (v1.1.4-130)
    [    9.442956] i40e 0000:04:00.0: fw 6.70.48768 api 1.7 nvm 10.2.5
    [    9.447237] smartpqi 0000:23:00.0: Microsemi Smart Family Controller found
            Starting dracut initqueue hook...
    [  OK  ] Started Show Plymouth Boot Scre[    9.471654] Broadcom NetXtreme-C/E driver
bnxt_en v1.9.1
    en.
    [  OK  ] Started Forward Password Requests to Plymouth Directory Watch.
    [[0;[    9.487108] smartpqi 0000:23:00.0: failed to allocate PQI error buffer
    ....
    [  139.050544] dracut-initqueue[949]: Warning: dracut-initqueue timeout - starting
timeout scripts
    [  139.589779] dracut-initqueue[949]: Warning: dracut-initqueue timeout - starting
timeout scripts

    Basically, the fact that the coherent DMA mask value wasn't set caused the
    driver to fall back to SWIOTLB when SME is active.

    For correct operation, lets call the dma_set_mask_and_coherent() to
    properly set the mask for both streaming and coherent, in order to inform
    the kernel about the devices DMA addressing capabilities.

    Signed-off-by: Lianbo Jiang <lijiang@redhat.com>
    Acked-by: Don Brace <don.brace@microsemi.com>
    Tested-by: Don Brace <don.brace@microsemi.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f0638d50b319890165c6bff037a40db1e28a467b
Author: Varun Prakash <varun@chelsio.com>
Date:   Wed May 22 20:10:55 2019 +0530

    scsi: libcxgbi: add a check for NULL pointer in cxgbi_check_route()

    [ Upstream commit cc555759117e8349088e0c5d19f2f2a500bafdbd ]

    ip_dev_find() can return NULL so add a check for NULL pointer.

    Signed-off-by: Varun Prakash <varun@chelsio.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b68f9cbff4448ee9bcddc91af7edbbfbaa2622ce
Author: Max Uvarov <muvarov@gmail.com>
Date:   Tue May 28 13:00:52 2019 +0300

    net: phy: dp83867: Set up RGMII TX delay

    [ Upstream commit 2b892649254fec01678c64f16427622b41fa27f4 ]

    PHY_INTERFACE_MODE_RGMII_RXID is less then TXID
    so code to set tx delay is never called.

    Fixes: 2a10154abcb75 ("net: phy: dp83867: Add TI dp83867 phy")
```

```
    Signed-off-by: Max Uvarov <muvarov@gmail.com>
    Cc: Florian Fainelli <f.fainelli@gmail.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8ee9f9c48f2d3944de54c2f0094157347230519b
Author: Max Uvarov <muvarov@gmail.com>
Date:   Tue May 28 13:00:50 2019 +0300

    net: phy: dp83867: increase SGMII autoneg timer duration

    [ Upstream commit 1a97a477e666cbdededab93bd3754e508f0c09d7 ]

    After reset SGMII Autoneg timer is set to 2us (bits 6 and 5 are 01).
    That is not enough to finalize autonegatiation on some devices.
    Increase this timer duration to maximum supported 16ms.

    Signed-off-by: Max Uvarov <muvarov@gmail.com>
    Cc: Heiner Kallweit <hkallweit1@gmail.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 315cb046c6d37ce32d1f06c1a9af1004c5e38221
Author: Max Uvarov <muvarov@gmail.com>
Date:   Tue May 28 13:00:49 2019 +0300

    net: phy: dp83867: fix speed 10 in sgmii mode

    [ Upstream commit 333061b924539c0de081339643f45514f5f1c1e6 ]

    For supporting 10Mps speed in SGMII mode DP83867_10M_SGMII_RATE_ADAPT bit
    of DP83867_10M_SGMII_CFG register has to be cleared by software.
    That does not affect speeds 100 and 1000 so can be done on init.

    Signed-off-by: Max Uvarov <muvarov@gmail.com>
    Cc: Heiner Kallweit <hkallweit1@gmail.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 132a01b4dc554231c4143bc78ae5058b3ef7502c
Author: Russell King <rmk+kernel@armlinux.org.uk>
Date:   Tue May 28 10:27:21 2019 +0100

    net: phylink: ensure consistent phy interface mode

    [ Upstream commit c678726305b9425454be7c8a7624290b602602fc ]

    Ensure that we supply the same phy interface mode to mac_link_down() as
    we did for the corresponding mac_link_up() call.  This ensures that MAC
    drivers that use the phy interface mode in these methods can depend on
    mac_link_down() always corresponding to a mac_link_up() call for the
    same interface mode.

    Signed-off-by: Russell King <rmk+kernel@armlinux.org.uk>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4c17461bff063b5fa2b71a4359cd83af8570cc29
Author: Jes Sorensen <jsorensen@fb.com>
Date:   Fri Apr 19 16:35:44 2019 -0400

    blk-mq: Fix memory leak in error handling
```

```
    [ Upstream commit 41de54c64811bf087c8464fdeb43c6ad8be2686b ]

    If blk_mq_init_allocated_queue() fails, make sure to free the poll
    stat callback struct allocated.

    Signed-off-by: Jes Sorensen <jsorensen@fb.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a6c6ea45529b8abd1b416f12de7914689d59de48
Author: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
Date:   Tue May 28 13:10:46 2019 +0900

    net: sh_eth: fix mdio access in sh_eth_close() for R-Car Gen2 and RZ/A1 SoCs

    [ Upstream commit 315ca92dd863fecbffc0bb52ae0ac11e0398726a ]

    The sh_eth_close() resets the MAC and then calls phy_stop()
    so that mdio read access result is incorrect without any error
    according to kernel trace like below:

    ifconfig-216   [003] .n..   109.133124: mdio_access: ee700000.ethernet-ffffffff read
phy:0x01 reg:0x00 val:0xffff

    According to the hardware manual, the RMII mode should be set to 1
    before operation the Ethernet MAC. However, the previous code was not
    set to 1 after the driver issued the soft_reset in sh_eth_dev_exit()
    so that the mdio read access result seemed incorrect. To fix the issue,
    this patch adds a condition and set the RMII mode register in
    sh_eth_dev_exit() for R-Car Gen2 and RZ/A1 SoCs.

    Note that when I have tried to move the sh_eth_dev_exit() calling
    after phy_stop() on sh_eth_close(), but it gets worse (kernel panic
    happened and it seems that a register is accessed while the clock is
    off).

    Signed-off-by: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2750ffb4b5b817057c6d7e1d3820f89fd58e012c
Author: Sami Tolvanen <samitolvanen@google.com>
Date:   Fri May 24 15:11:18 2019 -0700

    arm64: use the correct function type for __arm64_sys_ni_syscall

    [ Upstream commit 1e29ab3186e33c77dbb2d7566172a205b59fa390 ]

    Calling sys_ni_syscall through a syscall_fn_t pointer trips indirect
    call Control-Flow Integrity checking due to a function type
    mismatch. Use SYSCALL_DEFINE0 for __arm64_sys_ni_syscall instead and
    remove the now unnecessary casts.

    Signed-off-by: Sami Tolvanen <samitolvanen@google.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5d3c03a5064d4a871528bc42130771e6bcff63d4
Author: Sami Tolvanen <samitolvanen@google.com>
Date:   Fri May 24 15:11:17 2019 -0700

    arm64: use the correct function type in SYSCALL_DEFINE0

    [ Upstream commit 0e358bd7b7ebd27e491dabed938eae254c17fe3b ]
```

```
    Although a syscall defined using SYSCALL_DEFINE0 doesn't accept
    parameters, use the correct function type to avoid indirect call
    type mismatches with Control-Flow Integrity checking.

    Signed-off-by: Sami Tolvanen <samitolvanen@google.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5a72ffd5d5a18e58d7ea1e7fcbf7936b843b9d4b
Author: Sami Tolvanen <samitolvanen@google.com>
Date:   Fri May 24 15:11:16 2019 -0700

    arm64: fix syscall_fn_t type

    [ Upstream commit 8ef8f368ce72b5e17f7c1f1ef15c38dcfd0fef64 ]

    Syscall wrappers in <asm/syscall_wrapper.h> use const struct pt_regs *
    as the argument type. Use const in syscall_fn_t as well to fix indirect
    call type mismatches with Control-Flow Integrity checking.

    Signed-off-by: Sami Tolvanen <samitolvanen@google.com>
    Reviewed-by: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 95f47c8a8b96c74657f94c70a6cd6aea885a078b
Author: Geert Uytterhoeven <geert@linux-m68k.org>
Date:   Tue May 28 16:24:23 2019 +0200

    ALSA: fireface: Use ULL suffixes for 64-bit constants

    [ Upstream commit 6954158a16404e7091cea494cd0a435ca2f90388 ]

    With gcc 4.1:

        sound/firewire/fireface/ff-protocol-latter.c: In function
'latter_switch_fetching_mode':
        sound/firewire/fireface/ff-protocol-latter.c:97: warning: integer constant is too
large for 'long' type
        sound/firewire/fireface/ff-protocol-latter.c: In function 'latter_begin_session':
        sound/firewire/fireface/ff-protocol-latter.c:170: warning: integer constant is too
large for 'long' type
        sound/firewire/fireface/ff-protocol-latter.c:197: warning: integer constant is too
large for 'long' type
        sound/firewire/fireface/ff-protocol-latter.c:205: warning: integer constant is too
large for 'long' type
        sound/firewire/fireface/ff-protocol-latter.c: In function 'latter_finish_session':
        sound/firewire/fireface/ff-protocol-latter.c:214: warning: integer constant is too
large for 'long' type

    Fix this by adding the missing "ULL" suffixes.
    Add the same suffix to the last constant, to maintain consistency.

    Fixes: fd1cc9de64c2ca6c ("ALSA: fireface: add support for Fireface UCX")
    Signed-off-by: Geert Uytterhoeven <geert@linux-m68k.org>
    Reviewed-by: Takashi Sakamoto <o-takashi@sakamocchi.jp>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b25820b9bc5ebd0bca8c7f21458df318fe5339e0
Author: Paul Mackerras <paulus@ozlabs.org>
Date:   Thu May 23 16:36:32 2019 +1000

    KVM: PPC: Book3S HV: Don't take kvm->lock around kvm_for_each_vcpu
```

```
    [ Upstream commit 5a3f49364c3ffa1107bd88f8292406e98c5d206c ]

    Currently the HV KVM code takes the kvm->lock around calls to
    kvm_for_each_vcpu() and kvm_get_vcpu_by_id() (which can call
    kvm_for_each_vcpu() internally).  However, that leads to a lock
    order inversion problem, because these are called in contexts where
    the vcpu mutex is held, but the vcpu mutexes nest within kvm->lock
    according to Documentation/virtual/kvm/locking.txt.  Hence there
    is a possibility of deadlock.

    To fix this, we simply don't take the kvm->lock mutex around these
    calls.  This is safe because the implementations of kvm_for_each_vcpu()
    and kvm_get_vcpu_by_id() have been designed to be able to be called
    locklessly.

    Signed-off-by: Paul Mackerras <paulus@ozlabs.org>
    Reviewed-by: Cédric Le Goater <clg@kaod.org>
    Signed-off-by: Paul Mackerras <paulus@ozlabs.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9b3bd09b6f57270f085788182ba1297cbf760b3b
Author: Paul Mackerras <paulus@ozlabs.org>
Date:   Wed May 29 11:54:00 2019 +1000

    KVM: PPC: Book3S: Use new mutex to synchronize access to rtas token list

    [ Upstream commit 1659e27d2bc1ef47b6d031abe01b467f18cb72d9 ]

    Currently the Book 3S KVM code uses kvm->lock to synchronize access
    to the kvm->arch.rtas_tokens list.  Because this list is scanned
    inside kvmppc_rtas_hcall(), which is called with the vcpu mutex held,
    taking kvm->lock cause a lock inversion problem, which could lead to
    a deadlock.

    To fix this, we add a new mutex, kvm->arch.rtas_token_lock, which nests
    inside the vcpu mutexes, and use that instead of kvm->lock when
    accessing the rtas token list.

    This removes the lockdep_assert_held() in kvmppc_rtas_tokens_free().
    At this point we don't hold the new mutex, but that is OK because
    kvmppc_rtas_tokens_free() is only called when the whole VM is being
    destroyed, and at that point nothing can be looking up a token in
    the list.

    Signed-off-by: Paul Mackerras <paulus@ozlabs.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 818fab4491696728e51e3e0b249a448b59b690b4
Author: Paul Mackerras <paulus@ozlabs.org>
Date:   Thu May 23 16:35:34 2019 +1000

    KVM: PPC: Book3S HV: Use new mutex to synchronize MMU setup

    [ Upstream commit 0d4ee88d92884c661fcafd5576da243aa943dc24 ]

    Currently the HV KVM code uses kvm->lock in conjunction with a flag,
    kvm->arch.mmu_ready, to synchronize MMU setup and hold off vcpu
    execution until the MMU-related data structures are ready.  However,
    this means that kvm->lock is being taken inside vcpu->mutex, which
    is contrary to Documentation/virtual/kvm/locking.txt and results in
    lockdep warnings.

    To fix this, we add a new mutex, kvm->arch.mmu_setup_lock, which nests
    inside the vcpu mutexes, and is taken in the places where kvm->lock
```

```
    was taken that are related to MMU setup.

    Additionally we take the new mutex in the vcpu creation code at the
    point where we are creating a new vcore, in order to provide mutual
    exclusion with kvmppc_update_lpcr() and ensure that an update to
    kvm->arch.lpcr doesn't get missed, which could otherwise lead to a
    stale vcore->lpcr value.

    Signed-off-by: Paul Mackerras <paulus@ozlabs.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 560e20f2307aeac798fb52145199a36d751d3d8b
Author: Gen Zhang <blackgod016574@gmail.com>
Date:   Tue May 28 09:12:39 2019 +0800

    dfs_cache: fix a wrong use of kfree in flush_cache_ent()

    [ Upstream commit 50fbc13dc12666f3604dc2555a47fc8c4e29162b ]

    In flush_cache_ent(), 'ce->ce_path' is allocated by kstrdup_const().
    It should be freed by kfree_const(), rather than kfree().

    Signed-off-by: Gen Zhang <blackgod016574@gmail.com>
    Reviewed-by: Paulo Alcantara <palcantara@suse.de>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 68a376d453bfd79f21bfb964eecfc13353eb8108
Author: Ross Lagerwall <ross.lagerwall@citrix.com>
Date:   Mon May 13 14:56:35 2019 +0100

    xenbus: Avoid deadlock during suspend due to open transactions

    [ Upstream commit d10e0cc113c9e1b64b5c6e3db37b5c839794f3df ]

    During a suspend/resume, the xenwatch thread waits for all outstanding
    xenstore requests and transactions to complete. This does not work
    correctly for transactions started by userspace because it waits for
    them to complete after freezing userspace threads which means the
    transactions have no way of completing, resulting in a deadlock. This is
    trivial to reproduce by running this script and then suspending the VM:

        import pyxs, time
        c = pyxs.client.Client(xen_bus_path="/dev/xen/xenbus")
        c.connect()
        c.transaction()
        time.sleep(3600)

    Even if this deadlock were resolved, misbehaving userspace should not
    prevent a VM from being migrated. So, instead of waiting for these
    transactions to complete before suspending, store the current generation
    id for each transaction when it is started. The global generation id is
    incremented during resume. If the caller commits the transaction and the
    generation id does not match the current generation id, return EAGAIN so
    that they try again. If the transaction was instead discarded, return OK
    since no changes were made anyway.

    This only affects users of the xenbus file interface. In-kernel users of
    xenbus are assumed to be well-behaved and complete all transactions
    before freezing.

    Signed-off-by: Ross Lagerwall <ross.lagerwall@citrix.com>
    Reviewed-by: Juergen Gross <jgross@suse.com>
    Signed-off-by: Boris Ostrovsky <boris.ostrovsky@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 30e2039d6bf6a133265da020221c3a787f3b06c6
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Sat May 25 22:21:51 2019 +0800

    xen/pvcalls: Remove set but not used variable

    [ Upstream commit 41349672e3cbc2e8349831f21253509c3415aa2b ]

    Fixes gcc '-Wunused-but-set-variable' warning:

    drivers/xen/pvcalls-front.c: In function pvcalls_front_sendmsg:
    drivers/xen/pvcalls-front.c:543:25: warning: variable bedata set but not used [-Wunused-
but-set-variable]
    drivers/xen/pvcalls-front.c: In function pvcalls_front_recvmsg:
    drivers/xen/pvcalls-front.c:638:25: warning: variable bedata set but not used [-Wunused-
but-set-variable]

    They are never used since introduction.

    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Reviewed-by: Juergen Gross <jgross@suse.com>
    Signed-off-by: Boris Ostrovsky <boris.ostrovsky@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b2d4f385490e30729a295f7fe3be101adf7ec4fc
Author: Madalin Bucur <madalin.bucur@nxp.com>
Date:   Mon May 27 15:24:05 2019 +0300

    dpaa_eth: use only online CPU portals

    [ Upstream commit 7aae703f8096d21e34ce5f34f16715587bc30902 ]

    Make sure only the portals for the online CPUs are used.
    Without this change, there are issues when someone boots with
    maxcpus=n, with n < actual number of cores available as frames
    either received or corresponding to the transmit confirmation
    path would be offered for dequeue to the offline CPU portals,
    getting lost.

    Signed-off-by: Madalin Bucur <madalin.bucur@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 71af3faddd52334c8adc5b8dbd67e9bb6fe95e74
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Tue May 28 09:14:30 2019 -0700

    ia64: fix build errors by exporting paddr_to_nid()

    [ Upstream commit 9a626c4a6326da4433a0d4d4a8a7d1571caf1ed3 ]

    Fix build errors on ia64 when DISCONTIGMEM=y and NUMA=y by
    exporting paddr_to_nid().

    Fixes these build errors:

    ERROR: "paddr_to_nid" [sound/core/snd-pcm.ko] undefined!
    ERROR: "paddr_to_nid" [net/sunrpc/sunrpc.ko] undefined!
    ERROR: "paddr_to_nid" [fs/cifs/cifs.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/video/fbdev/core/fb.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/usb/mon/usbmon.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/usb/core/usbcore.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/md/raid1.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/md/dm-mod.ko] undefined!
```

```
    ERROR: "paddr_to_nid" [drivers/md/dm-crypt.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/md/dm-bufio.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/ide/ide-core.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/ide/ide-cd_mod.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/gpu/drm/drm.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/char/agp/agpgart.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/block/nbd.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/block/loop.ko] undefined!
    ERROR: "paddr_to_nid" [drivers/block/brd.ko] undefined!
    ERROR: "paddr_to_nid" [crypto/ccm.ko] undefined!

    Reported-by: kbuild test robot <lkp@intel.com>
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Cc: Tony Luck <tony.luck@intel.com>
    Cc: Fenghua Yu <fenghua.yu@intel.com>
    Cc: linux-ia64@vger.kernel.org
    Signed-off-by: Tony Luck <tony.luck@intel.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 15bac827297cede1c6b5cab62d6d124286a11226
Author: Thomas Richter <tmricht@linux.ibm.com>
Date:   Wed May 22 16:46:01 2019 +0200

    perf record: Fix s390 missing module symbol and warning for non-root users

    [ Upstream commit 6738028dd57df064b969d8392c943ef3b3ae705d ]

    Command 'perf record' and 'perf report' on a system without kernel
    debuginfo packages uses /proc/kallsyms and /proc/modules to find
    addresses for kernel and module symbols. On x86 this works for root and
    non-root users.

    On s390, when invoked as non-root user, many of the following warnings
    are shown and module symbols are missing:

        proc/{kallsyms,modules} inconsistency while looking for
            "[sha1_s390]" module!

    Command 'perf record' creates a list of module start addresses by
    parsing the output of /proc/modules and creates a PERF_RECORD_MMAP
    record for the kernel and each module. The following function call
    sequence is executed:

      machine__create_kernel_maps
        machine__create_module
          modules__parse
            machine__create_module --> for each line in /proc/modules
              arch__fix_module_text_start

    Function arch__fix_module_text_start() is s390 specific. It opens
    file /sys/module/<name>/sections/.text to extract the module's .text
    section start address. On s390 the module loader prepends a header
    before the first section, whereas on x86 the module's text section
    address is identical the the module's load address.

    However module section files are root readable only. For non-root the
    read operation fails and machine__create_module() returns an error.
    Command perf record does not generate any PERF_RECORD_MMAP record
    for loaded modules. Later command perf report complains about missing
    module maps.

    To fix this function arch__fix_module_text_start() always returns
    success. For root users there is no change, for non-root users
    the module's load address is used as module's text start address
```

```
    (the prepended header then counts as part of the text section).

    This enable non-root users to use module symbols and avoid the
    warning when perf report is executed.

    Output before:

      [tmricht@m83lp54 perf]$ ./perf report -D | fgrep MMAP
      0 0x168 [0x50]: PERF_RECORD_MMAP ... x [kernel.kallsyms]_text

    Output after:

      [tmricht@m83lp54 perf]$ ./perf report -D | fgrep MMAP
      0 0x168 [0x50]: PERF_RECORD_MMAP ... x [kernel.kallsyms]_text
      0 0x1b8 [0x98]: PERF_RECORD_MMAP ... x /lib/modules/.../autofs4.ko.xz
      0 0x250 [0xa8]: PERF_RECORD_MMAP ... x /lib/modules/.../sha_common.ko.xz
      0 0x2f8 [0x98]: PERF_RECORD_MMAP ... x /lib/modules/.../des_generic.ko.xz

    Signed-off-by: Thomas Richter <tmricht@linux.ibm.com>
    Reviewed-by: Hendrik Brueckner <brueckner@linux.ibm.com>
    Cc: Heiko Carstens <heiko.carstens@de.ibm.com>
    Link: http://lkml.kernel.org/r/20190522144601.50763-4-tmricht@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9b73f5b89ebbc8ae75409f9b9320a1d21e54d574
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Wed May 22 14:32:48 2019 +0900

    perf namespace: Protect reading thread's namespace

    [ Upstream commit 6584140ba9e6762dd7ec73795243289b914f31f9 ]

    It seems that the current code lacks holding the namespace lock in
    thread__namespaces().  Otherwise it can see inconsistent results.

    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Hari Bathini <hbathini@linux.vnet.ibm.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Krister Johansen <kjlx@templeofstupid.com>
    Link: http://lkml.kernel.org/r/20190522053250.207156-2-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4a705e5ce7206bf020401f0c7909060babe858ac
Author: Harald Freudenberger <freude@linux.ibm.com>
Date:   Tue May 21 13:50:09 2019 +0200

    s390/zcrypt: Fix wrong dispatching for control domain CPRBs

    [ Upstream commit 7379e652797c0b9b5f6caea1576f2dff9ce6a708 ]

    The zcrypt device driver does not handle CPRBs which address
    a control domain correctly. This fix introduces a workaround:
    The domain field of the request CPRB is checked if there is
    a valid domain value in there. If this is true and the value
    is a control only domain (a domain which is enabled in the
    crypto config ADM mask but disabled in the AQM mask) the
    CPRB is forwarded to the default usage domain. If there is
    no default domain, the request is rejected with an ENODEV.

    This fix is important for maintaining crypto adapters. For
    example one LPAR can use a crypto adapter domain ('Control
    and Usage') but another LPAR needs to be able to maintain
    this adapter domain ('Control'). Scenarios like this did
```

```
    not work properly and the patch enables this.

    Signed-off-by: Harald Freudenberger <freude@linux.ibm.com>
    Signed-off-by: Heiko Carstens <heiko.carstens@de.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc6c4cbde736bccd22287ee6f95375d6d59d428a
Author: Shawn Landden <shawn@git.icu>
Date:   Sat May 18 15:32:38 2019 -0300

    perf data: Fix 'strncat may truncate' build failure with recent gcc

    [ Upstream commit 97acec7df172cd1e450f81f5e293c0aa145a2797 ]

    This strncat() is safe because the buffer was allocated with zalloc(),
    however gcc doesn't know that. Since the string always has 4 non-null
    bytes, just use memcpy() here.

        CC       /home/shawn/linux/tools/perf/util/data-convert-bt.o
      In file included from /usr/include/string.h:494,
                       from /home/shawn/linux/tools/lib/traceevent/event-parse.h:27,
                       from util/data-convert-bt.c:22:
      In function ‘strncat’,
          inlined from ‘string_set_value’ at util/data-convert-bt.c:274:4:
      /usr/include/powerpc64le-linux-gnu/bits/string_fortified.h:136:10: error:
‘__builtin_strncat’ output may be truncated copying 4 bytes from a string of length 4 [-
Werror=stringop-truncation]
        136 |   return __builtin___strncat_chk (__dest, __src, __len, __bos (__dest));
            |          ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    Signed-off-by: Shawn Landden <shawn@git.icu>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Wang Nan <wangnan0@huawei.com>
    LPU-Reference: 20190518183238.10954-1-shawn@git.icu
    Link: https://lkml.kernel.org/n/tip-289f1jice17ta7tr3tstm9jm@git.kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3f81d5ff634022c14b9f359f14698adeb1226a7a
Author: Sahitya Tummala <stummala@codeaurora.org>
Date:   Thu Jan 3 16:48:15 2019 +0530

    configfs: Fix use-after-free when accessing sd->s_dentry

    [ Upstream commit f6122ed2a4f9c9c1c073ddf6308d1b2ac10e0781 ]

    In the vfs_statx() context, during path lookup, the dentry gets
    added to sd->s_dentry via configfs_attach_attr(). In the end,
    vfs_statx() kills the dentry by calling path_put(), which invokes
    configfs_d_iput(). Ideally, this dentry must be removed from
    sd->s_dentry but it doesn't if the sd->s_count >= 3. As a result,
    sd->s_dentry is holding reference to a stale dentry pointer whose
    memory is already freed up. This results in use-after-free issue,
    when this stale sd->s_dentry is accessed later in
    configfs_readdir() path.

    This issue can be easily reproduced, by running the LTP test case -
    sh fs_racer_file_list.sh /config
    (https://github.com/linux-test-
project/ltp/blob/master/testcases/kernel/fs/racer/fs_racer_file_list.sh)

    Fixes: 76ae281f6307 ('configfs: fix race between dentry put and lookup')
    Signed-off-by: Sahitya Tummala <stummala@codeaurora.org>
```

```
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0cad0ddd2c766d2200214c4d00d2cbd644c05711
Author: Bard Liao <yung-chuan.liao@linux.intel.com>
Date:   Mon May 27 00:58:32 2019 +0800

    ALSA: hda - Force polling mode on CNL for fixing codec communication

    [ Upstream commit fa763f1b2858752e6150ffff46886a1b7faffc82 ]

    We observed the same issue as reported by commit a8d7bde23e7130686b7662
    ("ALSA: hda - Force polling mode on CFL for fixing codec communication")
    We don't have a better solution. So apply the same workaround to CNL.

    Signed-off-by: Bard Liao <yung-chuan.liao@linux.intel.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bafddfd69436e264abd883f7e3c1a724bcbfd98d
Author: Yingjoe Chen <yingjoe.chen@mediatek.com>
Date:   Tue May 7 22:20:32 2019 +0800

    i2c: dev: fix potential memory leak in i2cdev_ioctl_rdwr

    [ Upstream commit a0692f0eef91354b62c2b4c94954536536be5425 ]

    If I2C_M_RECV_LEN check failed, msgs[i].buf allocated by memdup_user
    will not be freed. Pump index up so it will be freed.

    Fixes: 838bfa6049fb ("i2c-dev: Add support for I2C_M_RECV_LEN")
    Signed-off-by: Yingjoe Chen <yingjoe.chen@mediatek.com>
    Signed-off-by: Wolfram Sang <wsa@the-dreams.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9385e669d4532aea92c0780da0be97ba1e23aaad
Author: Dmitry Bogdanov <dmitry.bogdanov@aquantia.com>
Date:   Sat May 25 09:58:03 2019 +0000

    net: aquantia: fix LRO with FCS error

    [ Upstream commit eaeb3b7494ba9159323814a8ce8af06a9277d99b ]

    Driver stops producing skbs on ring if a packet with FCS error
    was coalesced into LRO session. Ring gets hang forever.

    Thats a logical error in driver processing descriptors:
    When rx_stat indicates MAC Error, next pointer and eop flags
    are not filled. This confuses driver so it waits for descriptor 0
    to be filled by HW.

    Solution is fill next pointer and eop flag even for packets with FCS error.

    Fixes: bab6de8fd180b ("net: ethernet: aquantia: Atlantic A0 and B0 specific functions.")
    Signed-off-by: Igor Russkikh <igor.russkikh@aquantia.com>
    Signed-off-by: Dmitry Bogdanov <dmitry.bogdanov@aquantia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4a6ff3e2df5cf1131d51a35c650f96928c248ebb
Author: Igor Russkikh <Igor.Russkikh@aquantia.com>
Date:   Sat May 25 09:57:59 2019 +0000

    net: aquantia: tx clean budget logic error
```

```
    [ Upstream commit 31bafc49a7736989e4c2d9f7280002c66536e590 ]

    In case no other traffic happening on the ring, full tx cleanup
    may not be completed. That may cause socket buffer to overflow
    and tx traffic to stuck until next activity on the ring happens.

    This is due to logic error in budget variable decrementor.
    Variable is compared with zero, and then post decremented,
    causing it to become MAX_INT. Solution is remove decrementor
    from the `for` statement and rewrite it in a clear way.

    Fixes: b647d3980948e ("net: aquantia: Add tx clean budget and valid budget handling
logic")
    Signed-off-by: Igor Russkikh <igor.russkikh@aquantia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7251bb77528e14ba3e08fe7f0b5e4be6fd0cba0f
Author: Lucas Stach <l.stach@pengutronix.de>
Date:   Tue May 21 14:53:40 2019 +0200

    drm/etnaviv: lock MMU while dumping core

    [ Upstream commit 1396500d673bd027683a0609ff84dca7eb6ea2e7 ]

    The devcoredump needs to operate on a stable state of the MMU while
    it is writing the MMU state to the coredump. The missing lock
    allowed both the userspace submit, as well as the GPU job finish
    paths to mutate the MMU state while a coredump is under way.

    Fixes: a8c21a5451d8 (drm/etnaviv: add initial etnaviv DRM driver)
    Reported-by: David Jander <david@protonic.nl>
    Signed-off-by: Lucas Stach <l.stach@pengutronix.de>
    Tested-by: David Jander <david@protonic.nl>
    Reviewed-by: Philipp Zabel <p.zabel@pengutronix.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cc8ba5f3b9755882555762a2c4fadc56f4517228
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Thu May 16 12:42:20 2019 +0200

    ACPI/PCI: PM: Add missing wakeup.flags.valid checks

    [ Upstream commit 9a51c6b1f9e0239a9435db036b212498a2a3b75c ]

    Both acpi_pci_need_resume() and acpi_dev_needs_resume() check if the
    current ACPI wakeup configuration of the device matches what is
    expected as far as system wakeup from sleep states is concerned, as
    reflected by the device_may_wakeup() return value for the device.

    However, they only should do that if wakeup.flags.valid is set for
    the device's ACPI companion, because otherwise the wakeup.prepare_count
    value for it is meaningless.

    Add the missing wakeup.flags.valid checks to these functions.

    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Reviewed-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ff6f3ec62db15bc8b8d07af351e0ffa50ea35acd
Author: Kees Cook <keescook@chromium.org>
Date:   Fri May 24 13:20:19 2019 -0700

    net: tulip: de4x5: Drop redundant MODULE_DEVICE_TABLE()
```

```
    [ Upstream commit 3e66b7cc50ef921121babc91487e1fb98af1ba6e ]

    Building with Clang reports the redundant use of MODULE_DEVICE_TABLE():

    drivers/net/ethernet/dec/tulip/de4x5.c:2110:1: error: redefinition of
'__mod_eisa__de4x5_eisa_ids_device_table'
    MODULE_DEVICE_TABLE(eisa, de4x5_eisa_ids);
    ^
    ./include/linux/module.h:229:21: note: expanded from macro 'MODULE_DEVICE_TABLE'
    extern typeof(name) __mod_##type##__##name##_device_table               \
                        ^
    <scratch space>:90:1: note: expanded from here
    __mod_eisa__de4x5_eisa_ids_device_table
    ^
    drivers/net/ethernet/dec/tulip/de4x5.c:2100:1: note: previous definition is here
    MODULE_DEVICE_TABLE(eisa, de4x5_eisa_ids);
    ^
    ./include/linux/module.h:229:21: note: expanded from macro 'MODULE_DEVICE_TABLE'
    extern typeof(name) __mod_##type##__##name##_device_table               \
                        ^
    <scratch space>:85:1: note: expanded from here
    __mod_eisa__de4x5_eisa_ids_device_table
    ^

    This drops the one further from the table definition to match the common
    use of MODULE_DEVICE_TABLE().

    Fixes: 07563c711fbc ("EISA bus MODALIAS attributes support")
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 632ebf6d706070821e2fcf8a764e23adf4251d18
Author: Ioana Radulescu <ruxandra.radulescu@nxp.com>
Date:   Fri May 24 18:15:16 2019 +0300

    dpaa2-eth: Use PTR_ERR_OR_ZERO where appropriate

    [ Upstream commit bd8460fa4de46e9d6177af4fe33bf0763a7af4b7 ]

    Use PTR_ERR_OR_ZERO instead of PTR_ERR in cases where
    zero is a valid input. Reported by smatch.

    Signed-off-by: Ioana Radulescu <ruxandra.radulescu@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 399a7963f8b3d863f87405f1f4ef339be779ed69
Author: Ioana Radulescu <ruxandra.radulescu@nxp.com>
Date:   Fri May 24 18:15:15 2019 +0300

    dpaa2-eth: Fix potential spectre issue

    [ Upstream commit 5a20a093d965560f632b2ec325f8876918f78165 ]

    Smatch reports a potential spectre vulnerability in the dpaa2-eth
    driver, where the value of rxnfc->fs.location (which is provided
    from user-space) is used as index in an array.

    Add a call to array_index_nospec() to sanitize the access.

    Signed-off-by: Ioana Radulescu <ruxandra.radulescu@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit fd670f49dd3054679d35f7da5ec9f31619f16dd0
Author: Pavel Begunkov <asml.silence@gmail.com>
Date:   Sun May 26 12:35:47 2019 +0300

    io_uring: Fix __io_uring_register() false success

    [ Upstream commit a278682dad37fd2f8d2f30d8e84e376a856ab472 ]

    If io_copy_iov() fails, it will break the loop and report success,
    albeit partially completed operation.

    Signed-off-by: Pavel Begunkov <asml.silence@gmail.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 84296dc648061f565ca99676aab575dd465c6d3c
Author: Biao Huang <biao.huang@mediatek.com>
Date:   Fri May 24 14:26:09 2019 +0800

    net: stmmac: dwmac-mediatek: modify csr_clk value to fix mdio read/write fail

    [ Upstream commit f4ca7a9260dfe700f2a16f0881825de625067515 ]

    1. the frequency of csr clock is 66.5MHz, so the csr_clk value should
    be 0 other than 5.
    2. the csr_clk can be got from device tree, so remove initialization here.

    Fixes: 9992f37e346b ("stmmac: dwmac-mediatek: add support for mt2712")
    Signed-off-by: Biao Huang <biao.huang@mediatek.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 32e577bc4bb57fd5f5225adb229f7443ccaa74af
Author: Biao Huang <biao.huang@mediatek.com>
Date:   Fri May 24 14:26:08 2019 +0800

    net: stmmac: fix csr_clk can't be zero issue

    [ Upstream commit 5e7f7fc538d894b2d9aa41876b8dcf35f5fe11e6 ]

    The specific clk_csr value can be zero, and
    stmmac_clk is necessary for MDC clock which can be set dynamically.
    So, change the condition from plat->clk_csr to plat->stmmac_clk to
    fix clk_csr can't be zero issue.

    Fixes: cd7201f477b9 ("stmmac: MDC clock dynamically based on the csr clock input")
    Signed-off-by: Biao Huang <biao.huang@mediatek.com>
    Acked-by: Alexandre TORGUE <alexandre.torgue@st.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 88bf99a6fe9b824cf1f37610f48bcc26198a676a
Author: Biao Huang <biao.huang@mediatek.com>
Date:   Fri May 24 14:26:07 2019 +0800

    net: stmmac: update rx tail pointer register to fix rx dma hang issue.

    [ Upstream commit 4523a5611526709ec9b4e2574f1bb7818212651e ]

    Currently we will not update the receive descriptor tail pointer in
    stmmac_rx_refill. Rx dma will think no available descriptors and stop
    once received packets exceed DMA_RX_SIZE, so that the rx only test will fail.

    Update the receive tail pointer in stmmac_rx_refill to add more descriptors
```

```
    to the rx channel, so packets can be received continually

    Fixes: 54139cf3bb33 ("net: stmmac: adding multiple buffers for rx")
    Signed-off-by: Biao Huang <biao.huang@mediatek.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fc113d17aedc4a7000a30c1d16fa358d8053c4f3
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Thu May 23 15:00:41 2019 -0700

    gpio: fix gpio-adp5588 build errors

    [ Upstream commit e9646f0f5bb62b7d43f0968f39d536cfe7123b53 ]

    The gpio-adp5588 driver uses interfaces that are provided by
    GPIOLIB_IRQCHIP, so select that symbol in its Kconfig entry.

    Fixes these build errors:

    ../drivers/gpio/gpio-adp5588.c: In function ‘adp5588_irq_handler’:
    ../drivers/gpio/gpio-adp5588.c:266:26: error: ‘struct gpio_chip’ has no member named
‘irq’
                dev->gpio_chip.irq.domain, gpio));
                         ^
    ../drivers/gpio/gpio-adp5588.c: In function ‘adp5588_irq_setup’:
    ../drivers/gpio/gpio-adp5588.c:298:2: error: implicit declaration of function
‘gpiochip_irqchip_add_nested’ [-Werror=implicit-function-declaration]
      ret = gpiochip_irqchip_add_nested(&dev->gpio_chip,
      ^
    ../drivers/gpio/gpio-adp5588.c:307:2: error: implicit declaration of function
‘gpiochip_set_nested_irqchip’ [-Werror=implicit-function-declaration]
      gpiochip_set_nested_irqchip(&dev->gpio_chip,
      ^

    Fixes: 459773ae8dbb ("gpio: adp5588-gpio: support interrupt controller")
    Reported-by: kbuild test robot <lkp@intel.com>
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Cc: linux-gpio@vger.kernel.org
    Reviewed-by: Bartosz Golaszewski <bgolaszewski@baylibre.com>
    Acked-by: Michael Hennerich <michael.hennerich@analog.com>
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f808486448a2157a6254683822548c806312bde0
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Fri May 17 13:52:33 2019 +0200

    perf/ring-buffer: Always use {READ,WRITE}_ONCE() for rb->user_page data

    [ Upstream commit 4d839dd9e4356bbacf3eb0ab13a549b83b008c21 ]

    We must use {READ,WRITE}_ONCE() on rb->user_page data such that
    concurrent usage will see whole values. A few key sites were missing
    this.

    Suggested-by: Yabin Cui <yabinc@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
```

```
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: acme@kernel.org
    Cc: mark.rutland@arm.com
    Cc: namhyung@kernel.org
    Fixes: 7b732a750477 ("perf_counter: new output ABI - part 1")
    Link: http://lkml.kernel.org/r/20190517115418.394192145@infradead.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fc1b40e55d389e9c55a42ad88a796df4f5dbb1ac
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Fri May 17 13:52:32 2019 +0200

    perf/ring_buffer: Add ordering to rb->nest increment

    [ Upstream commit 3f9fbe9bd86c534eba2faf5d840fd44c6049f50e ]

    Similar to how decrementing rb->next too early can cause data_head to
    (temporarily) be observed to go backward, so too can this happen when
    we increment too late.

    This barrier() ensures the rb->head load happens after the increment,
    both the one in the 'goto again' path, as the one from
    perf_output_get_handle() -- albeit very unlikely to matter for the
    latter.

    Suggested-by: Yabin Cui <yabinc@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: acme@kernel.org
    Cc: mark.rutland@arm.com
    Cc: namhyung@kernel.org
    Fixes: ef60777c9abd ("perf: Optimize the perf_output() path by removing IRQ-disables")
    Link: http://lkml.kernel.org/r/20190517115418.309516009@infradead.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6421f2ef8525b9836971d3e17cd07fb4462a44c9
Author: Yabin Cui <yabinc@google.com>
Date:   Fri May 17 13:52:31 2019 +0200

    perf/ring_buffer: Fix exposing a temporarily decreased data_head

    [ Upstream commit 1b038c6e05ff70a1e66e3e571c2e6106bdb75f53 ]

    In perf_output_put_handle(), an IRQ/NMI can happen in below location and
    write records to the same ring buffer:

            ...
            local_dec_and_test(&rb->nest)
            ...                          <-- an IRQ/NMI can happen here
            rb->user_page->data_head = head;
            ...

    In this case, a value A is written to data_head in the IRQ, then a value
    B is written to data_head after the IRQ. And A > B. As a result,
    data_head is temporarily decreased from A to B. And a reader may see
    data_head < data_tail if it read the buffer frequently enough, which
```

```
    creates unexpected behaviors.

    This can be fixed by moving dec(&rb->nest) to after updating data_head,
    which prevents the IRQ/NMI above from updating data_head.

    [ Split up by peterz. ]

    Signed-off-by: Yabin Cui <yabinc@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@kernel.org>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: mark.rutland@arm.com
    Fixes: ef60777c9abd ("perf: Optimize the perf_output() path by removing IRQ-disables")
    Link: http://lkml.kernel.org/r/20190517115418.224478157@infradead.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 436b1cb41173c73a1768f2e23cb8bb09c5a6deea
Author: Frank van der Linden <fllinden@amazon.com>
Date:   Wed May 22 22:17:45 2019 +0000

    x86/CPU/AMD: Don't force the CPB cap when running under a hypervisor

    [ Upstream commit 2ac44ab608705948564791ce1d15d43ba81a1e38 ]

    For F17h AMD CPUs, the CPB capability ('Core Performance Boost') is forcibly set,
    because some versions of that chip incorrectly report that they do not have it.

    However, a hypervisor may filter out the CPB capability, for good
    reasons. For example, KVM currently does not emulate setting the CPB
    bit in MSR_K7_HWCR, and unchecked MSR access errors will be thrown
    when trying to set it as a guest:

            unchecked MSR access error: WRMSR to 0xc0010015 (tried to write
0x0000000001000011) at rIP: 0xffffffff890638f4 (native_write_msr+0x4/0x20)

            Call Trace:
            boost_set_msr+0x50/0x80 [acpi_cpufreq]
            cpuhp_invoke_callback+0x86/0x560
            sort_range+0x20/0x20
            cpuhp_thread_fun+0xb0/0x110
            smpboot_thread_fn+0xef/0x160
            kthread+0x113/0x130
            kthread_create_worker_on_cpu+0x70/0x70
            ret_from_fork+0x35/0x40

    To avoid this issue, don't forcibly set the CPB capability for a CPU
    when running under a hypervisor.

    Signed-off-by: Frank van der Linden <fllinden@amazon.com>
    Acked-by: Borislav Petkov <bp@suse.de>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: bp@alien8.de
    Cc: jiaxun.yang@flygoat.com
```

```
    Fixes: 0237199186e7 ("x86/CPU/AMD: Set the CPB bit unconditionally on F17h")
    Link: http://lkml.kernel.org/r/20190522221745.GA15789@dev-dsk-fllinden-2c-c1893d73.us-
west-2.amazon.com
    [ Minor edits to the changelog. ]
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9f9fb3a6bbde028236be007d688d02e91d812a5f
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Wed May 22 11:45:13 2019 +0300

    mISDN: make sure device name is NUL terminated

    [ Upstream commit ccfb62f27beb295103e9392462b20a6ed807d0ea ]

    The user can change the device_name with the IMSETDEVNAME ioctl, but we
    need to ensure that the user's name is NUL terminated.  Otherwise it
    could result in a buffer overflow when we copy the name back to the user
    with IMGETDEVINFO ioctl.

    I also changed two strcpy() calls which handle the name to strscpy().
    Hopefully, there aren't any other ways to create a too long name, but
    it's nice to do this as a kernel hardening measure.

    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fc0193f074635325859cab6ea59eb2cd1ef3da85
Author: Jia-Ju Bai <baijiaju1990@gmail.com>
Date:   Wed May 22 14:33:58 2019 +0300

    usb: xhci: Fix a potential null pointer dereference in xhci_debugfs_create_endpoint()

    [ Upstream commit 5bce256f0b528624a34fe907db385133bb7be33e ]

    In xhci_debugfs_create_slot(), kzalloc() can fail and
    dev->debugfs_private will be NULL.
    In xhci_debugfs_create_endpoint(), dev->debugfs_private is used without
    any null-pointer check, and can cause a null pointer dereference.

    To fix this bug, a null-pointer check is added in
    xhci_debugfs_create_endpoint().

    This bug is found by a runtime fuzzing tool named FIZZER written by us.

    [subjet line change change, add potential -Mathais]
    Signed-off-by: Jia-Ju Bai <baijiaju1990@gmail.com>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Mathias Nyman <mathias.nyman@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b53f68045b8b668660a8ccacbd2616589c1c4479
Author: Anju T Sudhakar <anju@linux.vnet.ibm.com>
Date:   Mon May 20 14:27:53 2019 +0530

    powerpc/powernv: Return for invalid IMC domain

    [ Upstream commit b59bd3527fe3c1939340df558d7f9d568fc9f882 ]

    Currently init_imc_pmu() can fail either because we try to register an
    IMC unit with an invalid domain (i.e an IMC node not supported by the
    kernel) or something went wrong while registering a valid IMC unit. In
    both the cases kernel provides a 'Register failed' error message.
```

```
    For example when trace-imc node is not supported by the kernel, but
    skiboot advertises a trace-imc node we print:

      IMC Unknown Device type
      IMC PMU (null) Register failed

    To avoid confusion just print the unknown device type message, before
    attempting PMU registration, so the second message isn't printed.

    Fixes: 8f95faaac56c ("powerpc/powernv: Detect and create IMC device")
    Reported-by: Pavaman Subramaniyam <pavsubra@in.ibm.com>
    Signed-off-by: Anju T Sudhakar <anju@linux.vnet.ibm.com>
    Reviewed-by: Madhavan Srinivasan <maddy@linux.vnet.ibm.com>
    [mpe: Reword change log a bit]
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 85984f9ba6d62a10a7e1bc71e391c579425cdc09
Author: Tony Lindgren <tony@atomide.com>
Date:   Mon May 6 14:08:54 2019 -0700

    clk: ti: clkctrl: Fix clkdm_clk handling

    [ Upstream commit 1cc54078d104f5b4d7e9f8d55362efa5a8daffdb ]

    We need to always call clkdm_clk_enable() and clkdm_clk_disable() even
    the clkctrl clock(s) enabled for the domain do not have any gate register
    bits. Otherwise clockdomains may never get enabled except when devices get
    probed with the legacy "ti,hwmods" devicetree property.

    Fixes: 88a172526c32 ("clk: ti: add support for clkctrl clocks")
    Signed-off-by: Tony Lindgren <tony@atomide.com>
    Signed-off-by: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9141d4bfc8bb2ee2da9b5528d4e2c83d804bf77e
Author: Jeffrin Jose T <jeffrin@rajagiritech.edu.in>
Date:   Wed May 15 12:14:04 2019 +0530

    selftests: netfilter: missing error check when setting up veth interface

    [ Upstream commit 82ce6eb1dd13fd12e449b2ee2c2ec051e6f52c43 ]

    A test for the basic NAT functionality uses ip command which needs veth
    device. There is a condition where the kernel support for veth is not
    compiled into the kernel and the test script breaks. This patch contains
    code for reasonable error display and correct code exit.

    Signed-off-by: Jeffrin Jose T <jeffrin@rajagiritech.edu.in>
    Acked-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1a9e8e2e591080d1f358fd4f481b715ddec83ccc
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Fri May 17 22:31:49 2019 +0800

    ipvs: Fix use-after-free in ip_vs_in

    [ Upstream commit 719c7d563c17b150877cee03a4b812a424989dfa ]

    BUG: KASAN: use-after-free in ip_vs_in.part.29+0xe8/0xd20 [ip_vs]
    Read of size 4 at addr ffff8881e9b26e2c by task sshd/5603
```

```
     CPU: 0 PID: 5603 Comm: sshd Not tainted 4.19.39+ #30
     Hardware name: Red Hat KVM, BIOS 0.5.1 01/01/2011
     Call Trace:
      dump_stack+0x71/0xab
      print_address_description+0x6a/0x270
      kasan_report+0x179/0x2c0
      ip_vs_in.part.29+0xe8/0xd20 [ip_vs]
      ip_vs_in+0xd8/0x170 [ip_vs]
      nf_hook_slow+0x5f/0xe0
      __ip_local_out+0x1d5/0x250
      ip_local_out+0x19/0x60
      __tcp_transmit_skb+0xba1/0x14f0
      tcp_write_xmit+0x41f/0x1ed0
      ? _copy_from_iter_full+0xca/0x340
      __tcp_push_pending_frames+0x52/0x140
      tcp_sendmsg_locked+0x787/0x1600
      ? tcp_sendpage+0x60/0x60
      ? inet_sk_set_state+0xb0/0xb0
      tcp_sendmsg+0x27/0x40
      sock_sendmsg+0x6d/0x80
      sock_write_iter+0x121/0x1c0
      ? sock_sendmsg+0x80/0x80
      __vfs_write+0x23e/0x370
      vfs_write+0xe7/0x230
      ksys_write+0xa1/0x120
      ? __ia32_sys_read+0x50/0x50
      ? __audit_syscall_exit+0x3ce/0x450
      do_syscall_64+0x73/0x200
      entry_SYSCALL_64_after_hwframe+0x44/0xa9
     RIP: 0033:0x7ff6f6147c60
     Code: 73 01 c3 48 8b 0d 28 12 2d 00 f7 d8 64 89 01 48 83 c8 ff c3 66 0f 1f 44 00 00 83
  3d 5d 73 2d 00 00 75 10 b8 01 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 31 c3 48 83
     RSP: 002b:00007ffd772ead18 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
     RAX: ffffffffffffffda RBX: 0000000000000034 RCX: 00007ff6f6147c60
     RDX: 0000000000000034 RSI: 000055df30a31270 RDI: 0000000000000003
     RBP: 000055df30a31270 R08: 0000000000000000 R09: 0000000000000000
     R10: 00007ffd772ead70 R11: 0000000000000246 R12: 00007ffd772ead74
     R13: 00007ffd772eae20 R14: 00007ffd772eae24 R15: 000055df2f12ddc0

     Allocated by task 6052:
      kasan_kmalloc+0xa0/0xd0
      __kmalloc+0x10a/0x220
      ops_init+0x97/0x190
      register_pernet_operations+0x1ac/0x360
      register_pernet_subsys+0x24/0x40
      0xffffffffc0ea016d
      do_one_initcall+0x8b/0x253
      do_init_module+0xe3/0x335
      load_module+0x2fc0/0x3890
      __do_sys_finit_module+0x192/0x1c0
      do_syscall_64+0x73/0x200
      entry_SYSCALL_64_after_hwframe+0x44/0xa9

     Freed by task 6067:
      __kasan_slab_free+0x130/0x180
      kfree+0x90/0x1a0
      ops_free_list.part.7+0xa6/0xc0
      unregister_pernet_operations+0x18b/0x1f0
      unregister_pernet_subsys+0x1d/0x30
      ip_vs_cleanup+0x1d/0xd2f [ip_vs]
      __x64_sys_delete_module+0x20c/0x300
      do_syscall_64+0x73/0x200
      entry_SYSCALL_64_after_hwframe+0x44/0xa9

     The buggy address belongs to the object at ffff8881e9b26600 which belongs to the cache
```

```
  kmalloc-4096 of size 4096
      The buggy address is located 2092 bytes inside of 4096-byte region [ffff8881e9b26600,
  ffff8881e9b27600)
      The buggy address belongs to the page:
      page:ffffea0007a6c800 count:1 mapcount:0 mapping:ffff888107c0e600 index:0x0
  compound_mapcount: 0
      flags: 0x17ffffc0008100(slab|head)
      raw: 0017ffffc0008100 dead000000000100 dead000000000200 ffff888107c0e600
      raw: 0000000000000000 0000000080070007 00000001ffffffff 0000000000000000
      page dumped because: kasan: bad access detected

      while unregistering ipvs module, ops_free_list calls
      __ip_vs_cleanup, then nf_unregister_net_hooks be called to
      do remove nf hook entries. It need a RCU period to finish,
      however net->ipvs is set to NULL immediately, which will
      trigger NULL pointer dereference when a packet is hooked
      and handled by ip_vs_in where net->ipvs is dereferenced.

      Another scene is ops_free_list call ops_free to free the
      net_generic directly while __ip_vs_cleanup finished, then
      calling ip_vs_in will triggers use-after-free.

      This patch moves nf_unregister_net_hooks from __ip_vs_cleanup()
      to __ip_vs_dev_cleanup(),  where rcu_barrier() is called by
      unregister_pernet_device -> unregister_pernet_operations,
      that will do the needed grace period.

      Reported-by: Hulk Robot <hulkci@huawei.com>
      Fixes: efe41606184e ("ipvs: convert to use pernet nf_hook api")
      Suggested-by: Julian Anastasov <ja@ssi.bg>
      Signed-off-by: YueHaibing <yuehaibing@huawei.com>
      Acked-by: Julian Anastasov <ja@ssi.bg>
      Signed-off-by: Simon Horman <horms@verge.net.au>
      Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 4e7ec2e90f45e85f69179658576e322f0b261403
  Author: Phil Sutter <phil@nwl.cc>
  Date:   Wed May 15 20:15:32 2019 +0200

      netfilter: nft_fib: Fix existence check support

      [ Upstream commit e633508a95289489d28faacb68b32c3e7e68ef6f ]

      NFTA_FIB_F_PRESENT flag was not always honored since eval functions did
      not call nft_fib_store_result in all cases.

      Given that in all callsites there is a struct net_device pointer
      available which holds the interface data to be stored in destination
      register, simplify nft_fib_store_result() to just accept that pointer
      instead of the nft_pktinfo pointer and interface index. This also
      allows to drop the index to interface lookup previously needed to get
      the name associated with given index.

      Fixes: 055c4b34b94f6 ("netfilter: nft_fib: Support existence check")
      Signed-off-by: Phil Sutter <phil@nwl.cc>
      Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit eaa5eb41f71600172ed1e76de26b7fe0a6902ad0
  Author: Jagdish Motwani <jagdish.motwani@sophos.com>
  Date:   Mon May 13 23:47:40 2019 +0530

      netfilter: nf_queue: fix reinject verdict handling
```

```
    [ Upstream commit 946c0d8e6ed43dae6527e878d0077c1e11015db0 ]

    This patch fixes netfilter hook traversal when there are more than 1 hooks
    returning NF_QUEUE verdict. When the first queue reinjects the packet,
    'nf_reinject' starts traversing hooks with a proper hook_index. However,
    if it again receives a NF_QUEUE verdict (by some other netfilter hook), it
    queues the packet with a wrong hook_index. So, when the second queue
    reinjects the packet, it re-executes hooks in between.

    Fixes: 960632ece694 ("netfilter: convert hook list to an array")
    Signed-off-by: Jagdish Motwani <jagdish.motwani@sophos.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1159921769f280a7e96c1cf726b78ec4a05e6837
Author: Stephane Eranian <eranian@google.com>
Date:   Mon May 20 17:52:46 2019 -0700

    perf/x86/intel/ds: Fix EVENT vs. UEVENT PEBS constraints

    [ Upstream commit 23e3983a466cd540ffdd2bbc6e0c51e31934f941 ]

    This patch fixes an bug revealed by the following commit:

      6b89d4c1ae85 ("perf/x86/intel: Fix INTEL_FLAGS_EVENT_CONSTRAINT* masking")

    That patch modified INTEL_FLAGS_EVENT_CONSTRAINT() to only look at the event code
    when matching a constraint. If code+umask were needed, then the
    INTEL_FLAGS_UEVENT_CONSTRAINT() macro was needed instead.
    This broke with some of the constraints for PEBS events.

    Several of them, including the one used for cycles:p, cycles:pp, cycles:ppp
    fell in that category and caused the event to be rejected in PEBS mode.
    In other words, on some platforms a cmdline such as:

      $ perf top -e cycles:pp

    would fail with -EINVAL.

    This patch fixes this bug by properly using INTEL_FLAGS_UEVENT_CONSTRAINT()
    when needed in the PEBS constraint tables.

    Reported-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Stephane Eranian <eranian@google.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: kan.liang@intel.com
    Link: http://lkml.kernel.org/r/20190521005246.423-1-eranian@google.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ac453193740951d77f3e680c1d1049191317a35b
Author: Florian Westphal <fw@strlen.de>
Date:   Tue Apr 30 14:53:11 2019 +0200

    netfilter: nf_tables: fix oops during rule dump

    [ Upstream commit 2c82c7e724ff51cab78e1afd5c2aaa31994fe41e ]

    We can oops in nf_tables_fill_rule_info().
```

```
    Its not possible to fetch previous element in rcu-protected lists
    when deletions are not prevented somehow: list_del_rcu poisons
    the ->prev pointer value.

    Before rcu-conversion this was safe as dump operations did hold
    nfnetlink mutex.

    Pass previous rule as argument, obtained by keeping a pointer to
    the previous rule during traversal.

    Fixes: d9adf22a291883 ("netfilter: nf_tables: use call_rcu in netlink dumps")
    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 347f026eba70ecce85998fc48cad72f465911a52
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Tue Apr 30 16:37:53 2019 +0800

    pinctrl: intel: Clear interrupt status in mask/unmask callback

    [ Upstream commit 670784fb4ebe54434e263837390e358405031d9e ]

    Commit a939bb57cd47 ("pinctrl: intel: implement gpio_irq_enable") was
    added because clearing interrupt status bit is required to avoid
    unexpected behavior.

    Turns out the unmask callback also needs the fix, which can solve weird
    IRQ triggering issues on I2C touchpad ELAN1200.

    Signed-off-by: Kai-Heng Feng <kai.heng.feng@canonical.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5c454589e0fc859300c4152a85fc88b6dc0c5315
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Wed May 15 12:52:23 2019 +0300

    staging: wilc1000: Fix some double unlock bugs in wilc_wlan_cleanup()

    [ Upstream commit fea69916360468e364a4988db25a5afa835f3406 ]

    If ->hif_read_reg() or ->hif_write_reg() fail then the code unlocks
    and keeps executing.  It should just return.

    Fixes: c5c77ba18ea6 ("staging: wilc1000: Add SDIO/SPI 802.11 driver")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 24e560bb1ba2c4192c6a23fc55c2eb888efc293e
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Mon May 13 14:07:18 2019 +0300

    Staging: vc04_services: Fix a couple error codes

    [ Upstream commit ca4e4efbefbbdde0a7bb3023ea08d491f4daf9b9 ]

    These are accidentally returning positive EINVAL instead of negative
    -EINVAL.  Some of the callers treat positive values as success.

    Fixes: 7b3ad5abf027 ("staging: Import the BCM2835 MMAL-based V4L2 camera driver.")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Acked-by: Stefan Wahren <stefan.wahren@i2se.com>
```

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

commit 41fa707720c9e71f79833c0960be2d01de65c1e7
Author: Chengguang Xu <cgxu519@gmail.com>
Date:   Mon May 6 19:01:02 2019 +0800

    staging: erofs: set sb->s_root to NULL when failing from __getname()

    [ Upstream commit f2dcb8841e6b155da098edae09125859ef7e853d ]

    Set sb->s_root to NULL when failing from __getname(),
    so that we can avoid double dput and unnecessary operations
    in generic_shutdown_super().

    Signed-off-by: Chengguang Xu <cgxu519@gmail.com>
    Reviewed-by: Chao Yu <yuchao0@huawei.com>
    Reviewed-by: Gao Xiang <gaoxiang25@huawei.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 83907b7c58ddbf80574d154d12f5e4e1701989a2
Author: Steve Moskovchenko <stevemo@skydio.com>
Date:   Tue Apr 2 23:28:56 2019 -0700

    iio: imu: mpu6050: Fix FIFO layout for ICM20602

    [ Upstream commit 1615fe41a1959a2ee2814ba62736b2bb54e9802a ]

    The MPU6050 driver has recently gained support for the
    ICM20602 IMU, which is very similar to MPU6xxx. However,
    the ICM20602's FIFO data specifically includes temperature
    readings, which were not present on MPU6xxx parts. As a
    result, the driver will under-read the ICM20602's FIFO
    register, causing the same (partial) sample to be returned
    for all reads, until the FIFO overflows.

    Fix this by adding a table of scan elements specifically
    for the ICM20602, which takes the extra temperature data
    into consideration.

    While we're at it, fix the temperature offset and scaling
    on ICM20602, since it uses different scale/offset constants
    than the rest of the MPU6xxx devices.

    Signed-off-by: Steve Moskovchenko <stevemo@skydio.com>
    Fixes: 22904bdff978 ("iio: imu: mpu6050: Add support for the ICM 20602 IMU")
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a8c94e6bdd026d57885f899333c04b01778535e
Author: Alaa Hleihel <alaa@mellanox.com>
Date:   Sun May 26 11:56:27 2019 +0300

    net/mlx5e: Avoid detaching non-existing netdev under switchdev mode

    After introducing dedicated uplink representor, the netdev instance
    set over the esw manager vport (PF) became no longer in use, so it was
    removed in the cited commit once we're on switchdev mode.
    However, the mlx5e_detach function was not updated accordingly, and it
    still tries to detach a non-existing netdev, causing a kernel crash.

    This patch fixes this issue.

    Fixes: aec002f6f82c ("net/mlx5e: Uninstantiate esw manager vport netdev on switchdev
```

```
mode")
    Signed-off-by: Alaa Hleihel <alaa@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0ab34cf296c3c7d326e57a38f61a5a86686bdda1
Author: Willem de Bruijn <willemb@google.com>
Date:   Fri Jun 7 17:57:48 2019 -0400

    net: correct udp zerocopy refcnt also when zerocopy only on append

    [ Upstream commit 522924b583082f51b8a2406624a2f27c22119b20 ]

    The below patch fixes an incorrect zerocopy refcnt increment when
    appending with MSG_MORE to an existing zerocopy udp skb.

      send(.., MSG_ZEROCOPY | MSG_MORE);    // refcnt 1
      send(.., MSG_ZEROCOPY | MSG_MORE);    // refcnt still 1 (bar frags)

    But it missed that zerocopy need not be passed at the first send. The
    right test whether the uarg is newly allocated and thus has extra
    refcnt 1 is not !skb, but !skb_zcopy.

      send(.., MSG_MORE);                   // <no uarg>
      send(.., MSG_ZEROCOPY);               // refcnt 1

    Fixes: 100f6d8e09905 ("net: correct zerocopy refcnt with udp MSG_MORE")
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Willem de Bruijn <willemb@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 284bc559f332f77a08cea0f0f23ff7024419e191
Author: Eli Britstein <elibr@mellanox.com>
Date:   Sun Jun 2 13:47:59 2019 +0000

    net/mlx5e: Support tagged tunnel over bond

    Stacked devices like bond interface may have a VLAN device on top of
    them. Detect lag state correctly under this condition, and return the
    correct routed net device, according to it the encap header is built.

    Fixes: e32ee6c78efa ("net/mlx5e: Support tunnel encap over tagged Ethernet")
    Signed-off-by: Eli Britstein <elibr@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1bb4eb4efe3afbd66d80168ce6968ecabc2ffab9
Author: Petr Machata <petrm@mellanox.com>
Date:   Tue Jun 11 10:19:45 2019 +0300

    mlxsw: spectrum_buffers: Reduce pool size on Spectrum-2

    Due to an issue on Spectrum-2, in front-panel ports split four ways, 2 out
    of 32 port buffers cannot be used. To work around this, the next FW release
    will mark them as unused, and will report correspondingly lower total
    shared buffer size. mlxsw will pick up the new value through a query to
    cap_total_buffer_size resource. However the initial size for shared buffer
    pool 0 is hard-coded and therefore needs to be updated.

    Thus reduce the pool size by 2.7 MiB (which corresponds to 2/32 of the
    total size of 42 MiB), and round down to the whole number of cells.
```

```
    Fixes: fe099bf682ab ("mlxsw: spectrum_buffers: Add Spectrum-2 shared buffer
configuration")
    Signed-off-by: Petr Machata <petrm@mellanox.com>
    Acked-by: Jiri Pirko <jiri@mellanox.com>
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b43417db5e3b932275692db1497a20619056d261
Author: Raed Salem <raeds@mellanox.com>
Date:   Sun Jun 2 12:04:08 2019 +0300

    net/mlx5e: Fix source port matching in fdb peer flow rule

    The cited commit changed the initialization placement of the eswitch
    attributes so it is done prior to parse tc actions function call,
    including among others the in_rep and in_mdev fields which are mistakenly
    reassigned inside the parse actions function.

    This breaks the source port matching criteria of the peer redirect rule.

    Fix by removing the now redundant reassignment of the already initialized
    fields.

    Fixes: 988ab9c7363a ("net/mlx5e: Introduce mlx5e_flow_esw_attr_init() helper")
    Signed-off-by: Raed Salem <raeds@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit acec2a3945117249dbf8909670036c73de16bc5c
Author: Jiri Pirko <jiri@mellanox.com>
Date:   Tue Jun 11 10:19:43 2019 +0300

    mlxsw: spectrum_flower: Fix TOS matching

    The TOS value was not extracted correctly. Fix it.

    Fixes: 87996f91f739 ("mlxsw: spectrum_flower: Add support for ip tos")
    Reported-by: Alexander Petrovskiy <alexpe@mellanox.com>
    Signed-off-by: Jiri Pirko <jiri@mellanox.com>
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dd3690f21ac3fd4bd9b69428f1c5f274282f14ab
Author: Chris Mi <chrism@mellanox.com>
Date:   Thu May 16 17:36:43 2019 +0800

    net/mlx5e: Add ndo_set_feature for uplink representor

    After we have a dedicated uplink representor, the new netdev ops
    doesn't support ndo_set_feature. Because of that, we can't change
    some features, eg. rxvlan. Now add it back.

    In this patch, I also do a cleanup for the features flag handling,
    eg. remove duplicate NETIF_F_HW_TC flag setting.

    Fixes: aec002f6f82c ("net/mlx5e: Uninstantiate esw manager vport netdev on switchdev
mode")
    Signed-off-by: Chris Mi <chrism@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Reviewed-by: Vlad Buslov <vladbu@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 01bf40c9cad61857b8058b666867ad83e6ff5fc9
Author: Ido Schimmel <idosch@mellanox.com>
Date:   Tue Jun 11 10:19:41 2019 +0300

    mlxsw: spectrum_router: Refresh nexthop neighbour when it becomes dead

    The driver tries to periodically refresh neighbours that are used to
    reach nexthops. This is done by periodically calling neigh_event_send().

    However, if the neighbour becomes dead, there is nothing we can do to
    return it to a connected state and the above function call is basically
    a NOP.

    This results in the nexthop never being written to the device's
    adjacency table and therefore never used to forward packets.

    Fix this by dropping our reference from the dead neighbour and
    associating the nexthop with a new neigbhour which we will try to
    refresh.

    Fixes: a7ff87acd995 ("mlxsw: spectrum_router: Implement next-hop routing")
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Reported-by: Alex Veber <alexve@mellanox.com>
    Tested-by: Alex Veber <alexve@mellanox.com>
    Acked-by: Jiri Pirko <jiri@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 37afcc01660c17ce3ac1529cd55d7b7216a0aa0d
Author: Edward Srouji <edwards@mellanox.com>
Date:   Thu May 23 19:45:38 2019 +0300

    net/mlx5: Update pci error handler entries and command translation

    Add missing entries for create/destroy UCTX and UMEM commands.
    This could get us wrong "unknown FW command" error in flows
    where we unbind the device or reset the driver.

    Also the translation of these commands from opcodes to string
    was missing.

    Fixes: 6e3722baac04 ("IB/mlx5: Use the correct commands for UMEM and UCTX allocation")
    Signed-off-by: Edward Srouji <edwards@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a21291c7a3ae13a6173d65990a9079aecfd27b2f
Author: Maxime Chevallier <maxime.chevallier@bootlin.com>
Date:   Wed Jun 12 17:18:38 2019 +0200

    net: ethtool: Allow matching on vlan DEI bit

    [ Upstream commit f0d2ca1531377e7da888913e277eefac05a59b6f ]

    Using ethtool, users can specify a classification action matching on the
    full vlan tag, which includes the DEI bit (also previously called CFI).

    However, when converting the ethool_flow_spec to a flow_rule, we use
    dissector keys to represent the matching patterns.

    Since the vlan dissector key doesn't include the DEI bit, this
    information was silently discarded when translating the ethtool
    flow spec in to a flow_rule.
```

```
    This commit adds the DEI bit into the vlan dissector key, and allows
    propagating the information to the driver when parsing the ethtool flow
    spec.

    Fixes: eca4205f9ec3 ("ethtool: add ethtool_rx_flow_spec to flow_rule structure
translator")
    Reported-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Signed-off-by: Maxime Chevallier <maxime.chevallier@bootlin.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 02b6efb81f2ca80e95d801dab0516ef0075ec1f0
Author: Robert Hancock <hancock@sedsystems.ca>
Date:   Wed Jun 12 14:33:32 2019 -0600

    net: dsa: microchip: Don't try to read stats for unused ports

    [ Upstream commit 6bb9e376c2a4cc5120c3bf5fd3048b9a0a6ec1f8 ]

    If some of the switch ports were not listed in the device tree, due to
    being unused, the ksz_mib_read_work function ended up accessing a NULL
    dp->slave pointer and causing an oops. Skip checking statistics for any
    unused ports.

    Fixes: 7c6ff470aa867f53 ("net: dsa: microchip: add MIB counter reading support")
    Signed-off-by: Robert Hancock <hancock@sedsystems.ca>
    Reviewed-by: Vivien Didelot <vivien.didelot@gmail.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f61bd3c7c2d81f02bb1baf52f7aa5d219ce19dc8
Author: Maxime Chevallier <maxime.chevallier@bootlin.com>
Date:   Tue Jun 11 11:51:43 2019 +0200

    net: mvpp2: prs: Use the correct helpers when removing all VID filters

    [ Upstream commit 6b7a3430c163455cf8a514d636bda52b04654972 ]

    When removing all VID filters, the mvpp2_prs_vid_entry_remove would be
    called with the TCAM id incorrectly used as a VID, causing the wrong
    TCAM entries to be invalidated.

    Fix this by directly invalidating entries in the VID range.

    Fixes: 56beda3db602 ("net: mvpp2: Add hardware offloading for VLAN filtering")
    Suggested-by: Yuri Chipchev <yuric@marvell.com>
    Signed-off-by: Maxime Chevallier <maxime.chevallier@bootlin.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 60c653e2f1ac36bf2b2cde21c1baff337a18c9b3
Author: Maxime Chevallier <maxime.chevallier@bootlin.com>
Date:   Tue Jun 11 11:51:42 2019 +0200

    net: mvpp2: prs: Fix parser range for VID filtering

    [ Upstream commit 46b0090a6636cf34c0e856f15dd03e15ba4cdda6 ]

    VID filtering is implemented in the Header Parser, with one range of 11
    vids being assigned for each no-loopback port.

    Make sure we use the per-port range when looking for existing entries in
    the Parser.
```

```
    Since we used a global range instead of a per-port one, this causes VIDs
    to be removed from the whitelist from all ports of the same PPv2
    instance.

    Fixes: 56beda3db602 ("net: mvpp2: Add hardware offloading for VLAN filtering")
    Suggested-by: Yuri Chipchev <yuric@marvell.com>
    Signed-off-by: Maxime Chevallier <maxime.chevallier@bootlin.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit af1c05e6c51fc92d43c3e0d911633f51c5b2c90b
Author: Stefano Brivio <sbrivio@redhat.com>
Date:   Tue Jun 11 00:27:06 2019 +0200

    geneve: Don't assume linear buffers in error handler

    [ Upstream commit eccc73a6b2cb6c04bfbc40a0769f3c428dfba232 ]

    In commit a07966447f39 ("geneve: ICMP error lookup handler") I wrongly
    assumed buffers from icmp_socket_deliver() would be linear. This is not
    the case: icmp_socket_deliver() only guarantees we have 8 bytes of linear
    data.

    Eric fixed this same issue for fou and fou6 in commits 26fc181e6cac
    ("fou, fou6: do not assume linear skbs") and 5355ed6388e2 ("fou, fou6:
    avoid uninit-value in gue_err() and gue6_err()").

    Use pskb_may_pull() instead of checking skb->len, and take into account
    the fact we later access the GENEVE header with udp_hdr(), so we also
    need to sum skb_transport_header() here.

    Reported-by: Guillaume Nault <gnault@redhat.com>
    Fixes: a07966447f39 ("geneve: ICMP error lookup handler")
    Signed-off-by: Stefano Brivio <sbrivio@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0cb7d3242192df201be97c61e394c87e18bf8a9d
Author: Stefano Brivio <sbrivio@redhat.com>
Date:   Tue Jun 11 00:27:05 2019 +0200

    vxlan: Don't assume linear buffers in error handler

    [ Upstream commit 8399a6930d12f5965230f4ff058228a4cc80c0b9 ]

    In commit c3a43b9fec8a ("vxlan: ICMP error lookup handler") I wrongly
    assumed buffers from icmp_socket_deliver() would be linear. This is not
    the case: icmp_socket_deliver() only guarantees we have 8 bytes of linear
    data.

    Eric fixed this same issue for fou and fou6 in commits 26fc181e6cac
    ("fou, fou6: do not assume linear skbs") and 5355ed6388e2 ("fou, fou6:
    avoid uninit-value in gue_err() and gue6_err()").

    Use pskb_may_pull() instead of checking skb->len, and take into account
    the fact we later access the VXLAN header with udp_hdr(), so we also
    need to sum skb_transport_header() here.

    Reported-by: Guillaume Nault <gnault@redhat.com>
    Fixes: c3a43b9fec8a ("vxlan: ICMP error lookup handler")
    Signed-off-by: Stefano Brivio <sbrivio@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 251c58f821872ba654b3e411c28ea7cc12a53e15
Author: Alaa Hleihel <alaa@mellanox.com>
Date:   Sun May 19 11:11:49 2019 +0300

    net/mlx5: Avoid reloading already removed devices

    Prior to reloading a device we must first verify that it was not already
    removed. Otherwise, the attempt to remove the device will do nothing, and
    in that case we will end up proceeding with adding an new device that no
    one was expecting to remove, leaving behind used resources such as EQs that
    causes a failure to destroy comp EQs and syndrome (0x30f433).

    Fix that by making sure that we try to remove and add a device (based on a
    protocol) only if the device is already added.

    Fixes: c5447c70594b ("net/mlx5: E-Switch, Reload IB interface when switching devlink
modes")
    Signed-off-by: Alaa Hleihel <alaa@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8882d7e99b47cfdf880c3af701becff29718bec8
Author: Stephen Barber <smbarber@chromium.org>
Date:   Fri Jun 14 23:42:37 2019 -0700

    vsock/virtio: set SOCK_DONE on peer shutdown

    [ Upstream commit 42f5cda5eaf4396a939ae9bb43bb8d1d09c1b15c ]

    Set the SOCK_DONE flag to match the TCP_CLOSING state when a peer has
    shut down and there is nothing left to read.

    This fixes the following bug:
    1) Peer sends SHUTDOWN(RDWR).
    2) Socket enters TCP_CLOSING but SOCK_DONE is not set.
    3) read() returns -ENOTCONN until close() is called, then returns 0.

    Signed-off-by: Stephen Barber <smbarber@chromium.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 72788b8541deaab1fe13a95c958aa3da98618633
Author: Xin Long <lucien.xin@gmail.com>
Date:   Sun Jun 16 17:24:07 2019 +0800

    tipc: purge deferredq list for each grp member in tipc_group_delete

    [ Upstream commit 5cf02612b33f104fe1015b2dfaf1758ad3675588 ]

    Syzbot reported a memleak caused by grp members' deferredq list not
    purged when the grp is be deleted.

    The issue occurs when more(msg_grp_bc_seqno(hdr), m->bc_rcv_nxt) in
    tipc_group_filter_msg() and the skb will stay in deferredq.

    So fix it by calling __skb_queue_purge for each member's deferredq
    in tipc_group_delete() when a tipc sk leaves the grp.

    Fixes: b87a5ea31c93 ("tipc: guarantee group unicast doesn't bypass group broadcast")
    Reported-by: syzbot+78fbe679c8ca8d264a8d@syzkaller.appspotmail.com
    Signed-off-by: Xin Long <lucien.xin@gmail.com>
    Acked-by: Ying Xue <ying.xue@windriver.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit aecf98cd498f79a31fc91e5f239f9151ba524cac
Author: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
Date:   Tue Jun 11 17:38:37 2019 +0200

    sunhv: Fix device naming inconsistency between sunhv_console and sunhv_reg

    [ Upstream commit 07a6d63eb1b54b5fb38092780fe618dfe1d96e23 ]

    In d5a2aa24, the name in struct console sunhv_console was changed from "ttyS"
    to "ttyHV" while the name in struct uart_ops sunhv_pops remained unchanged.

    This results in the hypervisor console device to be listed as "ttyHV0" under
    /proc/consoles while the device node is still named "ttyS0":

    root@osaka:~# cat /proc/consoles
    ttyHV0               -W- (EC p  )    4:64
    tty0                 -WU (E     )    4:1
    root@osaka:~# readlink /sys/dev/char/4:64
    ../../devices/root/f02836f0/f0285690/tty/ttyS0
    root@osaka:~#

    This means that any userland code which tries to determine the name of the
    device file of the hypervisor console device can not rely on the information
    provided by /proc/consoles. In particular, booting current versions of debian-
    installer inside a SPARC LDOM will fail with the installer unable to determine
    the console device.

    After renaming the device in struct uart_ops sunhv_pops to "ttyHV" as well,
    the inconsistency is fixed and it is possible again to determine the name
    of the device file of the hypervisor console device by reading the contents
    of /proc/console:

    root@osaka:~# cat /proc/consoles
    ttyHV0               -W- (EC p  )    4:64
    tty0                 -WU (E     )    4:1
    root@osaka:~# readlink /sys/dev/char/4:64
    ../../devices/root/f02836f0/f0285690/tty/ttyHV0
    root@osaka:~#

    With this change, debian-installer works correctly when installing inside
    a SPARC LDOM.

    Signed-off-by: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9cb8eddfe7d65f6573dc76b64b9edc04aa042db7
Author: Neil Horman <nhorman@tuxdriver.com>
Date:   Thu Jun 13 06:35:59 2019 -0400

    sctp: Free cookie before we memdup a new one

    [ Upstream commit ce950f1050cece5e406a5cde723c69bba60e1b26 ]

    Based on comments from Xin, even after fixes for our recent syzbot
    report of cookie memory leaks, its possible to get a resend of an INIT
    chunk which would lead to us leaking cookie memory.

    To ensure that we don't leak cookie memory, free any previously
    allocated cookie first.

    Change notes
    v1->v2
    update subsystem tag in subject (davem)
    repeat kfree check for peer_random and peer_hmacs (xin)
```

```
    v2->v3
    net->sctp
    also free peer_chunks

    v3->v4
    fix subject tags

    v4->v5
    remove cut line

    Signed-off-by: Neil Horman <nhorman@tuxdriver.com>
    Reported-by: syzbot+f7e9153b037eac9b1df8@syzkaller.appspotmail.com
    CC: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    CC: Xin Long <lucien.xin@gmail.com>
    CC: "David S. Miller" <davem@davemloft.net>
    CC: netdev@vger.kernel.org
    Acked-by: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4c85012ba97ae0ebb7c33922a38b64f1b98938cd
Author: Young Xiao <92siuyang@gmail.com>
Date:   Fri Jun 14 15:13:02 2019 +0800

    nfc: Ensure presence of required attributes in the deactivate_target handler

    [ Upstream commit 385097a3675749cbc9e97c085c0e5dfe4269ca51 ]

    Check that the NFC_ATTR_TARGET_INDEX attributes (in addition to
    NFC_ATTR_DEVICE_INDEX) are provided by the netlink client prior to
    accessing them. This prevents potential unhandled NULL pointer dereference
    exceptions which can be triggered by malicious user-mode programs,
    if they omit one or both of these attributes.

    Signed-off-by: Young Xiao <92siuyang@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 20d3d270e5c2c2e159d6b9e3eeb992726fe8cea6
Author: John Fastabend <john.fastabend@gmail.com>
Date:   Wed Jun 12 17:23:57 2019 +0000

    net: tls, correctly account for copied bytes with multiple sk_msgs

    [ Upstream commit 648ee6cea7dde4a5cdf817e5d964fd60b22006a4 ]

    tls_sw_do_sendpage needs to return the total number of bytes sent
    regardless of how many sk_msgs are allocated. Unfortunately, copied
    (the value we return up the stack) is zero'd before each new sk_msg
    is allocated so we only return the copied size of the last sk_msg used.

    The caller (splice, etc.) of sendpage will then believe only part
    of its data was sent and send the missing chunks again. However,
    because the data actually was sent the receiver will get multiple
    copies of the same data.

    To reproduce this do multiple sendfile calls with a length close to
    the max record size. This will in turn call splice/sendpage, sendpage
    may use multiple sk_msg in this case and then returns the incorrect
    number of bytes. This will cause splice to resend creating duplicate
    data on the receiver. Andre created a C program that can easily
    generate this case so we will push a similar selftest for this to
    bpf-next shortly.
```

```
    The fix is to _not_ zero the copied field so that the total sent
    bytes is returned.

    Reported-by: Steinar H. Gunderson <steinar+kernel@gunderson.no>
    Reported-by: Andre Tomt <andre@tomt.net>
    Tested-by: Andre Tomt <andre@tomt.net>
    Fixes: d829e9c4112b ("tls: convert to generic sk_msg interface")
    Signed-off-by: John Fastabend <john.fastabend@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 39ea88e3c0feef68dc416fa70e43fcde27a2437c
Author: Taehee Yoo <ap420073@gmail.com>
Date:   Sun Jun 9 23:26:21 2019 +0900

    net: openvswitch: do not free vport if register_netdevice() is failed.

    [ Upstream commit 309b66970ee2abf721ecd0876a48940fa0b99a35 ]

    In order to create an internal vport, internal_dev_create() is used and
    that calls register_netdevice() internally.
    If register_netdevice() fails, it calls dev->priv_destructor() to free
    private data of netdev. actually, a private data of this is a vport.

    Hence internal_dev_create() should not free and use a vport after failure
    of register_netdevice().

    Test command
        ovs-dpctl add-dp bonding_masters

    Splat looks like:
    [ 1035.667767] kasan: GPF could be caused by NULL-ptr deref or user memory access
    [ 1035.675958] general protection fault: 0000 [#1] SMP DEBUG_PAGEALLOC KASAN PTI
    [ 1035.676916] CPU: 1 PID: 1028 Comm: ovs-vswitchd Tainted: G    B             5.2.0-
rc3+ #240
    [ 1035.676916] RIP: 0010:internal_dev_create+0x2e5/0x4e0 [openvswitch]
    [ 1035.676916] Code: 48 c1 ea 03 80 3c 02 00 0f 85 9f 01 00 00 4c 8b 23 48 b8 00 00 00
00 00 fc ff df 49 8d bc 24 60 05 00 00 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 86 01 00 00
49 8b bc 24 60 05 00 00 e8 e4 68 f4
    [ 1035.713720] RSP: 0018:ffff88810dcb7578 EFLAGS: 00010206
    [ 1035.713720] RAX: dfffc0000000000 RBX: ffff88810d13fe08 RCX: ffffffff84297704
    [ 1035.713720] RDX: 00000000000000ac RSI: 0000000000000000 RDI: 0000000000000560
    [ 1035.713720] RBP: 00000000ffffffef R08: fffffbfff0d3b881 R09: fffffbfff0d3b881
    [ 1035.713720] R10: 0000000000000001 R11: fffffbfff0d3b880 R12: 0000000000000000
    [ 1035.768776] R13: 0000607ee460b900 R14: ffff88810dcb7690 R15: ffff88810dcb7698
    [ 1035.777709] FS:  00007f02095fc980(0000) GS:ffff88811b400000(0000)
knlGS:0000000000000000
    [ 1035.777709] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [ 1035.777709] CR2: 00007ffdf01d2f28 CR3: 0000000108258000 CR4: 00000000001006e0
    [ 1035.777709] Call Trace:
    [ 1035.777709]  ovs_vport_add+0x267/0x4f0 [openvswitch]
    [ 1035.777709]  new_vport+0x15/0x1e0 [openvswitch]
    [ 1035.777709]  ovs_vport_cmd_new+0x567/0xd10 [openvswitch]
    [ 1035.777709]  ? ovs_dp_cmd_dump+0x490/0x490 [openvswitch]
    [ 1035.777709]  ? __kmalloc+0x131/0x2e0
    [ 1035.777709]  ? genl_family_rcv_msg+0xa54/0x1030
    [ 1035.777709]  genl_family_rcv_msg+0x63a/0x1030
    [ 1035.777709]  ? genl_unregister_family+0x630/0x630
    [ 1035.841681]  ? debug_show_all_locks+0x2d0/0x2d0
    [ ... ]

    Fixes: cf124db566e6 ("net: Fix inconsistent teardown and release of private netdev
state.")
    Signed-off-by: Taehee Yoo <ap420073@gmail.com>
    Reviewed-by: Greg Rose <gvrose8192@gmail.com>
```

```
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 22da034c885ff9ea6862780a9a1687630deb08f6
Author: Linus Walleij <linus.walleij@linaro.org>
Date:   Fri Jun 14 00:25:20 2019 +0200

    net: dsa: rtl8366: Fix up VLAN filtering

    [ Upstream commit 760c80b70bed2cd01630e8595d1bbde910339f31 ]

    We get this regression when using RTL8366RB as part of a bridge
    with OpenWrt:

    WARNING: CPU: 0 PID: 1347 at net/switchdev/switchdev.c:291
             switchdev_port_attr_set_now+0x80/0xa4
    lan0: Commit of attribute (id=7) failed.
    (...)
    realtek-smi switch lan0: failed to initialize vlan filtering on this port

    This is because it is trying to disable VLAN filtering
    on VLAN0, as we have forgot to add 1 to the port number
    to get the right VLAN in rtl8366_vlan_filtering(): when
    we initialize the VLAN we associate VLAN1 with port 0,
    VLAN2 with port 1 etc, so we need to add 1 to the port
    offset.

    Fixes: d8652956cf37 ("net: dsa: realtek-smi: Add Realtek SMI driver")
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d97ce9c779ac98da71e2639cb13ea04e46c94912
Author: Eric Dumazet <edumazet@google.com>
Date:   Sat Jun 15 16:28:48 2019 -0700

    neigh: fix use-after-free read in pneigh_get_next

    [ Upstream commit f3e92cb8e2eb8c27d109e6fd73d3a69a8c09e288 ]

    Nine years ago, I added RCU handling to neighbours, not pneighbours.
    (pneigh are not commonly used)

    Unfortunately I missed that /proc dump operations would use a
    common entry and exit point : neigh_seq_start() and neigh_seq_stop()

    We need to read_lock(tbl->lock) or risk use-after-free while
    iterating the pneigh structures.

    We might later convert pneigh to RCU and revert this patch.

    sysbot reported :

    BUG: KASAN: use-after-free in pneigh_get_next.isra.0+0x24b/0x280
net/core/neighbour.c:3158
    Read of size 8 at addr ffff888097f2a700 by task syz-executor.0/9825

    CPU: 1 PID: 9825 Comm: syz-executor.0 Not tainted 5.2.0-rc4+ #32
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
01/01/2011
    Call Trace:
     __dump_stack lib/dump_stack.c:77 [inline]
     dump_stack+0x172/0x1f0 lib/dump_stack.c:113
     print_address_description.cold+0x7c/0x20d mm/kasan/report.c:188
     __kasan_report.cold+0x1b/0x40 mm/kasan/report.c:317
```

```
      kasan_report+0x12/0x20 mm/kasan/common.c:614
      __asan_report_load8_noabort+0x14/0x20 mm/kasan/generic_report.c:132
      pneigh_get_next.isra.0+0x24b/0x280 net/core/neighbour.c:3158
      neigh_seq_next+0xdb/0x210 net/core/neighbour.c:3240
      seq_read+0x9cf/0x1110 fs/seq_file.c:258
      proc_reg_read+0x1fc/0x2c0 fs/proc/inode.c:221
      do_loop_readv_writev fs/read_write.c:714 [inline]
      do_loop_readv_writev fs/read_write.c:701 [inline]
      do_iter_read+0x4a4/0x660 fs/read_write.c:935
      vfs_readv+0xf0/0x160 fs/read_write.c:997
      kernel_readv fs/splice.c:359 [inline]
      default_file_splice_read+0x475/0x890 fs/splice.c:414
      do_splice_to+0x127/0x180 fs/splice.c:877
      splice_direct_to_actor+0x2d2/0x970 fs/splice.c:954
      do_splice_direct+0x1da/0x2a0 fs/splice.c:1063
      do_sendfile+0x597/0xd00 fs/read_write.c:1464
      __do_sys_sendfile64 fs/read_write.c:1525 [inline]
      __se_sys_sendfile64 fs/read_write.c:1511 [inline]
      __x64_sys_sendfile64+0x1dd/0x220 fs/read_write.c:1511
      do_syscall_64+0xfd/0x680 arch/x86/entry/common.c:301
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
    RIP: 0033:0x4592c9
    Code: fd b7 fb ff c3 66 2e 0f 1f 84 00 00 00 00 00 66 90 48 89 f8 48 89 f7 48 89 d6 48
 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 0f 83 cb b7 fb ff c3 66 2e
 0f 1f 84 00 00 00 00
    RSP: 002b:00007f4aab51dc78 EFLAGS: 00000246 ORIG_RAX: 0000000000000028
    RAX: ffffffffffffffda RBX: 0000000000000004 RCX: 00000000004592c9
    RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000005
    RBP: 000000000075bf20 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000080000000 R11: 0000000000000246 R12: 00007f4aab51e6d4
    R13: 00000000004c689d R14: 00000000004db828 R15: 00000000ffffffff

    Allocated by task 9827:
      save_stack+0x23/0x90 mm/kasan/common.c:71
      set_track mm/kasan/common.c:79 [inline]
      __kasan_kmalloc mm/kasan/common.c:489 [inline]
      __kasan_kmalloc.constprop.0+0xcf/0xe0 mm/kasan/common.c:462
      kasan_kmalloc+0x9/0x10 mm/kasan/common.c:503
      __do_kmalloc mm/slab.c:3660 [inline]
      __kmalloc+0x15c/0x740 mm/slab.c:3669
      kmalloc include/linux/slab.h:552 [inline]
      pneigh_lookup+0x19c/0x4a0 net/core/neighbour.c:731
      arp_req_set_public net/ipv4/arp.c:1010 [inline]
      arp_req_set+0x613/0x720 net/ipv4/arp.c:1026
      arp_ioctl+0x652/0x7f0 net/ipv4/arp.c:1226
      inet_ioctl+0x2a0/0x340 net/ipv4/af_inet.c:926
      sock_do_ioctl+0xd8/0x2f0 net/socket.c:1043
      sock_ioctl+0x3ed/0x780 net/socket.c:1194
      vfs_ioctl fs/ioctl.c:46 [inline]
      file_ioctl fs/ioctl.c:509 [inline]
      do_vfs_ioctl+0xd5f/0x1380 fs/ioctl.c:696
      ksys_ioctl+0xab/0xd0 fs/ioctl.c:713
      __do_sys_ioctl fs/ioctl.c:720 [inline]
      __se_sys_ioctl fs/ioctl.c:718 [inline]
      __x64_sys_ioctl+0x73/0xb0 fs/ioctl.c:718
      do_syscall_64+0xfd/0x680 arch/x86/entry/common.c:301
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

    Freed by task 9824:
      save_stack+0x23/0x90 mm/kasan/common.c:71
      set_track mm/kasan/common.c:79 [inline]
      __kasan_slab_free+0x102/0x150 mm/kasan/common.c:451
      kasan_slab_free+0xe/0x10 mm/kasan/common.c:459
      __cache_free mm/slab.c:3432 [inline]
      kfree+0xcf/0x220 mm/slab.c:3755
```

```
      pneigh_ifdown_and_unlock net/core/neighbour.c:812 [inline]
      __neigh_ifdown+0x236/0x2f0 net/core/neighbour.c:356
      neigh_ifdown+0x20/0x30 net/core/neighbour.c:372
      arp_ifdown+0x1d/0x21 net/ipv4/arp.c:1274
      inetdev_destroy net/ipv4/devinet.c:319 [inline]
      inetdev_event+0xa14/0x11f0 net/ipv4/devinet.c:1544
      notifier_call_chain+0xc2/0x230 kernel/notifier.c:95
      __raw_notifier_call_chain kernel/notifier.c:396 [inline]
      raw_notifier_call_chain+0x2e/0x40 kernel/notifier.c:403
      call_netdevice_notifiers_info+0x3f/0x90 net/core/dev.c:1749
      call_netdevice_notifiers_extack net/core/dev.c:1761 [inline]
      call_netdevice_notifiers net/core/dev.c:1775 [inline]
      rollback_registered_many+0x9b9/0xfc0 net/core/dev.c:8178
      rollback_registered+0x109/0x1d0 net/core/dev.c:8220
      unregister_netdevice_queue net/core/dev.c:9267 [inline]
      unregister_netdevice_queue+0x1ee/0x2c0 net/core/dev.c:9260
      unregister_netdevice include/linux/netdevice.h:2631 [inline]
      __tun_detach+0xd8a/0x1040 drivers/net/tun.c:724
      tun_detach drivers/net/tun.c:741 [inline]
      tun_chr_close+0xe0/0x180 drivers/net/tun.c:3451
      __fput+0x2ff/0x890 fs/file_table.c:280
      ____fput+0x16/0x20 fs/file_table.c:313
      task_work_run+0x145/0x1c0 kernel/task_work.c:113
      tracehook_notify_resume include/linux/tracehook.h:185 [inline]
      exit_to_usermode_loop+0x273/0x2c0 arch/x86/entry/common.c:168
      prepare_exit_to_usermode arch/x86/entry/common.c:199 [inline]
      syscall_return_slowpath arch/x86/entry/common.c:279 [inline]
      do_syscall_64+0x58e/0x680 arch/x86/entry/common.c:304
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

    The buggy address belongs to the object at ffff888097f2a700
     which belongs to the cache kmalloc-64 of size 64
    The buggy address is located 0 bytes inside of
     64-byte region [ffff888097f2a700, ffff888097f2a740)
    The buggy address belongs to the page:
    page:ffffea00025fca80 refcount:1 mapcount:0 mapping:ffff8880aa400340 index:0x0
    flags: 0x1fffc0000000200(slab)
    raw: 01fffc0000000200 ffffea000250d548 ffffea00025726c8 ffff8880aa400340
    raw: 0000000000000000 ffff888097f2a000 0000000100000020 0000000000000000
    page dumped because: kasan: bad access detected

    Memory state around the buggy address:
     ffff888097f2a600: 00 00 00 00 00 00 00 00 fc fc fc fc fc fc fc fc
     ffff888097f2a680: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc
    >ffff888097f2a700: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc
                      ^
     ffff888097f2a780: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc
     ffff888097f2a800: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc

    Fixes: 767e97e1e0db ("neigh: RCU conversion of struct neighbour")
    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: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit da930ec93201692aa9ff3b072985978fc6047006
Author: Jeremy Sowden <jeremy@azazel.net>
Date:   Sun Jun 16 16:54:37 2019 +0100

    lapb: fixed leak of control-blocks.

    [ Upstream commit 6be8e297f9bcea666ea85ac7a6cd9d52d6deaf92 ]

    lapb_register calls lapb_create_cb, which initializes the control-
    block's ref-count to one, and __lapb_insert_cb, which increments it when
```

```
    adding the new block to the list of blocks.

    lapb_unregister calls __lapb_remove_cb, which decrements the ref-count
    when removing control-block from the list of blocks, and calls lapb_put
    itself to decrement the ref-count before returning.

    However, lapb_unregister also calls __lapb_devtostruct to look up the
    right control-block for the given net_device, and __lapb_devtostruct
    also bumps the ref-count, which means that when lapb_unregister returns
    the ref-count is still 1 and the control-block is leaked.

    Call lapb_put after __lapb_devtostruct to fix leak.

    Reported-by: syzbot+afb980676c836b4a0afa@syzkaller.appspotmail.com
    Signed-off-by: Jeremy Sowden <jeremy@azazel.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a7f938c9af445559764a7714984fa43c5409b112
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Jun 6 14:32:34 2019 -0700

    ipv6: flowlabel: fl6_sock_lookup() must use atomic_inc_not_zero

    [ Upstream commit 65a3c497c0e965a552008db8bc2653f62bc925a1 ]

    Before taking a refcount, make sure the object is not already
    scheduled for deletion.

    Same fix is needed in ipv6_flowlabel_opt()

    Fixes: 18367681a10b ("ipv6 flowlabel: Convert np->ipv6_fl_list to RCU.")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Willem de Bruijn <willemb@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3514df79eb234b2dfd17755086ebff794cbb7aba
Author: Haiyang Zhang <haiyangz@microsoft.com>
Date:   Thu Jun 13 21:06:53 2019 +0000

    hv_netvsc: Set probe mode to sync

    [ Upstream commit 9a33629ba6b26caebd73e3c581ba1e6068c696a7 ]

    For better consistency of synthetic NIC names, we set the probe mode to
    PROBE_FORCE_SYNCHRONOUS. So the names can be aligned with the vmbus
    channel offer sequence.

    Fixes: af0a5646cb8d ("use the new async probing feature for the hyperv drivers")
    Signed-off-by: Haiyang Zhang <haiyangz@microsoft.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1a976c6c586e4ba4ae1affd0aadba2cd56b5cefb
Author: Ivan Vecera <ivecera@redhat.com>
Date:   Fri Jun 14 17:48:36 2019 +0200

    be2net: Fix number of Rx queues used for flow hashing

    [ Upstream commit 718f4a2537089ea41903bf357071306163bc7c04 ]

    Number of Rx queues used for flow hashing returned by the driver is
    incorrect and this bug prevents user to use the last Rx queue in
    indirection table.
```

```
    Let's say we have a NIC with 6 combined queues:

    [root@sm-03 ~]# ethtool -l enp4s0f0
    Channel parameters for enp4s0f0:
    Pre-set maximums:
    RX:             5
    TX:             5
    Other:          0
    Combined:       6
    Current hardware settings:
    RX:             0
    TX:             0
    Other:          0
    Combined:       6

    Default indirection table maps all (6) queues equally but the driver
    reports only 5 rings available.

    [root@sm-03 ~]# ethtool -x enp4s0f0
    RX flow hash indirection table for enp4s0f0 with 5 RX ring(s):
        0:      0     1     2     3     4     5     0     1
        8:      2     3     4     5     0     1     2     3
       16:      4     5     0     1     2     3     4     5
       24:      0     1     2     3     4     5     0     1
    ...

    Now change indirection table somehow:

    [root@sm-03 ~]# ethtool -X enp4s0f0 weight 1 1
    [root@sm-03 ~]# ethtool -x enp4s0f0
    RX flow hash indirection table for enp4s0f0 with 6 RX ring(s):
        0:      0     0     0     0     0     0     0     0
    ...
       64:      1     1     1     1     1     1     1     1
    ...

    Now it is not possible to change mapping back to equal (default) state:

    [root@sm-03 ~]# ethtool -X enp4s0f0 equal 6
    Cannot set RX flow hash configuration: Invalid argument

    Fixes: 594ad54a2c3b ("be2net: Add support for setting and getting rx flow hash options")
    Reported-by: Tianhao <tizhao@redhat.com>
    Signed-off-by: Ivan Vecera <ivecera@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5f9c0c152eacfa182c4625d6f4e28554ac9e6e06
Author: Eric Dumazet <edumazet@google.com>
Date:   Sat Jun 15 16:40:52 2019 -0700

    ax25: fix inconsistent lock state in ax25_destroy_timer

    [ Upstream commit d4d5d8e83c9616aeef28a2869cea49cc3fb35526 ]

    Before thread in process context uses bh_lock_sock()
    we must disable bh.

    sysbot reported :

    WARNING: inconsistent lock state
    5.2.0-rc3+ #32 Not tainted

    inconsistent {SOFTIRQ-ON-W} -> {IN-SOFTIRQ-W} usage.
```

```
     blkid/26581 [HC0[0]:SC1[1]:HE1:SE0] takes:
     00000000e0da85ee (slock-AF_AX25){+.?.}, at: spin_lock include/linux/spinlock.h:338
  [inline]
     00000000e0da85ee (slock-AF_AX25){+.?.}, at: ax25_destroy_timer+0x53/0xc0
  net/ax25/af_ax25.c:275
     {SOFTIRQ-ON-W} state was registered at:
       lock_acquire+0x16f/0x3f0 kernel/locking/lockdep.c:4303
       __raw_spin_lock include/linux/spinlock_api_smp.h:142 [inline]
       _raw_spin_lock+0x2f/0x40 kernel/locking/spinlock.c:151
       spin_lock include/linux/spinlock.h:338 [inline]
       ax25_rt_autobind+0x3ca/0x720 net/ax25/ax25_route.c:429
       ax25_connect.cold+0x30/0xa4 net/ax25/af_ax25.c:1221
       __sys_connect+0x264/0x330 net/socket.c:1834
       __do_sys_connect net/socket.c:1845 [inline]
       __se_sys_connect net/socket.c:1842 [inline]
       __x64_sys_connect+0x73/0xb0 net/socket.c:1842
       do_syscall_64+0xfd/0x680 arch/x86/entry/common.c:301
       entry_SYSCALL_64_after_hwframe+0x49/0xbe
     irq event stamp: 2272
     hardirqs last  enabled at (2272): [<ffffffff810065f3>] trace_hardirqs_on_thunk+0x1a/0x1c
     hardirqs last disabled at (2271): [<ffffffff8100660f>]
  trace_hardirqs_off_thunk+0x1a/0x1c
     softirqs last  enabled at (1522): [<ffffffff87400654>] __do_softirq+0x654/0x94c
  kernel/softirq.c:320
     softirqs last disabled at (2267): [<ffffffff81449010>] invoke_softirq
  kernel/softirq.c:374 [inline]
     softirqs last disabled at (2267): [<ffffffff81449010>] irq_exit+0x180/0x1d0
  kernel/softirq.c:414

     other info that might help us debug this:
      Possible unsafe locking scenario:

            CPU0
            ----
       lock(slock-AF_AX25);
       <Interrupt>
         lock(slock-AF_AX25);

      *** DEADLOCK ***

     1 lock held by blkid/26581:
      #0: 0000000010fd154d ((&ax25->dtimer)){+.-.}, at: lockdep_copy_map
  include/linux/lockdep.h:175 [inline]
      #0: 0000000010fd154d ((&ax25->dtimer)){+.-.}, at: call_timer_fn+0xe0/0x720
  kernel/time/timer.c:1312

     stack backtrace:
     CPU: 1 PID: 26581 Comm: blkid Not tainted 5.2.0-rc3+ #32
     Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
  01/01/2011
     Call Trace:
      <IRQ>
      __dump_stack lib/dump_stack.c:77 [inline]
      dump_stack+0x172/0x1f0 lib/dump_stack.c:113
      print_usage_bug.cold+0x393/0x4a2 kernel/locking/lockdep.c:2935
      valid_state kernel/locking/lockdep.c:2948 [inline]
      mark_lock_irq kernel/locking/lockdep.c:3138 [inline]
      mark_lock+0xd46/0x1370 kernel/locking/lockdep.c:3513
      mark_irqflags kernel/locking/lockdep.c:3391 [inline]
      __lock_acquire+0x159f/0x5490 kernel/locking/lockdep.c:3745
      lock_acquire+0x16f/0x3f0 kernel/locking/lockdep.c:4303
      __raw_spin_lock include/linux/spinlock_api_smp.h:142 [inline]
      _raw_spin_lock+0x2f/0x40 kernel/locking/spinlock.c:151
      spin_lock include/linux/spinlock.h:338 [inline]
      ax25_destroy_timer+0x53/0xc0 net/ax25/af_ax25.c:275
```

```
       call_timer_fn+0x193/0x720 kernel/time/timer.c:1322
       expire_timers kernel/time/timer.c:1366 [inline]
       __run_timers kernel/time/timer.c:1685 [inline]
       __run_timers kernel/time/timer.c:1653 [inline]
       run_timer_softirq+0x66f/0x1740 kernel/time/timer.c:1698
       __do_softirq+0x25c/0x94c kernel/softirq.c:293
       invoke_softirq kernel/softirq.c:374 [inline]
       irq_exit+0x180/0x1d0 kernel/softirq.c:414
       exiting_irq arch/x86/include/asm/apic.h:536 [inline]
       smp_apic_timer_interrupt+0x13b/0x550 arch/x86/kernel/apic/apic.c:1068
       apic_timer_interrupt+0xf/0x20 arch/x86/entry/entry_64.S:806
       </IRQ>
     RIP: 0033:0x7f858d5c3232
     Code: 8b 61 08 48 8b 84 24 d8 00 00 00 4c 89 44 24 28 48 8b ac 24 d0 00 00 00 4c 8b b4
  24 e8 00 00 00 48 89 7c 24 68 48 89 4c 24 78 <48> 89 44 24 58 8b 84 24 e0 00 00 00 89 84 24
  84 00 00 00 8b 84 24
     RSP: 002b:00007ffcaf0cf5c0 EFLAGS: 00000206 ORIG_RAX: ffffffffffffff13
     RAX: 00007f858d7d27a8 RBX: 00007f858d7d8820 RCX: 00007f858d3940d8
     RDX: 00007ffcaf0cf798 RSI: 00000000f5e616f3 RDI: 00007f858d394fee
     RBP: 0000000000000000 R08: 00007ffcaf0cf780 R09: 00007f858d7db480
     R10: 0000000000000000 R11: 0000000009691a75 R12: 0000000000000005
     R13: 00000000f5e616f3 R14: 0000000000000000 R15: 00007ffcaf0cf798

     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: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit f51ec54da907ffca5de7cd645f28a4d9224d0c93
  Author: Florian Westphal <fw@strlen.de>
  Date:   Mon May 20 13:48:10 2019 +0200

     netfilter: nat: fix udp checksum corruption

     commit 6bac76db1da3cb162c425d58ae421486f8e43955 upstream.

     Due to copy&paste error nf_nat_mangle_udp_packet passes IPPROTO_TCP,
     resulting in incorrect udp checksum when payload had to be mangled.

     Fixes: dac3fe72596f9 ("netfilter: nat: remove csum_recalc hook")
     Reported-by: Marc Haber <mh+netdev@zugschlus.de>
     Tested-by: Marc Haber <mh+netdev@zugschlus.de>
     Signed-off-by: Florian Westphal <fw@strlen.de>
     Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```