```
commit 60b6aa2b71efa7e0bd5393ce292ace4a0cf2e71b
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Jan 29 16:45:34 2020 +0100

    Linux 5.4.16

commit d8a11e0f4e26f033a217a24617c11bcd24b8f65a
Author: Martin Schiller <ms@dev.tdt.de>
Date:   Thu Jan 9 07:31:14 2020 +0100

    net/x25: fix nonblocking connect

    commit e21dba7a4df4d93da237da65a096084b4f2e87b4 upstream.

    This patch fixes 2 issues in x25_connect():

    1. It makes absolutely no sense to reset the neighbour and the
    connection state after a (successful) nonblocking call of x25_connect.
    This prevents any connection from being established, since the response
    (call accept) cannot be processed.

    2. Any further calls to x25_connect() while a call is pending should
    simply return, instead of creating new Call Request (on different
    logical channels).

    This patch should also fix the "KASAN: null-ptr-deref Write in
    x25_connect" and "BUG: unable to handle kernel NULL pointer dereference
    in x25_connect" bugs reported by syzbot.

    Signed-off-by: Martin Schiller <ms@dev.tdt.de>
    Reported-by: syzbot+429c200ffc8772bfe070@syzkaller.appspotmail.com
    Reported-by: syzbot+eec0c87f31a7c3b66f7b@syzkaller.appspotmail.com
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ce75dd3abbc8f1be531ac5de559d59df9098020c
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Tue Jan 21 16:48:03 2020 +0100

    netfilter: nf_tables: autoload modules from the abort path

    commit eb014de4fd418de1a277913cba244e47274fe392 upstream.

    This patch introduces a list of pending module requests. This new module
    list is composed of nft_module_request objects that contain the module
    name and one status field that tells if the module has been already
    loaded (the 'done' field).

    In the first pass, from the preparation phase, the netlink command finds
    that a module is missing on this list. Then, a module request is
    allocated and added to this list and nft_request_module() returns
    -EAGAIN. This triggers the abort path with the autoload parameter set on
    from nfnetlink, request_module() is called and the module request enters
    the 'done' state. Since the mutex is released when loading modules from
    the abort phase, the module list is zapped so this is iteration occurs
    over a local list. Therefore, the request_module() calls happen when
    object lists are in consistent state (after fulling aborting the
    transaction) and the commit list is empty.

    On the second pass, the netlink command will find that it already tried
    to load the module, so it does not request it again and
    nft_request_module() returns 0. Then, there is a look up to find the
    object that the command was missing. If the module was successfully
    loaded, the command proceeds normally since it finds the missing object
```

```
    in place, otherwise -ENOENT is reported to userspace.

    This patch also updates nfnetlink to include the reason to enter the
    abort phase, which is required for this new autoload module rationale.

    Fixes: ec7470b834fe ("netfilter: nf_tables: store transaction list locally while
requesting module")
    Reported-by: syzbot+29125d208b3dae9a7019@syzkaller.appspotmail.com
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 07ac418120aed1d8bb98f36b9bc17460ce37a036
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Tue Jan 21 16:07:00 2020 +0100

    netfilter: nf_tables: add __nft_chain_type_get()

    commit 826035498ec14b77b62a44f0cb6b94d45530db6f upstream.

    This new helper function validates that unknown family and chain type
    coming from userspace do not trigger an out-of-bound array access. Bail
    out in case __nft_chain_type_get() returns NULL from
    nft_chain_parse_hook().

    Fixes: 9370761c56b6 ("netfilter: nf_tables: convert built-in tables/chains to chain
types")
    Reported-by: syzbot+156a04714799b1d480bc@syzkaller.appspotmail.com
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ea52197c9c4b9e1722e4ec0b798e1ab3744dd5a2
Author: Kadlecsik József <kadlec@blackhole.kfki.hu>
Date:   Sun Jan 19 22:06:49 2020 +0100

    netfilter: ipset: use bitmap infrastructure completely

    commit 32c72165dbd0e246e69d16a3ad348a4851afd415 upstream.

    The bitmap allocation did not use full unsigned long sizes
    when calculating the required size and that was triggered by KASAN
    as slab-out-of-bounds read in several places. The patch fixes all
    of them.

    Reported-by: syzbot+fabca5cbf5e54f3fe2de@syzkaller.appspotmail.com
    Reported-by: syzbot+827ced406c9a1d9570ed@syzkaller.appspotmail.com
    Reported-by: syzbot+190d63957b22ef673ea5@syzkaller.appspotmail.com
    Reported-by: syzbot+dfccdb2bdb4a12ad425e@syzkaller.appspotmail.com
    Reported-by: syzbot+df0d0f5895ef1f41a65b@syzkaller.appspotmail.com
    Reported-by: syzbot+b08bd19bb37513357fd4@syzkaller.appspotmail.com
    Reported-by: syzbot+53cdd0ec0bbabd53370a@syzkaller.appspotmail.com
    Signed-off-by: Jozsef Kadlecsik <kadlec@netfilter.org>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d5fd8ac3fcc52dce384c54130a56e13719b2eaeb
Author: Hans Verkuil <hverkuil-cisco@xs4all.nl>
Date:   Sun Nov 10 07:27:04 2019 +0100

    media: v4l2-ioctl.c: zero reserved fields for S/TRY_FMT

    commit ee8951e56c0f960b9621636603a822811cef3158 upstream.

    v4l2_vbi_format, v4l2_sliced_vbi_format and v4l2_sdr_format
    have a reserved array at the end that should be zeroed by drivers
    as per the V4L2 spec. Older drivers often do not do this, so just
```

```
    handle this in the core.

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 40b1747b03684f03827b6323a17e4aa67af1e307
Author: Wen Huang <huangwenabc@gmail.com>
Date:   Thu Nov 28 18:51:04 2019 +0800

    libertas: Fix two buffer overflows at parsing bss descriptor

    commit e5e884b42639c74b5b57dc277909915c0aefc8bb upstream.

    add_ie_rates() copys rates without checking the length
    in bss descriptor from remote AP.when victim connects to
    remote attacker, this may trigger buffer overflow.
    lbs_ibss_join_existing() copys rates without checking the length
    in bss descriptor from remote IBSS node.when victim connects to
    remote attacker, this may trigger buffer overflow.
    Fix them by putting the length check before performing copy.

    This fix addresses CVE-2019-14896 and CVE-2019-14897.
    This also fix build warning of mixed declarations and code.

    Reported-by: kbuild test robot <lkp@intel.com>
    Signed-off-by: Wen Huang <huangwenabc@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 74389183b67101570ec25e80694758b5903fd543
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Prevent tx watchdog timeout

    commit 686f85d71d095f1d26b807e23b0f0bfd22042c45 upstream.

    Section 5.5.3.2 of the datasheet says,

        If FIFO Underrun, Byte Count Mismatch, Excessive Collision, or
        Excessive Deferral (if enabled) errors occur, transmission ceases.

    In this situation, the chip asserts a TXER interrupt rather than TXDN.
    But the handler for the TXDN is the only way that the transmit queue
    gets restarted. Hence, an aborted transmission can result in a watchdog
    timeout.

    This problem can be reproduced on congested link, as that can result in
    excessive transmitter collisions. Another way to reproduce this is with
    a FIFO Underrun, which may be caused by DMA latency.

    In event of a TXER interrupt, prevent a watchdog timeout by restarting
    transmission.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 58c0c4b122b1681c0f3b0b5b568468a2c00cd906
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100
```

```
    net/sonic: Fix CAM initialization

    commit 772f66421d5aa0b9f256056f513bbc38ac132271 upstream.

    Section 4.3.1 of the datasheet says,

        This bit [TXP] must not be set if a Load CAM operation is in
        progress (LCAM is set). The SONIC will lock up if both bits are
        set simultaneously.

    Testing has shown that the driver sometimes attempts to set LCAM
    while TXP is set. Avoid this by waiting for command completion
    before and after giving the LCAM command.

    After issuing the Load CAM command, poll for !SONIC_CR_LCAM rather than
    SONIC_INT_LCD, because the SONIC_CR_TXP bit can't be used until
    !SONIC_CR_LCAM.

    When in reset mode, take the opportunity to reset the CAM Enable
    register.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 079e7e775895b7472de6c4d3be62b6e80997d354
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Fix command register usage

    commit 27e0c31c5f27c1d1a1d9d135c123069f60dcf97b upstream.

    There are several issues relating to command register usage during
    chip initialization.

    Firstly, the SONIC sometimes comes out of software reset with the
    Start Timer bit set. This gets logged as,

        macsonic macsonic eth0: sonic_init: status=24, i=101

    Avoid this by giving the Stop Timer command earlier than later.

    Secondly, the loop that waits for the Read RRA command to complete has
    the break condition inverted. That's why the for loop iterates until
    its termination condition. Call the helper for this instead.

    Finally, give the Receiver Enable command after clearing interrupts,
    not before, to avoid the possibility of losing an interrupt.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4d391c40c29521907d484700836c97d2264982d5
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Quiesce SONIC before re-initializing descriptor memory

    commit 3f4b7e6a2be982fd8820a2b54d46dd9c351db899 upstream.
```

```
    Make sure the SONIC's DMA engine is idle before altering the transmit
    and receive descriptors. Add a helper for this as it will be needed
    again.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9f638adb9f1a6373bd8369c70254c9d37ddd7885
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Fix receive buffer replenishment

    commit 89ba879e95582d3bba55081e45b5409e883312ca upstream.

    As soon as the driver is finished with a receive buffer it allocs a new
    one and overwrites the corresponding RRA entry with a new buffer pointer.

    Problem is, the buffer pointer is split across two word-sized registers.
    It can't be updated in one atomic store. So this operation races with the
    chip while it stores received packets and advances its RRP register.
    This could result in memory corruption by a DMA write.

    Avoid this problem by adding buffers only at the location given by the
    RWP register, in accordance with the National Semiconductor datasheet.

    Re-factor this code into separate functions to calculate a RRA pointer
    and to update the RWP.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a0c502d4535bac2e01c6f6fd99bb837ea09d770
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Improve receive descriptor status flag check

    commit 94b166349503957079ef5e7d6f667f157aea014a upstream.

    After sonic_tx_timeout() calls sonic_init(), it can happen that
    sonic_rx() will subsequently encounter a receive descriptor with no
    flags set. Remove the comment that says that this can't happen.

    When giving a receive descriptor to the SONIC, clear the descriptor
    status field. That way, any rx descriptor with flags set can only be
    a newly received packet.

    Don't process a descriptor without the LPKT bit set. The buffer is
    still in use by the SONIC.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a21378463475077b0176f3949964aa72e33765b
```

```
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Avoid needless receive descriptor EOL flag updates

    commit eaabfd19b2c787bbe88dc32424b9a43d67293422 upstream.

    The while loop in sonic_rx() traverses the rx descriptor ring. It stops
    when it reaches a descriptor that the SONIC has not used. Each iteration
    advances the EOL flag so the SONIC can keep using more descriptors.
    Therefore, the while loop has no definite termination condition.

    The algorithm described in the National Semiconductor literature is quite
    different. It consumes descriptors up to the one with its EOL flag set
    (which will also have its "in use" flag set). All freed descriptors are
    then returned to the ring at once, by adjusting the EOL flags (and link
    pointers).

    Adopt the algorithm from datasheet as it's simpler, terminates quickly
    and avoids a lot of pointless descriptor EOL flag changes.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f6ecc13bea2b391aab55105d768c584b8757b0cd
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Fix receive buffer handling

    commit 9e311820f67e740f4fb8dcb82b4c4b5b05bdd1a5 upstream.

    The SONIC can sometimes advance its rx buffer pointer (RRP register)
    without advancing its rx descriptor pointer (CRDA register). As a result
    the index of the current rx descriptor may not equal that of the current
    rx buffer. The driver mistakenly assumes that they are always equal.
    This assumption leads to incorrect packet lengths and possible packet
    duplication. Avoid this by calling a new function to locate the buffer
    corresponding to a given descriptor.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e7092fbffc349c1330e08c42eaf948acb8503cb7
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Fix interface error stats collection

    commit 427db97df1ee721c20bdc9a66db8a9e1da719855 upstream.

    The tx_aborted_errors statistic should count packets flagged with EXD,
    EXC, FU, or BCM bits because those bits denote an aborted transmission.
    That corresponds to the bitmask 0x0446, not 0x0642. Use macros for these
    constants to avoid mistakes. Better to leave out FIFO Underruns (FU) as
    there's a separate counter for that purpose.

    Don't lump all these errors in with the general tx_errors counter as
    that's used for tx timeout events.
```

```
    On the rx side, don't count RDE and RBAE interrupts as dropped packets.
    These interrupts don't indicate a lost packet, just a lack of resources.
    When a lack of resources results in a lost packet, this gets reported
    in the rx_missed_errors counter (along with RFO events).

    Don't double-count rx_frame_errors and rx_crc_errors.

    Don't use the general rx_errors counter for events that already have
    special counters.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c5a85f05e4626de6441cd5afaa46d07c9e287c7c
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Use MMIO accessors

    commit e3885f576196ddfc670b3d53e745de96ffcb49ab upstream.

    The driver accesses descriptor memory which is simultaneously accessed by
    the chip, so the compiler must not be allowed to re-order CPU accesses.
    sonic_buf_get() used 'volatile' to prevent that. sonic_buf_put() should
    have done so too but was overlooked.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c463d5a17032ff66a221e705ece62bed80a47978
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Clear interrupt flags immediately

    commit 5fedabf5a70be26b19d7520f09f12a62274317c6 upstream.

    The chip can change a packet's descriptor status flags at any time.
    However, an active interrupt flag gets cleared rather late. This
    allows a race condition that could theoretically lose an interrupt.
    Fix this by clearing asserted interrupt flags immediately.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 091c1715f9dde8eaf3bf513d8fe73fe032365510
Author: Finn Thain <fthain@telegraphics.com.au>
Date:   Thu Jan 23 09:07:26 2020 +1100

    net/sonic: Add mutual exclusion for accessing shared state

    commit 865ad2f2201dc18685ba2686f13217f8b3a9c52c upstream.

    The netif_stop_queue() call in sonic_send_packet() races with the
    netif_wake_queue() call in sonic_interrupt(). This causes issues
```

```
    like "NETDEV WATCHDOG: eth0 (macsonic): transmit queue 0 timed out".
    Fix this by disabling interrupts when accessing tx_skb[] and next_tx.
    Update a comment to clarify the synchronization properties.

    Fixes: efcce839360f ("[PATCH] macsonic/jazzsonic network drivers update")
    Tested-by: Stan Johnson <userm57@yahoo.com>
    Signed-off-by: Finn Thain <fthain@telegraphics.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ab94448beeab30c30c4399b244e65db838be4d64
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Thu Jan 23 10:05:05 2020 -0800

    readdir: be more conservative with directory entry names

    commit 2c6b7bcd747201441923a0d3062577a8d1fbd8f8 upstream.

    Commit 8a23eb804ca4 ("Make filldir[64]() verify the directory entry
    filename is valid") added some minimal validity checks on the directory
    entries passed to filldir[64]().  But they really were pretty minimal.

    This fleshes out at least the name length check: we used to disallow
    zero-length names, but really, negative lengths or oevr-long names
    aren't ok either.  Both could happen if there is some filesystem
    corruption going on.

    Now, most filesystems tend to use just an "unsigned char" or similar for
    the length of a directory entry name, so even with a corrupt filesystem
    you should never see anything odd like that.  But since we then use the
    name length to create the directory entry record length, let's make sure
    it actually is half-way sensible.

    Note how POSIX states that the size of a path component is limited by
    NAME_MAX, but we actually use PATH_MAX for the check here.  That's
    because while NAME_MAX is generally the correct maximum name length
    (it's 255, for the same old "name length is usually just a byte on
    disk"), there's nothing in the VFS layer that really cares.

    So the real limitation at a VFS layer is the total pathname length you
    can pass as a filename: PATH_MAX.

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

commit 454759886d0b463213fad0f1c733469e2c501ab9
Author: Al Viro <viro@zeniv.linux.org.uk>
Date:   Sun Jan 26 09:29:34 2020 -0500

    do_last(): fetch directory ->i_mode and ->i_uid before it's too late

    commit d0cb50185ae942b03c4327be322055d622dc79f6 upstream.

    may_create_in_sticky() call is done when we already have dropped the
    reference to dir.

    Fixes: 30aba6656f61e (namei: allow restricted O_CREAT of FIFOs and regular files)
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5d001c4f74e0417c25e1da6fc861e8f9ee0ba290
Author: Jakub Sitnicki <jakub@cloudflare.com>
Date:   Tue Jan 21 13:31:47 2020 +0100

    net, sk_msg: Don't check if sock is locked when tearing down psock
```

```
    commit 58c8db929db1c1d785a6f5d8f8692e5dbcc35e84 upstream.

    As John Fastabend reports [0], psock state tear-down can happen on receive
    path *after* unlocking the socket, if the only other psock user, that is
    sockmap or sockhash, releases its psock reference before tcp_bpf_recvmsg
    does so:

     tcp_bpf_recvmsg()
      psock = sk_psock_get(sk)                        <- refcnt 2
      lock_sock(sk);
      ...
                                    sock_map_free()  <- refcnt 1
      release_sock(sk)
      sk_psock_put()                                  <- refcnt 0

    Remove the lockdep check for socket lock in psock tear-down that got
    introduced in 7e81a3530206 ("bpf: Sockmap, ensure sock lock held during
    tear down").

    [0] https://lore.kernel.org/netdev/5e25dc995d7d_74082aaee6e465b441@john-XPS-13-
9370.notmuch/

    Fixes: 7e81a3530206 ("bpf: Sockmap, ensure sock lock held during tear down")
    Reported-by: syzbot+d73682fcf7fee6982fe3@syzkaller.appspotmail.com
    Suggested-by: John Fastabend <john.fastabend@gmail.com>
    Signed-off-by: Jakub Sitnicki <jakub@cloudflare.com>
    Acked-by: John Fastabend <john.fastabend@gmail.com>
    Acked-by: Daniel Borkmann <daniel@iogearbox.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2e061912c3ec2843e149208fc0160c2bed877cb
Author: Ulrich Weber <ulrich.weber@gmail.com>
Date:   Wed Jan 15 12:11:29 2020 +0100

    xfrm: support output_mark for offload ESP packets

    commit 4e4362d2bf2a49ff44dbbc9585207977ca3d71d0 upstream.

    Commit 9b42c1f179a6 ("xfrm: Extend the output_mark") added output_mark
    support but missed ESP offload support.

    xfrm_smark_get() is not called within xfrm_input() for packets coming
    from esp4_gro_receive() or esp6_gro_receive(). Therefore call
    xfrm_smark_get() directly within these functions.

    Fixes: 9b42c1f179a6 ("xfrm: Extend the output_mark to support input direction and
masking.")
    Signed-off-by: Ulrich Weber <ulrich.weber@gmail.com>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4e1572b19c64244cac4a8aee395b02a2b819e72d
Author: Matthew Auld <matthew.auld@intel.com>
Date:   Fri Jan 17 13:24:13 2020 +0000

    drm/i915/userptr: fix size calculation

    commit ecc4d2a52df65479de5e333a9065ed02202a400f upstream.

    If we create a rather large userptr object(e.g 1ULL << 32) we might
    shift past the type-width of num_pages: (int)num_pages << PAGE_SHIFT,
    resulting in a totally bogus sg_table, which fortunately will eventually
    manifest as:
```

```
    gen8_ppgtt_insert_huge:463 GEM_BUG_ON(iter->sg->length < page_size)
    kernel BUG at drivers/gpu/drm/i915/gt/gen8_ppgtt.c:463!

    v2: more unsigned long
        prefer I915_GTT_PAGE_SIZE

    Fixes: 5cc9ed4b9a7a ("drm/i915: Introduce mapping of user pages into video memory
(userptr) ioctl")
    Signed-off-by: Matthew Auld <matthew.auld@intel.com>
    Cc: Chris Wilson <chris@chris-wilson.co.uk>
    Reviewed-by: Chris Wilson <chris@chris-wilson.co.uk>
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200117132413.1170563-2-
matthew.auld@intel.com
    (cherry picked from commit 8e78871bc1e5efec22c950d3fd24ddb63d4ff28a)
    Signed-off-by: Joonas Lahtinen <joonas.lahtinen@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 503b54ab9666567e1aff98dc207cf3c89b7d2ab3
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Wed Dec 11 10:15:52 2019 +0100

    iwlwifi: mvm: fix potential SKB leak on TXQ TX

    commit df2378ab0f2a9dd4cf4501268af1902cc4ebacd8 upstream.

    When we transmit after TXQ dequeue, we aren't paying attention to
    the return value of the transmit functions, leading to a potential
    SKB leak.

    Refactor the code a bit (and rename ..._tx to ..._tx_sta) to check
    for this happening.

    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Fixes: cfbc6c4c5b91 ("iwlwifi: mvm: support mac80211 TXQs model")
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1e3c082df32052e77f132a5bb4b2249ef471152c
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Wed Dec 11 10:09:56 2019 +0100

    iwlwifi: mvm: fix SKB leak on invalid queue

    commit b9f726c94224e863d4d3458dfec2e7e1284a39ce upstream.

    It used to be the case that if we got here, we wouldn't warn
    but instead allocate the queue (DQA). With using the mac80211
    TXQs model this changed, and we really have nothing to do with
    the frame here anymore, hence the warning now.

    However, clearly we missed in coding & review that this is now
    a pure error path and leaks the SKB if we return 0 instead of
    an indication that the SKB needs to be freed. Fix this.

    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Fixes: cfbc6c4c5b91 ("iwlwifi: mvm: support mac80211 TXQs model")
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 68ee631ba110f03d6c415ffce399c9f0e987562e
Author: Changbin Du <changbin.du@intel.com>
Date:   Sun Jan 12 11:42:31 2020 +0800
```

```
    tracing: xen: Ordered comparison of function pointers

    commit d0695e2351102affd8efae83989056bc4b275917 upstream.

    Just as commit 0566e40ce7 ("tracing: initcall: Ordered comparison of
    function pointers"), this patch fixes another remaining one in xen.h
    found by clang-9.

    In file included from arch/x86/xen/trace.c:21:
    In file included from ./include/trace/events/xen.h:475:
    In file included from ./include/trace/define_trace.h:102:
    In file included from ./include/trace/trace_events.h:473:
    ./include/trace/events/xen.h:69:7: warning: ordered comparison of function \
    pointers ('xen_mc_callback_fn_t' (aka 'void (*)(void *)') and 'xen_mc_callback_fn_t') [-
Wordered-compare-function-pointers]
                        __field(xen_mc_callback_fn_t, fn)
                        ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ./include/trace/trace_events.h:421:29: note: expanded from macro '__field'
                                    ^
    ./include/trace/trace_events.h:407:6: note: expanded from macro '__field_ext'
                                     is_signed_type(type), filter_type);    \
                                     ^
    ./include/linux/trace_events.h:554:44: note: expanded from macro 'is_signed_type'
                                                ^


    Fixes: c796f213a6934 ("xen/trace: add multicall tracing")
    Signed-off-by: Changbin Du <changbin.du@gmail.com>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3c6a183d3108229b1f4d8d378061d0dca1c17838
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Wed Jan 15 20:47:37 2020 -0800

    scsi: RDMA/isert: Fix a recently introduced regression related to logout

    commit 04060db41178c7c244f2c7dcd913e7fd331de915 upstream.

    iscsit_close_connection() calls isert_wait_conn(). Due to commit
    e9d3009cb936 both functions call target_wait_for_sess_cmds() although that
    last function should be called only once. Fix this by removing the
    target_wait_for_sess_cmds() call from isert_wait_conn() and by only calling
    isert_wait_conn() after target_wait_for_sess_cmds().

    Fixes: e9d3009cb936 ("scsi: target: iscsi: Wait for all commands to finish before
freeing a session").
    Link: https://lore.kernel.org/r/20200116044737.19507-1-bvanassche@acm.org
    Reported-by: Rahul Kundu <rahul.kundu@chelsio.com>
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Tested-by: Mike Marciniszyn <mike.marciniszyn@intel.com>
    Acked-by: Sagi Grimberg <sagi@grimberg.me>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bbf0d37947f71abbabb659d8e2114a86648eb83b
Author: Gilles Buloz <gilles.buloz@kontron.com>
Date:   Fri Nov 29 10:56:05 2019 +0100

    hwmon: (nct7802) Fix non-working alarm on voltages

    commit e51a7dda299815e92f43960d620cdfc8dfc144f2 upstream.

    No alarm is reported by /sys/.../inX_alarm

    In detail:
```

```
    The SMI Voltage status register is the only register giving a status
    for voltages, but it does not work like the non-SMI status registers
    used for temperatures and fans.
    A bit is set for each input crossing a threshold, in both direction,
    but the "inside" or "outside" limits info is not available.
    Also this register is cleared on read.
    Note : this is not explicitly spelled out in the datasheet, but from
    experiment.
    As a result if an input is crossing a threshold (min or max in any
    direction), the alarm is reported only once even if the input is
    still outside limits. Also if the alarm for another input is read
    before the one of this input, no alarm is reported at all.

    Signed-off-by: Gilles Buloz <gilles.buloz@kontron.com>
    Link: https://lore.kernel.org/r/5de0f566.tBga5POKAgHlmd0p%gilles.buloz@kontron.com
    Fixes: 3434f3783580 ("hwmon: Driver for Nuvoton NCT7802Y")
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 515b97c73447807a30f71267d15b329b194d4056
Author: Gilles Buloz <gilles.buloz@kontron.com>
Date:   Wed Nov 27 18:09:34 2019 +0100

    hwmon: (nct7802) Fix voltage limits to wrong registers

    commit 7713e62c8623c54dac88d1fa724aa487a38c3efb upstream.

    in0 thresholds are written to the in2 thresholds registers
    in2 thresholds to in3 thresholds
    in3 thresholds to in4 thresholds
    in4 thresholds to in0 thresholds

    Signed-off-by: Gilles Buloz <gilles.buloz@kontron.com>
    Link: https://lore.kernel.org/r/5de0f509.rc0oEvPOMjbfPW1w%gilles.buloz@kontron.com
    Fixes: 3434f3783580 ("hwmon: Driver for Nuvoton NCT7802Y")
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 54b818246b03163a99e569a4cf11c0e3a312696e
Author: xiaofeng.yan <yanxiaofeng7@jd.com>
Date:   Mon Jan 20 14:26:39 2020 +0800

    hsr: Fix a compilation error

    commit 80892772c4edac88c538165d26a0105f19b61c1c upstream.

    A compliation error happen when building branch 5.5-rc7

    In file included from net/hsr/hsr_main.c:12:0:
    net/hsr/hsr_main.h:194:20: error: two or more data types in declaration specifiers
     static inline void void hsr_debugfs_rename(struct net_device *dev)

    So Removed one void.

    Fixes: 4c2d5e33dcd3 ("hsr: rename debugfs file when interface name is changed")
    Signed-off-by: xiaofeng.yan <yanxiaofeng7@jd.com>
    Acked-by: Taehee Yoo <ap420073@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5440678b5acbfea878d30922ba44ffecfb9487b7
Author: Jacek Anaszewski <jacek.anaszewski@gmail.com>
Date:   Thu Dec 5 22:25:01 2019 +0100
```

```
    leds: gpio: Fix uninitialized gpio label for fwnode based probe

    commit 90a8e82d3ca8c1f85ac63f4a94c9b034f05af4ee upstream.

    When switching to using generic LED name composition mechanism via
    devm_led_classdev_register_ext() API the part of code initializing
    struct gpio_led's template name property was removed alongside.
    It was however overlooked that the property was also passed to
    devm_fwnode_get_gpiod_from_child() in place of "label" parameter,
    which when set to NULL, results in gpio label being initialized to '?'.

    It could be observed in debugfs and failed to properly identify
    gpio association with LED consumer.

    Fix this shortcoming by updating the GPIO label after the LED is
    registered and its final name is known.

    Fixes: d7235f5feaa0 ("leds: gpio: Use generic support for composing LED names")
    Cc: Russell King <linux@armlinux.org.uk>
    Reviewed-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Jacek Anaszewski <jacek.anaszewski@gmail.com>
    [fixed comment]
    Signed-off-by: Pavel Machek <pavel@ucw.cz>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 376b860334d2b2a0db65396093724b21be2e18a9
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed Jan 22 12:37:25 2020 -0800

    readdir: make user_access_begin() use the real access range

    commit 3c2659bd1db81ed6a264a9fc6262d51667d655ad upstream.

    In commit 9f79b78ef744 ("Convert filldir[64]() from __put_user() to
    unsafe_put_user()") I changed filldir to not do individual __put_user()
    accesses, but instead use unsafe_put_user() surrounded by the proper
    user_access_begin/end() pair.

    That make them enormously faster on modern x86, where the STAC/CLAC
    games make individual user accesses fairly heavy-weight.

    However, the user_access_begin() range was not really the exact right
    one, since filldir() has the unfortunate problem that it needs to not
    only fill out the new directory entry, it also needs to fix up the
    previous one to contain the proper file offset.

    It's unfortunate, but the "d_off" field in "struct dirent" is _not_ the
    file offset of the directory entry itself - it's the offset of the next
    one.  So we end up backfilling the offset in the previous entry as we
    walk along.

    But since x86 didn't really care about the exact range, and used to be
    the only architecture that did anything fancy in user_access_begin() to
    begin with, the filldir[64]() changes did something lazy, and even
    commented on it:

            /*
             * Note! This range-checks 'previous' (which may be NULL).
             * The real range was checked in getdents
             */
            if (!user_access_begin(dirent, sizeof(*dirent)))
                    goto efault;

    and it all worked fine.
```

```
    But now 32-bit ppc is starting to also implement user_access_begin(),
    and the fact that we faked the range to only be the (possibly not even
    valid) previous directory entry becomes a problem, because ppc32 will
    actually be using the range that is passed in for more than just "check
    that it's user space".

    This is a complete rewrite of Christophe's original patch.

    By saving off the record length of the previous entry instead of a
    pointer to it in the filldir data structures, we can simplify the range
    check and the writing of the previous entry d_off field.  No need for
    any conditionals in the user accesses themselves, although we retain the
    conditional EINTR checking for the "was this the first directory entry"
    signal handling latency logic.

    Fixes: 9f79b78ef744 ("Convert filldir[64]() from __put_user() to unsafe_put_user()")
    Link:
https://lore.kernel.org/lkml/a02d3426f93f7eb04960a4d9140902d278cab0bb.1579697910.git.christo
phe.leroy@c-s.fr/
    Link:
https://lore.kernel.org/lkml/408c90c4068b00ea8f1c41cca45b84ec23d4946b.1579783936.git.christo
phe.leroy@c-s.fr/
    Reported-and-tested-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 16aab32ff8b379f3e1e3b4f0d52f50cb52fc48bb
Author: Shuah Khan <skhan@linuxfoundation.org>
Date:   Thu Jan 23 15:32:14 2020 -0700

    iommu/amd: Fix IOMMU perf counter clobbering during init

    commit 8c17bbf6c8f70058a66305f2e1982552e6ea7f47 upstream.

    init_iommu_perf_ctr() clobbers the register when it checks write access
    to IOMMU perf counters and fails to restore when they are writable.

    Add save and restore to fix it.

    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Fixes: 30861ddc9cca4 ("perf/x86/amd: Add IOMMU Performance Counter resource management")
    Reviewed-by: Suravee Suthikulpanit <suravee.suthikulpanit@amd.com>
    Tested-by: Suravee Suthikulpanit <suravee.suthikulpanit@amd.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9f6216862a2075474fa985ea3b801c2ac4b41de3
Author: Christophe Leroy <christophe.leroy@c-s.fr>
Date:   Thu Jan 23 08:34:18 2020 +0000

    lib: Reduce user_access_begin() boundaries in strncpy_from_user() and strnlen_user()

    commit ab10ae1c3bef56c29bac61e1201c752221b87b41 upstream.

    The range passed to user_access_begin() by strncpy_from_user() and
    strnlen_user() starts at 'src' and goes up to the limit of userspace
    although reads will be limited by the 'count' param.

    On 32 bits powerpc (book3s/32) access has to be granted for each
    256Mbytes segment and the cost increases with the number of segments to
    unlock.

    Limit the range with 'count' param.

    Fixes: 594cc251fdd0 ("make 'user_access_begin()' do 'access_ok()'")
```

```
    Signed-off-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9e2e1a5abccd3a4cd331549399c6ff4008aa4bba
Author: Florian Westphal <fw@strlen.de>
Date:   Sat Jan 18 11:27:25 2020 +0100

    netfilter: nft_osf: add missing check for DREG attribute

    commit 7eaecf7963c1c8f62d62c6a8e7c439b0e7f2d365 upstream.

    syzbot reports just another NULL deref crash because of missing test
    for presence of the attribute.

    Reported-by: syzbot+cf23983d697c26c34f60@syzkaller.appspotmail.com
    Fixes:  b96af92d6eaf9fadd ("netfilter: nf_tables: implement Passive OS fingerprint
module in nft_osf")
    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>

commit 373ffce2337bb4ad8109cc4309fde532b0a02e4d
Author: Chuhong Yuan <hslester96@gmail.com>
Date:   Fri Jan 10 10:30:04 2020 -0800

    Input: sun4i-ts - add a check for devm_thermal_zone_of_sensor_register

    commit 97e24b095348a15ec08c476423c3b3b939186ad7 upstream.

    The driver misses a check for devm_thermal_zone_of_sensor_register().
    Add a check to fix it.

    Fixes: e28d0c9cd381 ("input: convert sun4i-ts to use
devm_thermal_zone_of_sensor_register")
    Signed-off-by: Chuhong Yuan <hslester96@gmail.com>
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 21aa01da9dba44e61700247c2813e78aeb3720cb
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 10 11:55:47 2020 -0800

    Input: pegasus_notetaker - fix endpoint sanity check

    commit bcfcb7f9b480dd0be8f0df2df17340ca92a03b98 upstream.

    The driver was checking the number of endpoints of the first alternate
    setting instead of the current one, something which could be used by a
    malicious device (or USB descriptor fuzzer) to trigger a NULL-pointer
    dereference.

    Fixes: 1afca2b66aac ("Input: add Pegasus Notetaker tablet driver")
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Acked-by: Martin Kepplinger <martink@posteo.de>
    Acked-by: Vladis Dronov <vdronov@redhat.com>
    Link: https://lore.kernel.org/r/20191210113737.4016-2-johan@kernel.org
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c65dfef2780316a1f916eec9689da1a09e6f77ee
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 10 11:59:32 2020 -0800

    Input: aiptek - fix endpoint sanity check
```

```
    commit 3111491fca4f01764e0c158c5e0f7ced808eef51 upstream.

    The driver was checking the number of endpoints of the first alternate
    setting instead of the current one, something which could lead to the
    driver binding to an invalid interface.

    This in turn could cause the driver to misbehave or trigger a WARN() in
    usb_submit_urb() that kernels with panic_on_warn set would choke on.

    Fixes: 8e20cf2bce12 ("Input: aiptek - fix crash on detecting device without endpoints")
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Acked-by: Vladis Dronov <vdronov@redhat.com>
    Link: https://lore.kernel.org/r/20191210113737.4016-3-johan@kernel.org
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 56ded4adf16e91ee522549e3d724fa3e1cdcdb4b
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 10 12:00:18 2020 -0800

    Input: gtco - fix endpoint sanity check

    commit a8eeb74df5a6bdb214b2b581b14782c5f5a0cf83 upstream.

    The driver was checking the number of endpoints of the first alternate
    setting instead of the current one, something which could lead to the
    driver binding to an invalid interface.

    This in turn could cause the driver to misbehave or trigger a WARN() in
    usb_submit_urb() that kernels with panic_on_warn set would choke on.

    Fixes: 162f98dea487 ("Input: gtco - fix crash on detecting device without endpoints")
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Acked-by: Vladis Dronov <vdronov@redhat.com>
    Link: https://lore.kernel.org/r/20191210113737.4016-5-johan@kernel.org
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3e629da150b53225d8f27cca5d27e3da783e40be
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 10 12:01:27 2020 -0800

    Input: sur40 - fix interface sanity checks

    commit 6b32391ed675827f8425a414abbc6fbd54ea54fe upstream.

    Make sure to use the current alternate setting when verifying the
    interface descriptors to avoid binding to an invalid interface.

    This in turn could cause the driver to misbehave or trigger a WARN() in
    usb_submit_urb() that kernels with panic_on_warn set would choke on.

    Fixes: bdb5c57f209c ("Input: add sur40 driver for Samsung SUR40 (aka MS Surface
2.0/Pixelsense)")
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Acked-by: Vladis Dronov <vdronov@redhat.com>
    Link: https://lore.kernel.org/r/20191210113737.4016-8-johan@kernel.org
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 68c2bc4da0986fb2add4e76cf1f41444ddc6e1b1
Author: Stephan Gerhold <stephan@gerhold.net>
Date:   Fri Jan 17 13:40:36 2020 -0800
```

```
    Input: pm8xxx-vib - fix handling of separate enable register

    commit 996d5d5f89a558a3608a46e73ccd1b99f1b1d058 upstream.

    Setting the vibrator enable_mask is not implemented correctly:

    For regmap_update_bits(map, reg, mask, val) we give in either
    regs->enable_mask or 0 (= no-op) as mask and "val" as value.
    But "val" actually refers to the vibrator voltage control register,
    which has nothing to do with the enable_mask.

    So we usually end up doing nothing when we really wanted
    to enable the vibrator.

    We want to set or clear the enable_mask (to enable/disable the vibrator).
    Therefore, change the call to always modify the enable_mask
    and set the bits only if we want to enable the vibrator.

    Fixes: d4c7c5c96c92 ("Input: pm8xxx-vib - handle separate enable register")
    Signed-off-by: Stephan Gerhold <stephan@gerhold.net>
    Link: https://lore.kernel.org/r/20200114183442.45720-1-stephan@gerhold.net
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c3f6abfe2f117390bd66488bf88059e704f38f39
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Fri Jan 10 04:38:32 2020 -0800

    net/tls: fix async operation

    commit db885e66d268884dc72967279b7e84f522556abc upstream.

    Mallesham reports the TLS with async accelerator was broken by
    commit d10523d0b3d7 ("net/tls: free the record on encryption error")
    because encryption can return -EINPROGRESS in such setups, which
    should not be treated as an error.

    The error is also present in the BPF path (likely copied from there).

    Reported-by: Mallesham Jatharakonda <mallesham.jatharakonda@oneconvergence.com>
    Fixes: d3b18ad31f93 ("tls: add bpf support to sk_msg handling")
    Fixes: d10523d0b3d7 ("net/tls: free the record on encryption error")
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Reviewed-by: Simon Horman <simon.horman@netronome.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c9cb4f2bea447ca21fc959005105b76b3455e83c
Author: Ido Schimmel <idosch@mellanox.com>
Date:   Wed Jan 15 13:53:46 2020 +0200

    mlxsw: switchx2: Do not modify cloned SKBs during xmit

    commit 63963d0f9d17be83d0e419e03282847ecc2c3715 upstream.

    The driver needs to prepend a Tx header to each packet it is
    transmitting. The header includes information such as the egress port
    and traffic class.

    The addition of the header requires the driver to modify the SKB's
    header and therefore it must not be shared. Otherwise, we risk hitting
    various race conditions.

    For example, when a packet is flooded (cloned) by the bridge driver to
    two switch ports swp1 and swp2:
```

```
    t0 - mlxsw_sp_port_xmit() is called for swp1. Tx header is prepended with
         swp1's port number
    t1 - mlxsw_sp_port_xmit() is called for swp2. Tx header is prepended with
         swp2's port number, overwriting swp1's port number
    t2 - The device processes data buffer from t0. Packet is transmitted via
         swp2
    t3 - The device processes data buffer from t1. Packet is transmitted via
         swp2

    Usually, the device is fast enough and transmits the packet before its
    Tx header is overwritten, but this is not the case in emulated
    environments.

    Fix this by making sure the SKB's header is writable by calling
    skb_cow_head(). Since the function ensures we have headroom to push the
    Tx header, the check further in the function can be removed.

    v2:
    * Use skb_cow_head() instead of skb_unshare() as suggested by Jakub
    * Remove unnecessary check regarding headroom

    Fixes: 31557f0f9755 ("mlxsw: Introduce Mellanox SwitchX-2 ASIC support")
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Reported-by: Shalom Toledo <shalomt@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 77d323cc3783d50f177a7b5b8ba8c9271b118048
Author: Faiz Abbas <faiz_abbas@ti.com>
Date:   Wed Jan 8 20:03:00 2020 +0530

    mmc: sdhci_am654: Reset Command and Data line after tuning

    commit de31f6ab68a3f548d88686d53514f252d78f61d5 upstream.

    The tuning data is leftover in the buffer after tuning. This can cause
    issues in future data commands, especially with CQHCI. Reset the command
    and data lines after tuning to continue from a clean state.

    Fixes: 41fd4caeb00b ("mmc: sdhci_am654: Add Initial Support for AM654 SDHCI driver")
    Cc: stable@vger.kernel.org
    Signed-off-by: Faiz Abbas <faiz_abbas@ti.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Link: https://lore.kernel.org/r/20200108143301.1929-3-faiz_abbas@ti.com
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d5bb81660456610a6b3e0b0f54df5a09898c0847
Author: Faiz Abbas <faiz_abbas@ti.com>
Date:   Wed Jan 8 20:02:59 2020 +0530

    mmc: sdhci_am654: Remove Inverted Write Protect flag

    commit 4d627c88546a697b07565dbb70d2f9f46a5ee76f upstream.

    The MMC/SD controllers on am65x and j721e don't in fact detect the write
    protect line as inverted. No issues were detected because of this
    because the sdwp line is not connected on any of the evms. Fix this by
    removing the flag.

    Fixes: 1accbced1c32 ("mmc: sdhci_am654: Add Support for 4 bit IP on J721E")
    Cc: stable@vger.kernel.org
    Signed-off-by: Faiz Abbas <faiz_abbas@ti.com>
```

```
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Link: https://lore.kernel.org/r/20200108143301.1929-2-faiz_abbas@ti.com
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d8bc6ac2db28ce03816263933a52adadbf68e984
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Wed Jan 15 10:54:35 2020 +0100

    mmc: sdhci: fix minimum clock rate for v3 controller

    commit 2a187d03352086e300daa2044051db00044cd171 upstream.

    For SDHCIv3+ with programmable clock mode, minimal clock frequency is
    still base clock / max(divider). Minimal programmable clock frequency is
    always greater than minimal divided clock frequency. Without this patch,
    SDHCI uses out-of-spec initial frequency when multiplier is big enough:

    mmc1: mmc_rescan_try_freq: trying to init card at 468750 Hz
    [for 480 MHz source clock divided by 1024]

    The code in sdhci_calc_clk() already chooses a correct SDCLK clock mode.

    Fixes: c3ed3877625f ("mmc: sdhci: add support for programmable clock mode")
    Cc: <stable@vger.kernel.org> # 4f6aa3264af4: mmc: tegra: Only advertise UHS modes if IO
regulator is present
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Link:
https://lore.kernel.org/r/ffb489519a446caffe7a0a05c4b9372bd52397bb.1579082031.git.mirq-
linux@rere.qmqm.pl
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 723f200e19bdb42bf0b1cd5c129d599b23e93db0
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Tue Jan 7 10:47:34 2020 +0100

    mmc: tegra: fix SDR50 tuning override

    commit f571389c0b015e76f91c697c4c1700aba860d34f upstream.

    Commit 7ad2ed1dfcbe inadvertently mixed up a quirk flag's name and
    broke SDR50 tuning override. Use correct NVQUIRK_ name.

    Fixes: 7ad2ed1dfcbe ("mmc: tegra: enable UHS-I modes")
    Cc: <stable@vger.kernel.org>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Reviewed-by: Thierry Reding <treding@nvidia.com>
    Tested-by: Thierry Reding <treding@nvidia.com>
    Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Link:
https://lore.kernel.org/r/9aff1d859935e59edd81e4939e40d6c55e0b55f6.1578390388.git.mirq-
linux@rere.qmqm.pl
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bccb1568bae8b7adf4de980b2843fecbe9c52c5e
Author: Alex Sverdlin <alexander.sverdlin@nokia.com>
Date:   Wed Jan 8 15:57:47 2020 +0100

    ARM: 8950/1: ftrace/recordmcount: filter relocation types

    commit 927d780ee371d7e121cea4fc7812f6ef2cea461c upstream.
```

```
Scenario 1, ARMv7
=================

If code in arch/arm/kernel/ftrace.c would operate on mcount() pointer
the following may be generated:

00000230 <prealloc_fixed_plts>:
 230:   b5f8            push    {r3, r4, r5, r6, r7, lr}
 232:   b500            push    {lr}
 234:   f7ff fffe       bl      0 <__gnu_mcount_nc>
                        234: R_ARM_THM_CALL     __gnu_mcount_nc
 238:   f240 0600       movw    r6, #0
                        238: R_ARM_THM_MOVW_ABS_NC      __gnu_mcount_nc
 23c:   f8d0 1180       ldr.w   r1, [r0, #384]  ; 0x180

FTRACE currently is not able to deal with it:

WARNING: CPU: 0 PID: 0 at .../kernel/trace/ftrace.c:1979 ftrace_bug+0x1ad/0x230()
...
CPU: 0 PID: 0 Comm: swapper/0 Not tainted 4.4.116-... #1
...
[<c0314e3d>] (unwind_backtrace) from [<c03115e9>] (show_stack+0x11/0x14)
[<c03115e9>] (show_stack) from [<c051a7f1>] (dump_stack+0x81/0xa8)
[<c051a7f1>] (dump_stack) from [<c0321c5d>] (warn_slowpath_common+0x69/0x90)
[<c0321c5d>] (warn_slowpath_common) from [<c0321cf3>] (warn_slowpath_null+0x17/0x1c)
[<c0321cf3>] (warn_slowpath_null) from [<c038ee9d>] (ftrace_bug+0x1ad/0x230)
[<c038ee9d>] (ftrace_bug) from [<c038f1f9>] (ftrace_process_locs+0x27d/0x444)
[<c038f1f9>] (ftrace_process_locs) from [<c08915bd>] (ftrace_init+0x91/0xe8)
[<c08915bd>] (ftrace_init) from [<c0885a67>] (start_kernel+0x34b/0x358)
[<c0885a67>] (start_kernel) from [<00308095>] (0x308095)
---[ end trace cb88537fdc8fa200 ]---
ftrace failed to modify [<c031266c>] prealloc_fixed_plts+0x8/0x60
 actual: 44:f2:e1:36
ftrace record flags: 0
 (0)   expected tramp: c03143e9

Scenario 2, ARMv4T
==================

ftrace: allocating 14435 entries in 43 pages
------------[ cut here ]------------
WARNING: CPU: 0 PID: 0 at kernel/trace/ftrace.c:2029 ftrace_bug+0x204/0x310
CPU: 0 PID: 0 Comm: swapper Not tainted 4.19.5 #1
Hardware name: Cirrus Logic EDB9302 Evaluation Board
[<c0010a24>] (unwind_backtrace) from [<c000ecb0>] (show_stack+0x20/0x2c)
[<c000ecb0>] (show_stack) from [<c03c72e8>] (dump_stack+0x20/0x30)
[<c03c72e8>] (dump_stack) from [<c0021c18>] (__warn+0xdc/0x104)
[<c0021c18>] (__warn) from [<c0021d7c>] (warn_slowpath_null+0x4c/0x5c)
[<c0021d7c>] (warn_slowpath_null) from [<c0095360>] (ftrace_bug+0x204/0x310)
[<c0095360>] (ftrace_bug) from [<c04dabac>] (ftrace_init+0x3b4/0x4d4)
[<c04dabac>] (ftrace_init) from [<c04cef4c>] (start_kernel+0x20c/0x410)
[<c04cef4c>] (start_kernel) from [<00000000>] (  (null))
---[ end trace 0506a2f5dae6b341 ]---
ftrace failed to modify
[<c000c350>] perf_trace_sys_exit+0x5c/0xe8
 actual:   1e:ff:2f:e1
Initializing ftrace call sites
ftrace record flags: 0
 (0)
 expected tramp: c000fb24

The analysis for this problem has been already performed previously,
refer to the link below.
```

```
    Fix the above problems by allowing only selected reloc types in
    __mcount_loc. The list itself comes from the legacy recordmcount.pl
    script.

    Link: https://lore.kernel.org/lkml/56961010.6000806@pengutronix.de/
    Cc: stable@vger.kernel.org
    Fixes: ed60453fa8f8 ("ARM: 6511/1: ftrace: add ARM support for C version of
recordmcount")
    Signed-off-by: Alexander Sverdlin <alexander.sverdlin@nokia.com>
    Acked-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Russell King <rmk+kernel@armlinux.org.uk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ff847d466bdb0f17608083185586bb3c60f7cb78
Author: Hans Verkuil <hverkuil-cisco@xs4all.nl>
Date:   Thu Jan 16 20:12:27 2020 -0800

    Revert "Input: synaptics-rmi4 - don't increment rmiaddr for SMBus transfers"

    commit 8ff771f8c8d55d95f102cf88a970e541a8bd6bcf upstream.

    This reverts commit a284e11c371e446371675668d8c8120a27227339.

    This causes problems (drifting cursor) with at least the F11 function that
    reads more than 32 bytes.

    The real issue is in the F54 driver, and so this should be fixed there, and
    not in rmi_smbus.c.

    So first revert this bad commit, then fix the real problem in F54 in another
    patch.

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Reported-by: Timo Kaufmann <timokau@zoho.com>
    Fixes: a284e11c371e ("Input: synaptics-rmi4 - don't increment rmiaddr for SMBus
transfers")
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200115124819.3191024-2-hverkuil-cisco@xs4all.nl
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cec754961758e4e7331d98a82140d64d2702aa7e
Author: Johan Hovold <johan@kernel.org>
Date:   Mon Jan 13 10:38:57 2020 -0800

    Input: keyspan-remote - fix control-message timeouts

    commit ba9a103f40fc4a3ec7558ec9b0b97d4f92034249 upstream.

    The driver was issuing synchronous uninterruptible control requests
    without using a timeout. This could lead to the driver hanging on probe
    due to a malfunctioning (or malicious) device until the device is
    physically disconnected. While sleeping in probe the driver prevents
    other devices connected to the same hub from being added to (or removed
    from) the bus.

    The USB upper limit of five seconds per request should be more than
    enough.

    Fixes: 99f83c9c9ac9 ("[PATCH] USB: add driver for Keyspan Digital Remote")
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: stable <stable@vger.kernel.org>     # 2.6.13
    Link: https://lore.kernel.org/r/20200113171715.30621-1-johan@kernel.org
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
```

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

commit 5d1973adcfffa180cb474ee55db0d990eb833378
Author: Jerry Snitselaar <jsnitsel@redhat.com>
Date:   Tue Jan 21 17:34:26 2020 -0700

    iommu/vt-d: Call __dmar_remove_one_dev_info with valid pointer

    commit bf708cfb2f4811d1948a88c41ab96587e84ad344 upstream.

    It is possible for archdata.iommu to be set to
    DEFER_DEVICE_DOMAIN_INFO or DUMMY_DEVICE_DOMAIN_INFO so check for
    those values before calling __dmar_remove_one_dev_info. Without a
    check it can result in a null pointer dereference. This has been seen
    while booting a kdump kernel on an HP dl380 gen9.

    Cc: Joerg Roedel <joro@8bytes.org>
    Cc: Lu Baolu <baolu.lu@linux.intel.com>
    Cc: David Woodhouse <dwmw2@infradead.org>
    Cc: stable@vger.kernel.org # 5.3+
    Cc: linux-kernel@vger.kernel.org
    Fixes: ae23bfb68f28 ("iommu/vt-d: Detach domain before using a private one")
    Signed-off-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d4616a9f8ee9d8a05fa9f36901d91ed6a0a49e1e
Author: Boyan Ding <boyan.j.ding@gmail.com>
Date:   Wed Jan 1 12:41:20 2020 -0800

    pinctrl: sunrisepoint: Add missing Interrupt Status register offset

    commit 9608ea6c6613ced75b2c41703d99f44e6f8849f1 upstream.

    Commit 179e5a6114cc ("pinctrl: intel: Remove default Interrupt Status
    offset") removes default interrupt status offset of GPIO controllers,
    with previous commits explicitly providing the previously default
    offsets. However, the is_offset value in SPTH_COMMUNITY is missing,
    preventing related irq from being properly detected and handled.

    Fixes: f702e0b93cdb ("pinctrl: sunrisepoint: Provide Interrupt Status register offset")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=205745
    Cc: stable@vger.kernel.org
    Signed-off-by: Boyan Ding <boyan.j.ding@gmail.com>
    Acked-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dd05cf12c72f11b7841d4ffeca29e5190606df1b
Author: Matthew Wilcox (Oracle) <willy@infradead.org>
Date:   Fri Jan 17 22:13:21 2020 -0500

    XArray: Fix xas_find returning too many entries

    commit c44aa5e8ab58b5f4cf473970ec784c3333496a2e upstream.

    If you call xas_find() with the initial index > max, it should have
    returned NULL but was returning the entry at index.

    Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db38561288b75082b5e839decaa15ed253bd2298
```

```
Author: Matthew Wilcox (Oracle) <willy@infradead.org>
Date:   Fri Jan 17 22:00:41 2020 -0500

    XArray: Fix xa_find_after with multi-index entries

    commit 19c30f4dd0923ef191f35c652ee4058e91e89056 upstream.

    If the entry is of an order which is a multiple of XA_CHUNK_SIZE,
    the current detection of sibling entries does not work.  Factor out
    an xas_sibling() function to make xa_find_after() a little more
    understandable, and write a new implementation that doesn't suffer from
    the same bug.

    Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a5135ca1f92a7b201b7f8297f42b8579f92bc55d
Author: Matthew Wilcox (Oracle) <willy@infradead.org>
Date:   Fri Jan 17 17:45:12 2020 -0500

    XArray: Fix infinite loop with entry at ULONG_MAX

    commit 430f24f94c8a174d411a550d7b5529301922e67a upstream.

    If there is an entry at ULONG_MAX, xa_for_each() will overflow the
    'index + 1' in xa_find_after() and wrap around to 0.  Catch this case
    and terminate the loop by returning NULL.

    Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8560e3d90afad8c5ee1ae4fea41949a76b5bb8b
Author: Emmanuel Grumbach <emmanuel.grumbach@intel.com>
Date:   Tue Dec 3 10:08:49 2019 +0200

    iwlwifi: mvm: don't send the IWL_MVM_RXQ_NSSN_SYNC notif to Rx queues

    commit d829229e35f302fd49c052b5c5906c90ecf9911d upstream.

    The purpose of this was to keep all the queues updated with
    the Rx sequence numbers because unlikely yet possible
    situations where queues can't understand if a specific
    packet needs to be dropped or not.

    Unfortunately, it was reported that this caused issues in
    our DMA engine. We don't fully understand how this is related,
    but this is being currently debugged. For now, just don't send
    this notification to the Rx queues. This de-facto reverts my
    commit 3c514bf831ac12356b695ff054bef641b9e99593:

    iwlwifi: mvm: add a loose synchronization of the NSSN across Rx queues

    This issue was reported here:
    https://bugzilla.kernel.org/show_bug.cgi?id=204873
    https://bugzilla.kernel.org/show_bug.cgi?id=205001
    and others maybe.

    Fixes: 3c514bf831ac ("iwlwifi: mvm: add a loose synchronization of the NSSN across Rx
queues")
    CC: <stable@vger.kernel.org> # 5.3+
    Signed-off-by: Emmanuel Grumbach <emmanuel.grumbach@intel.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 4e7d8cd6a1a8959a5a3dcb289162041c2eb49124
Author: Mehmet Akif Tasova <makiftasova@gmail.com>
Date:   Fri Dec 13 23:35:10 2019 +0300

    Revert "iwlwifi: mvm: fix scan config command size"

    commit 205608749e1ef394f513888091e613c5bfccbcca upstream.

    Since v5.4-rc1 was released, iwlwifi started throwing errors when scan
    commands were sent to the firmware with certain devices (depending on
    the OTP burned in the device, which contains the list of available
    channels).  For instance:

    iwlwifi 0000:00:14.3: FW error in SYNC CMD SCAN_CFG_CMD

    This bug was reported in the ArchLinux bug tracker:
    https://bugs.archlinux.org/task/64703

    And also in a specific case in bugzilla, when the lar_disabled option
    was set: https://bugzilla.kernel.org/show_bug.cgi?id=205193

    Revert the commit that introduced this error, by using the number of
    channels from the OTP instead of the number of channels that is
    specified in the FW TLV that tells us how many channels it supports.

    This reverts commit 06eb547c4ae4382e70d556ba213d13c95ca1801b.

    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Mehmet Akif Tasova <makiftasova@gmail.com>
    [ Luca: reworded the commit message a bit. ]
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7cf7a7d025879086bd69570398832e00d3613dd5
Author: Frederic Barrat <fbarrat@linux.ibm.com>
Date:   Mon Jan 13 14:01:18 2020 +0100

    powerpc/xive: Discard ESB load value when interrupt is invalid

    commit 17328f218fb760c9c6accc5b52494889243a6b98 upstream.

    A load on an ESB page returning all 1's means that the underlying
    device has invalidated the access to the PQ state of the interrupt
    through mmio. It may happen, for example when querying a PHB interrupt
    while the PHB is in an error state.

    In that case, we should consider the interrupt to be invalid when
    checking its state in the irq_get_irqchip_state() handler.

    Fixes: da15c03b047d ("powerpc/xive: Implement get_irqchip_state method for XIVE to fix
shutdown race")
    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Frederic Barrat <fbarrat@linux.ibm.com>
    [clg: wrote a commit log, introduced XIVE_ESB_INVALID ]
    Signed-off-by: Cédric Le Goater <clg@kaod.org>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/20200113130118.27969-1-clg@kaod.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6145307fa69332ab98cfd0efd6c63394418ba23b
Author: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
Date:   Wed Jan 8 11:14:22 2020 +0530

    powerpc/mm/hash: Fix sharing context ids between kernel & userspace
```

```
    commit 5d2e5dd5849b4ef5e8ec35e812cdb732c13cd27e upstream.

    Commit 0034d395f89d ("powerpc/mm/hash64: Map all the kernel regions in
    the same 0xc range") has a bug in the definition of MIN_USER_CONTEXT.

    The result is that the context id used for the vmemmap and the lowest
    context id handed out to userspace are the same. The context id is
    essentially the process identifier as far as the first stage of the
    MMU translation is concerned.

    This can result in multiple SLB entries with the same VSID (Virtual
    Segment ID), accessible to the kernel and some random userspace
    process that happens to get the overlapping id, which is not expected
    eg:

      07 c00c000008000000 40066bdea7000500  1T  ESID=   c00c00  VSID=      66bdea7 LLP:100
      12 0002000008000000 40066bdea7000d80  1T  ESID=      200  VSID=      66bdea7 LLP:100

    Even though the user process and the kernel use the same VSID, the
    permissions in the hash page table prevent the user process from
    reading or writing to any kernel mappings.

    It can also lead to SLB entries with different base page size
    encodings (LLP), eg:

      05 c00c000008000000 00006bde0053b500 256M ESID=c00c00000  VSID=    6bde0053b LLP:100
      09 0000000008000000 00006bde0053bc80 256M ESID=        0  VSID=    6bde0053b LLP:  0

    Such SLB entries can result in machine checks, eg. as seen on a G5:

      Oops: Machine check, sig: 7 [#1]
      BE PAGE SIZE=64K MU-Hash SMP NR_CPUS=4 NUMA Power Mac
      NIP: c00000000026f248 LR: c000000000295e58 CTR: 0000000000000000
      REGS: c0000000erfd3d70 TRAP: 0200 Tainted: G M (5.5.0-rcl-gcc-8.2.0-00010-
g228b667d8ea1)
      MSR: 9000000000109032 <SF,HV,EE,ME,IR,DR,RI> CR: 24282048 XER: 00000000
      DAR: c00c000000612c80 DSISR: 00000400 IRQMASK: 0
      ...
      NIP [c00000000026f248] .kmem_cache_free+0x58/0x140
      LR  [c088000008295e58] .putname 8x88/0xa
      Call Trace:
        .putname+0xB8/0xa
        .filename_lookup.part.76+0xbe/0x160
        .do_faccessat+0xe0/0x380
        system_call+0x5c/ex68

    This happens with 256MB segments and 64K pages, as the duplicate VSID
    is hit with the first vmemmap segment and the first user segment, and
    older 32-bit userspace maps things in the first user segment.

    On other CPUs a machine check is not seen. Instead the userspace
    process can get stuck continuously faulting, with the fault never
    properly serviced, due to the kernel not understanding that there is
    already a HPTE for the address but with inaccessible permissions.

    On machines with 1T segments we've not seen the bug hit other than by
    deliberately exercising it. That seems to be just a matter of luck
    though, due to the typical layout of the user virtual address space
    and the ranges of vmemmap that are typically populated.

    To fix it we add 2 to MIN_USER_CONTEXT. This ensures the lowest
    context given to userspace doesn't overlap with the VMEMMAP context,
    or with the context for INVALID_REGION_ID.
```

```
    Fixes: 0034d395f89d ("powerpc/mm/hash64: Map all the kernel regions in the same 0xc
range")
    Cc: stable@vger.kernel.org # v5.2+
    Reported-by: Christian Marillat <marillat@debian.org>
    Reported-by: Romain Dolbeau <romain@dolbeau.org>
    Signed-off-by: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
    [mpe: Account for INVALID_REGION_ID, mostly rewrite change log]
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/20200123102547.11623-1-mpe@ellerman.id.au
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bf470f051556b610265ea02ad2102de75e8e619c
Author: Steven Rostedt (VMware) <rostedt@goodmis.org>
Date:   Mon Jan 20 13:07:31 2020 -0500

    tracing: Fix histogram code when expression has same var as value

    commit 8bcebc77e85f3d7536f96845a0fe94b1dddb6af0 upstream.

    While working on a tool to convert SQL syntex into the histogram language of
    the kernel, I discovered the following bug:

     # echo 'first u64 start_time u64 end_time pid_t pid u64 delta' >> synthetic_events
     # echo 'hist:keys=pid:start=common_timestamp' > events/sched/sched_waking/trigger
     # echo
'hist:keys=next_pid:delta=common_timestamp-$start,start2=$start:onmatch(sched.sched_waking).
trace(first,$start2,common_timestamp,next_pid,$delta)' > events/sched/sched_switch/trigger

    Would not display any histograms in the sched_switch histogram side.

    But if I were to swap the location of

      "delta=common_timestamp-$start" with "start2=$start"

    Such that the last line had:

     # echo
'hist:keys=next_pid:start2=$start,delta=common_timestamp-$start:onmatch(sched.sched_waking).
trace(first,$start2,common_timestamp,next_pid,$delta)' > events/sched/sched_switch/trigger

    The histogram works as expected.

    What I found out is that the expressions clear out the value once it is
    resolved. As the variables are resolved in the order listed, when
    processing:

      delta=common_timestamp-$start

    The $start is cleared. When it gets to "start2=$start", it errors out with
    "unresolved symbol" (which is silent as this happens at the location of the
    trace), and the histogram is dropped.

    When processing the histogram for variable references, instead of adding a
    new reference for a variable used twice, use the same reference. That way,
    not only is it more efficient, but the order will no longer matter in
    processing of the variables.

    From Tom Zanussi:

     "Just to clarify some more about what the problem was is that without
      your patch, we would have two separate references to the same variable,
      and during resolve_var_refs(), they'd both want to be resolved
      separately, so in this case, since the first reference to start wasn't
      part of an expression, it wouldn't get the read-once flag set, so would
      be read normally, and then the second reference would do the read-once
```

```
        read and also be read but using read-once.  So everything worked and
        you didn't see a problem:

         from: start2=$start,delta=common_timestamp-$start

        In the second case, when you switched them around, the first reference
        would be resolved by doing the read-once, and following that the second
        reference would try to resolve and see that the variable had already
        been read, so failed as unset, which caused it to short-circuit out and
        not do the trigger action to generate the synthetic event:

         to: delta=common_timestamp-$start,start2=$start

        With your patch, we only have the single resolution which happens
        correctly the one time it's resolved, so this can't happen."

    Link: https://lore.kernel.org/r/20200116154216.58ca08eb@gandalf.local.home

    Cc: stable@vger.kernel.org
    Fixes: 067fe038e70f6 ("tracing: Add variable reference handling to hist triggers")
    Reviewed-by: Tom Zanuss <zanussi@kernel.org>
    Tested-by: Tom Zanussi <zanussi@kernel.org>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 07af0d03e8b2e888b07ce5ed353fa8b6497918e1
Author: Masami Ichikawa <masami256@gmail.com>
Date:   Thu Jan 16 22:12:36 2020 +0900

    tracing: Do not set trace clock if tracefs lockdown is in effect

    commit bf24daac8f2bd5b8affaec03c2be1d20bcdd6837 upstream.

    When trace_clock option is not set and unstable clcok detected,
    tracing_set_default_clock() sets trace_clock(ThinkPad A285 is one of
    case). In that case, if lockdown is in effect, null pointer
    dereference error happens in ring_buffer_set_clock().

    Link: http://lkml.kernel.org/r/20200116131236.3866925-1-masami256@gmail.com

    Cc: stable@vger.kernel.org
    Fixes: 17911ff38aa58 ("tracing: Add locked_down checks to the open calls of files
created for tracefs")
    Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=1788488
    Signed-off-by: Masami Ichikawa <masami256@gmail.com>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 67551f958458ba077c6237a14c4cf6a834cba667
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date:   Fri Jan 10 10:45:39 2020 +0900

    tracing/uprobe: Fix double perf_event linking on multiprobe uprobe

    commit 99c9a923e97a583a38050baa92c9377d73946330 upstream.

    Fix double perf_event linking to trace_uprobe_filter on
    multiple uprobe event by moving trace_uprobe_filter under
    trace_probe_event.

    In uprobe perf event, trace_uprobe_filter data structure is
    managing target mm filters (in perf_event) related to each
    uprobe event.

    Since commit 60d53e2c3b75 ("tracing/probe: Split trace_event
```

```
    related data from trace_probe") left the trace_uprobe_filter
    data structure in trace_uprobe, if a trace_probe_event has
    multiple trace_uprobe (multi-probe event), a perf_event is
    added to different trace_uprobe_filter on each trace_uprobe.
    This leads a linked list corruption.

    To fix this issue, move trace_uprobe_filter to trace_probe_event
    and link it once on each event instead of each probe.

    Link: http://lkml.kernel.org/r/157862073931.1800.3800576241181489174.stgit@devnote2

    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: "Naveen N . Rao" <naveen.n.rao@linux.ibm.com>
    Cc: Anil S Keshavamurthy <anil.s.keshavamurthy@intel.com>
    Cc: "David S . Miller" <davem@davemloft.net>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: =?utf-8?q?Toke_H=C3=B8iland-J?= =?utf-8?b?w7hyZ2Vuc2Vu?= <thoiland@redhat.com>
    Cc: Jean-Tsung Hsiao <jhsiao@redhat.com>
    Cc: Jesper Dangaard Brouer <brouer@redhat.com>
    Cc: stable@vger.kernel.org
    Fixes: 60d53e2c3b75 ("tracing/probe: Split trace_event related data from trace_probe")
    Link: https://lkml.kernel.org/r/20200108171611.GA8472@kernel.org
    Reported-by: Arnaldo Carvalho de Melo <acme@kernel.org>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 44eebad1049644e6200d6a08c7f6d56436c55021
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date:   Fri Dec 20 11:31:43 2019 +0900

    tracing: trigger: Replace unneeded RCU-list traversals

    commit aeed8aa3874dc15b9d82a6fe796fd7cfbb684448 upstream.

    With CONFIG_PROVE_RCU_LIST, I had many suspicious RCU warnings
    when I ran ftracetest trigger testcases.

    -----
      # dmesg -c > /dev/null
      # ./ftracetest test.d/trigger
      ...
      # dmesg | grep "RCU-list traversed" | cut -f 2 -d ] | cut -f 2 -d " "
      kernel/trace/trace_events_hist.c:6070
      kernel/trace/trace_events_hist.c:1760
      kernel/trace/trace_events_hist.c:5911
      kernel/trace/trace_events_trigger.c:504
      kernel/trace/trace_events_hist.c:1810
      kernel/trace/trace_events_hist.c:3158
      kernel/trace/trace_events_hist.c:3105
      kernel/trace/trace_events_hist.c:5518
      kernel/trace/trace_events_hist.c:5998
      kernel/trace/trace_events_hist.c:6019
      kernel/trace/trace_events_hist.c:6044
      kernel/trace/trace_events_trigger.c:1500
      kernel/trace/trace_events_trigger.c:1540
      kernel/trace/trace_events_trigger.c:539
      kernel/trace/trace_events_trigger.c:584
    -----

    I investigated those warnings and found that the RCU-list
    traversals in event trigger and hist didn't need to use
```

```
    RCU version because those were called only under event_mutex.

    I also checked other RCU-list traversals related to event
    trigger list, and found that most of them were called from
    event_hist_trigger_func() or hist_unregister_trigger() or
    register/unregister functions except for a few cases.

    Replace these unneeded RCU-list traversals with normal list
    traversal macro and lockdep_assert_held() to check the
    event_mutex is held.

    Link: http://lkml.kernel.org/r/157680910305.11685.15110237954275915782.stgit@devnote2

    Cc: stable@vger.kernel.org
    Fixes: 30350d65ac567 ("tracing: Add variable support to hist triggers")
    Reviewed-by: Tom Zanussi <zanussi@kernel.org>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b1302cbe530b26ce6ba4a1a4019513393a566f6f
Author: Alexander Potapenko <glider@google.com>
Date:   Thu Jan 16 12:09:34 2020 +0100

    PM: hibernate: fix crashes with init_on_free=1

    commit 18451f9f9e5810b8bd1245c5ae166f257e0e2b9d upstream.

    Upon resuming from hibernation, free pages may contain stale data from
    the kernel that initiated the resume. This breaks the invariant
    inflicted by init_on_free=1 that freed pages must be zeroed.

    To deal with this problem, make clear_free_pages() also clear the free
    pages when init_on_free is enabled.

    Fixes: 6471384af2a6 ("mm: security: introduce init_on_alloc=1 and init_on_free=1 boot
options")
    Reported-by: Johannes Stezenbach <js@sig21.net>
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Cc: 5.3+ <stable@vger.kernel.org> # 5.3+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d63b777d8145d351d1e8f991156a98fe51bea964
Author: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
Date:   Thu Jan 16 13:45:08 2020 +0000

    drm/i915: Align engine->uabi_class/instance with i915_drm.h

    commit 5eec71829ad7749a8c918f66a91a9bcf6fb4462a upstream.

    In our ABI we have defined I915_ENGINE_CLASS_INVALID_NONE and
    I915_ENGINE_CLASS_INVALID_VIRTUAL as negative values which creates
    implicit coupling with type widths used in, also ABI, struct
    i915_engine_class_instance.

    One place where we export engine->uabi_class
    I915_ENGINE_CLASS_INVALID_VIRTUAL is from our our tracepoints. Because the
    type of the former is u8 in contrast to u16 defined in the ABI, 254 will
    be returned instead of 65534 which userspace would legitimately expect.

    Another place is I915_CONTEXT_PARAM_ENGINES.

    Therefore we need to align the type used to store engine ABI class and
    instance.
```

```
    v2:
     * Update the commit message mentioning get_engines and cc stable.
       (Chris)

    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Fixes: 6d06779e8672 ("drm/i915: Load balancing across a virtual engine")
    Cc: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: <stable@vger.kernel.org> # v5.3+
    Reviewed-by: Chris Wilson <chris@chris-wilson.co.uk>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200116134508.25211-1-
tvrtko.ursulin@linux.intel.com
    (cherry picked from commit 0b3bd0cdc329a1e2e00995cffd61aacf58c87cb4)
    Signed-off-by: Joonas Lahtinen <joonas.lahtinen@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3e041c27b9909704a54bc673f9110391e740f583
Author: Boris Brezillon <boris.brezillon@collabora.com>
Date:   Wed Jan 15 20:15:54 2020 -0600

    drm/panfrost: Add the panfrost_gem_mapping concept

    commit bdefca2d8dc0f80bbe49e08bf52a717146490706 upstream.

    With the introduction of per-FD address space, the same BO can be mapped
    in different address space if the BO is globally visible (GEM_FLINK)
    and opened in different context or if the dmabuf is self-imported. The
    current implementation does not take case into account, and attaches the
    mapping directly to the panfrost_gem_object.

    Let's create a panfrost_gem_mapping struct and allow multiple mappings
    per BO.

    The mappings are refcounted which helps solve another problem where
    mappings were torn down (GEM handle closed by userspace) while GPU
    jobs accessing those BOs were still in-flight. Jobs now keep a
    reference on the mappings they use.

    v2 (robh):
    - Minor review comment clean-ups from Steven
    - Use list_is_singular helper
    - Just WARN if we add a mapping when madvise state is not WILLNEED.
      With that, drop the use of object_name_lock.

    v3 (robh):
    - Revert returning list iterator in panfrost_gem_mapping_get()

    Fixes: a5efb4c9a562 ("drm/panfrost: Restructure the GEM object creation")
    Fixes: 7282f7645d06 ("drm/panfrost: Implement per FD address spaces")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Boris Brezillon <boris.brezillon@collabora.com>
    Signed-off-by: Rob Herring <robh@kernel.org>
    Acked-by: Boris Brezillon <boris.brezillon@collabora.com>
    Reviewed-by: Steven Price <steven.price@arm.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200116021554.15090-1-
robh@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8c40a8d7d38444ce8f0ba7be9689ff5d254d71a3
Author: Alex Deucher <alexander.deucher@amd.com>
Date:   Tue Jan 14 17:09:28 2020 -0600

    PCI: Mark AMD Navi14 GPU rev 0xc5 ATS as broken

    commit 5e89cd303e3a4505752952259b9f1ba036632544 upstream.
```

```
    To account for parts of the chip that are "harvested" (disabled) due to
    silicon flaws, caches on some AMD GPUs must be initialized before ATS is
    enabled.

    ATS is normally enabled by the IOMMU driver before the GPU driver loads, so
    this cache initialization would have to be done in a quirk, but that's too
    complex to be practical.

    For Navi14 (device ID 0x7340), this initialization is done by the VBIOS,
    but apparently some boards went to production with an older VBIOS that
    doesn't do it.  Disable ATS for those boards.

    Link: https://lore.kernel.org/r/20200114205523.1054271-3-alexander.deucher@amd.com
    Bug: https://gitlab.freedesktop.org/drm/amd/issues/1015
    See-also: d28ca864c493 ("PCI: Mark AMD Stoney Radeon R7 GPU ATS as broken")
    See-also: 9b44b0b09dec ("PCI: Mark AMD Stoney GPU ATS as broken")
    [bhelgaas: squash into one patch, simplify slightly, commit log]
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fdd0f3b0e98badb7eb2428252f83c2acb494e176
Author: Jeff Layton <jlayton@kernel.org>
Date:   Wed Apr 3 13:16:01 2019 -0400

    ceph: hold extra reference to r_parent over life of request

    commit 9c1c2b35f1d94de8325344c2777d7ee67492db3b upstream.

    Currently, we just assume that it will stick around by virtue of the
    submitter's reference, but later patches will allow the syscall to
    return early and we can't rely on that reference at that point.

    While I'm not aware of any reports of it, Xiubo pointed out that this
    may fix a use-after-free.  If the wait for a reply times out or is
    canceled via signal, and then the reply comes in after the syscall
    returns, the client can end up trying to access r_parent without a
    reference.

    Take an extra reference to the inode when setting r_parent and release
    it when releasing the request.

    Cc: stable@vger.kernel.org
    Signed-off-by: Jeff Layton <jlayton@kernel.org>
    Reviewed-by: "Yan, Zheng" <zyan@redhat.com>
    Signed-off-by: Ilya Dryomov <idryomov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 19d0cd2127428aa755907aa05828f6698e48cf87
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Thu Jan 16 10:44:17 2020 -0800

    hwmon: (core) Do not use device managed functions for memory allocations

    commit 3bf8bdcf3bada771eb12b57f2a30caee69e8ab8d upstream.

    The hwmon core uses device managed functions, tied to the hwmon parent
    device, for various internal memory allocations. This is problematic
    since hwmon device lifetime does not necessarily match its parent's
    device lifetime. If there is a mismatch, memory leaks will accumulate
    until the parent device is released.

    Fix the problem by managing all memory allocations internally. The only
```

```
    exception is memory allocation for thermal device registration, which
    can be tied to the hwmon device, along with thermal device registration
    itself.

    Fixes: d560168b5d0f ("hwmon: (core) New hwmon registration API")
    Cc: stable@vger.kernel.org # v4.14.x: 47c332deb8e8: hwmon: Deal with errors from the
thermal subsystem
    Cc: stable@vger.kernel.org # v4.14.x: 74e3512731bd: hwmon: (core) Fix double-free in
__hwmon_device_register()
    Cc: stable@vger.kernel.org # v4.9.x: 3a412d5e4a1c: hwmon: (core) Simplify sysfs
attribute name allocation
    Cc: stable@vger.kernel.org # v4.9.x: 47c332deb8e8: hwmon: Deal with errors from the
thermal subsystem
    Cc: stable@vger.kernel.org # v4.9.x: 74e3512731bd: hwmon: (core) Fix double-free in
__hwmon_device_register()
    Cc: stable@vger.kernel.org # v4.9+
    Cc: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0387ee1b95d1cf21156dc73ccb7a34a8884b0958
Author: Luuk Paulussen <luuk.paulussen@alliedtelesis.co.nz>
Date:   Fri Dec 6 12:16:59 2019 +1300

    hwmon: (adt7475) Make volt2reg return same reg as reg2volt input

    commit cf3ca1877574a306c0207cbf7fdf25419d9229df upstream.

    reg2volt returns the voltage that matches a given register value.
    Converting this back the other way with volt2reg didn't return the same
    register value because it used truncation instead of rounding.

    This meant that values read from sysfs could not be written back to sysfs
    to set back the same register value.

    With this change, volt2reg will return the same value for every voltage
    previously returned by reg2volt (for the set of possible input values)

    Signed-off-by: Luuk Paulussen <luuk.paulussen@alliedtelesis.co.nz>
    Link: https://lore.kernel.org/r/20191205231659.1301-1-luuk.paulussen@alliedtelesis.co.nz
    cc: stable@vger.kernel.org
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2d00fec60c8d5c19b0c85936b47ec50ac7917bc5
Author: David Howells <dhowells@redhat.com>
Date:   Sun Jan 26 01:02:53 2020 +0000

    afs: Fix characters allowed into cell names

    commit a45ea48e2bcd92c1f678b794f488ca0bda9835b8 upstream.

    The afs filesystem needs to prohibit certain characters from cell names,
    such as '/', as these are used to form filenames in procfs, leading to
    the following warning being generated:

            WARNING: CPU: 0 PID: 3489 at fs/proc/generic.c:178

    Fix afs_alloc_cell() to disallow nonprintable characters, '/', '@' and
    names that begin with a dot.

    Remove the check for "@cell" as that is then redundant.

    This can be tested by running:
```

```
             echo add foo/.bar 1.2.3.4 >/proc/fs/afs/cells

     Note that we will also need to deal with:

      - Names ending in ".invalid" shouldn't be passed to the DNS.

      - Names that contain non-valid domainname chars shouldn't be passed to
        the DNS.

      - DNS replies that say "your-dns-needs-immediate-attention.<gTLD>" and
        replies containing A records that say 127.0.53.53 should be
        considered invalid.
        [https://www.icann.org/en/system/files/files/name-collision-mitigation-01aug14-
en.pdf]

     but these need to be dealt with by the kafs-client DNS program rather
     than the kernel.

     Reported-by: syzbot+b904ba7c947a37b4b291@syzkaller.appspotmail.com
     Cc: stable@kernel.org
     Signed-off-by: David Howells <dhowells@redhat.com>
     Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b29d143a690a660b27e7e84c01dd541d38fe9372
Author: Jens Axboe <axboe@kernel.dk>
Date:   Sun Jan 26 09:53:12 2020 -0700

     Revert "io_uring: only allow submit from owning task"

     commit 73e08e711d9c1d79fae01daed4b0e1fee5f8a275 upstream.

     This ends up being too restrictive for tasks that willingly fork and
     share the ring between forks. Andres reports that this breaks his
     postgresql work. Since we're close to 5.5 release, revert this change
     for now.

     Cc: stable@vger.kernel.org
     Fixes: 44d282796f81 ("io_uring: only allow submit from owning task")
     Reported-by: Andres Freund <andres@anarazel.de>
     Signed-off-by: Jens Axboe <axboe@kernel.dk>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 29ce06b6b6d45ce461ddb723502a0827b8d8c9fb
Author: David Ahern <dsahern@gmail.com>
Date:   Fri Jan 10 09:03:58 2020 -0800

     ipv4: Detect rollover in specific fib table dump

     [ Upstream commit 9827c0634e461703abf81e8cc8b7adf5da5886d0 ]

     Sven-Haegar reported looping on fib dumps when 255.255.255.255 route has
     been added to a table. The looping is caused by the key rolling over from
     FFFFFFFF to 0. When dumping a specific table only, we need a means to detect
     when the table dump is done. The key and count saved to cb args are both 0
     only at the start of the table dump. If key is 0 and count > 0, then we are
     in the rollover case. Detect and return to avoid looping.

     This only affects dumps of a specific table; for dumps of all tables
     (the case prior to the change in the Fixes tag) inet_dump_fib moved
     the entry counter to the next table and reset the cb args used by
     fib_table_dump and fn_trie_dump_leaf, so the rollover ffffffff back
     to 0 did not cause looping with the dumps.

     Fixes: effe67926624 ("net: Enable kernel side filtering of route dumps")
```

```
    Reported-by: Sven-Haegar Koch <haegar@sdinet.de>
    Signed-off-by: David Ahern <dsahern@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9bd7ae7eab7cddbb7fa68cc9319829e5374316fa
Author: Tariq Toukan <tariqt@mellanox.com>
Date:   Mon Jan 20 13:42:00 2020 +0200

    net/mlx5e: kTLS, Do not send decrypted-marked SKBs via non-accel path

    commit 342508c1c7540e281fd36151c175ba5ff954a99f upstream.

    When TCP out-of-order is identified (unexpected tcp seq mismatch), driver
    analyzes the packet and decides what handling should it get:
    1. go to accelerated path (to be encrypted in HW),
    2. go to regular xmit path (send w/o encryption),
    3. drop.

    Packets marked with skb->decrypted by the TLS stack in the TX flow skips
    SW encryption, and rely on the HW offload.
    Verify that such packets are never sent un-encrypted on the wire.
    Add a WARN to catch such bugs, and prefer dropping the packet in these cases.

    Fixes: 46a3ea98074e ("net/mlx5e: kTLS, Enhance TX resync flow")
    Signed-off-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Boris Pismenny <borisp@mellanox.com>
    Reviewed-by: Boris Pismenny <borisp@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 43eda24c6f75cdeaf7780fdb18142d5a45601d02
Author: Tariq Toukan <tariqt@mellanox.com>
Date:   Mon Jan 13 14:46:09 2020 +0200

    net/mlx5e: kTLS, Remove redundant posts in TX resync flow

    commit 1e92899791358dba94a9db7cc3b6004636b5a2f6 upstream.

    The call to tx_post_resync_params() is done earlier in the flow,
    the post of the control WQEs is unnecessarily repeated. Remove it.

    Fixes: 700ec4974240 ("net/mlx5e: kTLS, Fix missing SQ edge fill")
    Signed-off-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Boris Pismenny <borisp@mellanox.com>
    Reviewed-by: Boris Pismenny <borisp@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 384faad2ad5a49c56e4e7c931533f7e2a6368f84
Author: Tariq Toukan <tariqt@mellanox.com>
Date:   Sun Jan 12 16:22:14 2020 +0200

    net/mlx5e: kTLS, Fix corner-case checks in TX resync flow

    commit ffbd9ca94e2ebbfe802d4b28bab5ba19818de853 upstream.

    There are the following cases:

    1. Packet ends before start marker: bypass offload.
    2. Packet starts before start marker and ends after it: drop,
       not supported, breaks contract with kernel.
    3. packet ends before tls record info starts: drop,
       this packet was already acknowledged and its record info
       was released.
```

```
    Add the above as comment in code.

    Mind possible wraparounds of the TCP seq, replace the simple comparison
    with a call to the TCP before() method.

    In addition, remove logic that handles negative sync_len values,
    as it became impossible.

    Fixes: d2ead1f360e8 ("net/mlx5e: Add kTLS TX HW offload support")
    Fixes: 46a3ea98074e ("net/mlx5e: kTLS, Enhance TX resync flow")
    Signed-off-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Boris Pismenny <borisp@mellanox.com>
    Reviewed-by: Boris Pismenny <borisp@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7ae31c620dcc6cc8efbe3db5e2f261af42403814
Author: Erez Shitrit <erezsh@mellanox.com>
Date:   Sun Jan 12 08:57:59 2020 +0200

    net/mlx5: DR, use non preemptible call to get the current cpu number

    commit c0702a4bd41829f05638ec2dab70f6bb8d8010ce upstream.

    Use raw_smp_processor_id instead of smp_processor_id() otherwise we will
    get the following trace in debug-kernel:
            BUG: using smp_processor_id() in preemptible [00000000] code: devlink
            caller is dr_create_cq.constprop.2+0x31d/0x970 [mlx5_core]
            Call Trace:
            dump_stack+0x9a/0xf0
            debug_smp_processor_id+0x1f3/0x200
            dr_create_cq.constprop.2+0x31d/0x970
            genl_family_rcv_msg+0x5fd/0x1170
            genl_rcv_msg+0xb8/0x160
            netlink_rcv_skb+0x11e/0x340

    Fixes: 297cccebdc5a ("net/mlx5: DR, Expose an internal API to issue RDMA operations")
    Signed-off-by: Erez Shitrit <erezsh@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8a3b1a86d4f3828160f48a56a33c090e557e4539
Author: Eli Cohen <eli@mellanox.com>
Date:   Sun Jan 12 13:43:37 2020 +0200

    net/mlx5: E-Switch, Prevent ingress rate configuration of uplink rep

    commit e401a1848be87123a2b2049addbf21138cb47081 upstream.

    Since the implementation relies on limiting the VF transmit rate to
    simulate ingress rate limiting, and since either uplink representor or
    ecpf are not associated with a VF, we limit the rate limit configuration
    for those ports.

    Fixes: fcb64c0f5640 ("net/mlx5: E-Switch, add ingress rate support")
    Signed-off-by: Eli Cohen <eli@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 b20f5fb7d473ef06a7c19d7f193ec5c4b34342c2
Author: Erez Shitrit <erezsh@mellanox.com>
Date:   Wed Jan 8 14:17:32 2020 +0200
```

```
    net/mlx5: DR, Enable counter on non-fwd-dest objects

    commmit b850a82114df9b0ec1d191dc64eed1f20a772e0f upstream.

    The current code handles only counters that attached to dest, we still
    have the cases where we have counter on non-dest, like over drop etc.

    Fixes: 6a48faeeca10 ("net/mlx5: Add direct rule fs_cmd implementation")
    Signed-off-by: Hamdan Igbaria <hamdani@mellanox.com>
    Signed-off-by: Erez Shitrit <erezsh@mellanox.com>
    Reviewed-by: Alex Vesker <valex@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ffc574c95e5bb3c247def6da00b2ce6358496ac0
Author: Meir Lichtinger <meirl@mellanox.com>
Date:   Thu Dec 12 16:09:33 2019 +0200

    net/mlx5: Update the list of the PCI supported devices

    commit 505a7f5478062c6cd11e22022d9f1bf64cd8eab3 upstream

    Add the upcoming ConnectX-7 device ID.

    Fixes: 85327a9c4150 ("net/mlx5: Update the list of the PCI supported devices")
    Signed-off-by: Meir Lichtinger <meirl@mellanox.com>
    Reviewed-by: Eran Ben Elisha <eranbe@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 36940c4daad0f8668234eb6f4f33c87fc8a01eef
Author: Paul Blakey <paulb@mellanox.com>
Date:   Tue Dec 31 17:04:15 2019 +0200

    net/mlx5: Fix lowest FDB pool size

    commit 93b8a7ecb7287cc9b0196f12a25b57c2462d11dc upstream.

    The pool sizes represent the pool sizes in the fw. when we request
    a pool size from fw, it will return the next possible group.
    We track how many pools the fw has left and start requesting groups
    from the big to the small.
    When we start request 4k group, which doesn't exists in fw, fw
    wants to allocate the next possible size, 64k, but will fail since
    its exhausted. The correct smallest pool size in fw is 128 and not 4k.

    Fixes: e52c28024008 ("net/mlx5: E-Switch, Add chains and priorities")
    Signed-off-by: Paul Blakey <paulb@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 d18d22ce8f62839365c984b1df474d3975ed4eb2
Author: Maxim Mikityanskiy <maximmi@mellanox.com>
Date:   Tue Jan 21 15:09:40 2020 +0000

    net: Fix packet reordering caused by GRO and listified RX cooperation

    [ Upstream commit c80794323e82ac6ab45052ebba5757ce47b4b588 ]

    Commit 323ebb61e32b ("net: use listified RX for handling GRO_NORMAL
    skbs") introduces batching of GRO_NORMAL packets in napi_frags_finish,
    and commit 6570bc79c0df ("net: core: use listified Rx for GRO_NORMAL in
    napi_gro_receive()") adds the same to napi_skb_finish. However,
    dev_gro_receive (that is called just before napi_{frags,skb}_finish) can
```

```
    also pass skbs to the networking stack: e.g., when the GRO session is
    flushed, napi_gro_complete is called, which passes pp directly to
    netif_receive_skb_internal, skipping napi->rx_list. It means that the
    packet stored in pp will be handled by the stack earlier than the
    packets that arrived before, but are still waiting in napi->rx_list. It
    leads to TCP reorderings that can be observed in the TCPOFOQueue counter
    in netstat.

    This commit fixes the reordering issue by making napi_gro_complete also
    use napi->rx_list, so that all packets going through GRO will keep their
    order. In order to keep napi_gro_flush working properly, gro_normal_list
    calls are moved after the flush to clear napi->rx_list.

    iwlwifi calls napi_gro_flush directly and does the same thing that is
    done by gro_normal_list, so the same change is applied there:
    napi_gro_flush is moved to be before the flush of napi->rx_list.

    A few other drivers also use napi_gro_flush (brocade/bna/bnad.c,
    cortina/gemini.c, hisilicon/hns3/hns3_enet.c). The first two also use
    napi_complete_done afterwards, which performs the gro_normal_list flush,
    so they are fine. The latter calls napi_gro_receive right after
    napi_gro_flush, so it can end up with non-empty napi->rx_list anyway.

    Fixes: 323ebb61e32b ("net: use listified RX for handling GRO_NORMAL skbs")
    Signed-off-by: Maxim Mikityanskiy <maximmi@mellanox.com>
    Cc: Alexander Lobakin <alobakin@dlink.ru>
    Cc: Edward Cree <ecree@solarflare.com>
    Acked-by: Alexander Lobakin <alobakin@dlink.ru>
    Acked-by: Saeed Mahameed <saeedm@mellanox.com>
    Acked-by: Edward Cree <ecree@solarflare.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8bc3025206165141cf7937f487fd0275479d5562
Author: Kristian Evensen <kristian.evensen@gmail.com>
Date:   Thu Jan 23 13:20:18 2020 +0100

    fou: Fix IPv6 netlink policy

    [ Upstream commit bb48eb9b12a95db9d679025927269d4adda6dbd1 ]

    When submitting v2 of "fou: Support binding FoU socket" (1713cb37bf67),
    I accidentally sent the wrong version of the patch and one fix was
    missing. In the initial version of the patch, as well as the version 2
    that I submitted, I incorrectly used ".type" for the two V6-attributes.
    The correct is to use ".len".

    Reported-by: Dmitry Vyukov <dvyukov@google.com>
    Fixes: 1713cb37bf67 ("fou: Support binding FoU socket")
    Signed-off-by: Kristian Evensen <kristian.evensen@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e7b6ef675c46d275250c66a516accd41cde7c0e6
Author: Ido Schimmel <idosch@mellanox.com>
Date:   Wed Jan 22 20:09:52 2020 +0200

    mlxsw: spectrum_acl: Fix use-after-free during reload

    [ Upstream commit 971de2e572118c1128bff295341e37b6c8b8f108 ]

    During reload (or module unload), the router block is de-initialized.
    Among other things, this results in the removal of a default multicast
    route from each active virtual router (VRF). These default routes are
    configured during initialization to trap packets to the CPU. In
```

```
    Spectrum-2, unlike Spectrum-1, multicast routes are implemented using
    ACL rules.

    Since the router block is de-initialized before the ACL block, it is
    possible that the ACL rules corresponding to the default routes are
    deleted while being accessed by the ACL delayed work that queries rules'
    activity from the device. This can result in a rare use-after-free [1].

    Fix this by protecting the rules list accessed by the delayed work with
    a lock. We cannot use a spinlock as the activity read operation is
    blocking.

    [1]
    [  123.331662] ==================================================================
    [  123.339920] BUG: KASAN: use-after-free in
mlxsw_sp_acl_rule_activity_update_work+0x330/0x3b0
    [  123.349381] Read of size 8 at addr ffff8881f3bb4520 by task kworker/0:2/78
    [  123.357080]
    [  123.358773] CPU: 0 PID: 78 Comm: kworker/0:2 Not tainted 5.5.0-rc5-custom-33108-
gf5df95d3ef41 #2209
    [  123.368898] Hardware name: Mellanox Technologies Ltd. MSN3700C/VMOD0008, BIOS 5.11
10/10/2018
    [  123.378456] Workqueue: mlxsw_core mlxsw_sp_acl_rule_activity_update_work
    [  123.385970] Call Trace:
    [  123.388734]  dump_stack+0xc6/0x11e
    [  123.392568]  print_address_description.constprop.4+0x21/0x340
    [  123.403236]  __kasan_report.cold.8+0x76/0xb1
    [  123.414884]  kasan_report+0xe/0x20
    [  123.418716]  mlxsw_sp_acl_rule_activity_update_work+0x330/0x3b0
    [  123.444034]  process_one_work+0xb06/0x19a0
    [  123.453731]  worker_thread+0x91/0xe90
    [  123.467348]  kthread+0x348/0x410
    [  123.476847]  ret_from_fork+0x24/0x30
    [  123.480863]
    [  123.482545] Allocated by task 73:
    [  123.486273]  save_stack+0x19/0x80
    [  123.490000]  __kasan_kmalloc.constprop.6+0xc1/0xd0
    [  123.495379]  mlxsw_sp_acl_rule_create+0xa7/0x230
    [  123.500566]  mlxsw_sp2_mr_tcam_route_create+0xf6/0x3e0
    [  123.506334]  mlxsw_sp_mr_tcam_route_create+0x5b4/0x820
    [  123.512102]  mlxsw_sp_mr_table_create+0x3b5/0x690
    [  123.517389]  mlxsw_sp_vr_get+0x289/0x4d0
    [  123.521797]  mlxsw_sp_fib_node_get+0xa2/0x990
    [  123.526692]  mlxsw_sp_router_fib4_event_work+0x54c/0x2d60
    [  123.532752]  process_one_work+0xb06/0x19a0
    [  123.537352]  worker_thread+0x91/0xe90
    [  123.541471]  kthread+0x348/0x410
    [  123.545103]  ret_from_fork+0x24/0x30
    [  123.549113]
    [  123.550795] Freed by task 518:
    [  123.554231]  save_stack+0x19/0x80
    [  123.557958]  __kasan_slab_free+0x125/0x170
    [  123.562556]  kfree+0xd7/0x3a0
    [  123.565895]  mlxsw_sp_acl_rule_destroy+0x63/0xd0
    [  123.571081]  mlxsw_sp2_mr_tcam_route_destroy+0xd5/0x130
    [  123.576946]  mlxsw_sp_mr_tcam_route_destroy+0xba/0x260
    [  123.582714]  mlxsw_sp_mr_table_destroy+0x1ab/0x290
    [  123.588091]  mlxsw_sp_vr_put+0x1db/0x350
    [  123.592496]  mlxsw_sp_fib_node_put+0x298/0x4c0
    [  123.597486]  mlxsw_sp_vr_fib_flush+0x15b/0x360
    [  123.602476]  mlxsw_sp_router_fib_flush+0xba/0x470
    [  123.607756]  mlxsw_sp_vrs_fini+0xaa/0x120
    [  123.612260]  mlxsw_sp_router_fini+0x137/0x384
    [  123.617152]  mlxsw_sp_fini+0x30a/0x4a0
    [  123.621374]  mlxsw_core_bus_device_unregister+0x159/0x600
```

```
     [  123.627435]  mlxsw_devlink_core_bus_device_reload_down+0x7e/0xb0
     [  123.634176]  devlink_reload+0xb4/0x380
     [  123.638391]  devlink_nl_cmd_reload+0x610/0x700
     [  123.643382]  genl_rcv_msg+0x6a8/0xdc0
     [  123.647497]  netlink_rcv_skb+0x134/0x3a0
     [  123.651904]  genl_rcv+0x29/0x40
     [  123.655436]  netlink_unicast+0x4d4/0x700
     [  123.659843]  netlink_sendmsg+0x7c0/0xc70
     [  123.664251]  __sys_sendto+0x265/0x3c0
     [  123.668367]  __x64_sys_sendto+0xe2/0x1b0
     [  123.672773]  do_syscall_64+0xa0/0x530
     [  123.676892]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
     [  123.682552]
     [  123.684238] The buggy address belongs to the object at ffff8881f3bb4500
     [  123.684238]  which belongs to the cache kmalloc-128 of size 128
     [  123.698261] The buggy address is located 32 bytes inside of
     [  123.698261]  128-byte region [ffff8881f3bb4500, ffff8881f3bb4580)
     [  123.711303] The buggy address belongs to the page:
     [  123.716682] page:ffffea0007ceed00 refcount:1 mapcount:0 mapping:ffff888236403500
 index:0x0
     [  123.725958] raw: 0200000000000200 dead000000000100 dead000000000122 ffff888236403500
     [  123.734646] raw: 0000000000000000 0000000000100010 00000001ffffffff 0000000000000000
     [  123.743315] page dumped because: kasan: bad access detected
     [  123.749562]
     [  123.751241] Memory state around the buggy address:
     [  123.756620]  ffff8881f3bb4400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
     [  123.764716]  ffff8881f3bb4480: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [  123.772812] >ffff8881f3bb4500: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     [  123.780904]                                    ^
     [  123.785697]  ffff8881f3bb4580: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [  123.793793]  ffff8881f3bb4600: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
     [  123.801883] ==================================================================
```

```
     Fixes: cf7221a4f5a5 ("mlxsw: spectrum_router: Add Multicast routing support for
 Spectrum-2")
     Signed-off-by: Ido Schimmel <idosch@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 ef3c0a1660a7d52fbcd1a417915bb5f06c72117a
 Author: Michael Ellerman <mpe@ellerman.id.au>
 Date:   Wed Jan 22 15:07:28 2020 +1100

     airo: Add missing CAP_NET_ADMIN check in AIROOLDIOCTL/SIOCDEVPRIVATE

     [ Upstream commit 78f7a7566f5eb59321e99b55a6fdb16ea05b37d1 ]

     The driver for Cisco Aironet 4500 and 4800 series cards (airo.c),
     implements AIROOLDIOCTL/SIOCDEVPRIVATE in airo_ioctl().

     The ioctl handler copies an aironet_ioctl struct from userspace, which
     includes a command. Some of the commands are handled in readrids(),
     where the user controlled command is converted into a driver-internal
     value called "ridcode".

     There are two command values, AIROGWEPKTMP and AIROGWEPKNV, which
     correspond to ridcode values of RID_WEP_TEMP and RID_WEP_PERM
     respectively. These commands both have checks that the user has
     CAP_NET_ADMIN, with the comment that "Only super-user can read WEP
     keys", otherwise they return -EPERM.

     However there is another command value, AIRORRID, that lets the user
     specify the ridcode value directly, with no other checks. This means
     the user can bypass the CAP_NET_ADMIN check on AIROGWEPKTMP and
```

```
    AIROGWEPKNV.

    Fix it by moving the CAP_NET_ADMIN check out of the command handling
    and instead do it later based on the ridcode. That way regardless of
    whether the ridcode is set via AIROGWEPKTMP or AIROGWEPKNV, or passed
    in using AIRORID, we always do the CAP_NET_ADMIN check.

    Found by Ilja by code inspection, not tested as I don't have the
    required hardware.

    Reported-by: Ilja Van Sprundel <ivansprundel@ioactive.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9975b19b8d82841077af1d5643c5a0a5599d2e02
Author: Michael Ellerman <mpe@ellerman.id.au>
Date:   Wed Jan 22 15:07:27 2020 +1100

    airo: Fix possible info leak in AIROOLDIOCTL/SIOCDEVPRIVATE

    [ Upstream commit d6bce2137f5d6bb1093e96d2f801479099b28094 ]

    The driver for Cisco Aironet 4500 and 4800 series cards (airo.c),
    implements AIROOLDIOCTL/SIOCDEVPRIVATE in airo_ioctl().

    The ioctl handler copies an aironet_ioctl struct from userspace, which
    includes a command and a length. Some of the commands are handled in
    readrids(), which kmalloc()'s a buffer of RIDSIZE (2048) bytes.

    That buffer is then passed to PC4500_readrid(), which has two cases.
    The else case does some setup and then reads up to RIDSIZE bytes from
    the hardware into the kmalloc()'ed buffer.

    Here len == RIDSIZE, pBuf is the kmalloc()'ed buffer:

            // read the rid length field
            bap_read(ai, pBuf, 2, BAP1);
            // length for remaining part of rid
            len = min(len, (int)le16_to_cpu(*(__le16*)pBuf)) - 2;
            ...
            // read remainder of the rid
            rc = bap_read(ai, ((__le16*)pBuf)+1, len, BAP1);

    PC4500_readrid() then returns to readrids() which does:

            len = comp->len;
            if (copy_to_user(comp->data, iobuf, min(len, (int)RIDSIZE))) {

    Where comp->len is the user controlled length field.

    So if the "rid length field" returned by the hardware is < 2048, and
    the user requests 2048 bytes in comp->len, we will leak the previous
    contents of the kmalloc()'ed buffer to userspace.

    Fix it by kzalloc()'ing the buffer.

    Found by Ilja by code inspection, not tested as I don't have the
    required hardware.

    Reported-by: Ilja Van Sprundel <ivansprundel@ioactive.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit a802ddbf41a3ba8dc5b05120f173c096dc54bd30
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Jan 22 09:07:35 2020 -0800

    tun: add mutex_unlock() call and napi.skb clearing in tun_get_user()

    [ Upstream commit 1efba987c48629c0c64703bb4ea76ca1a3771d17 ]

    If both IFF_NAPI_FRAGS mode and XDP are enabled, and the XDP program
    consumes the skb, we need to clear the napi.skb (or risk
    a use-after-free) and release the mutex (or risk a deadlock)

    WARNING: lock held when returning to user space!
    5.5.0-rc6-syzkaller #0 Not tainted
    ------------------------------------------------
    syz-executor.0/455 is leaving the kernel with locks still held!
    1 lock held by syz-executor.0/455:
     #0: ffff888098f6e748 (&tfile->napi_mutex){+.+.}, at: tun_get_user+0x1604/0x3fc0
drivers/net/tun.c:1835

    Fixes: 90e33d459407 ("tun: enable napi_gro_frags() for TUN/TAP driver")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Cc: Petar Penkov <ppenkov@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 69486bfa064e77895392be3ad6a7b29d22556668
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Jan 22 21:03:00 2020 -0800

    tcp: do not leave dangling pointers in tp->highest_sack

    [ Upstream commit 2bec445f9bf35e52e395b971df48d3e1e5dc704a ]

    Latest commit 853697504de0 ("tcp: Fix highest_sack and highest_sack_seq")
    apparently allowed syzbot to trigger various crashes in TCP stack [1]

    I believe this commit only made things easier for syzbot to find
    its way into triggering use-after-frees. But really the bugs
    could lead to bad TCP behavior or even plain crashes even for
    non malicious peers.

    I have audited all calls to tcp_rtx_queue_unlink() and
    tcp_rtx_queue_unlink_and_free() and made sure tp->highest_sack would be updated
    if we are removing from rtx queue the skb that tp->highest_sack points to.

    These updates were missing in three locations :

    1) tcp_clean_rtx_queue() [This one seems quite serious,
                              I have no idea why this was not caught earlier]

    2) tcp_rtx_queue_purge() [Probably not a big deal for normal operations]

    3) tcp_send_synack()     [Probably not a big deal for normal operations]

    [1]
    BUG: KASAN: use-after-free in tcp_highest_sack_seq include/net/tcp.h:1864 [inline]
    BUG: KASAN: use-after-free in tcp_highest_sack_seq include/net/tcp.h:1856 [inline]
    BUG: KASAN: use-after-free in tcp_check_sack_reordering+0x33c/0x3a0
net/ipv4/tcp_input.c:891
    Read of size 4 at addr ffff8880a488d068 by task ksoftirqd/1/16

    CPU: 1 PID: 16 Comm: ksoftirqd/1 Not tainted 5.5.0-rc5-syzkaller #0
```

```
     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+0x197/0x210 lib/dump_stack.c:118
      print_address_description.constprop.0.cold+0xd4/0x30b mm/kasan/report.c:374
      __kasan_report.cold+0x1b/0x41 mm/kasan/report.c:506
      kasan_report+0x12/0x20 mm/kasan/common.c:639
      __asan_report_load4_noabort+0x14/0x20 mm/kasan/generic_report.c:134
      tcp_highest_sack_seq include/net/tcp.h:1864 [inline]
      tcp_highest_sack_seq include/net/tcp.h:1856 [inline]
      tcp_check_sack_reordering+0x33c/0x3a0 net/ipv4/tcp_input.c:891
      tcp_try_undo_partial net/ipv4/tcp_input.c:2730 [inline]
      tcp_fastretrans_alert+0xf74/0x23f0 net/ipv4/tcp_input.c:2847
      tcp_ack+0x2577/0x5bf0 net/ipv4/tcp_input.c:3710
      tcp_rcv_established+0x6dd/0x1e90 net/ipv4/tcp_input.c:5706
      tcp_v4_do_rcv+0x619/0x8d0 net/ipv4/tcp_ipv4.c:1619
      tcp_v4_rcv+0x307f/0x3b40 net/ipv4/tcp_ipv4.c:2001
      ip_protocol_deliver_rcu+0x5a/0x880 net/ipv4/ip_input.c:204
      ip_local_deliver_finish+0x23b/0x380 net/ipv4/ip_input.c:231
      NF_HOOK include/linux/netfilter.h:307 [inline]
      NF_HOOK include/linux/netfilter.h:301 [inline]
      ip_local_deliver+0x1e9/0x520 net/ipv4/ip_input.c:252
      dst_input include/net/dst.h:442 [inline]
      ip_rcv_finish+0x1db/0x2f0 net/ipv4/ip_input.c:428
      NF_HOOK include/linux/netfilter.h:307 [inline]
      NF_HOOK include/linux/netfilter.h:301 [inline]
      ip_rcv+0xe8/0x3f0 net/ipv4/ip_input.c:538
      __netif_receive_skb_one_core+0x113/0x1a0 net/core/dev.c:5148
      __netif_receive_skb+0x2c/0x1d0 net/core/dev.c:5262
      process_backlog+0x206/0x750 net/core/dev.c:6093
      napi_poll net/core/dev.c:6530 [inline]
      net_rx_action+0x508/0x1120 net/core/dev.c:6598
      __do_softirq+0x262/0x98c kernel/softirq.c:292
      run_ksoftirqd kernel/softirq.c:603 [inline]
      run_ksoftirqd+0x8e/0x110 kernel/softirq.c:595
      smpboot_thread_fn+0x6a3/0xa40 kernel/smpboot.c:165
      kthread+0x361/0x430 kernel/kthread.c:255
      ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:352

     Allocated by task 10091:
      save_stack+0x23/0x90 mm/kasan/common.c:72
      set_track mm/kasan/common.c:80 [inline]
      __kasan_kmalloc mm/kasan/common.c:513 [inline]
      __kasan_kmalloc.constprop.0+0xcf/0xe0 mm/kasan/common.c:486
      kasan_slab_alloc+0xf/0x20 mm/kasan/common.c:521
      slab_post_alloc_hook mm/slab.h:584 [inline]
      slab_alloc_node mm/slab.c:3263 [inline]
      kmem_cache_alloc_node+0x138/0x740 mm/slab.c:3575
      __alloc_skb+0xd5/0x5e0 net/core/skbuff.c:198
      alloc_skb_fclone include/linux/skbuff.h:1099 [inline]
      sk_stream_alloc_skb net/ipv4/tcp.c:875 [inline]
      sk_stream_alloc_skb+0x113/0xc90 net/ipv4/tcp.c:852
      tcp_sendmsg_locked+0xcf9/0x3470 net/ipv4/tcp.c:1282
      tcp_sendmsg+0x30/0x50 net/ipv4/tcp.c:1432
      inet_sendmsg+0x9e/0xe0 net/ipv4/af_inet.c:807
      sock_sendmsg_nosec net/socket.c:652 [inline]
      sock_sendmsg+0xd7/0x130 net/socket.c:672
      __sys_sendto+0x262/0x380 net/socket.c:1998
      __do_sys_sendto net/socket.c:2010 [inline]
      __se_sys_sendto net/socket.c:2006 [inline]
      __x64_sys_sendto+0xe1/0x1a0 net/socket.c:2006
      do_syscall_64+0xfa/0x790 arch/x86/entry/common.c:294
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
```

```
    Freed by task 10095:
     save_stack+0x23/0x90 mm/kasan/common.c:72
     set_track mm/kasan/common.c:80 [inline]
     kasan_set_free_info mm/kasan/common.c:335 [inline]
     __kasan_slab_free+0x102/0x150 mm/kasan/common.c:474
     kasan_slab_free+0xe/0x10 mm/kasan/common.c:483
     __cache_free mm/slab.c:3426 [inline]
     kmem_cache_free+0x86/0x320 mm/slab.c:3694
     kfree_skbmem+0x178/0x1c0 net/core/skbuff.c:645
     __kfree_skb+0x1e/0x30 net/core/skbuff.c:681
     sk_eat_skb include/net/sock.h:2453 [inline]
     tcp_recvmsg+0x1252/0x2930 net/ipv4/tcp.c:2166
     inet_recvmsg+0x136/0x610 net/ipv4/af_inet.c:838
     sock_recvmsg_nosec net/socket.c:886 [inline]
     sock_recvmsg net/socket.c:904 [inline]
     sock_recvmsg+0xce/0x110 net/socket.c:900
     __sys_recvfrom+0x1ff/0x350 net/socket.c:2055
     __do_sys_recvfrom net/socket.c:2073 [inline]
     __se_sys_recvfrom net/socket.c:2069 [inline]
     __x64_sys_recvfrom+0xe1/0x1a0 net/socket.c:2069
     do_syscall_64+0xfa/0x790 arch/x86/entry/common.c:294
     entry_SYSCALL_64_after_hwframe+0x49/0xbe

    The buggy address belongs to the object at ffff8880a488d040
     which belongs to the cache skbuff_fclone_cache of size 456
    The buggy address is located 40 bytes inside of
     456-byte region [ffff8880a488d040, ffff8880a488d208)
    The buggy address belongs to the page:
    page:ffffea0002922340 refcount:1 mapcount:0 mapping:ffff88821b057000 index:0x0
    raw: 00fffe0000000200 ffffea00022a5788 ffffea0002624a48 ffff88821b057000
    raw: 0000000000000000 ffff8880a488d040 0000000100000006 0000000000000000
    page dumped because: kasan: bad access detected

    Memory state around the buggy address:
     ffff8880a488cf00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     ffff8880a488cf80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
    >ffff8880a488d000: fc fc fc fc fc fc fc fc fb fb fb fb fb fb fb fb
                                                      ^
     ffff8880a488d080: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     ffff8880a488d100: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb

    Fixes: 853697504de0 ("tcp: Fix highest_sack and highest_sack_seq")
    Fixes: 50895b9de1d3 ("tcp: highest_sack fix")
    Fixes: 737ff314563c ("tcp: use sequence distance to detect reordering")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Cambda Zhu <cambda@linux.alibaba.com>
    Cc: Yuchung Cheng <ycheng@google.com>
    Cc: Neal Cardwell <ncardwell@google.com>
    Acked-by: Neal Cardwell <ncardwell@google.com>
    Acked-by: Yuchung Cheng <ycheng@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 562a7c35049c82f2422366f24c4e4a9520a5a450
Author: Wen Yang <wenyang@linux.alibaba.com>
Date:   Mon Jan 20 18:04:56 2020 +0800

    tcp_bbr: improve arithmetic division in bbr_update_bw()

    [ Upstream commit 5b2f1f3070b6447b76174ea8bfb7390dc6253ebd ]

    do_div() does a 64-by-32 division. Use div64_long() instead of it
    if the divisor is long, to avoid truncation to 32-bit.
    And as a nice side effect also cleans up the function a bit.
```

```
    Signed-off-by: Wen Yang <wenyang@linux.alibaba.com>
    Cc: Eric Dumazet <edumazet@google.com>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>
    Cc: Hideaki YOSHIFUJI <yoshfuji@linux-ipv6.org>
    Cc: netdev@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 41c230ae2aa4ff9ca33afef22e705d67d45b16f6
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Tue Jan 21 16:50:49 2020 +0100

    Revert "udp: do rmem bulk free even if the rx sk queue is empty"

    [ Upstream commit d39ca2590d10712f412add7a88e1dd467a7246f4 ]

    This reverts commit 0d4a6608f68c7532dcbfec2ea1150c9761767d03.

    Willem reported that after commit 0d4a6608f68c ("udp: do rmem bulk
    free even if the rx sk queue is empty") the memory allocated by
    an almost idle system with many UDP sockets can grow a lot.

    For stable kernel keep the solution as simple as possible and revert
    the offending commit.

    Reported-by: Willem de Bruijn <willemdebruijn.kernel@gmail.com>
    Diagnosed-by: Eric Dumazet <eric.dumazet@gmail.com>
    Fixes: 0d4a6608f68c ("udp: do rmem bulk free even if the rx sk queue is empty")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Acked-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 63635790a8911c1cc284508e8b9709cedf17cdb7
Author: James Hughes <james.hughes@raspberrypi.org>
Date:   Mon Jan 20 11:12:40 2020 +0000

    net: usb: lan78xx: Add .ndo_features_check

    [ Upstream commit ce896476c65d72b4b99fa09c2f33436b4198f034 ]

    As reported by Eric Dumazet, there are still some outstanding
    cases where the driver does not handle TSO correctly when skb's
    are over a certain size. Most cases have been fixed, this patch
    should ensure that forwarded SKB's that are greater than
    MAX_SINGLE_PACKET_SIZE - TX_OVERHEAD are software segmented
    and handled correctly.

    Signed-off-by: James Hughes <james.hughes@raspberrypi.org>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d6f7ed61fa6e64fd25b8488463fad96966d87497
Author: Jouni Hogander <jouni.hogander@unikie.com>
Date:   Mon Jan 20 09:51:03 2020 +0200

    net-sysfs: Fix reference count leak

    [ Upstream commit cb626bf566eb4433318d35681286c494f04fedcc ]

    Netdev_register_kobject is calling device_initialize. In case of error
```

```
    reference taken by device_initialize is not given up.

    Drivers are supposed to call free_netdev in case of error. In non-error
    case the last reference is given up there and device release sequence
    is triggered. In error case this reference is kept and the release
    sequence is never started.

    Fix this by setting reg_state as NETREG_UNREGISTERED if registering
    fails.

    This is the rootcause for couple of memory leaks reported by Syzkaller:

    BUG: memory leak unreferenced object 0xffff8880675ca008 (size 256):
      comm "netdev_register", pid 281, jiffies 4294696663 (age 6.808s)
      hex dump (first 32 bytes):
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
      backtrace:
        [<0000000058ca4711>] kmem_cache_alloc_trace+0x167/0x280
        [<000000002340019b>] device_add+0x882/0x1750
        [<000000001d588c3a>] netdev_register_kobject+0x128/0x380
        [<0000000011ef5535>] register_netdevice+0xa1b/0xf00
        [<000000007fcf1c99>] __tun_chr_ioctl+0x20d5/0x3dd0
        [<000000006a5b7b2b>] tun_chr_ioctl+0x2f/0x40
        [<00000000f30f834a>] do_vfs_ioctl+0x1c7/0x1510
        [<00000000fba062ea>] ksys_ioctl+0x99/0xb0
        [<00000000b1c1b8d2>] __x64_sys_ioctl+0x78/0xb0
        [<00000000984cabb9>] do_syscall_64+0x16f/0x580
        [<000000000bde033d>] entry_SYSCALL_64_after_hwframe+0x44/0xa9
        [<00000000e6ca2d9f>] 0xffffffffffffffff

    BUG: memory leak
    unreferenced object 0xffff8880668ba588 (size 8):
      comm "kobject_set_nam", pid 286, jiffies 4294725297 (age 9.871s)
      hex dump (first 8 bytes):
        6e 72 30 00 cc be df 2b                          nr0....+
      backtrace:
        [<00000000a322332a>] __kmalloc_track_caller+0x16e/0x290
        [<000000000236fd26b>] kstrdup+0x3e/0x70
        [<00000000dd4a2815>] kstrdup_const+0x3e/0x50
        [<0000000049a377fc>] kvasprintf_const+0x10e/0x160
        [<00000000627fc711>] kobject_set_name_vargs+0x5b/0x140
        [<0000000019eeab06>] dev_set_name+0xc0/0xf0
        [<0000000069cb12bc>] netdev_register_kobject+0xc8/0x320
        [<00000000f2e83732>] register_netdevice+0xa1b/0xf00
        [<000000009e1f57cc>] __tun_chr_ioctl+0x20d5/0x3dd0
        [<000000009c560784>] tun_chr_ioctl+0x2f/0x40
        [<000000000d759e02>] do_vfs_ioctl+0x1c7/0x1510
        [<00000000351d7c31>] ksys_ioctl+0x99/0xb0
        [<000000008390040a>] __x64_sys_ioctl+0x78/0xb0
        [<0000000052d196b7>] do_syscall_64+0x16f/0x580
        [<0000000019af9236>] entry_SYSCALL_64_after_hwframe+0x44/0xa9
        [<00000000bc384531>] 0xffffffffffffffff

    v3 -> v4:
      Set reg_state to NETREG_UNREGISTERED if registering fails

    v2 -> v3:
    * Replaced BUG_ON with WARN_ON in free_netdev and netdev_release

    v1 -> v2:
    * Relying on driver calling free_netdev rather than calling
      put_device directly in error path

    Reported-by: syzbot+ad8ca40ecd77896d51e2@syzkaller.appspotmail.com
```

```
    Cc: David Miller <davem@davemloft.net>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Lukas Bulwahn <lukas.bulwahn@gmail.com>
    Signed-off-by: Jouni Hogander <jouni.hogander@unikie.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9b60a32108e2ec263b5c27b2396e92ee8d773f3b
Author: Eric Dumazet <edumazet@google.com>
Date:   Tue Jan 21 11:02:20 2020 -0800

    net_sched: use validated TCA_KIND attribute in tc_new_tfilter()

    [ Upstream commit 36d79af7fb59d6d9106feb9c1855eb93d6d53fe6 ]

    sysbot found another issue in tc_new_tfilter().
    We probably should use @name which contains the sanitized
    version of TCA_KIND.

    BUG: KMSAN: uninit-value in string_nocheck lib/vsprintf.c:608 [inline]
    BUG: KMSAN: uninit-value in string+0x522/0x690 lib/vsprintf.c:689
    CPU: 1 PID: 10753 Comm: syz-executor.1 Not tainted 5.5.0-rc5-syzkaller #0
    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+0x1c9/0x220 lib/dump_stack.c:118
     kmsan_report+0xf7/0x1e0 mm/kmsan/kmsan_report.c:118
     __msan_warning+0x58/0xa0 mm/kmsan/kmsan_instr.c:215
     string_nocheck lib/vsprintf.c:608 [inline]
     string+0x522/0x690 lib/vsprintf.c:689
     vsnprintf+0x207d/0x31b0 lib/vsprintf.c:2574
     __request_module+0x2ad/0x11c0 kernel/kmod.c:143
     tcf_proto_lookup_ops+0x241/0x720 net/sched/cls_api.c:139
     tcf_proto_create net/sched/cls_api.c:262 [inline]
     tc_new_tfilter+0x2a4e/0x5010 net/sched/cls_api.c:2058
     rtnetlink_rcv_msg+0xcb7/0x1570 net/core/rtnetlink.c:5415
     netlink_rcv_skb+0x451/0x650 net/netlink/af_netlink.c:2477
     rtnetlink_rcv+0x50/0x60 net/core/rtnetlink.c:5442
     netlink_unicast_kernel net/netlink/af_netlink.c:1302 [inline]
     netlink_unicast+0xf9e/0x1100 net/netlink/af_netlink.c:1328
     netlink_sendmsg+0x1248/0x14d0 net/netlink/af_netlink.c:1917
     sock_sendmsg_nosec net/socket.c:639 [inline]
     sock_sendmsg net/socket.c:659 [inline]
     ____sys_sendmsg+0x12b6/0x1350 net/socket.c:2330
     ___sys_sendmsg net/socket.c:2384 [inline]
     __sys_sendmsg+0x451/0x5f0 net/socket.c:2417
     __do_sys_sendmsg net/socket.c:2426 [inline]
     __se_sys_sendmsg+0x97/0xb0 net/socket.c:2424
     __x64_sys_sendmsg+0x4a/0x70 net/socket.c:2424
     do_syscall_64+0xb8/0x160 arch/x86/entry/common.c:296
     entry_SYSCALL_64_after_hwframe+0x44/0xa9
    RIP: 0033:0x45b349
    Code: ad b6 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 7b b6 fb ff c3 66 2e
0f 1f 84 00 00 00 00
    RSP: 002b:00007f88b3948c78 EFLAGS: 00000246 ORIG_RAX: 000000000000002e
    RAX: ffffffffffffffda RBX: 00007f88b39496d4 RCX: 000000000045b349
    RDX: 0000000000000000 RSI: 00000000200001c0 RDI: 0000000000000003
    RBP: 000000000075bfc8 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000246 R12: 00000000ffffffff
    R13: 000000000000099f R14: 00000000004cb163 R15: 000000000075bfd4

    Uninit was created at:
     kmsan_save_stack_with_flags mm/kmsan/kmsan.c:144 [inline]
```

```
      kmsan_internal_poison_shadow+0x66/0xd0 mm/kmsan/kmsan.c:127
      kmsan_slab_alloc+0x8a/0xe0 mm/kmsan/kmsan_hooks.c:82
      slab_alloc_node mm/slub.c:2774 [inline]
      __kmalloc_node_track_caller+0xb40/0x1200 mm/slub.c:4382
      __kmalloc_reserve net/core/skbuff.c:141 [inline]
      __alloc_skb+0x2fd/0xac0 net/core/skbuff.c:209
      alloc_skb include/linux/skbuff.h:1049 [inline]
      netlink_alloc_large_skb net/netlink/af_netlink.c:1174 [inline]
      netlink_sendmsg+0x7d3/0x14d0 net/netlink/af_netlink.c:1892
      sock_sendmsg_nosec net/socket.c:639 [inline]
      sock_sendmsg net/socket.c:659 [inline]
      ____sys_sendmsg+0x12b6/0x1350 net/socket.c:2330
      ___sys_sendmsg net/socket.c:2384 [inline]
      __sys_sendmsg+0x451/0x5f0 net/socket.c:2417
      __do_sys_sendmsg net/socket.c:2426 [inline]
      __se_sys_sendmsg+0x97/0xb0 net/socket.c:2424
      __x64_sys_sendmsg+0x4a/0x70 net/socket.c:2424
      do_syscall_64+0xb8/0x160 arch/x86/entry/common.c:296
      entry_SYSCALL_64_after_hwframe+0x44/0xa9

    Fixes: 6f96c3c6904c ("net_sched: fix backward compatibility for TCA_KIND")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Cc: Cong Wang <xiyou.wangcong@gmail.com>
    Cc: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Cc: Jamal Hadi Salim <jhs@mojatatu.com>
    Cc: Jiri Pirko <jiri@resnulli.us>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b6a9a954b3fea24386b46be1bf9f5b0ba32470db
Author: Cong Wang <xiyou.wangcong@gmail.com>
Date:   Wed Jan 22 15:42:02 2020 -0800

    net_sched: fix datalen for ematch

    [ Upstream commit 61678d28d4a45ef376f5d02a839cc37509ae9281 ]

    syzbot reported an out-of-bound access in em_nbyte. As initially
    analyzed by Eric, this is because em_nbyte sets its own em->datalen
    in em_nbyte_change() other than the one specified by user, but this
    value gets overwritten later by its caller tcf_em_validate().
    We should leave em->datalen untouched to respect their choices.

    I audit all the in-tree ematch users, all of those implement
    ->change() set em->datalen, so we can just avoid setting it twice
    in this case.

    Reported-and-tested-by: syzbot+5af9a90dad568aa9f611@syzkaller.appspotmail.com
    Reported-by: syzbot+2f07903a5b05e7f36410@syzkaller.appspotmail.com
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Cc: Eric Dumazet <eric.dumazet@gmail.com>
    Signed-off-by: Cong Wang <xiyou.wangcong@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 888934af187cd7168545711027db64006e7dad9d
Author: Eric Dumazet <edumazet@google.com>
Date:   Tue Jan 21 22:47:29 2020 -0800

    net: rtnetlink: validate IFLA_MTU attribute in rtnl_create_link()

    [ Upstream commit d836f5c69d87473ff65c06a6123e5b2cf5e56f5b ]
```

```
    rtnl_create_link() needs to apply dev->min_mtu and dev->max_mtu
    checks that we apply in do_setlink()

    Otherwise malicious users can crash the kernel, for example after
    an integer overflow :

    BUG: KASAN: use-after-free in memset include/linux/string.h:365 [inline]
    BUG: KASAN: use-after-free in __alloc_skb+0x37b/0x5e0 net/core/skbuff.c:238
    Write of size 32 at addr ffff88819f20b9c0 by task swapper/0/0

    CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.5.0-rc1-syzkaller #0
    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+0x197/0x210 lib/dump_stack.c:118
     print_address_description.constprop.0.cold+0xd4/0x30b mm/kasan/report.c:374
     __kasan_report.cold+0x1b/0x41 mm/kasan/report.c:506
     kasan_report+0x12/0x20 mm/kasan/common.c:639
     check_memory_region_inline mm/kasan/generic.c:185 [inline]
     check_memory_region+0x134/0x1a0 mm/kasan/generic.c:192
     memset+0x24/0x40 mm/kasan/common.c:108
     memset include/linux/string.h:365 [inline]
     __alloc_skb+0x37b/0x5e0 net/core/skbuff.c:238
     alloc_skb include/linux/skbuff.h:1049 [inline]
     alloc_skb_with_frags+0x93/0x590 net/core/skbuff.c:5664
     sock_alloc_send_pskb+0x7ad/0x920 net/core/sock.c:2242
     sock_alloc_send_skb+0x32/0x40 net/core/sock.c:2259
     mld_newpack+0x1d7/0x7f0 net/ipv6/mcast.c:1609
     add_grhead.isra.0+0x299/0x370 net/ipv6/mcast.c:1713
     add_grec+0x7db/0x10b0 net/ipv6/mcast.c:1844
     mld_send_cr net/ipv6/mcast.c:1970 [inline]
     mld_ifc_timer_expire+0x3d3/0x950 net/ipv6/mcast.c:2477
     call_timer_fn+0x1ac/0x780 kernel/time/timer.c:1404
     expire_timers kernel/time/timer.c:1449 [inline]
     __run_timers kernel/time/timer.c:1773 [inline]
     __run_timers kernel/time/timer.c:1740 [inline]
     run_timer_softirq+0x6c3/0x1790 kernel/time/timer.c:1786
     __do_softirq+0x262/0x98c kernel/softirq.c:292
     invoke_softirq kernel/softirq.c:373 [inline]
     irq_exit+0x19b/0x1e0 kernel/softirq.c:413
     exiting_irq arch/x86/include/asm/apic.h:536 [inline]
     smp_apic_timer_interrupt+0x1a3/0x610 arch/x86/kernel/apic/apic.c:1137
     apic_timer_interrupt+0xf/0x20 arch/x86/entry/entry_64.S:829
     </IRQ>
    RIP: 0010:native_safe_halt+0xe/0x10 arch/x86/include/asm/irqflags.h:61
    Code: 98 6b ea f9 eb 8a cc cc cc cc cc cc e9 07 00 00 00 0f 00 2d 44 1c 60 00 f4 c3 66
90 e9 07 00 00 00 0f 00 2d 34 1c 60 00 fb f4 <c3> cc 55 48 89 e5 41 57 41 56 41 55 41 54 53
e8 4e 5d 9a f9 e8 79
    RSP: 0018:ffffffff89807ce8 EFLAGS: 00000286 ORIG_RAX: ffffffffffffff13
    RAX: 1ffffffff13266ae RBX: ffffffff8987a1c0 RCX: 0000000000000000
    RDX: dffffc0000000000 RSI: 0000000000000006 RDI: ffffffff8987aa54
    RBP: ffffffff89807d18 R08: ffffffff8987a1c0 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000000 R12: dffffc0000000000
    R13: ffffffff8a799980 R14: 0000000000000000 R15: 0000000000000000
     arch_cpu_idle+0xa/0x10 arch/x86/kernel/process.c:690
     default_idle_call+0x84/0xb0 kernel/sched/idle.c:94
     cpuidle_idle_call kernel/sched/idle.c:154 [inline]
     do_idle+0x3c8/0x6e0 kernel/sched/idle.c:269
     cpu_startup_entry+0x1b/0x20 kernel/sched/idle.c:361
     rest_init+0x23b/0x371 init/main.c:451
     arch_call_rest_init+0xe/0x1b
     start_kernel+0x904/0x943 init/main.c:784
     x86_64_start_reservations+0x29/0x2b arch/x86/kernel/head64.c:490
```

```
      x86_64_start_kernel+0x77/0x7b arch/x86/kernel/head64.c:471
      secondary_startup_64+0xa4/0xb0 arch/x86/kernel/head_64.S:242

    The buggy address belongs to the page:
    page:ffffea00067c82c0 refcount:0 mapcount:0 mapping:0000000000000000 index:0x0
    raw: 057ffe0000000000 ffffea00067c82c8 ffffea00067c82c8 0000000000000000
    raw: 0000000000000000 0000000000000000 00000000ffffffff 0000000000000000
    page dumped because: kasan: bad access detected

    Memory state around the buggy address:
     ffff88819f20b880: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
     ffff88819f20b900: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
    >ffff88819f20b980: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
                                           ^
     ffff88819f20ba00: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
     ffff88819f20ba80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

    Fixes: 61e84623ace3 ("net: centralize net_device min/max MTU checking")
    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 500869d71822a4e05cef536f01b9005395369e65
Author: William Dauchy <w.dauchy@criteo.com>
Date:   Tue Jan 21 15:26:24 2020 +0100

    net, ip_tunnel: fix namespaces move

    [ Upstream commit d0f418516022c32ecceaf4275423e5bd3f8743a9 ]

    in the same manner as commit 690afc165bb3 ("net: ip6_gre: fix moving
    ip6gre between namespaces"), fix namespace moving as it was broken since
    commit 2e15ea390e6f ("ip_gre: Add support to collect tunnel metadata.").
    Indeed, the ip6_gre commit removed the local flag for collect_md
    condition, so there is no reason to keep it for ip_gre/ip_tunnel.

    this patch will fix both ip_tunnel and ip_gre modules.

    Fixes: 2e15ea390e6f ("ip_gre: Add support to collect tunnel metadata.")
    Signed-off-by: William Dauchy <w.dauchy@criteo.com>
    Acked-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ead0377f601ec3aae7965fb16da28685963a097b
Author: William Dauchy <w.dauchy@criteo.com>
Date:   Tue Jan 21 21:49:54 2020 +0100

    net, ip6_tunnel: fix namespaces move

    [ Upstream commit 5311a69aaca30fa849c3cc46fb25f75727fb72d0 ]

    in the same manner as commit d0f418516022 ("net, ip_tunnel: fix
    namespaces move"), fix namespace moving as it was broken since commit
    8d79266bc48c ("ip6_tunnel: add collect_md mode to IPv6 tunnel"), but for
    ipv6 this time; there is no reason to keep it for ip6_tunnel.

    Fixes: 8d79266bc48c ("ip6_tunnel: add collect_md mode to IPv6 tunnel")
    Signed-off-by: William Dauchy <w.dauchy@criteo.com>
    Acked-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7943bb0f06365cf5e32f3cf8a6b29eeae981fb8a
```

```
Author: Niko Kortstrom <niko.kortstrom@nokia.com>
Date:   Thu Jan 16 11:43:27 2020 +0200

    net: ip6_gre: fix moving ip6gre between namespaces

    [ Upstream commit 690afc165bb314354667f67157c1a1aea7dc797a ]

    Support for moving IPv4 GRE tunnels between namespaces was added in
    commit b57708add314 ("gre: add x-netns support"). The respective change
    for IPv6 tunnels, commit 22f08069e8b4 ("ip6gre: add x-netns support")
    did not drop NETIF_F_NETNS_LOCAL flag so moving them from one netns to
    another is still denied in IPv6 case. Drop NETIF_F_NETNS_LOCAL flag from
    ip6gre tunnels to allow moving ip6gre tunnel endpoints between network
    namespaces.

    Signed-off-by: Niko Kortstrom <niko.kortstrom@nokia.com>
    Acked-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Acked-by: William Tu <u9012063@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c6e524f4fad3e25e18e14a02cbd2f325d6375558
Author: Michael Ellerman <mpe@ellerman.id.au>
Date:   Fri Jan 24 20:41:44 2020 +1100

    net: cxgb3_main: Add CAP_NET_ADMIN check to CHELSIO_GET_MEM

    [ Upstream commit 3546d8f1bbe992488ed91592cf6bf76e7114791a =

    The cxgb3 driver for "Chelsio T3-based gigabit and 10Gb Ethernet
    adapters" implements a custom ioctl as SIOCCHIOCTL/SIOCDEVPRIVATE in
    cxgb_extension_ioctl().

    One of the subcommands of the ioctl is CHELSIO_GET_MEM, which appears
    to read memory directly out of the adapter and return it to userspace.
    It's not entirely clear what the contents of the adapter memory
    contains, but the assumption is that it shouldn't be accessible to all
    users.

    So add a CAP_NET_ADMIN check to the CHELSIO_GET_MEM case. Put it after
    the is_offload() check, which matches two of the other subcommands in
    the same function which also check for is_offload() and CAP_NET_ADMIN.

    Found by Ilja by code inspection, not tested as I don't have the
    required hardware.

    Reported-by: Ilja Van Sprundel <ivansprundel@ioactive.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 32df8d0bf461586b57a7ae3d3c0714584b03e7e6
Author: Florian Fainelli <f.fainelli@gmail.com>
Date:   Thu Jan 23 09:49:34 2020 -0800

    net: bcmgenet: Use netif_tx_napi_add() for TX NAPI

    [ Upstream commit 148965df1a990af98b2c84092c2a2274c7489284 ]

    Before commit 7587935cfa11 ("net: bcmgenet: move NAPI initialization to
    ring initialization") moved the code, this used to be
    netif_tx_napi_add(), but we lost that small semantic change in the
    process, restore that.

    Fixes: 7587935cfa11 ("net: bcmgenet: move NAPI initialization to ring initialization")
```

```
    Signed-off-by: Florian Fainelli <f.fainelli@gmail.com>
    Acked-by: Doug Berger <opendmb@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e2bd33937c507df324176da01a99448bf3864b4c
Author: Yuki Taguchi <tagyounit@gmail.com>
Date:   Mon Jan 20 13:48:37 2020 +0900

    ipv6: sr: remove SKB_GSO_IPXIP6 on End.D* actions

    [ Upstream commit 62ebaeaedee7591c257543d040677a60e35c7aec ]

    After LRO/GRO is applied, SRv6 encapsulated packets have
    SKB_GSO_IPXIP6 feature flag, and this flag must be removed right after
    decapulation procedure.

    Currently, SKB_GSO_IPXIP6 flag is not removed on End.D* actions, which
    creates inconsistent packet state, that is, a normal TCP/IP packets
    have the SKB_GSO_IPXIP6 flag. This behavior can cause unexpected
    fallback to GSO on routing to netdevices that do not support
    SKB_GSO_IPXIP6. For example, on inter-VRF forwarding, decapsulated
    packets separated into small packets by GSO because VRF devices do not
    support TSO for packets with SKB_GSO_IPXIP6 flag, and this degrades
    forwarding performance.

    This patch removes encapsulation related GSO flags from the skb right
    after the End.D* action is applied.

    Fixes: d7a669dd2f8b ("ipv6: sr: add helper functions for seg6local")
    Signed-off-by: Yuki Taguchi <tagyounit@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6cbb9683d7437f9c1ae96a7a8e3add38e0ed2560
Author: Eric Dumazet <edumazet@google.com>
Date:   Tue Jan 21 23:17:14 2020 -0800

    gtp: make sure only SOCK_DGRAM UDP sockets are accepted

    [ Upstream commit 940ba14986657a50c15f694efca1beba31fa568f ]

    A malicious user could use RAW sockets and fool
    GTP using them as standard SOCK_DGRAM UDP sockets.

    BUG: KMSAN: uninit-value in udp_tunnel_encap_enable include/net/udp_tunnel.h:174
[inline]
    BUG: KMSAN: uninit-value in setup_udp_tunnel_sock+0x45e/0x6f0 net/ipv4/udp_tunnel.c:85
    CPU: 0 PID: 11262 Comm: syz-executor613 Not tainted 5.5.0-rc5-syzkaller #0
    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+0x1c9/0x220 lib/dump_stack.c:118
     kmsan_report+0xf7/0x1e0 mm/kmsan/kmsan_report.c:118
     __msan_warning+0x58/0xa0 mm/kmsan/kmsan_instr.c:215
     udp_tunnel_encap_enable include/net/udp_tunnel.h:174 [inline]
     setup_udp_tunnel_sock+0x45e/0x6f0 net/ipv4/udp_tunnel.c:85
     gtp_encap_enable_socket+0x37f/0x5a0 drivers/net/gtp.c:827
     gtp_encap_enable drivers/net/gtp.c:844 [inline]
     gtp_newlink+0xfb/0x1e50 drivers/net/gtp.c:666
     __rtnl_newlink net/core/rtnetlink.c:3305 [inline]
     rtnl_newlink+0x2973/0x3920 net/core/rtnetlink.c:3363
     rtnetlink_rcv_msg+0x1153/0x1570 net/core/rtnetlink.c:5424
     netlink_rcv_skb+0x451/0x650 net/netlink/af_netlink.c:2477
```

```
      rtnetlink_rcv+0x50/0x60 net/core/rtnetlink.c:5442
      netlink_unicast_kernel net/netlink/af_netlink.c:1302 [inline]
      netlink_unicast+0xf9e/0x1100 net/netlink/af_netlink.c:1328
      netlink_sendmsg+0x1248/0x14d0 net/netlink/af_netlink.c:1917
      sock_sendmsg_nosec net/socket.c:639 [inline]
      sock_sendmsg net/socket.c:659 [inline]
      ____sys_sendmsg+0x12b6/0x1350 net/socket.c:2330
      ___sys_sendmsg net/socket.c:2384 [inline]
      __sys_sendmsg+0x451/0x5f0 net/socket.c:2417
      __do_sys_sendmsg net/socket.c:2426 [inline]
      __se_sys_sendmsg+0x97/0xb0 net/socket.c:2424
      __x64_sys_sendmsg+0x4a/0x70 net/socket.c:2424
      do_syscall_64+0xb8/0x160 arch/x86/entry/common.c:296
      entry_SYSCALL_64_after_hwframe+0x44/0xa9
    RIP: 0033:0x441359
    Code: e8 ac e8 ff ff 48 83 c4 18 c3 0f 1f 80 00 00 00 00 48 89 f8 48 89 f7 48 89 d6 48
 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 0f 83 eb 08 fc ff c3 66 2e
 0f 1f 84 00 00 00 00
    RSP: 002b:00007fff1cd0ac28 EFLAGS: 00000246 ORIG_RAX: 000000000000002e
    RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 0000000000441359
    RDX: 0000000000000000 RSI: 0000000020000100 RDI: 0000000000000003
    RBP: 00000000006cb018 R08: 00000000004002c8 R09: 00000000004002c8
    R10: 00000000004002c8 R11: 0000000000000246 R12: 00000000004020d0
    R13: 0000000000402160 R14: 0000000000000000 R15: 0000000000000000

    Uninit was created at:
      kmsan_save_stack_with_flags+0x3c/0x90 mm/kmsan/kmsan.c:144
      kmsan_internal_alloc_meta_for_pages mm/kmsan/kmsan_shadow.c:307 [inline]
      kmsan_alloc_page+0x12a/0x310 mm/kmsan/kmsan_shadow.c:336
      __alloc_pages_nodemask+0x57f2/0x5f60 mm/page_alloc.c:4800
      alloc_pages_current+0x67d/0x990 mm/mempolicy.c:2207
      alloc_pages include/linux/gfp.h:534 [inline]
      alloc_slab_page+0x111/0x12f0 mm/slub.c:1511
      allocate_slab mm/slub.c:1656 [inline]
      new_slab+0x2bc/0x1130 mm/slub.c:1722
      new_slab_objects mm/slub.c:2473 [inline]
      ___slab_alloc+0x1533/0x1f30 mm/slub.c:2624
      __slab_alloc mm/slub.c:2664 [inline]
      slab_alloc_node mm/slub.c:2738 [inline]
      slab_alloc mm/slub.c:2783 [inline]
      kmem_cache_alloc+0xb23/0xd70 mm/slub.c:2788
      sk_prot_alloc+0xf2/0x620 net/core/sock.c:1597
      sk_alloc+0xf0/0xbe0 net/core/sock.c:1657
      inet_create+0x7c7/0x1370 net/ipv4/af_inet.c:321
      __sock_create+0x8eb/0xf00 net/socket.c:1420
      sock_create net/socket.c:1471 [inline]
      __sys_socket+0x1a1/0x600 net/socket.c:1513
      __do_sys_socket net/socket.c:1522 [inline]
      __se_sys_socket+0x8d/0xb0 net/socket.c:1520
      __x64_sys_socket+0x4a/0x70 net/socket.c:1520
      do_syscall_64+0xb8/0x160 arch/x86/entry/common.c:296
      entry_SYSCALL_64_after_hwframe+0x44/0xa9

    Fixes: 459aa660eb1d ("gtp: add initial driver for datapath of GPRS Tunneling Protocol
 (GTP-U)")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Pablo Neira <pablo@netfilter.org>
    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 80a4c81c67918f8f879b8ca63c0301e22b7e8b7a
 Author: Wenwen Wang <wenwen@cs.uga.edu>
 Date:   Sat Jan 25 14:33:29 2020 +0000
```

```
    firestream: fix memory leaks

    [ Upstream commit fa865ba183d61c1ec8cbcab8573159c3b72b89a4 ]

    In fs_open(), 'vcc' is allocated through kmalloc() and assigned to
    'atm_vcc->dev_data.' In the following execution, if an error occurs, e.g.,
    there is no more free channel, an error code EBUSY or ENOMEM will be
    returned. However, 'vcc' is not deallocated, leading to memory leaks. Note
    that, in normal cases where fs_open() returns 0, 'vcc' will be deallocated
    in fs_close(). But, if fs_open() fails, there is no guarantee that
    fs_close() will be invoked.

    To fix this issue, deallocate 'vcc' before the error code is returned.

    Signed-off-by: Wenwen Wang <wenwen@cs.uga.edu>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 34545cad8e0476aa6843f132e1177fe1517b2814
Author: Richard Palethorpe <rpalethorpe@suse.com>
Date:   Tue Jan 21 14:42:58 2020 +0100

    can, slip: Protect tty->disc_data in write_wakeup and close with RCU

    [ Upstream commit 0ace17d56824165c7f4c68785d6b58971db954dd ]

    write_wakeup can happen in parallel with close/hangup where tty->disc_data
    is set to NULL and the netdevice is freed thus also freeing
    disc_data. write_wakeup accesses disc_data so we must prevent close from
    freeing the netdev while write_wakeup has a non-NULL view of
    tty->disc_data.

    We also need to make sure that accesses to disc_data are atomic. Which can
    all be done with RCU.

    This problem was found by Syzkaller on SLCAN, but the same issue is
    reproducible with the SLIP line discipline using an LTP test based on the
    Syzkaller reproducer.

    A fix which didn't use RCU was posted by Hillf Danton.

    Fixes: 661f7fda21b1 ("slip: Fix deadlock in write_wakeup")
    Fixes: a8e83b17536a ("slcan: Port write_wakeup deadlock fix from slip")
    Reported-by: syzbot+017e491ae13c0068598a@syzkaller.appspotmail.com
    Signed-off-by: Richard Palethorpe <rpalethorpe@suse.com>
    Cc: Wolfgang Grandegger <wg@grandegger.com>
    Cc: Marc Kleine-Budde <mkl@pengutronix.de>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: Tyler Hall <tylerwhall@gmail.com>
    Cc: linux-can@vger.kernel.org
    Cc: netdev@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Cc: syzkaller@googlegroups.com
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```