```
commit f7d5b3dc4792a5fe0a4d6b8106a8f3eb20c3c24c
Author: Sasha Levin <sashal@kernel.org>
Date:   Sun Aug 25 10:13:54 2019 -0400

    Linux 5.2.10

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

commit cabd470b9e137c8f8040b67fe26815eedf6d0d48
Author: David Ahern <dsahern@gmail.com>
Date:   Mon Aug 12 13:07:07 2019 -0700

    netlink: Fix nlmsg_parse as a wrapper for strict message parsing

    [ Upstream commit d00ee64e1dcf09b3afefd1340f3e9eb637272714 ]

    Eric reported a syzbot warning:

    BUG: KMSAN: uninit-value in nh_valid_get_del_req+0x6f1/0x8c0 net/ipv4/nexthop.c:1510
    CPU: 0 PID: 11812 Comm: syz-executor444 Not tainted 5.3.0-rc3+ #17
    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+0x191/0x1f0 lib/dump_stack.c:113
     kmsan_report+0x162/0x2d0 mm/kmsan/kmsan_report.c:109
     __msan_warning+0x75/0xe0 mm/kmsan/kmsan_instr.c:294
     nh_valid_get_del_req+0x6f1/0x8c0 net/ipv4/nexthop.c:1510
     rtm_del_nexthop+0x1b1/0x610 net/ipv4/nexthop.c:1543
     rtnetlink_rcv_msg+0x115a/0x1580 net/core/rtnetlink.c:5223
     netlink_rcv_skb+0x431/0x620 net/netlink/af_netlink.c:2477
     rtnetlink_rcv+0x50/0x60 net/core/rtnetlink.c:5241
     netlink_unicast_kernel net/netlink/af_netlink.c:1302 [inline]
     netlink_unicast+0xf6c/0x1050 net/netlink/af_netlink.c:1328
     netlink_sendmsg+0x110f/0x1330 net/netlink/af_netlink.c:1917
     sock_sendmsg_nosec net/socket.c:637 [inline]
     sock_sendmsg net/socket.c:657 [inline]
     ___sys_sendmsg+0x14ff/0x1590 net/socket.c:2311
     __sys_sendmmsg+0x53a/0xae0 net/socket.c:2413
     __do_sys_sendmmsg net/socket.c:2442 [inline]
     __se_sys_sendmmsg+0xbd/0xe0 net/socket.c:2439
     __x64_sys_sendmmsg+0x56/0x70 net/socket.c:2439
     do_syscall_64+0xbc/0xf0 arch/x86/entry/common.c:297
     entry_SYSCALL_64_after_hwframe+0x63/0xe7

    The root cause is nlmsg_parse calling __nla_parse which means the
    header struct size is not checked.

    nlmsg_parse should be a wrapper around __nlmsg_parse with
    NL_VALIDATE_STRICT for the validate argument very much like
    nlmsg_parse_deprecated is for NL_VALIDATE_LIBERAL.

    Fixes: 3de6440354465 ("netlink: re-add parse/validate functions in strict mode")
    Reported-by: Eric Dumazet <edumazet@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: David Ahern <dsahern@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9a31192e1ca12e5e7c80c3600ede8c660728c8f4
Author: Heiner Kallweit <hkallweit1@gmail.com>
Date:   Mon Aug 12 21:20:02 2019 +0200
```

```
    net: phy: consider AN_RESTART status when reading link status

    [ Upstream commit c36757eb9dee13681227ad3676d37f14b3a2b2af ]

    After configuring and restarting aneg we immediately try to read the
    link status. On some systems the PHY may not yet have cleared the
    "aneg complete" and "link up" bits, resulting in a false link-up
    signal. See [0] for a report.
    Clause 22 and 45 both require the PHY to keep the AN_RESTART
    bit set until the PHY actually starts auto-negotiation.
    Let's consider this in the generic functions for reading link status.
    The commit marked as fixed is the first one where the patch applies
    cleanly.

    [0] https://marc.info/?t=156518400300003&r=1&w=2

    Fixes: c1164bb1a631 ("net: phy: check PMAPMD link status only in genphy_c45_read_link")
    Tested-by: Yonglong Liu <liuyonglong@huawei.com>
    Signed-off-by: Heiner Kallweit <hkallweit1@gmail.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bc110443a73f275de83d691bccbba9ecd25ca9be
Author: Jakub Kicinski <jakub.kicinski@netronome.com>
Date:   Wed Aug 7 17:03:59 2019 -0700

    net/tls: prevent skb_orphan() from leaking TLS plain text with offload

    [ Upstream commit 414776621d1006e57e80e6db7fdc3837897aaa64 ]

    sk_validate_xmit_skb() and drivers depend on the sk member of
    struct sk_buff to identify segments requiring encryption.
    Any operation which removes or does not preserve the original TLS
    socket such as skb_orphan() or skb_clone() will cause clear text
    leaks.

    Make the TCP socket underlying an offloaded TLS connection
    mark all skbs as decrypted, if TLS TX is in offload mode.
    Then in sk_validate_xmit_skb() catch skbs which have no socket
    (or a socket with no validation) and decrypted flag set.

    Note that CONFIG_SOCK_VALIDATE_XMIT, CONFIG_TLS_DEVICE and
    sk->sk_validate_xmit_skb are slightly interchangeable right now,
    they all imply TLS offload. The new checks are guarded by
    CONFIG_TLS_DEVICE because that's the option guarding the
    sk_buff->decrypted member.

    Second, smaller issue with orphaning is that it breaks
    the guarantee that packets will be delivered to device
    queues in-order. All TLS offload drivers depend on that
    scheduling property. This means skb_orphan_partial()'s
    trick of preserving partial socket references will cause
    issues in the drivers. We need a full orphan, and as a
    result netem delay/throttling will cause all TLS offload
    skbs to be dropped.

    Reusing the sk_buff->decrypted flag also protects from
    leaking clear text when incoming, decrypted skb is redirected
    (e.g. by TC).

    See commit 0608c69c9a80 ("bpf: sk_msg, sock{map|hash} redirect
    through ULP") for justification why the internal flag is safe.
    The only location which could leak the flag in is tcp_bpf_sendmsg(),
    which is taken care of by clearing the previously unused bit.
```

```
    v2:
     - remove superfluous decrypted mark copy (Willem);
     - remove the stale doc entry (Boris);
     - rely entirely on EOR marking to prevent coalescing (Boris);
     - use an internal sendpages flag instead of marking the socket
       (Boris).
    v3 (Willem):
     - reorganize the can_skb_orphan_partial() condition;
     - fix the flag leak-in through tcp_bpf_sendmsg.

    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Acked-by: Willem de Bruijn <willemb@google.com>
    Reviewed-by: Boris Pismenny <borisp@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bfdbef8aca0339bc8b3428d17790e2e87cb9036a
Author: Maxim Mikityanskiy <maximmi@mellanox.com>
Date:   Fri Jul 5 17:59:28 2019 +0300

    net/mlx5e: Use flow keys dissector to parse packets for ARFS

    [ Upstream commit 405b93eb764367a670e729da18e54dc42db32620 ]

    The current ARFS code relies on certain fields to be set in the SKB
    (e.g. transport_header) and extracts IP addresses and ports by custom
    code that parses the packet. The necessary SKB fields, however, are not
    always set at that point, which leads to an out-of-bounds access. Use
    skb_flow_dissect_flow_keys() to get the necessary information reliably,
    fix the out-of-bounds access and reuse the code.

    Fixes: 18c908e477dc ("net/mlx5e: Add accelerated RFS support")
    Signed-off-by: Maxim Mikityanskiy <maximmi@mellanox.com>
    Reviewed-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7c682c96051f209c3c2a420c4e086d05c085be22
Author: Aya Levin <ayal@mellanox.com>
Date:   Thu Aug 8 15:55:48 2019 +0300

    net/mlx5e: Remove redundant check in CQE recovery flow of tx reporter

    [ Upstream commit a4e508cab623951dc4754f346e5673714f3bbade ]

    Remove check of recovery bit, in the beginning of the CQE recovery
    function. This test is already performed right before the reporter
    is invoked, when CQE error is detected.

    Fixes: de8650a82071 ("net/mlx5e: Add tx reporter support")
    Signed-off-by: Aya Levin <ayal@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0fd12075148352e3948819cc96c3f3f262b0d075
Author: Aya Levin <ayal@mellanox.com>
Date:   Wed Aug 7 15:59:06 2019 +0300

    net/mlx5e: Fix false negative indication on tx reporter CQE recovery

    [ Upstream commit d9a2fcf53c76a7edb2bcf99e94507935561a83d5 ]

    Remove wrong error return value when SQ is not in error state.
    CQE recovery on TX reporter queries the sq state. If the sq is not in
```

```
    error state, the sq is either in ready or reset state. Ready state is
    good state which doesn't require recovery and reset state is a temporal
    state which ends in ready state. With this patch, CQE recovery in this
    scenario is successful.

    Fixes: de8650a82071 ("net/mlx5e: Add tx reporter support")
    Signed-off-by: Aya Levin <ayal@mellanox.com>
    Reviewed-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c02f176650560dceb79e3939a88fd12b0fde9e91
Author: Mohamad Heib <mohamadh@mellanox.com>
Date:   Tue Apr 23 21:13:48 2019 +0300

    net/mlx5e: ethtool, Avoid setting speed to 56GBASE when autoneg off

    [ Upstream commit 5faf5b70c51dd9c9905bf8209e33cbd867486607 ]

    Setting speed to 56GBASE is allowed only with auto-negotiation enabled.

    This patch prevent setting speed to 56GBASE when auto-negotiation disabled.

    Fixes: f62b8bb8f2d3 ("net/mlx5: Extend mlx5_core to support ConnectX-4 Ethernet
functionality")
    Signed-off-by: Mohamad Heib <mohamadh@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit beb811bd390d1518d391699d8b0087394da530bd
Author: David Ahern <dsahern@gmail.com>
Date:   Tue Aug 6 12:15:17 2019 -0700

    netdevsim: Restore per-network namespace accounting for fib entries

    [ Upstream commit 59c84b9fcf42c99a945d5fdc49220d854e539690 ]

    Prior to the commit in the fixes tag, the resource controller in netdevsim
    tracked fib entries and rules per network namespace. Restore that behavior.

    Fixes: 5fc494225c1e ("netdevsim: create devlink instance per netdevsim instance")
    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 e0b3ec04b778d1e4ede1f1dc76467055321e78db
Author: Roman Mashak <mrv@mojatatu.com>
Date:   Wed Aug 7 15:57:29 2019 -0400

    tc-testing: updated skbedit action tests with batch create/delete

    [ Upstream commit 7bc161846dcf4af0485f260930d17fdd892a4980 ]

    Update TDC tests with cases varifying ability of TC to install or delete
    batches of skbedit actions.

    Signed-off-by: Roman Mashak <mrv@mojatatu.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a237148b87a74875b387febae23362b26f2aa142
Author: Roman Mashak <mrv@mojatatu.com>
Date:   Wed Aug 7 15:57:28 2019 -0400

    net sched: update skbedit action for batched events operations
```

```
    [ Upstream commit e1fea322fc6d4075254ca9c5f2afdace0281da2a ]

    Add get_fill_size() routine used to calculate the action size
    when building a batch of events.

    Fixes: ca9b0e27e ("pkt_action: add new action skbedit")
    Signed-off-by: Roman Mashak <mrv@mojatatu.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 13ac261e86b5f6cdd6b7aaf2cb57ed48a2b8793d
Author: Somnath Kotur <somnath.kotur@broadcom.com>
Date:   Sat Aug 17 17:04:52 2019 -0400

    bnxt_en: Fix to include flow direction in L2 key

    [ Upstream commit 9bf46566e80fd94845527d01ebd888eb49313551 ]

    FW expects the driver to provide unique flow reference handles
    for Tx or Rx flows. When a Tx flow and an Rx flow end up sharing
    a reference handle, flow offload does not seem to work.
    This could happen in the case of 2 flows having their L2 fields
    wildcarded but in different direction.
    Fix to incorporate the flow direction as part of the L2 key

    v2: Move the dir field to the end of the bnxt_tc_l2_key struct to
    fix the warning reported by kbuild test robot <lkp@intel.com>.
    There is existing code that initializes the structure using
    nested initializer and will warn with the new u8 field added to
    the beginning.  The structure also packs nicer when this new u8 is
    added to the end of the structure [MChan].

    Fixes: abd43a13525d ("bnxt_en: Support for 64-bit flow handle.")
    Signed-off-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 58516d32c2871e277ceaaa0ab1dec5c56f29034d
Author: Venkat Duvvuru <venkatkumar.duvvuru@broadcom.com>
Date:   Sat Aug 17 17:04:51 2019 -0400

    bnxt_en: Use correct src_fid to determine direction of the flow

    [ Upstream commit 685ec6a81bb0d47faf1dba49437d5bdaede2733d ]

    Direction of the flow is determined using src_fid. For an RX flow,
    src_fid is PF's fid and for TX flow, src_fid is VF's fid. Direction
    of the flow must be specified, when getting statistics for that flow.
    Currently, for DECAP flow, direction is determined incorrectly, i.e.,
    direction is initialized as TX for DECAP flow, instead of RX. Because
    of which, stats are not reported for this DECAP flow, though it is
    offloaded and there is traffic for that flow, resulting in flow age out.

    This patch fixes the problem by determining the DECAP flow's direction
    using correct fid.  Set the flow direction in all cases for consistency
    even if 64-bit flow handle is not used.

    Fixes: abd43a13525d ("bnxt_en: Support for 64-bit flow handle.")
    Signed-off-by: Venkat Duvvuru <venkatkumar.duvvuru@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 45ad3556e9edcc9ee505ca8544dad7480736a713
Author: Vasundhara Volam <vasundhara-v.volam@broadcom.com>
Date:   Sat Aug 17 17:04:50 2019 -0400

    bnxt_en: Suppress HWRM errors for HWRM_NVM_GET_VARIABLE command

    [ Upstream commit b703ba751dbb4bcd086509ed4b28102bc1670b35 ]

    For newly added NVM parameters, older firmware may not have the support.
    Suppress the error message to avoid the unncessary error message which is
    triggered when devlink calls the driver during initialization.

    Fixes: 782a624d00fa ("bnxt_en: Add bnxt_en initial params table and register it.")
    Signed-off-by: Vasundhara Volam <vasundhara-v.volam@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d54cfa9da8fec22421fb3c1f2f5c0ad1c225dcab
Author: Vasundhara Volam <vasundhara-v.volam@broadcom.com>
Date:   Sat Aug 17 17:04:49 2019 -0400

    bnxt_en: Fix handling FRAG_ERR when NVM_INSTALL_UPDATE cmd fails

    [ Upstream commit dd2ebf3404c7c295014bc025dea23960960ceb1a ]

    If FW returns FRAG_ERR in response error code, driver is resending the
    command only when HWRM command returns success. Fix the code to resend
    NVM_INSTALL_UPDATE command with DEFRAG install flags, if FW returns
    FRAG_ERR in its response error code.

    Fixes: cb4d1d626145 ("bnxt_en: Retry failed NVM_INSTALL_UPDATE with defragmentation flag
enabled.")
    Signed-off-by: Vasundhara Volam <vasundhara-v.volam@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 889e8658d99a2fbab7a0fd7925720e341e1e15d9
Author: Michael Chan <michael.chan@broadcom.com>
Date:   Sat Aug 17 17:04:48 2019 -0400

    bnxt_en: Improve RX doorbell sequence.

    [ Upstream commit e8f267b063208372f7a329c6d5288d58944d873c ]

    When both RX buffers and RX aggregation buffers have to be
    replenished at the end of NAPI, post the RX aggregation buffers first
    before RX buffers.  Otherwise, we may run into a situation where
    there are only RX buffers without RX aggregation buffers for a split
    second.  This will cause the hardware to abort the RX packet and
    report buffer errors, which will cause unnecessary cleanup by the
    driver.

    Ringing the Aggregation ring doorbell first before the RX ring doorbell
    will prevent some of these buffer errors.  Use the same sequence during
    ring initialization as well.

    Fixes: 697197e5a173 ("bnxt_en: Re-structure doorbells.")
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6fdedaf3ad0c6f76ddc3436705b48714930f8bf0
Author: Michael Chan <michael.chan@broadcom.com>
```

```
Date:   Sat Aug 17 17:04:47 2019 -0400

    bnxt_en: Fix VNIC clearing logic for 57500 chips.

    [ Upstream commit a46ecb116fb7f722fa8cb2da01959c36e4e10c41 ]

    During device shutdown, the VNIC clearing sequence needs to be modified
    to free the VNIC first before freeing the RSS contexts.  The current
    code is doing the reverse and we can get mis-directed RX completions
    to CP ring ID 0 when the RSS contexts are freed and zeroed.  The clearing
    of RSS contexts is not required with the new sequence.

    Refactor the VNIC clearing logic into a new function bnxt_clear_vnic()
    and do the chip specific VNIC clearing sequence.

    Fixes: 7b3af4f75b81 ("bnxt_en: Add RSS support for 57500 chips.")
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 168c2657487adda4239321ba22d38dd8b51d2fb7
Author: Huy Nguyen <huyn@mellanox.com>
Date:   Thu Aug 1 11:10:19 2019 -0500

    net/mlx5e: Only support tx/rx pause setting for port owner

    [ Upstream commit 466df6eb4a9e813b3cfc674363316450c57a89c5 ]

    Only support changing tx/rx pause frame setting if the net device
    is the vport group manager.

    Fixes: 3c2d18ef22df ("net/mlx5e: Support ethtool get/set_pauseparam")
    Signed-off-by: Huy Nguyen <huyn@mellanox.com>
    Reviewed-by: Parav Pandit <parav@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 34f2824a236d2083ebb12ec970629264dac93dcc
Author: Ross Lagerwall <ross.lagerwall@citrix.com>
Date:   Mon Aug 5 16:34:34 2019 +0100

    xen/netback: Reset nr_frags before freeing skb

    [ Upstream commit 3a0233ddec554b886298de2428edb5c50a20e694 ]

    At this point nr_frags has been incremented but the frag does not yet
    have a page assigned so freeing the skb results in a crash. Reset
    nr_frags before freeing the skb to prevent this.

    Signed-off-by: Ross Lagerwall <ross.lagerwall@citrix.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e5cdd65c1534e6aab6c2378b0605298c7cd60b67
Author: Chris Packham <chris.packham@alliedtelesis.co.nz>
Date:   Mon Aug 12 08:18:25 2019 +1200

    tipc: initialise addr_trail_end when setting node addresses

    [ Upstream commit 8874ecae2977e5a2d4f0ba301364435b81c05938 ]

    We set the field 'addr_trial_end' to 'jiffies', instead of the current
    value 0, at the moment the node address is initialized. This guarantees
    we don't inadvertently enter an address trial period when the node
    address is explicitly set by the user.
```

```
    Signed-off-by: Chris Packham <chris.packham@alliedtelesis.co.nz>
    Acked-by: Jon Maloy <jon.maloy@ericsson.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 55cd9b92a4e8d38ae5f641e046a9cde2607845ad
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Thu Aug 8 14:22:47 2019 +0800

    team: Add vlan tx offload to hw_enc_features

    [ Upstream commit 227f2f030e28d8783c3d10ce70ff4ba79cad653f ]

    We should also enable team's vlan tx offload in hw_enc_features,
    pass the vlan packets to the slave devices with vlan tci, let the
    slave handle vlan tunneling offload implementation.

    Fixes: 3268e5cb494d ("team: Advertise tunneling offload features")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1a04318d6813808e36a5459d456219b81fdbadaa
Author: Xin Long <lucien.xin@gmail.com>
Date:   Mon Aug 12 20:49:12 2019 +0800

    sctp: fix the transport error_count check

    [ Upstream commit a1794de8b92ea6bc2037f445b296814ac826693e ]

    As the annotation says in sctp_do_8_2_transport_strike():

      "If the transport error count is greater than the pf_retrans
       threshold, and less than pathmaxrtx ..."

    It should be transport->error_count checked with pathmaxrxt,
    instead of asoc->pf_retrans.

    Fixes: 5aa93bcf66f4 ("sctp: Implement quick failover draft from tsvwg")
    Signed-off-by: Xin Long <lucien.xin@gmail.com>
    Acked-by: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 07a1e1551689701ac7d229ade34e1727012d4637
Author: zhengbin <zhengbin13@huawei.com>
Date:   Tue Aug 13 22:05:50 2019 +0800

    sctp: fix memleak in sctp_send_reset_streams

    [ Upstream commit 6d5afe20397b478192ed8c38ec0ee10fa3aec649 ]

    If the stream outq is not empty, need to kfree nstr_list.

    Fixes: d570a59c5b5f ("sctp: only allow the out stream reset when the stream outq is
empty")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: zhengbin <zhengbin13@huawei.com>
    Acked-by: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Acked-by: Neil Horman <nhorman@tuxdriver.com>
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aa21b3e2fdb375b4e7a2a81c440c4fb519b430a5
```

```
Author: Ivan Khoronzhuk <ivan.khoronzhuk@linaro.org>
Date:   Wed Aug 7 01:45:40 2019 +0300

    net: sched: sch_taprio: fix memleak in error path for sched list parse

    [ Upstream commit 51650d33b2771acd505068da669cf85cffac369a ]

    In error case, all entries should be freed from the sched list
    before deleting it. For simplicity use rcu way.

    Fixes: 5a781ccbd19e46 ("tc: Add support for configuring the taprio scheduler")
    Acked-by: Vinicius Costa Gomes <vinicius.gomes@intel.com>
    Signed-off-by: Ivan Khoronzhuk <ivan.khoronzhuk@linaro.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bd15d4663b99e8d5ecbbc5783a7b423f1eb4b1c1
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 14 02:11:57 2019 -0700

    net/packet: fix race in tpacket_snd()

    [ Upstream commit 32d3182cd2cd29b2e7e04df7b0db350fbe11289f ]

    packet_sendmsg() checks tx_ring.pg_vec to decide
    if it must call tpacket_snd().

    Problem is that the check is lockless, meaning another thread
    can issue a concurrent setsockopt(PACKET_TX_RING ) to flip
    tx_ring.pg_vec back to NULL.

    Given that tpacket_snd() grabs pg_vec_lock mutex, we can
    perform the check again to solve the race.

    syzbot reported :

    kasan: CONFIG_KASAN_INLINE enabled
    kasan: GPF could be caused by NULL-ptr deref or user memory access
    general protection fault: 0000 [#1] PREEMPT SMP KASAN
    CPU: 1 PID: 11429 Comm: syz-executor394 Not tainted 5.3.0-rc4+ #101
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
01/01/2011
    RIP: 0010:packet_lookup_frame+0x8d/0x270 net/packet/af_packet.c:474
    Code: c1 ee 03 f7 73 0c 80 3c 0e 00 0f 85 cb 01 00 00 48 8b 0b 89 c0 4c 8d 24 c1 48 b8
00 00 00 00 00 fc ff df 4c 89 e1 48 c1 e9 03 <80> 3c 01 00 0f 85 94 01 00 00 48 8d 7b 10 4d
8b 3c 24 48 b8 00 00
    RSP: 0018:ffff88809f82f7b8 EFLAGS: 00010246
    RAX: dfffc0000000000 RBX: ffff8880a45c7030 RCX: 0000000000000000
    RDX: 0000000000000000 RSI: 1ffff110148b8e06 RDI: ffff8880a45c703c
    RBP: ffff88809f82f7e8 R08: ffff888087aea200 R09: fffffbfff134ae50
    R10: fffffbfff134ae4f R11: ffffffff89a5727f R12: 0000000000000000
    R13: 0000000000000001 R14: ffff8880a45c6ac0 R15: 0000000000000000
    FS:  00007fa04716f700(0000) GS:ffff8880ae900000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007fa04716edb8 CR3: 0000000091eb4000 CR4: 00000000001406e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     packet_current_frame net/packet/af_packet.c:487 [inline]
     tpacket_snd net/packet/af_packet.c:2667 [inline]
     packet_sendmsg+0x590/0x6250 net/packet/af_packet.c:2975
     sock_sendmsg_nosec net/socket.c:637 [inline]
     sock_sendmsg+0xd7/0x130 net/socket.c:657
     ___sys_sendmsg+0x3e2/0x920 net/socket.c:2311
     __sys_sendmmsg+0x1bf/0x4d0 net/socket.c:2413
```

```
      __do_sys_sendmmsg net/socket.c:2442 [inline]
      __se_sys_sendmmsg net/socket.c:2439 [inline]
      __x64_sys_sendmmsg+0x9d/0x100 net/socket.c:2439
      do_syscall_64+0xfd/0x6a0 arch/x86/entry/common.c:296
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

    Fixes: 69e3c75f4d54 ("net: TX_RING and packet mmap")
    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 ea4b1cf56fdfccd27397860d4e5049859b430e58
Author: Wenwen Wang <wenwen@cs.uga.edu>
Date:   Mon Aug 12 14:11:35 2019 -0500

    net/mlx4_en: fix a memory leak bug

    [ Upstream commit 48ec7014c56e5eb2fbf6f479896143622d834f3b ]

    In mlx4_en_config_rss_steer(), 'rss_map->indir_qp' is allocated through
    kzalloc(). After that, mlx4_qp_alloc() is invoked to configure RSS
    indirection. However, if mlx4_qp_alloc() fails, the allocated
    'rss_map->indir_qp' is not deallocated, leading to a memory leak bug.

    To fix the above issue, add the 'qp_alloc_err' label to free
    'rss_map->indir_qp'.

    Fixes: 4931c6ef04b4 ("net/mlx4_en: Optimized single ring steering")
    Signed-off-by: Wenwen Wang <wenwen@cs.uga.edu>
    Reviewed-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4c3e9cdbcbf533300fd409d63110c253cef2f6fe
Author: Chen-Yu Tsai <wens@csie.org>
Date:   Sun Aug 11 22:18:25 2019 +0800

    net: dsa: Check existence of .port_mdb_add callback before calling it

    [ Upstream commit 58799865be84e2a895dab72de0e1b996ed943f22 ]

    The dsa framework has optional .port_mdb_{prepare,add,del} callback fields
    for drivers to handle multicast database entries. When adding an entry, the
    framework goes through a prepare phase, then a commit phase. Drivers not
    providing these callbacks should be detected in the prepare phase.

    DSA core may still bypass the bridge layer and call the dsa_port_mdb_add
    function directly with no prepare phase or no switchdev trans object,
    and the framework ends up calling an undefined .port_mdb_add callback.
    This results in a NULL pointer dereference, as shown in the log below.

    The other functions seem to be properly guarded. Do the same for
    .port_mdb_add in dsa_switch_mdb_add_bitmap() as well.

        8<--- cut here ---
        Unable to handle kernel NULL pointer dereference at virtual address 00000000
        pgd = (ptrval)
        [00000000] *pgd=00000000
        Internal error: Oops: 80000005 [#1] SMP ARM
        Modules linked in: rtl8xxxu rtl8192cu rtl_usb rtl8192c_common rtlwifi mac80211
cfg80211
        CPU: 1 PID: 134 Comm: kworker/1:2 Not tainted 5.3.0-rc1-00247-gd3519030752a #1
        Hardware name: Allwinner sun7i (A20) Family
        Workqueue: events switchdev_deferred_process_work
```

```
        PC is at 0x0
        LR is at dsa_switch_event+0x570/0x620
        pc : [<00000000>]    lr : [<c08533ec>]    psr: 80070013
        sp : ee871db8  ip : 00000000  fp : ee98d0a4
        r10: 0000000c  r9 : 00000008  r8 : ee89f710
        r7 : ee98d040  r6 : ee98d088  r5 : c0f04c48  r4 : ee98d04c
        r3 : 00000000  r2 : ee89f710  r1 : 00000008  r0 : ee98d040
        Flags: Nzcv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
        Control: 10c5387d  Table: 6deb406a  DAC: 00000051
        Process kworker/1:2 (pid: 134, stack limit = 0x(ptrval))
        Stack: (0xee871db8 to 0xee872000)
        1da0:                                                       ee871e14 103ace2d
        1dc0: 00000000 ffffffff 00000000 ee871e14 00000005 00000000 c08524a0 00000000
        1de0: ffffe000 c014bdfc c0f04c48 ee871e98 c0f04c48 ee9e5000 c0851120 c014bef0
        1e00: 00000000 b643aea2 ee9b4068 c08509a8 ee2bf940 ee89f710 ee871ecb 00000000
        1e20: 00000008 103ace2d 00000000 c087e248 ee29c868 103ace2d 00000001 ffffffff
        1e40: 00000000 ee871e98 00000006 00000000 c0fb2a50 c087e2d0 ffffffff c08523c4
        1e60: ffffffff c014bdfc 00000006 c0fad2d0 ee871e98 ee89f710 00000000 c014c500
        1e80: 00000000 ee89f3c0 c0f04c48 00000000 ee9e5000 c087dfb4 ee9e5000 00000000
        1ea0: ee89f710 ee871ecb 00000001 103ace2d 00000000 c0f04c48 00000000 c087e0a8
        1ec0: 00000000 efd9a3e0 0089f3c0 103ace2d ee89f700 ee89f710 ee9e5000 00000122
        1ee0: 00000100 c087e130 ee89f700 c0fad2c8 c1003ef0 c087de4c 2e928000 c0fad2ec
        1f00: c0fad2ec ee839580 ef7a62c0 ef7a9400 00000000 c087def8 c0fad2ec c01447dc
        1f20: ef315640 ef7a62c0 00000008 ee839580 ee839594 ef7a62c0 00000008 c0f03d00
        1f40: ef7a62d8 ef7a62c0 ffffe000 c0145b84 ffffe000 c0fb2420 c0bfaa8c 00000000
        1f60: ffffe000 ee84b600 ee84b5c0 00000000 ee870000 ee839580 c0145b40 ef0e5ea4
        1f80: ee84b61c c014a6f8 00000001 ee84b5c0 c014a5b0 00000000 00000000 00000000
        1fa0: 00000000 00000000 00000000 c01010e8 00000000 00000000 00000000 00000000
        1fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
        1fe0: 00000000 00000000 00000000 00000000 00000013 00000000 00000000 00000000
        [<c08533ec>] (dsa_switch_event) from [<c014bdfc>] (notifier_call_chain+0x48/0x84)
        [<c014bdfc>] (notifier_call_chain) from [<c014bef0>]
(raw_notifier_call_chain+0x18/0x20)
        [<c014bef0>] (raw_notifier_call_chain) from [<c08509a8>]
(dsa_port_mdb_add+0x48/0x74)
        [<c08509a8>] (dsa_port_mdb_add) from [<c087e248>]
(__switchdev_handle_port_obj_add+0x54/0xd4)
        [<c087e248>] (__switchdev_handle_port_obj_add) from [<c087e2d0>]
(switchdev_handle_port_obj_add+0x8/0x14)
        [<c087e2d0>] (switchdev_handle_port_obj_add) from [<c08523c4>]
(dsa_slave_switchdev_blocking_event+0x94/0xa4)
        [<c08523c4>] (dsa_slave_switchdev_blocking_event) from [<c014bdfc>]
(notifier_call_chain+0x48/0x84)
        [<c014bdfc>] (notifier_call_chain) from [<c014c500>]
(blocking_notifier_call_chain+0x50/0x68)
        [<c014c500>] (blocking_notifier_call_chain) from [<c087dfb4>]
(switchdev_port_obj_notify+0x44/0xa8)
        [<c087dfb4>] (switchdev_port_obj_notify) from [<c087e0a8>]
(switchdev_port_obj_add_now+0x90/0x104)
        [<c087e0a8>] (switchdev_port_obj_add_now) from [<c087e130>]
(switchdev_port_obj_add_deferred+0x14/0x5c)
        [<c087e130>] (switchdev_port_obj_add_deferred) from [<c087de4c>]
(switchdev_deferred_process+0x64/0x104)
        [<c087de4c>] (switchdev_deferred_process) from [<c087def8>]
(switchdev_deferred_process_work+0xc/0x14)
        [<c087def8>] (switchdev_deferred_process_work) from [<c01447dc>]
(process_one_work+0x218/0x50c)
        [<c01447dc>] (process_one_work) from [<c0145b84>] (worker_thread+0x44/0x5bc)
        [<c0145b84>] (worker_thread) from [<c014a6f8>] (kthread+0x148/0x150)
        [<c014a6f8>] (kthread) from [<c01010e8>] (ret_from_fork+0x14/0x2c)
        Exception stack(0xee871fb0 to 0xee871ff8)
        1fa0:                                     00000000 00000000 00000000 00000000
        1fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
        1fe0: 00000000 00000000 00000000 00000000 00000013 00000000
        Code: bad PC value
```

```
         ---[ end trace 1292c61abd17b130 ]---

         [<c08533ec>] (dsa_switch_event) from [<c014bdfc>] (notifier_call_chain+0x48/0x84)
         corresponds to

             $ arm-linux-gnueabihf-addr2line -C -i -e vmlinux c08533ec

             linux/net/dsa/switch.c:156
             linux/net/dsa/switch.c:178
             linux/net/dsa/switch.c:328

     Fixes: e6db98db8a95 ("net: dsa: add switch mdb bitmap functions")
     Signed-off-by: Chen-Yu Tsai <wens@csie.org>
     Reviewed-by: Vivien Didelot <vivien.didelot@gmail.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit d66635a0b601f04ef4da30eec8a218d53f79f9b7
 Author: YueHaibing <yuehaibing@huawei.com>
 Date:   Wed Aug 7 10:19:59 2019 +0800

     bonding: Add vlan tx offload to hw_enc_features

     [ Upstream commit d595b03de2cb0bdf9bcdf35ff27840cc3a37158f ]

     As commit 30d8177e8ac7 ("bonding: Always enable vlan tx offload")
     said, we should always enable bonding's vlan tx offload, pass the
     vlan packets to the slave devices with vlan tci, let them to handle
     vlan implementation.

     Now if encapsulation protocols like VXLAN is used, skb->encapsulation
     may be set, then the packet is passed to vlan device which based on
     bonding device. However in netif_skb_features(), the check of
     hw_enc_features:

              if (skb->encapsulation)
                      features &= dev->hw_enc_features;

     clears NETIF_F_HW_VLAN_CTAG_TX/NETIF_F_HW_VLAN_STAG_TX. This results
     in same issue in commit 30d8177e8ac7 like this:

     vlan_dev_hard_start_xmit
       -->dev_queue_xmit
         -->validate_xmit_skb
           -->netif_skb_features //NETIF_F_HW_VLAN_CTAG_TX is cleared
           -->validate_xmit_vlan
             -->__vlan_hwaccel_push_inside //skb->tci is cleared
     ...
      --> bond_start_xmit
        --> bond_xmit_hash //BOND_XMIT_POLICY_ENCAP34
          --> __skb_flow_dissect // nhoff point to IP header
             -->  case htons(ETH_P_8021Q)
                  // skb_vlan_tag_present is false, so
                  vlan = __skb_header_pointer(skb, nhoff, sizeof(_vlan),
                  //vlan point to ip header wrongly

     Fixes: b2a103e6d0af ("bonding: convert to ndo_fix_features")
     Signed-off-by: YueHaibing <yuehaibing@huawei.com>
     Acked-by: Jay Vosburgh <jay.vosburgh@canonical.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 5d6f83b9ed50d271bc6e95f901efc44c49d7cb7a
 Author: Manish Chopra <manishc@marvell.com>
 Date:   Sun Aug 18 07:25:48 2019 -0700
```

```
    bnx2x: Fix VF's VLAN reconfiguration in reload.

    [ Upstream commit 4a4d2d372fb9b9229327e2ed01d5d9572eddf4de ]

    Commit 04f05230c5c13 ("bnx2x: Remove configured vlans as
    part of unload sequence."), introduced a regression in driver
    that as a part of VF's reload flow, VLANs created on the VF
    doesn't get re-configured in hardware as vlan metadata/info
    was not getting cleared for the VFs which causes vlan PING to stop.

    This patch clears the vlan metadata/info so that VLANs gets
    re-configured back in the hardware in VF's reload flow and
    PING/traffic continues for VLANs created over the VFs.

    Fixes: 04f05230c5c13 ("bnx2x: Remove configured vlans as part of unload sequence.")
    Signed-off-by: Manish Chopra <manishc@marvell.com>
    Signed-off-by: Sudarsana Kalluru <skalluru@marvell.com>
    Signed-off-by: Shahed Shaikh <shshaikh@marvell.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e8904e5e4d43a297afc6a3e7c8881fda2039614c
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Tue Jul 16 20:17:20 2019 +0200

    Input: psmouse - fix build error of multiple definition

    commit 49e6979e7e92cf496105b5636f1df0ac17c159c0 upstream.

    trackpoint_detect() should be static inline while
    CONFIG_MOUSE_PS2_TRACKPOINT is not set, otherwise, we build fails:

    drivers/input/mouse/alps.o: In function `trackpoint_detect':
    alps.c:(.text+0x8e00): multiple definition of `trackpoint_detect'
    drivers/input/mouse/psmouse-base.o:psmouse-base.c:(.text+0x1b50): first defined here

    Reported-by: Hulk Robot <hulkci@huawei.com>
    Fixes: 55e3d9224b60 ("Input: psmouse - allow disabing certain protocol extensions")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Cc: Hui Wang <hui.wang@canonical.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 825169c942396de176858fddbda22fbc8bb42f10
Author: Haim Dreyfuss <haim.dreyfuss@intel.com>
Date:   Wed Feb 27 16:43:45 2019 +0200

    iwlwifi: Add support for SAR South Korea limitation

    commit 0c3d7282233c7b02c74400b49981d6fff1d683a8 upstream.

    South Korea is adding a more strict SAR limit called "Limb SAR".
    Currently, WGDS SAR offset group 3 is not used (not mapped to any country).
    In order to be able to comply with South Korea new restriction:
    - OEM will use WGDS SAR offset group 3 to South Korea limitation.
    - OEM will change WGDS revision to 1 (currently latest revision is 0)
            to notify that Korea Limb SAR applied.
    - Driver will read the WGDS table and pass the values to FW (as usual)
    - Driver will pass to FW an indication that Korea Limb SAR is applied
            in case table revision is 1.

    Signed-off-by: Haim Dreyfuss <haim.dreyfuss@intel.com>
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 9aaf224300b005ed87fae5550ed76a307e49579c
Author: Dirk Morris <dmorris@metaloft.com>
Date:   Thu Aug 8 13:57:51 2019 -0700

    netfilter: conntrack: Use consistent ct id hash calculation

    commit 656c8e9cc1badbc18eefe6ba01d33ebbcae61b9a upstream.

    Change ct id hash calculation to only use invariants.

    Currently the ct id hash calculation is based on some fields that can
    change in the lifetime on a conntrack entry in some corner cases. The
    current hash uses the whole tuple which contains an hlist pointer which
    will change when the conntrack is placed on the dying list resulting in
    a ct id change.

    This patch also removes the reply-side tuple and extension pointer from
    the hash calculation so that the ct id will will not change from
    initialization until confirmation.

    Fixes: 3c79107631db1f7 ("netfilter: ctnetlink: don't use conntrack/expect object
addresses as id")
    Signed-off-by: Dirk Morris <dmorris@metaloft.com>
    Acked-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 178398e1d821ba2cbeb0ffa636ceeb68c88d4f28
Author: Thiébaud Weksteen <tweek@google.com>
Date:   Tue Aug 6 13:00:50 2019 +0200

    usb: setup authorized_default attributes using usb_bus_notify

    commit 27709ae4e2fe6cf7da2ae45e718e190c5433342b upstream.

    Currently, the authorized_default and interface_authorized_default
    attributes for HCD are set up after the uevent has been sent to userland.
    This creates a race condition where userland may fail to access this
    file when processing the event. Move the appending of these attributes
    earlier relying on the usb_bus_notify dispatcher.

    Signed-off-by: Thiébaud Weksteen <tweek@google.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20190806110050.38918-1-tweek@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 24223b9c00f274adf453adfb57b4e04615492cb8
Author: Tony Lindgren <tony@atomide.com>
Date:   Thu Aug 15 01:26:02 2019 -0700

    USB: serial: option: Add Motorola modem UARTs

    commit 6caf0be40a707689e8ff8824fdb96ef77685b1ba upstream.

    On Motorola Mapphone devices such as Droid 4 there are five USB ports
    that do not use the same layout as Gobi 1K/2K/etc devices listed in
    qcserial.c. So we should use qcaux.c or option.c as noted by
    Dan Williams <dan.j.williams@intel.com>.

    As the Motorola USB serial ports have an interrupt endpoint as shown
    with lsusb -v, we should use option.c instead of qcaux.c as pointed out
    by Johan Hovold <johan@kernel.org>.

    The ff/ff/ff interfaces seem to always be UARTs on Motorola devices.
```

```
    For the other interfaces, class 0x0a (CDC Data) should not in general
    be added as they are typically part of a multi-interface function as
    noted earlier by Bjørn Mork <bjorn@mork.no>.

    However, looking at the Motorola mapphone kernel code, the mdm6600 0x0a
    class is only used for flashing the modem firmware, and there are no
    other interfaces. So I've added that too with more details below as it
    works just fine.

    The ttyUSB ports on Droid 4 are:

    ttyUSB0 DIAG, CQDM-capable
    ttyUSB1 MUX or NMEA, no response
    ttyUSB2 MUX or NMEA, no response
    ttyUSB3 TCMD
    ttyUSB4 AT-capable

    The ttyUSB0 is detected as QCDM capable by ModemManager. I think
    it's only used for debugging with ModemManager --debug for sending
    custom AT commands though. ModemManager already can manage data
    connection using the USB QMI ports that are already handled by the
    qmi_wwan.c driver.

    To enable the MUX or NMEA ports, it seems that something needs to be
    done additionally to enable them, maybe via the DIAG or TCMD port.
    It might be just a NVRAM setting somewhere, but I have no idea what
    NVRAM settings may need changing for that.

    The TCMD port seems to be a Motorola custom protocol for testing
    the modem and to configure it's NVRAM and seems to work just fine
    based on a quick test with a minimal tcmdrw tool I wrote.

    The voice modem AT-capable port seems to provide only partial
    support, and no PM support compared to the TS 27.010 based UART
    wired directly to the modem.

    The UARTs added with this change are the same product IDs as the
    Motorola Mapphone Android Linux kernel mdm6600_id_table. I don't
    have any mdm9600 based devices, so I have only tested these on
    mdm6600 based droid 4.

    Then for the class 0x0a (CDC Data) mode, the Motorola Mapphone Android
    Linux kernel driver moto_flashqsc.c just seems to change the
    port->bulk_out_size to 8K from the default. And is only used for
    flashing the modem firmware it seems.

    I've verified that flashing the modem with signed firmware works just
    fine with the option driver after manually toggling the GPIO pins, so
    I've added droid 4 modem flashing mode to the option driver. I've not
    added the other devices listed in moto_flashqsc.c in case they really
    need different port->bulk_out_size. Those can be added as they get
    tested to work for flashing the modem.

    After this patch the output of /sys/kernel/debug/usb/devices has
    the following for normal 22b8:2a70 mode including the related qmi_wwan
    interfaces:

    T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12   MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=22b8 ProdID=2a70 Rev= 0.00
    S:  Manufacturer=Motorola, Incorporated
    S:  Product=Flash MZ600
    C:* #Ifs= 9 Cfg#= 1 Atr=e0 MxPwr=500mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
    E:  Ad=81(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
```

```
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=82(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 2 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=83(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=03(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 3 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=84(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=04(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=85(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=86(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=05(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 5 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fb Prot=ff Driver=qmi_wwan
E:  Ad=87(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=88(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=06(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 6 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fb Prot=ff Driver=qmi_wwan
E:  Ad=89(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=8a(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=07(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 7 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fb Prot=ff Driver=qmi_wwan
E:  Ad=8b(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=8c(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=08(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
I:* If#= 8 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fb Prot=ff Driver=qmi_wwan
E:  Ad=8d(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=8e(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=09(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms

In 22b8:900e "qc_dload" mode the device shows up as:

T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12   MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=22b8 ProdID=900e Rev= 0.00
S:  Manufacturer=Motorola, Incorporated
S:  Product=Flash MZ600
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=500mA
I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=81(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms

And in 22b8:4281 "ram_downloader" mode the device shows up as:

T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12   MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=22b8 ProdID=4281 Rev= 0.00
S:  Manufacturer=Motorola, Incorporated
S:  Product=Flash MZ600
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=500mA
I:* If#= 0 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=fc Driver=option
E:  Ad=81(I) Atr=02(Bulk) MxPS=  64 Ivl=0ms
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms

Cc: Bjørn Mork <bjorn@mork.no>
Cc: Dan Williams <dan.j.williams@intel.com>
Cc: Lars Melin <larsm17@gmail.com>
Cc: Marcel Partap <mpartap@gmx.net>
Cc: Merlijn Wajer <merlijn@wizzup.org>
Cc: Michael Scott <hashcode0f@gmail.com>
Cc: NeKit <nekit1000@gmail.com>
Cc: Pavel Machek <pavel@ucw.cz>
Cc: Sebastian Reichel <sre@kernel.org>
Tested-by: Pavel Machek <pavel@ucw.cz>
```

```
    Signed-off-by: Tony Lindgren <tony@atomide.com>
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit efb1afa1c66c6d8dc3a35950eb18b9bde252a3e6
Author: Bob Ham <bob.ham@puri.sm>
Date:   Wed Jul 24 07:52:26 2019 -0700

    USB: serial: option: add the BroadMobi BM818 card

    commit e5d8badf37e6b547842f2fcde10361b29e08bd36 upstream.

    Add a VID:PID for the BroadMobi BM818 M.2 card

    T:  Bus=01 Lev=03 Prnt=40 Port=03 Cnt=01 Dev#= 44 Spd=480 MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=2020 ProdID=2060 Rev=00.00
    S:  Manufacturer=Qualcomm, Incorporated
    S:  Product=Qualcomm CDMA Technologies MSM
    C:  #Ifs= 5 Cfg#= 1 Atr=e0 MxPwr=500mA
    I:  If#=0x0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
    I:  If#=0x1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
    I:  If#=0x2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
    I:  If#=0x3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fe Prot=ff Driver=(none)
    I:  If#=0x4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)

    Signed-off-by: Bob Ham <bob.ham@puri.sm>
    Signed-off-by: Angus Ainslie (Purism) <angus@akkea.ca>
    Cc: stable <stable@vger.kernel.org>
    [ johan: use USB_DEVICE_INTERFACE_CLASS() ]
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 27894257310346285fd29ece637fe717ab428dcd
Author: Yoshiaki Okamoto <yokamoto@allied-telesis.co.jp>
Date:   Sat Jul 20 22:23:18 2019 +0900

    USB: serial: option: Add support for ZTE MF871A

    commit 7e7ae38bf928c5cfa6dd6e9a2cf8b42c84a27c92 upstream.

    This patch adds support for MF871A USB modem (aka Speed USB STICK U03)
    to option driver. This modem is manufactured by ZTE corporation, and
    sold by KDDI.

    Interface layout:
    0: AT
    1: MODEM

    usb-devices output:
    T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  9 Spd=480 MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=19d2 ProdID=1481 Rev=52.87
    S:  Manufacturer=ZTE,Incorporated
    S:  Product=ZTE Technologies MSM
    S:  SerialNumber=1234567890ABCDEF
    C:  #Ifs= 2 Cfg#= 1 Atr=80 MxPwr=500mA
    I:  If#= 0 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    I:  If#= 1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option

    Co-developed-by: Hiroyuki Yamamoto <hyamamo@allied-telesis.co.jp>
    Signed-off-by: Hiroyuki Yamamoto <hyamamo@allied-telesis.co.jp>
    Signed-off-by: Yoshiaki Okamoto <yokamoto@allied-telesis.co.jp>
    Cc: stable <stable@vger.kernel.org>
```

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

commit 57abf8f9824b2e290023135ed136495f1b3e6462
Author: Rogan Dawes <rogan@dawes.za.net>
Date:   Wed Jul 17 11:11:34 2019 +0200

    USB: serial: option: add D-Link DWM-222 device ID

    commit 552573e42aab5f75aff9bab855a9677979d9a7d5 upstream.

    Add device id for D-Link DWM-222 A2.

    MI_00 D-Link HS-USB Diagnostics
    MI_01 D-Link HS-USB Modem
    MI_02 D-Link HS-USB AT Port
    MI_03 D-Link HS-USB NMEA
    MI_04 D-Link HS-USB WWAN Adapter (qmi_wwan)
    MI_05 USB Mass Storage Device

    Cc: stable@vger.kernel.org
    Signed-off-by: Rogan Dawes <rogan@dawes.za.net>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 58ab4f8fcd9a0ae6c4399cb4076ba5f26e037436
Author: Oliver Neukum <oneukum@suse.com>
Date:   Tue Aug 13 11:35:41 2019 +0200

    USB: CDC: fix sanity checks in CDC union parser

    commit 54364278fb3cabdea51d6398b07c87415065b3fc upstream.

    A few checks checked for the size of the pointer to a structure
    instead of the structure itself. Copy & paste issue presumably.

    Fixes: e4c6fb7794982 ("usbnet: move the CDC parser into USB core")
    Cc: stable <stable@vger.kernel.org>
    Reported-by: syzbot+45a53506b65321c1fe91@syzkaller.appspotmail.com
    Signed-off-by: Oliver Neukum <oneukum@suse.com>
    Link: https://lore.kernel.org/r/20190813093541.18889-1-oneukum@suse.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 058a394e5a3aa98546034bb11f90ea67a14b9c80
Author: Oliver Neukum <oneukum@suse.com>
Date:   Thu Aug 8 16:21:19 2019 +0200

    usb: cdc-acm: make sure a refcount is taken early enough

    commit c52873e5a1ef72f845526d9f6a50704433f9c625 upstream.

    destroy() will decrement the refcount on the interface, so that
    it needs to be taken so early that it never undercounts.

    Fixes: 7fb57a019f94e ("USB: cdc-acm: Fix potential deadlock (lockdep warning)")
    Cc: stable <stable@vger.kernel.org>
    Reported-and-tested-by: syzbot+1b2449b7b5dc240d107a@syzkaller.appspotmail.com
    Signed-off-by: Oliver Neukum <oneukum@suse.com>
    Link: https://lore.kernel.org/r/20190808142119.7998-1-oneukum@suse.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit da395ccddeca35d8a157f3bb298ff05c1d9e10be
Author: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
Date:   Wed Jul 31 19:15:43 2019 +0900
```

```
    usb: gadget: udc: renesas_usb3: Fix sysfs interface of "role"

    commit 5dac665cf403967bb79a7aeb8c182a621fe617ff upstream.

    Since the role_store() uses strncmp(), it's possible to refer
    out-of-memory if the sysfs data size is smaller than strlen("host").
    This patch fixes it by using sysfs_streq() instead of strncmp().

    Fixes: cc995c9ec118 ("usb: gadget: udc: renesas_usb3: add support for usb role swap")
    Cc: <stable@vger.kernel.org> # v4.12+
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Signed-off-by: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
    Signed-off-by: Felipe Balbi <felipe.balbi@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6ee820f073c33e900273b496c1c1d48a30a67798
Author: Alan Stern <stern@rowland.harvard.edu>
Date:   Mon Aug 12 16:11:07 2019 -0400

    USB: core: Fix races in character device registration and deregistraion

    commit 303911cfc5b95d33687d9046133ff184cf5043ff upstream.

    The syzbot fuzzer has found two (!) races in the USB character device
    registration and deregistration routines.  This patch fixes the races.

    The first race results from the fact that usb_deregister_dev() sets
    usb_minors[intf->minor] to NULL before calling device_destroy() on the
    class device.  This leaves a window during which another thread can
    allocate the same minor number but will encounter a duplicate name
    error when it tries to register its own class device.  A typical error
    message in the system log would look like:

        sysfs: cannot create duplicate filename '/class/usbmisc/ldusb0'

    The patch fixes this race by destroying the class device first.

    The second race is in usb_register_dev().  When that routine runs, it
    first allocates a minor number, then drops minor_rwsem, and then
    creates the class device.  If the device creation fails, the minor
    number is deallocated and the whole routine returns an error.  But
    during the time while minor_rwsem was dropped, there is a window in
    which the minor number is allocated and so another thread can
    successfully open the device file.  Typically this results in
    use-after-free errors or invalid accesses when the other thread closes
    its open file reference, because the kernel then tries to release
    resources that were already deallocated when usb_register_dev()
    failed.  The patch fixes this race by keeping minor_rwsem locked
    throughout the entire routine.

    Reported-and-tested-by: syzbot+30cf45ebfe0b0c4847a1@syzkaller.appspotmail.com
    Signed-off-by: Alan Stern <stern@rowland.harvard.edu>
    CC: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/Pine.LNX.4.44L0.1908121607590.1659-
100000@iolanthe.rowland.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 90c191ca0d7464677e1c17d88f20c0a0f03597c2
Author: Jacopo Mondi <jacopo+renesas@jmondi.org>
Date:   Mon Aug 5 17:55:15 2019 +0200

    iio: adc: max9611: Fix temperature reading in probe

    commit b9ddd5091160793ee9fac10da765cf3f53d2aaf0 upstream.
```

```
    The max9611 driver reads the die temperature at probe time to validate
    the communication channel. Use the actual read value to perform the test
    instead of the read function return value, which was mistakenly used so
    far.

    The temperature reading test was only successful because the 0 return
    value is in the range of supported temperatures.

    Fixes: 69780a3bbc0b ("iio: adc: Add Maxim max9611 ADC driver")
    Signed-off-by: Jacopo Mondi <jacopo+renesas@jmondi.org>
    Cc: <Stable@vger.kernel.org>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2244a42351a428258d72f53f0a2cc064b037abad
Author: Ian Abbott <abbotti@mev.co.uk>
Date:   Mon Aug 12 13:08:14 2019 +0100

    staging: comedi: dt3000: Fix rounding up of timer divisor

    commit 8e2a589a3fc36ce858d42e767c3bcd8fc62a512b upstream.

    `dt3k_ns_to_timer()` determines the prescaler and divisor to use to
    produce a desired timing period.  It is influenced by a rounding mode
    and can round the divisor up, down, or to the nearest value.  However,
    the code for rounding up currently does the same as rounding down!  Fix
    ir by using the `DIV_ROUND_UP()` macro to calculate the divisor when
    rounding up.

    Also, change the types of the `divider`, `base` and `prescale` variables
    from `int` to `unsigned int` to avoid mixing signed and unsigned types
    in the calculations.

    Also fix a typo in a nearby comment: "improvment" => "improvement".

    Signed-off-by: Ian Abbott <abbotti@mev.co.uk>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20190812120814.21188-1-abbotti@mev.co.uk
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7961db77b2135740f683418397b06fbdae099a25
Author: Ian Abbott <abbotti@mev.co.uk>
Date:   Mon Aug 12 12:15:17 2019 +0100

    staging: comedi: dt3000: Fix signed integer overflow 'divider * base'

    commit b4d98bc3fc93ec3a58459948a2c0e0c9b501cd88 upstream.

    In `dt3k_ns_to_timer()` the following lines near the end of the function
    result in a signed integer overflow:

            prescale = 15;
            base = timer_base * (1 << prescale);
            divider = 65535;
            *nanosec = divider * base;

    (`divider`, `base` and `prescale` are type `int`, `timer_base` and
    `*nanosec` are type `unsigned int`.  The value of `timer_base` will be
    either 50 or 100.)

    The main reason for the overflow is that the calculation for `base` is
    completely wrong.  It should be:

            base = timer_base * (prescale + 1);
```

```
    which matches an earlier instance of this calculation in the same
    function.

    Reported-by: David Binderman <dcb314@hotmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Ian Abbott <abbotti@mev.co.uk>
    Link: https://lore.kernel.org/r/20190812111517.26803-1-abbotti@mev.co.uk
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4a91541877154901037d7a5b3f18287a13f2162e
Author: Anders Roxell <anders.roxell@linaro.org>
Date:   Fri Jul 26 13:27:05 2019 +0200

    arm64: KVM: regmap: Fix unexpected switch fall-through

    commit 3d584a3c85d6fe2cf878f220d4ad7145e7f89218 upstream.

    When fall-through warnings was enabled by default, commit d93512ef0f0e
    ("Makefile: Globally enable fall-through warning"), the following
    warnings was starting to show up:

    In file included from ../arch/arm64/include/asm/kvm_emulate.h:19,
                     from ../arch/arm64/kvm/regmap.c:13:
    ../arch/arm64/kvm/regmap.c: In function ‘vcpu_write_spsr32’:
    ../arch/arm64/include/asm/kvm_hyp.h:31:3: warning: this statement may fall
     through [-Wimplicit-fallthrough=]
       asm volatile(ALTERNATIVE(__msr_s(r##nvh, "%x0"), \
       ^~~
    ../arch/arm64/include/asm/kvm_hyp.h:46:31: note: in expansion of macro
‘write_sysreg_elx’
     #define write_sysreg_el1(v,r) write_sysreg_elx(v, r, _EL1, _EL12)
                                   ^~~~~~~~~~~~~~~
    ../arch/arm64/kvm/regmap.c:180:3: note: in expansion of macro ‘write_sysreg_el1’
       write_sysreg_el1(v, SYS_SPSR);
       ^~~~~~~~~~~~~~~
    ../arch/arm64/kvm/regmap.c:181:2: note: here
      case KVM_SPSR_ABT:
      ^~~~
    In file included from ../arch/arm64/include/asm/cputype.h:132,
                     from ../arch/arm64/include/asm/cache.h:8,
                     from ../include/linux/cache.h:6,
                     from ../include/linux/printk.h:9,
                     from ../include/linux/kernel.h:15,
                     from ../include/asm-generic/bug.h:18,
                     from ../arch/arm64/include/asm/bug.h:26,
                     from ../include/linux/bug.h:5,
                     from ../include/linux/mmdebug.h:5,
                     from ../include/linux/mm.h:9,
                     from ../arch/arm64/kvm/regmap.c:11:
    ../arch/arm64/include/asm/sysreg.h:837:2: warning: this statement may fall
     through [-Wimplicit-fallthrough=]
      asm volatile("msr " __stringify(r) ", %x0"  \
      ^~~
    ../arch/arm64/kvm/regmap.c:182:3: note: in expansion of macro ‘write_sysreg’
       write_sysreg(v, spsr_abt);
       ^~~~~~~~~~~~
    ../arch/arm64/kvm/regmap.c:183:2: note: here
      case KVM_SPSR_UND:
      ^~~~

    Rework to add a 'break;' in the swich-case since it didn't have that,
    leading to an interresting set of bugs.

    Cc: stable@vger.kernel.org # v4.17+
    Fixes: a892819560c4 ("KVM: arm64: Prepare to handle deferred save/restore of 32-bit
```

```
registers")
    Signed-off-by: Anders Roxell <anders.roxell@linaro.org>
    [maz: reworked commit message, fixed stable range]
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 591eb1c6e27f7a792172bc6d41d7fddfbf1bfd0b
Author: Nayna Jain <nayna@linux.ibm.com>
Date:   Thu Jul 11 12:13:35 2019 -0400

    tpm: tpm_ibm_vtpm: Fix unallocated banks

    [ Upstream commit fa4f99c05320eb28bf6ba52a9adf64d888da1f9e ]

    The nr_allocated_banks and allocated banks are initialized as part of
    tpm_chip_register. Currently, this is done as part of auto startup
    function. However, some drivers, like the ibm vtpm driver, do not run
    auto startup during initialization. This results in uninitialized memory
    issue and causes a kernel panic during boot.

    This patch moves the pcr allocation outside the auto startup function
    into tpm_chip_register. This ensures that allocated banks are initialized
    in any case.

    Fixes: 879b589210a9 ("tpm: retrieve digest size of unknown algorithms with PCR read")
    Reported-by: Michal Suchanek <msuchanek@suse.de>
    Signed-off-by: Nayna Jain <nayna@linux.ibm.com>
    Reviewed-by: Mimi Zohar <zohar@linux.ibm.com>
    Tested-by: Sachin Sant <sachinp@linux.vnet.ibm.com>
    Tested-by: Michal Suchánek <msuchanek@suse.de>
    Reviewed-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Signed-off-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 382cbf20a3877e5dad9ae82c784e9cdb8e2a1d4f
Author: Qian Cai <cai@lca.pw>
Date:   Fri Aug 2 21:49:19 2019 -0700

    asm-generic: fix -Wtype-limits compiler warnings

    [ Upstream commit cbedfe11347fe418621bd188d58a206beb676218 ]

    Commit d66acc39c7ce ("bitops: Optimise get_order()") introduced a
    compilation warning because "rx_frag_size" is an "ushort" while
    PAGE_SHIFT here is 16.

    The commit changed the get_order() to be a multi-line macro where
    compilers insist to check all statements in the macro even when
    __builtin_constant_p(rx_frag_size) will return false as "rx_frag_size"
    is a module parameter.

    In file included from ./arch/powerpc/include/asm/page_64.h:107,
                     from ./arch/powerpc/include/asm/page.h:242,
                     from ./arch/powerpc/include/asm/mmu.h:132,
                     from ./arch/powerpc/include/asm/lppaca.h:47,
                     from ./arch/powerpc/include/asm/paca.h:17,
                     from ./arch/powerpc/include/asm/current.h:13,
                     from ./include/linux/thread_info.h:21,
                     from ./arch/powerpc/include/asm/processor.h:39,
                     from ./include/linux/prefetch.h:15,
                     from drivers/net/ethernet/emulex/benet/be_main.c:14:
    drivers/net/ethernet/emulex/benet/be_main.c: In function 'be_rx_cqs_create':
    ./include/asm-generic/getorder.h:54:9: warning: comparison is always
    true due to limited range of data type [-Wtype-limits]
       (((n) < (1UL << PAGE_SHIFT)) ? 0 :  \
```

```
                ^
    drivers/net/ethernet/emulex/benet/be_main.c:3138:33: note: in expansion
    of macro 'get_order'
      adapter->big_page_size = (1 << get_order(rx_frag_size)) * PAGE_SIZE;
                                    ^~~~~~~~~

    Fix it by moving all of this multi-line macro into a proper function,
    and killing __get_order() off.

    [akpm@linux-foundation.org: remove __get_order() altogether]
    [cai@lca.pw: v2]
      Link: http://lkml.kernel.org/r/1564000166-31428-1-git-send-email-cai@lca.pw
    Link: http://lkml.kernel.org/r/1563914986-26502-1-git-send-email-cai@lca.pw
    Fixes: d66acc39c7ce ("bitops: Optimise get_order()")
    Signed-off-by: Qian Cai <cai@lca.pw>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Cc: David S. Miller <davem@davemloft.net>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Cc: David Howells <dhowells@redhat.com>
    Cc: Jakub Jelinek <jakub@redhat.com>
    Cc: Nick Desaulniers <ndesaulniers@google.com>
    Cc: Bill Wendling <morbo@google.com>
    Cc: James Y Knight <jyknight@google.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dcf7863f10783f1ec9a41df9d4be18f264923cf9
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Fri Aug 2 21:49:02 2019 -0700

    page flags: prioritize kasan bits over last-cpuid

    [ Upstream commit ee38d94a0ad89890b770f6c876263cf9fcbfde84 ]

    ARM64 randdconfig builds regularly run into a build error, especially
    when NUMA_BALANCING and SPARSEMEM are enabled but not SPARSEMEM_VMEMMAP:

      #error "KASAN: not enough bits in page flags for tag"

    The last-cpuid bits are already contitional on the available space, so
    the result of the calculation is a bit random on whether they were
    already left out or not.

    Adding the kasan tag bits before last-cpuid makes it much more likely to
    end up with a successful build here, and should be reliable for
    randconfig at least, as long as that does not randomize NR_CPUS or
    NODES_SHIFT but uses the defaults.

    In order for the modified check to not trigger in the x86 vdso32 code
    where all constants are wrong (building with -m32), enclose all the
    definitions with an #ifdef.

    [arnd@arndb.de: build fix]
      Link:
http://lkml.kernel.org/r/CAK8P3a3Mno1SWTcuAOT0Wa9VS15pdU6EfnkxLbDpyS55yO04+g@mail.gmail.com
    Link: http://lkml.kernel.org/r/20190722115520.3743282-1-arnd@arndb.de
    Link: https://lore.kernel.org/lkml/20190618095347.3850490-1-arnd@arndb.de/
    Fixes: 2813b9c02962 ("kasan, mm, arm64: tag non slab memory allocated via pagealloc")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Andrey Konovalov <andreyknvl@google.com>
    Reviewed-by: Andrey Ryabinin <aryabinin@virtuozzo.com>
    Cc: Andrey Konovalov <andreyknvl@google.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
```

```
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: Christoph Lameter <cl@linux.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8abc1d5f1c9e03bc929b3d1319404bfc1b22b2da
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Fri Aug 2 21:48:40 2019 -0700

    ocfs2: remove set but not used variable 'last_hash'

    [ Upstream commit 7bc36e3ce91471b6377c8eadc0a2f220a2280083 ]

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

      fs/ocfs2/xattr.c: In function ocfs2_xattr_bucket_find:
      fs/ocfs2/xattr.c:3828:6: warning: variable last_hash set but not used [-Wunused-but-
set-variable]

    It's never used and can be removed.

    Link: http://lkml.kernel.org/r/20190716132110.34836-1-yuehaibing@huawei.com
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Acked-by: Joseph Qi <joseph.qi@linux.alibaba.com>
    Cc: Mark Fasheh <mark@fasheh.com>
    Cc: Joel Becker <jlbec@evilplan.org>
    Cc: Junxiao Bi <junxiao.bi@oracle.com>
    Cc: Changwei Ge <gechangwei@live.cn>
    Cc: Gang He <ghe@suse.com>
    Cc: Jun Piao <piaojun@huawei.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b1d93b7227ec59f3430dbfd6405cc634282549d5
Author: Yang Shi <yang.shi@linux.alibaba.com>
Date:   Fri Aug 2 21:48:37 2019 -0700

    Revert "kmemleak: allow to coexist with fault injection"

    [ Upstream commit df9576def004d2cd5beedc00cb6e8901427634b9 ]

    When running ltp's oom test with kmemleak enabled, the below warning was
    triggerred since kernel detects __GFP_NOFAIL & ~__GFP_DIRECT_RECLAIM is
    passed in:

      WARNING: CPU: 105 PID: 2138 at mm/page_alloc.c:4608
__alloc_pages_nodemask+0x1c31/0x1d50
      Modules linked in: loop dax_pmem dax_pmem_core ip_tables x_tables xfs virtio_net
net_failover virtio_blk failover ata_generic virtio_pci virtio_ring virtio libata
      CPU: 105 PID: 2138 Comm: oom01 Not tainted 5.2.0-next-20190710+ #7
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.10.2-0-g5f4c7b1-
prebuilt.qemu-project.org 04/01/2014
      RIP: 0010:__alloc_pages_nodemask+0x1c31/0x1d50
      ...
       kmemleak_alloc+0x4e/0xb0
       kmem_cache_alloc+0x2a7/0x3e0
       mempool_alloc_slab+0x2d/0x40
       mempool_alloc+0x118/0x2b0
       bio_alloc_bioset+0x19d/0x350
       get_swap_bio+0x80/0x230
       __swap_writepage+0x5ff/0xb20
```

```
    The mempool_alloc_slab() clears __GFP_DIRECT_RECLAIM, however kmemleak
    has __GFP_NOFAIL set all the time due to d9570ee3bd1d4f2 ("kmemleak:
    allow to coexist with fault injection").  But, it doesn't make any sense
    to have __GFP_NOFAIL and ~__GFP_DIRECT_RECLAIM specified at the same
    time.

    According to the discussion on the mailing list, the commit should be
    reverted for short term solution.  Catalin Marinas would follow up with
    a better solution for longer term.

    The failure rate of kmemleak metadata allocation may increase in some
    circumstances, but this should be expected side effect.

    Link: http://lkml.kernel.org/r/1563299431-111710-1-git-send-email-
yang.shi@linux.alibaba.com
    Fixes: d9570ee3bd1d4f2 ("kmemleak: allow to coexist with fault injection")
    Signed-off-by: Yang Shi <yang.shi@linux.alibaba.com>
    Suggested-by: Catalin Marinas <catalin.marinas@arm.com>
    Acked-by: Michal Hocko <mhocko@suse.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Qian Cai <cai@lca.pw>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e058f41de0cdc516071925fff6daa2327859b0cf
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date:   Thu Aug 1 23:36:14 2019 +0900

    arm64: Make debug exception handlers visible from RCU

    [ Upstream commit d8bb6718c4db9bcd075dde7ff55d46091ccfae15 ]

    Make debug exceptions visible from RCU so that synchronize_rcu()
    correctly track the debug exception handler.

    This also introduces sanity checks for user-mode exceptions as same
    as x86's ist_enter()/ist_exit().

    The debug exception can interrupt in idle task. For example, it warns
    if we put a kprobe on a function called from idle task as below.
    The warning message showed that the rcu_read_lock() caused this
    problem. But actually, this means the RCU is lost the context which
    is already in NMI/IRQ.

      /sys/kernel/debug/tracing # echo p default_idle_call >> kprobe_events
      /sys/kernel/debug/tracing # echo 1 > events/kprobes/enable
      /sys/kernel/debug/tracing # [  135.122237]
      [  135.125035] =============================
      [  135.125310] WARNING: suspicious RCU usage
      [  135.125581] 5.2.0-08445-g9187c508bdc7 #20 Not tainted
      [  135.125904] -----------------------------
      [  135.126205] include/linux/rcupdate.h:594 rcu_read_lock() used illegally while idle!
      [  135.126839]
      [  135.126839] other info that might help us debug this:
      [  135.126839]
      [  135.127410]
      [  135.127410] RCU used illegally from idle CPU!
      [  135.127410] rcu_scheduler_active = 2, debug_locks = 1
      [  135.128114] RCU used illegally from extended quiescent state!
      [  135.128555] 1 lock held by swapper/0/0:
      [  135.128944]  #0: (____ptrval____) (rcu_read_lock){....}, at:
call_break_hook+0x0/0x178
```

```
        [  135.130499]
        [  135.130499] stack backtrace:
        [  135.131192] CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.2.0-08445-g9187c508bdc7 #20
        [  135.131841] Hardware name: linux,dummy-virt (DT)
        [  135.132224] Call trace:
        [  135.132491]  dump_backtrace+0x0/0x140
        [  135.132806]  show_stack+0x24/0x30
        [  135.133133]  dump_stack+0xc4/0x10c
        [  135.133726]  lockdep_rcu_suspicious+0xf8/0x108
        [  135.134171]  call_break_hook+0x170/0x178
        [  135.134486]  brk_handler+0x28/0x68
        [  135.134792]  do_debug_exception+0x90/0x150
        [  135.135051]  el1_dbg+0x18/0x8c
        [  135.135260]  default_idle_call+0x0/0x44
        [  135.135516]  cpu_startup_entry+0x2c/0x30
        [  135.135815]  rest_init+0x1b0/0x280
        [  135.136044]  arch_call_rest_init+0x14/0x1c
        [  135.136305]  start_kernel+0x4d4/0x500
        [  135.136597]

    So make debug exception visible to RCU can fix this warning.

    Reported-by: Naresh Kamboju <naresh.kamboju@linaro.org>
    Acked-by: Paul E. McKenney <paulmck@linux.ibm.com>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e38e8477654451a453740ac00693d509db03cf1c
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date:   Thu Aug 1 23:25:49 2019 +0900

    arm64: kprobes: Recover pstate.D in single-step exception handler

    [ Upstream commit b3980e48528c4d2a9e70b145a5bba328b73a0f93 ]

    kprobes manipulates the interrupted PSTATE for single step, and
    doesn't restore it. Thus, if we put a kprobe where the pstate.D
    (debug) masked, the mask will be cleared after the kprobe hits.

    Moreover, in the most complicated case, this can lead a kernel
    crash with below message when a nested kprobe hits.

    [  152.118921] Unexpected kernel single-step exception at EL1

    When the 1st kprobe hits, do_debug_exception() will be called.
    At this point, debug exception (= pstate.D) must be masked (=1).
    But if another kprobes hits before single-step of the first kprobe
    (e.g. inside user pre_handler), it unmask the debug exception
    (pstate.D = 0) and return.
    Then, when the 1st kprobe setting up single-step, it saves current
    DAIF, mask DAIF, enable single-step, and restore DAIF.
    However, since "D" flag in DAIF is cleared by the 2nd kprobe, the
    single-step exception happens soon after restoring DAIF.

    This has been introduced by commit 7419333fa15e ("arm64: kprobe:
    Always clear pstate.D in breakpoint exception handler")

    To solve this issue, this stores all DAIF bits and restore it
    after single stepping.

    Reported-by: Naresh Kamboju <naresh.kamboju@linaro.org>
    Fixes: 7419333fa15e ("arm64: kprobe: Always clear pstate.D in breakpoint exception
handler")
    Reviewed-by: James Morse <james.morse@arm.com>
```

```
    Tested-by: James Morse <james.morse@arm.com>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 49d9e6c8a32b3e355d1ab23cf33f08dbee2f15af
Author: Colin Ian King <colin.king@canonical.com>
Date:   Mon Jul 22 23:25:35 2019 +0100

    drm/exynos: fix missing decrement of retry counter

    [ Upstream commit 1bbbab097a05276e312dd2462791d32b21ceb1ee ]

    Currently the retry counter is not being decremented, leading to a
    potential infinite spin if the scalar_reads don't change state.

    Addresses-Coverity: ("Infinite loop")
    Fixes: 280e54c9f614 ("drm/exynos: scaler: Reset hardware before starting the operation")
    Signed-off-by: Colin Ian King <colin.king@canonical.com>
    Signed-off-by: Inki Dae <inki.dae@samsung.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1d48d90caacbbf906aeed7b23936c45a854c497a
Author: Wei Yongjun <weiyongjun1@huawei.com>
Date:   Thu Aug 1 01:27:25 2019 +0000

    RDMA/hns: Fix error return code in hns_roce_v1_rsv_lp_qp()

    [ Upstream commit 020fb3bebc224dfe9353a56ecbe2d5fac499dffc ]

    Fix to return error code -ENOMEM from the rdma_zalloc_drv_obj() error
    handling case instead of 0, as done elsewhere in this function.

    Fixes: e8ac9389f0d7 ("RDMA: Fix allocation failure on pointer pd")
    Fixes: 21a428a019c9 ("RDMA: Handle PD allocations by IB/core")
    Signed-off-by: Wei Yongjun <weiyongjun1@huawei.com>
    Reviewed-by: Leon Romanovsky <leonro@mellanox.com>
    Link: https://lore.kernel.org/r/20190801012725.150493-1-weiyongjun1@huawei.com
    Signed-off-by: Doug Ledford <dledford@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f9075dea4a7dce15df92fd5b409d8965b63ecd20
Author: Jeffrey Hugo <jeffrey.l.hugo@gmail.com>
Date:   Wed Jun 26 11:00:15 2019 -0700

    drm: msm: Fix add_gpu_components

    [ Upstream commit 9ca7ad6c7706edeae331c1632d0c63897418ebad ]

    add_gpu_components() adds found GPU nodes from the DT to the match list,
    regardless of the status of the nodes.  This is a problem, because if the
    nodes are disabled, they should not be on the match list because they will
    not be matched.  This prevents display from initing if a GPU node is
    defined, but it's status is disabled.

    Fix this by checking the node's status before adding it to the match list.

    Fixes: dc3ea265b856 (drm/msm: Drop the gpu binding)
    Reviewed-by: Rob Clark <robdclark@gmail.com>
    Signed-off-by: Jeffrey Hugo <jeffrey.l.hugo@gmail.com>
    Signed-off-by: Sean Paul <seanpaul@chromium.org>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190626180015.45242-1-
jeffrey.l.hugo@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit febe356e3589b1d4f294b6d74d60ae529524e077
Author: Leon Romanovsky <leonro@mellanox.com>
Date:   Wed Jul 31 11:38:52 2019 +0300

    RDMA/mlx5: Release locks during notifier unregister

    [ Upstream commit 23eaf3b5c1a755e3193480c76fb29414be648688 ]

    The below kernel panic was observed when created bond mode LACP
    with GRE tunnel on top. The reason to it was not released spinlock
    during mlx5 notify unregsiter sequence.

    [  234.562007] BUG: scheduling while atomic: sh/10900/0x00000002
    [  234.563005] Preemption disabled at:
    [  234.566864] ------------[ cut here ]------------
    [  234.567120] DEBUG_LOCKS_WARN_ON(val > preempt_count())
    [  234.567139] WARNING: CPU: 16 PID: 10900 at kernel/sched/core.c:3203
preempt_count_sub+0xca/0x170
    [  234.569550] CPU: 16 PID: 10900 Comm: sh Tainted: G        W 5.2.0-rc1-for-linust-dbg-
2019-05-25_04-57-33-60 #1
    [  234.569886] Hardware name: Dell Inc. PowerEdge R720/0X3D66, BIOS 2.6.1 02/12/2018
    [  234.570183] RIP: 0010:preempt_count_sub+0xca/0x170
    [  234.570404] Code: 03 38
    d0 7c 08 84 d2 0f 85 b0 00 00 00 8b 15 dd 02 03 04 85 d2 75 ba 48 c7 c6
    00 e1 88 83 48 c7 c7 40 e1 88 83 e8 76 11 f7 ff <0f> 0b 5b c3 65 8b 05
    d3 1f d8 7e 84 c0 75 82 e8 62 c3 c3 00 85 c0
    [  234.570911] RSP: 0018:ffff888b94477b08 EFLAGS: 00010286
    [  234.571133] RAX: 0000000000000000 RBX: 0000000000000001 RCX: 0000000000000000
    [  234.571391] RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000246
    [  234.571648] RBP: ffff888ba5560000 R08: fffffbfff08962d5 R09: fffffbfff08962d5
    [  234.571902] R10: 0000000000000001 R11: fffffbfff08962d4 R12: ffff888bac6e9548
    [  234.572157] R13: ffff888babfaf728 R14: ffff888bac6e9568 R15: ffff888babfaf750
    [  234.572412] FS: 00007fcafa59b740(0000) GS:ffff888bed200000(0000)
knlGS:0000000000000000
    [  234.572686] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [  234.572914] CR2: 00007f984f16b140 CR3: 0000000b2bf0a001 CR4: 00000000001606e0
    [  234.573172] Call Trace:
    [  234.573336] _raw_spin_unlock+0x2e/0x50
    [  234.573542] mlx5_ib_unbind_slave_port+0x1bc/0x690 [mlx5_ib]
    [  234.573793] mlx5_ib_cleanup_multiport_master+0x1d3/0x660 [mlx5_ib]
    [  234.574039] mlx5_ib_stage_init_cleanup+0x4c/0x360 [mlx5_ib]
    [  234.574271]  ? kfree+0xf5/0x2f0
    [  234.574465] __mlx5_ib_remove+0x61/0xd0 [mlx5_ib]
    [  234.574688]  ? __mlx5_ib_remove+0xd0/0xd0 [mlx5_ib]
    [  234.574951] mlx5_remove_device+0x234/0x300 [mlx5_core]
    [  234.575224] mlx5_unregister_device+0x4d/0x1e0 [mlx5_core]
    [  234.575493] remove_one+0x4f/0x160 [mlx5_core]
    [  234.575704] pci_device_remove+0xef/0x2a0
    [  234.581407]  ? pcibios_free_irq+0x10/0x10
    [  234.587143]  ? up_read+0xc1/0x260
    [  234.592785] device_release_driver_internal+0x1ab/0x430
    [  234.598442] unbind_store+0x152/0x200
    [  234.604064]  ? sysfs_kf_write+0x3b/0x180
    [  234.609441]  ? sysfs_file_ops+0x160/0x160
    [  234.615021] kernfs_fop_write+0x277/0x440
    [  234.620288]  ? __sb_start_write+0x1ef/0x2c0
    [  234.625512] vfs_write+0x15e/0x460
    [  234.630786] ksys_write+0x156/0x1e0
    [  234.635988]  ? __ia32_sys_read+0xb0/0xb0
    [  234.641120]  ? trace_hardirqs_off_thunk+0x1a/0x1c
    [  234.646163] do_syscall_64+0x95/0x470
    [  234.651106] entry_SYSCALL_64_after_hwframe+0x49/0xbe
    [  234.656004] RIP: 0033:0x7fcaf9c9cfd0
    [  234.660686] Code: 73 01
    c3 48 8b 0d c0 6e 2d 00 f7 d8 64 89 01 48 83 c8 ff c3 66 0f 1f 44 00 00
```

```
     83 3d cd cf 2d 00 00 75 10 b8 01 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73
     31 c3 48 83 ec 08 e8 ee cb 01 00 48 89 04 24
     [  234.670128] RSP: 002b:00007ffd3b01ddd8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
     [  234.674811] RAX: ffffffffffffffda RBX: 000000000000000d RCX: 00007fcaf9c9cfd0
     [  234.679387] RDX: 000000000000000d RSI: 00007fcafa5c1000 RDI: 0000000000000001
     [  234.683848] RBP: 00007fcafa5c1000 R08: 000000000000000a R09: 00007fcafa59b740
     [  234.688167] R10: 00007ffd3b01d8e0 R11: 0000000000000246 R12: 00007fcaf9f75400
     [  234.692386] R13: 000000000000000d R14: 0000000000000001 R15: 0000000000000000
     [  234.696495] irq event stamp: 153067
     [  234.700525] hardirqs last enabled at (153067): [<ffffffff83258c39>]
 _raw_spin_unlock_irqrestore+0x59/0x70
     [  234.704665] hardirqs last disabled at (153066): [<ffffffff83259382>]
 _raw_spin_lock_irqsave+0x22/0x90
     [  234.708722] softirqs last enabled at (153058): [<ffffffff836006c5>]
 __do_softirq+0x6c5/0xb4e
     [  234.712673] softirqs last disabled at (153051): [<ffffffff81227c1d>]
 irq_exit+0x17d/0x1d0
     [  234.716601] ---[ end trace 5dbf096843ee9ce6 ]---

     Fixes: df097a278c75 ("IB/mlx5: Use the new mlx5 core notifier API")
     Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
     Link: https://lore.kernel.org/r/20190731083852.584-1-leon@kernel.org
     Signed-off-by: Doug Ledford <dledford@redhat.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 649d927da902b27e9758c90b11d8d67ccf326e04
 Author: Jack Morgenstein <jackm@dev.mellanox.co.il>
 Date:   Thu Aug 1 15:14:49 2019 +0300

     IB/mad: Fix use-after-free in ib mad completion handling

     [ Upstream commit 770b7d96cfff6a8bf6c9f261ba6f135dc9edf484 ]

     We encountered a use-after-free bug when unloading the driver:

     [ 3562.116059] BUG: KASAN: use-after-free in ib_mad_post_receive_mads+0xddc/0xed0
 [ib_core]
     [ 3562.117233] Read of size 4 at addr ffff8882ca5aa868 by task kworker/u13:2/23862
     [ 3562.118385]
     [ 3562.119519] CPU: 2 PID: 23862 Comm: kworker/u13:2 Tainted: G           OE     5.1.0-
 for-upstream-dbg-2019-05-19_16-44-30-13 #1
     [ 3562.121806] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Ubuntu-1.8.2-
 1ubuntu2 04/01/2014
     [ 3562.123075] Workqueue: ib-comp-unb-wq ib_cq_poll_work [ib_core]
     [ 3562.124383] Call Trace:
     [ 3562.125640]  dump_stack+0x9a/0xeb
     [ 3562.126911]  print_address_description+0xe3/0x2e0
     [ 3562.128223]  ? ib_mad_post_receive_mads+0xddc/0xed0 [ib_core]
     [ 3562.129545]  __kasan_report+0x15c/0x1df
     [ 3562.130866]  ? ib_mad_post_receive_mads+0xddc/0xed0 [ib_core]
     [ 3562.132174]  kasan_report+0xe/0x20
     [ 3562.133514]  ib_mad_post_receive_mads+0xddc/0xed0 [ib_core]
     [ 3562.134835]  ? find_mad_agent+0xa00/0xa00 [ib_core]
     [ 3562.136158]  ? qlist_free_all+0x51/0xb0
     [ 3562.137498]  ? mlx4_ib_sqp_comp_worker+0x1970/0x1970 [mlx4_ib]
     [ 3562.138833]  ? quarantine_reduce+0x1fa/0x270
     [ 3562.140171]  ? kasan_unpoison_shadow+0x30/0x40
     [ 3562.141522]  ib_mad_recv_done+0xdf6/0x3000 [ib_core]
     [ 3562.142880]  ? _raw_spin_unlock_irqrestore+0x46/0x70
     [ 3562.144277]  ? ib_mad_send_done+0x1810/0x1810 [ib_core]
     [ 3562.145649]  ? mlx4_ib_destroy_cq+0x2a0/0x2a0 [mlx4_ib]
     [ 3562.147008]  ? _raw_spin_unlock_irqrestore+0x46/0x70
     [ 3562.148380]  ? debug_object_deactivate+0x2b9/0x4a0
     [ 3562.149814]  __ib_process_cq+0xe2/0x1d0 [ib_core]
     [ 3562.151195]  ib_cq_poll_work+0x45/0xf0 [ib_core]
```

```
     [ 3562.152577]  process_one_work+0x90c/0x1860
     [ 3562.153959]  ? pwq_dec_nr_in_flight+0x320/0x320
     [ 3562.155320]  worker_thread+0x87/0xbb0
     [ 3562.156687]  ? __kthread_parkme+0xb6/0x180
     [ 3562.158058]  ? process_one_work+0x1860/0x1860
     [ 3562.159429]  kthread+0x320/0x3e0
     [ 3562.161391]  ? kthread_park+0x120/0x120
     [ 3562.162744]  ret_from_fork+0x24/0x30
     ...
     [ 3562.187615] Freed by task 31682:
     [ 3562.188602]  save_stack+0x19/0x80
     [ 3562.189586]  __kasan_slab_free+0x11d/0x160
     [ 3562.190571]  kfree+0xf5/0x2f0
     [ 3562.191552]  ib_mad_port_close+0x200/0x380 [ib_core]
     [ 3562.192538]  ib_mad_remove_device+0xf0/0x230 [ib_core]
     [ 3562.193538]  remove_client_context+0xa6/0xe0 [ib_core]
     [ 3562.194514]  disable_device+0x14e/0x260 [ib_core]
     [ 3562.195488]  __ib_unregister_device+0x79/0x150 [ib_core]
     [ 3562.196462]  ib_unregister_device+0x21/0x30 [ib_core]
     [ 3562.197439]  mlx4_ib_remove+0x162/0x690 [mlx4_ib]
     [ 3562.198408]  mlx4_remove_device+0x204/0x2c0 [mlx4_core]
     [ 3562.199381]  mlx4_unregister_interface+0x49/0x1d0 [mlx4_core]
     [ 3562.200356]  mlx4_ib_cleanup+0xc/0x1d [mlx4_ib]
     [ 3562.201329]  __x64_sys_delete_module+0x2d2/0x400
     [ 3562.202288]  do_syscall_64+0x95/0x470
     [ 3562.203277]  entry_SYSCALL_64_after_hwframe+0x49/0xbe

     The problem was that the MAD PD was deallocated before the MAD CQ.
     There was completion work pending for the CQ when the PD got deallocated.
     When the mad completion handling reached procedure
     ib_mad_post_receive_mads(), we got a use-after-free bug in the following
     line of code in that procedure:
        sg_list.lkey = qp_info->port_priv->pd->local_dma_lkey;
     (the pd pointer in the above line is no longer valid, because the
     pd has been deallocated).

     We fix this by allocating the PD before the CQ in procedure
     ib_mad_port_open(), and deallocating the PD after freeing the CQ
     in procedure ib_mad_port_close().

     Since the CQ completion work queue is flushed during ib_free_cq(),
     no completions will be pending for that CQ when the PD is later
     deallocated.

     Note that freeing the CQ before deallocating the PD is the practice
     in the ULPs.

     Fixes: 4be90bc60df4 ("IB/mad: Remove ib_get_dma_mr calls")
     Signed-off-by: Jack Morgenstein <jackm@dev.mellanox.co.il>
     Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
     Link: https://lore.kernel.org/r/20190801121449.24973-1-leon@kernel.org
     Signed-off-by: Doug Ledford <dledford@redhat.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 421c77400ed0a04a84d4c3ac83f1d933ff044623
  Author: Gal Pressman <galpress@amazon.com>
  Date:   Thu Aug 1 13:43:54 2019 +0300

     RDMA/restrack: Track driver QP types in resource tracker

     [ Upstream commit 52e0a118a20308dd6aa531e20a5ab5907d2264c8 ]

     The check for QP type different than XRC has excluded driver QP
     types from the resource tracker.
     As a result, "rdma resource show" user command would not show opened
```

```
    driver QPs which does not reflect the real state of the system.

    Check QP type explicitly instead of assuming enum values/ordering.

    Fixes: 40909f664d27 ("RDMA/efa: Add EFA verbs implementation")
    Signed-off-by: Gal Pressman <galpress@amazon.com>
    Reviewed-by: Leon Romanovsky <leonro@mellanox.com>
    Link: https://lore.kernel.org/r/20190801104354.11417-1-galpress@amazon.com
    Signed-off-by: Doug Ledford <dledford@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b542fe8c78b417ab9160cd47493597f3299ad22c
Author: Guy Levi <guyle@mellanox.com>
Date:   Wed Jul 31 11:19:29 2019 +0300

    IB/mlx5: Fix MR registration flow to use UMR properly

    [ Upstream commit e5366d309a772fef264ec85e858f9ea46f939848 ]

    Driver shouldn't allow to use UMR to register a MR when
    umr_modify_atomic_disabled is set. Otherwise it will always end up with a
    failure in the post send flow which sets the UMR WQE to modify atomic access
    right.

    Fixes: c8d75a980fab ("IB/mlx5: Respect new UMR capabilities")
    Signed-off-by: Guy Levi <guyle@mellanox.com>
    Reviewed-by: Moni Shoua <monis@mellanox.com>
    Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
    Link: https://lore.kernel.org/r/20190731081929.32559-1-leon@kernel.org
    Signed-off-by: Doug Ledford <dledford@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7e1b53037f60a42379b90e9c38b605645c17eb58
Author: Luck, Tony <tony.luck@intel.com>
Date:   Tue Jul 30 21:39:57 2019 -0700

    IB/core: Add mitigation for Spectre V1

    [ Upstream commit 61f259821dd3306e49b7d42a3f90fb5a4ff3351b ]

    Some processors may mispredict an array bounds check and
    speculatively access memory that they should not. With
    a user supplied array index we like to play things safe
    by masking the value with the array size before it is
    used as an index.

    Signed-off-by: Tony Luck <tony.luck@intel.com>
    Link: https://lore.kernel.org/r/20190731043957.GA1600@agluck-desk2.amr.corp.intel.com
    Signed-off-by: Doug Ledford <dledford@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb67ebbc0f6e92a970e39735c3bdfd07a06cb219
Author: Qian Cai <cai@lca.pw>
Date:   Thu Aug 1 10:47:05 2019 -0400

    arm64/mm: fix variable 'tag' set but not used

    [ Upstream commit 7732d20a160c76006c7fe7bca5178aea6af1d2e8 ]

    When CONFIG_KASAN_SW_TAGS=n, set_tag() is compiled away. GCC throws a
    warning,

    mm/kasan/common.c: In function '__kasan_kmalloc':
    mm/kasan/common.c:464:5: warning: variable 'tag' set but not used
    [-Wunused-but-set-variable]
```

```
       u8 tag = 0xff;
          ^~~
```

    Fix it by making __tag_set() a static inline function the same as
    arch_kasan_set_tag() in mm/kasan/kasan.h for consistency because there
    is a macro in arch/arm64/include/asm/kasan.h,

```
     #define arch_kasan_set_tag(addr, tag) __tag_set(addr, tag)
```

    However, when CONFIG_DEBUG_VIRTUAL=n and CONFIG_SPARSEMEM_VMEMMAP=y,
    page_to_virt() will call __tag_set() with incorrect type of a
    parameter, so fix that as well. Also, still let page_to_virt() return
    "void *" instead of "const void *", so will not need to add a similar
    cast in lowmem_page_address().

    Signed-off-by: Qian Cai <cai@lca.pw>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c676c48cdc2065e387d634b4fea850fde986fe39
Author: Qian Cai <cai@lca.pw>
Date: Wed Jul 31 16:05:45 2019 -0400

    arm64/mm: fix variable 'pud' set but not used

    [Upstream commit 7d4e2dcf311d3b98421d1f119efe5964cafa32fc]

    GCC throws a warning,

```
    arch/arm64/mm/mmu.c: In function 'pud_free_pmd_page':
    arch/arm64/mm/mmu.c:1033:8: warning: variable 'pud' set but not used
    [-Wunused-but-set-variable]
      pud_t pud;
            ^~~
```

    because pud_table() is a macro and compiled away. Fix it by making it a
    static inline function and for pud_sect() as well.

    Signed-off-by: Qian Cai <cai@lca.pw>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f82fecbba30adc3cb3ec86dfdb69fdc5f6b30ed3
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date: Thu Jul 25 17:16:05 2019 +0900

    arm64: unwind: Prohibit probing on return_address()

    [Upstream commit ee07b93e7721ccd5d5b9fa6f0c10cb3fe2f1f4f9]

    Prohibit probing on return_address() and subroutines which
    is called from return_address(), since the it is invoked from
    trace_hardirqs_off() which is also kprobe blacklisted.

    Reported-by: Naresh Kamboju <naresh.kamboju@linaro.org>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5c15fca6df6f3917420e5dbd6c0a35000abb6a17
Author: Julien Thierry <julien.thierry.kdev@gmail.com>
Date: Mon Jul 29 15:57:46 2019 +0100

    arm64: Lower priority mask for GIC_PRIO_IRQON

```
    [ Upstream commit 677379bc9139ac24b310a281fcb21a2f04288353 ]

    On a system with two security states, if SCR_EL3.FIQ is cleared,
    non-secure IRQ priorities get shifted to fit the secure view but
    priority masks aren't.

    On such system, it turns out that GIC_PRIO_IRQON masks the priority of
    normal interrupts, which obviously ends up in a hang.

    Increase GIC_PRIO_IRQON value (i.e. lower priority) to make sure
    interrupts are not blocked by it.

    Cc: Oleg Nesterov <oleg@redhat.com>
    Fixes: bd82d4bd21880b7c ("arm64: Fix incorrect irqflag restore for priority masking")
    Acked-by: Marc Zyngier <marc.zyngier@arm.com>
    Signed-off-by: Julien Thierry <julien.thierry.kdev@gmail.com>
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    [will: fixed Fixes: tag]
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1b6336c844dc7f1f962086c25d6fac9dce178515
Author: Mao Han <han_mao@c-sky.com>
Date:   Thu Jul 11 10:38:40 2019 +0800

    riscv: Fix perf record without libelf support

    [ Upstream commit b399abe7c21e248dc6224cadc9a378a2beb10cfd ]

    This patch fix following perf record error by linking vdso.so with
    build id.

    perf.data      perf.data.old
    [ perf record: Woken up 1 times to write data ]
    free(): double free detected in tcache 2
    Aborted

    perf record use filename__read_build_id(util/symbol-minimal.c) to get
    build id when libelf is not supported. When vdso.so is linked without
    build id, the section size of PT_NOTE will be zero, buf size will
    realloc to zero and cause memory corruption.

    Signed-off-by: Mao Han <han_mao@c-sky.com>
    Cc: Paul Walmsley <paul.walmsley@sifive.com>
    Cc: Palmer Dabbelt <palmer@sifive.com>
    Cc: Albert Ou <aou@eecs.berkeley.edu>
    Signed-off-by: Paul Walmsley <paul.walmsley@sifive.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 174cde5be1a83dc59af9fbf89176ee091b0be612
Author: Qian Cai <cai@lca.pw>
Date:   Tue Jul 30 17:23:48 2019 -0400

    arm64/efi: fix variable 'si' set but not used

    [ Upstream commit f1d4836201543e88ebe70237e67938168d5fab19 ]

    GCC throws out this warning on arm64.

    drivers/firmware/efi/libstub/arm-stub.c: In function 'efi_entry':
    drivers/firmware/efi/libstub/arm-stub.c:132:22: warning: variable 'si'
    set but not used [-Wunused-but-set-variable]

    Fix it by making free_screen_info() a static inline function.
```

```
    Acked-by: Will Deacon <will@kernel.org>
    Signed-off-by: Qian Cai <cai@lca.pw>
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a94d43c0dadcd4f76f949af936e103f7c6a934e8
Author: Stephen Boyd <swboyd@chromium.org>
Date:   Tue Jul 30 09:48:03 2019 -0700

    kbuild: Check for unknown options with cc-option usage in Kconfig and clang

    [ Upstream commit e8de12fb7cde2c85bc31097cd098da79a4818305 ]

    If the particular version of clang a user has doesn't enable
    -Werror=unknown-warning-option by default, even though it is the
    default[1], then make sure to pass the option to the Kconfig cc-option
    command so that testing options from Kconfig files works properly.
    Otherwise, depending on the default values setup in the clang toolchain
    we will silently assume options such as -Wmaybe-uninitialized are
    supported by clang, when they really aren't.

    A compilation issue only started happening for me once commit
    589834b3a009 ("kbuild: Add -Werror=unknown-warning-option to
    CLANG_FLAGS") was applied on top of commit b303c6df80c9 ("kbuild:
    compute false-positive -Wmaybe-uninitialized cases in Kconfig"). This
    leads kbuild to try and test for the existence of the
    -Wmaybe-uninitialized flag with the cc-option command in
    scripts/Kconfig.include, and it doesn't see an error returned from the
    option test so it sets the config value to Y. Then the Makefile tries to
    pass the unknown option on the command line and
    -Werror=unknown-warning-option catches the invalid option and breaks the
    build. Before commit 589834b3a009 ("kbuild: Add
    -Werror=unknown-warning-option to CLANG_FLAGS") the build works fine,
    but any cc-option test of a warning option in Kconfig files silently
    evaluates to true, even if the warning option flag isn't supported on
    clang.

    Note: This doesn't change cc-option usages in Makefiles because those
    use a different rule that includes KBUILD_CFLAGS by default (see the
    __cc-option command in scripts/Kbuild.incluide). The KBUILD_CFLAGS
    variable already has the -Werror=unknown-warning-option flag set. Thanks
    to Doug for pointing out the different rule.

    [1] https://clang.llvm.org/docs/DiagnosticsReference.html#wunknown-warning-option
    Cc: Peter Smith <peter.smith@linaro.org>
    Cc: Nick Desaulniers <ndesaulniers@google.com>
    Cc: Douglas Anderson <dianders@chromium.org>
    Signed-off-by: Stephen Boyd <swboyd@chromium.org>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ab5565b2dfe5a08c7060022f2ba2e175d286af88
Author: Masahiro Yamada <yamada.masahiro@socionext.com>
Date:   Wed Jul 31 00:59:00 2019 +0900

    kbuild: modpost: handle KBUILD_EXTRA_SYMBOLS only for external modules

    [ Upstream commit cb4819934a7f9b87876f11ed05b8624c0114551b ]

    KBUILD_EXTRA_SYMBOLS makes sense only when building external modules.
    Moreover, the modpost sets 'external_module' if the -e option is given.

    I replaced $(patsubst %, -e %,...) with simpler $(addprefix -e,...)
    while I was here.
```

```
    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c07b8aab75d913e220caa7891a0791b97972f4a7
Author: Miquel Raynal <miquel.raynal@bootlin.com>
Date:   Wed Jul 31 14:26:51 2019 +0200

    ata: libahci: do not complain in case of deferred probe

    [ Upstream commit 090bb803708198e5ab6b0046398c7ed9f4d12d6b ]

    Retrieving PHYs can defer the probe, do not spawn an error when
    -EPROBE_DEFER is returned, it is normal behavior.

    Fixes: b1a9edbda040 ("ata: libahci: allow to use multiple PHYs")
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5003c12aeda59dfd789f6205b0670285b849ea2d
Author: Wang Xiayang <xywang.sjtu@sjtu.edu.cn>
Date:   Sat Jul 27 17:30:30 2019 +0800

    drm/amdgpu: fix a potential information leaking bug

    [ Upstream commit 929e571c04c285861e0bb049a396a2bdaea63282 ]

    Coccinelle reports a path that the array "data" is never initialized.
    The path skips the checks in the conditional branches when either
    of callback functions, read_wave_vgprs and read_wave_sgprs, is not
    registered. Later, the uninitialized "data" array is read
    in the while-loop below and passed to put_user().

    Fix the path by allocating the array with kcalloc().

    The patch is simplier than adding a fall-back branch that explicitly
    calls memset(data, 0, ...). Also it does not need the multiplication
    1024*sizeof(*data) as the size parameter for memset() though there is
    no risk of integer overflow.

    Signed-off-by: Wang Xiayang <xywang.sjtu@sjtu.edu.cn>
    Reviewed-by: Chunming Zhou <david1.zhou@amd.com>
    Reviewed-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 22a7a24d4999032fb1faec5f244477e03d67e14b
Author: Christian König <christian.koenig@amd.com>
Date:   Tue Jul 30 11:17:03 2019 +0200

    drm/amdgpu: fix error handling in amdgpu_cs_process_fence_dep

    [ Upstream commit 67d0859e2758ef992fd32499747ce4b1038a63c0 ]

    We always need to drop the ctx reference and should check
    for errors first and then dereference the fence pointer.

    Signed-off-by: Christian König <christian.koenig@amd.com>
    Reviewed-by: Chunming Zhou <david1.zhou@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 08283dd59130956ec8ffd92b88e08db6476c97f2
```

```
Author: Evan Quan <evan.quan@amd.com>
Date:   Thu Jul 25 12:10:34 2019 +0800

    drm/amd/powerplay: fix null pointer dereference around dpm state relates

    [ Upstream commit 479156f2e5540077377a823eaf5a4263bd329063 ]

    DPM state relates are not supported on the new SW SMU ASICs. But still
    it's not OK to trigger null pointer dereference on accessing them.

    Signed-off-by: Evan Quan <evan.quan@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cee79a3268a838fbb816834050a3c2d6e817eb6e
Author: Kent Russell <kent.russell@amd.com>
Date:   Tue Jul 23 10:18:01 2019 -0400

    drm/amdkfd: Fix byte align on VegaM

    [ Upstream commit d65848657c3da5c0d4b685f823d0230f151ab34e ]

    This was missed during the addition of VegaM support

    Reviewed-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Kent Russell <kent.russell@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1b0ab059b8cf8e976631d518290e9f37c97c8c27
Author: Masahiro Yamada <yamada.masahiro@socionext.com>
Date:   Sat Jul 20 19:39:43 2019 +0900

    tracing: Fix header include guards in trace event headers

    [ Upstream commit b1d45c23284e55a379f85554a27a548b7988d47a ]

    These include guards are broken.

    Match the #if !define() and #define lines so that they work correctly.

    Link: http://lkml.kernel.org/r/20190720103943.16982-1-yamada.masahiro@socionext.com

    Fixes: f54d1867005c3 ("dma-buf: Rename struct fence to dma_fence")
    Fixes: 2e26ca7150a4f ("tracing: Fix tracepoint.h DECLARE_TRACE() to allow more than one
header")
    Fixes: e543002f77f46 ("qdisc: add tracepoint qdisc:qdisc_dequeue for dequeued SKBs")
    Fixes: 95f295f9fe081 ("dmaengine: tegra: add tracepoints to driver")
    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 70025ef1e6eed4e36c642259d15007782bd4efdf
Author: Jia-Ju Bai <baijiaju1990@gmail.com>
Date:   Mon Jul 29 16:44:51 2019 +0800

    scsi: qla2xxx: Fix possible fcport null-pointer dereferences

    [ Upstream commit e82f04ec6ba91065fd33a6201ffd7cab840e1475 ]

    In qla2x00_alloc_fcport(), fcport is assigned to NULL in the error
    handling code on line 4880:
        fcport = NULL;
```

```
    Then fcport is used on lines 4883-4886:
        INIT_WORK(&fcport->del_work, qla24xx_delete_sess_fn);
            INIT_WORK(&fcport->reg_work, qla_register_fcport_fn);
            INIT_LIST_HEAD(&fcport->gnl_entry);
            INIT_LIST_HEAD(&fcport->list);

    Thus, possible null-pointer dereferences may occur.

    To fix these bugs, qla2x00_alloc_fcport() directly returns NULL
    in the error handling code.

    These bugs are found by a static analysis tool STCheck written by us.

    Signed-off-by: Jia-Ju Bai <baijiaju1990@gmail.com>
    Acked-by: Himanshu Madhani <hmadhani@marvell.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3980c1191b134f7d0fc33f42458ddb693fd32dd
Author: Don Brace <don.brace@microsemi.com>
Date:   Wed Jul 24 17:08:06 2019 -0500

    scsi: hpsa: correct scsi command status issue after reset

    [ Upstream commit eeebce1862970653cdf5c01e98bc669edd8f529a ]

    Reviewed-by: Bader Ali - Saleh <bader.alisaleh@microsemi.com>
    Reviewed-by: Scott Teel <scott.teel@microsemi.com>
    Reviewed-by: Scott Benesh <scott.benesh@microsemi.com>
    Reviewed-by: Kevin Barnett <kevin.barnett@microsemi.com>
    Signed-off-by: Don Brace <don.brace@microsemi.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 24a4b729176dd84ccea122447b73f391cf18b531
Author: Filipe Manana <fdmanana@suse.com>
Date:   Mon Jul 29 09:37:10 2019 +0100

    Btrfs: fix deadlock between fiemap and transaction commits

    [ Upstream commit a6d155d2e363f26290ffd50591169cb96c2a609e ]

    The fiemap handler locks a file range that can have unflushed delalloc,
    and after locking the range, it tries to attach to a running transaction.
    If the running transaction started its commit, that is, it is in state
    TRANS_STATE_COMMIT_START, and either the filesystem was mounted with the
    flushoncommit option or the transaction is creating a snapshot for the
    subvolume that contains the file that fiemap is operating on, we end up
    deadlocking. This happens because fiemap is blocked on the transaction,
    waiting for it to complete, and the transaction is waiting for the flushed
    dealloc to complete, which requires locking the file range that the fiemap
    task already locked. The following stack traces serve as an example of
    when this deadlock happens:

      (...)
      [404571.515510] Workqueue: btrfs-endio-write btrfs_endio_write_helper [btrfs]
      [404571.515956] Call Trace:
      [404571.516360]  ? __schedule+0x3ae/0x7b0
      [404571.516730]  schedule+0x3a/0xb0
      [404571.517104]  lock_extent_bits+0x1ec/0x2a0 [btrfs]
      [404571.517465]  ? remove_wait_queue+0x60/0x60
      [404571.517832]  btrfs_finish_ordered_io+0x292/0x800 [btrfs]
      [404571.518202]  normal_work_helper+0xea/0x530 [btrfs]
      [404571.518566]  process_one_work+0x21e/0x5c0
      [404571.518990]  worker_thread+0x4f/0x3b0
```

```
[404571.519413]  ? process_one_work+0x5c0/0x5c0
[404571.519829]  kthread+0x103/0x140
[404571.520191]  ? kthread_create_worker_on_cpu+0x70/0x70
[404571.520565]  ret_from_fork+0x3a/0x50
[404571.520915] kworker/u8:6    D    0 31651      2 0x80004000
[404571.521290] Workqueue: btrfs-flush_delalloc btrfs_flush_delalloc_helper [btrfs]
(...)
[404571.537000] fsstress        D    0 13117  13115 0x00004000
[404571.537263] Call Trace:
[404571.537524]  ? __schedule+0x3ae/0x7b0
[404571.537788]  schedule+0x3a/0xb0
[404571.538066]  wait_current_trans+0xc8/0x100 [btrfs]
[404571.538349]  ? remove_wait_queue+0x60/0x60
[404571.538680]  start_transaction+0x33c/0x500 [btrfs]
[404571.539076]  btrfs_check_shared+0xa3/0x1f0 [btrfs]
[404571.539513]  ? extent_fiemap+0x2ce/0x650 [btrfs]
[404571.539866]  extent_fiemap+0x2ce/0x650 [btrfs]
[404571.540170]  do_vfs_ioctl+0x526/0x6f0
[404571.540436]  ksys_ioctl+0x70/0x80
[404571.540734]  __x64_sys_ioctl+0x16/0x20
[404571.540997]  do_syscall_64+0x60/0x1d0
[404571.541279]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
(...)
[404571.543729] btrfs           D    0 14210  14208 0x00004000
[404571.544023] Call Trace:
[404571.544275]  ? __schedule+0x3ae/0x7b0
[404571.544526]  ? wait_for_completion+0x112/0x1a0
[404571.544795]  schedule+0x3a/0xb0
[404571.545064]  schedule_timeout+0x1ff/0x390
[404571.545351]  ? lock_acquire+0xa6/0x190
[404571.545638]  ? wait_for_completion+0x49/0x1a0
[404571.545890]  ? wait_for_completion+0x112/0x1a0
[404571.546228]  wait_for_completion+0x131/0x1a0
[404571.546503]  ? wake_up_q+0x70/0x70
[404571.546775]  btrfs_wait_ordered_extents+0x27c/0x400 [btrfs]
[404571.547159]  btrfs_commit_transaction+0x3b0/0xae0 [btrfs]
[404571.547449]  ? btrfs_mksubvol+0x4a4/0x640 [btrfs]
[404571.547703]  ? remove_wait_queue+0x60/0x60
[404571.547969]  btrfs_mksubvol+0x605/0x640 [btrfs]
[404571.548226]  ? __sb_start_write+0xd4/0x1c0
[404571.548512]  ? mnt_want_write_file+0x24/0x50
[404571.548789]  btrfs_ioctl_snap_create_transid+0x169/0x1a0 [btrfs]
[404571.549048]  btrfs_ioctl_snap_create_v2+0x11d/0x170 [btrfs]
[404571.549307]  btrfs_ioctl+0x133f/0x3150 [btrfs]
[404571.549549]  ? mem_cgroup_charge_statistics+0x4c/0xd0
[404571.549792]  ? mem_cgroup_commit_charge+0x84/0x4b0
[404571.550064]  ? __handle_mm_fault+0xe3e/0x11f0
[404571.550306]  ? do_raw_spin_unlock+0x49/0xc0
[404571.550608]  ? _raw_spin_unlock+0x24/0x30
[404571.550976]  ? __handle_mm_fault+0xedf/0x11f0
[404571.551319]  ? do_vfs_ioctl+0xa2/0x6f0
[404571.551659]  ? btrfs_ioctl_get_supported_features+0x30/0x30 [btrfs]
[404571.552087]  do_vfs_ioctl+0xa2/0x6f0
[404571.552355]  ksys_ioctl+0x70/0x80
[404571.552621]  __x64_sys_ioctl+0x16/0x20
[404571.552864]  do_syscall_64+0x60/0x1d0
[404571.553104]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
(...)
```

If we were joining the transaction instead of attaching to it, we would not risk a deadlock because a join only blocks if the transaction is in a state greater then or equals to TRANS_STATE_COMMIT_DOING, and the delalloc flush performed by a transaction is done before it reaches that state, when it is in the state TRANS_STATE_COMMIT_START. However a transaction join is intended for use cases where we do modify the filesystem, and

```
    fiemap only needs to peek at delayed references from the current
    transaction in order to determine if extents are shared, and, besides
    that, when there is no current transaction or when it blocks to wait for
    a current committing transaction to complete, it creates a new transaction
    without reserving any space. Such unnecessary transactions, besides doing
    unnecessary IO, can cause transaction aborts (-ENOSPC) and unnecessary
    rotation of the precious backup roots.

    So fix this by adding a new transaction join variant, named join_nostart,
    which behaves like the regular join, but it does not create a transaction
    when none currently exists or after waiting for a committing transaction
    to complete.

    Fixes: 03628cdbc64db6 ("Btrfs: do not start a transaction during fiemap")
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 89cdbb8eb647b9ef1e70187229adab41d39ec5b4
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Mon Jul 29 17:05:20 2019 +0800

    drm/bridge: tc358764: Fix build error

    [ Upstream commit e1ae72a21e5f0d1846e26e3f5963930664702071 ]

    If CONFIG_DRM_TOSHIBA_TC358764=y but CONFIG_DRM_KMS_HELPER=m,
    building fails:

    drivers/gpu/drm/bridge/tc358764.o:(.rodata+0x228): undefined reference to
`drm_atomic_helper_connector_reset'
    drivers/gpu/drm/bridge/tc358764.o:(.rodata+0x240): undefined reference to
`drm_helper_probe_single_connector_modes'
    drivers/gpu/drm/bridge/tc358764.o:(.rodata+0x268): undefined reference to
`drm_atomic_helper_connector_duplicate_state'
    drivers/gpu/drm/bridge/tc358764.o:(.rodata+0x270): undefined reference to
`drm_atomic_helper_connector_destroy_state'

    Like TC358767, select DRM_KMS_HELPER to fix this, and
    change to select DRM_PANEL to avoid recursive dependency.

    Reported-by: Hulk Robot <hulkci@huawei.com>
    Fixes: f38b7cca6d0e ("drm/bridge: tc358764: Add DSI to LVDS bridge driver")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Reviewed-by: Neil Armstrong <narmstrong@baylibre.com>
    Signed-off-by: Neil Armstrong <narmstrong@baylibre.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190729090520.25968-1-
yuehaibing@huawei.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ad19295202294de91d35502519149be220224e57
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Mon Jul 29 15:12:16 2019 +0800

    drm/bridge: lvds-encoder: Fix build error while CONFIG_DRM_KMS_HELPER=m

    [ Upstream commit f4cc743a98136df3c3763050a0e8223b52d9a960 ]

    If DRM_LVDS_ENCODER=y but CONFIG_DRM_KMS_HELPER=m,
    build fails:

    drivers/gpu/drm/bridge/lvds-encoder.o: In function `lvds_encoder_probe':
    lvds-encoder.c:(.text+0x155): undefined reference to `devm_drm_panel_bridge_add'
```

```
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Fixes: dbb58bfd9ae6 ("drm/bridge: Fix lvds-encoder since the panel_bridge rework.")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Reviewed-by: Neil Armstrong <narmstrong@baylibre.com>
    Signed-off-by: Neil Armstrong <narmstrong@baylibre.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190729071216.27488-1-
yuehaibing@huawei.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c5afac52e87e4759467e6bb0e196f809da58fd17
Author: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
Date:   Mon Jul 29 15:21:28 2019 +0530

    powerpc/nvdimm: Pick nearby online node if the device node is not online

    [ Upstream commit da1115fdbd6e86c62185cdd2b4bf7add39f2f82b ]

    Currently, nvdimm subsystem expects the device numa node for SCM device to be
    an online node. It also doesn't try to bring the device numa node online. Hence
    if we use a non-online numa node as device node we hit crashes like below. This
    is because we try to access uninitialized NODE_DATA in different code paths.

    cpu 0x0: Vector: 300 (Data Access) at [c0000000fac53170]
        pc: c0000000004bbc50: ___slab_alloc+0x120/0xca0
        lr: c0000000004bc834: __slab_alloc+0x64/0xc0
        sp: c0000000fac53400
       msr: 8000000002009033
       dar: 73e8
     dsisr: 80000
      current = 0xc0000000fabb6d80
      paca    = 0xc000000003870000   irqmask: 0x03   irq_happened: 0x01
        pid   = 7, comm = kworker/u16:0
    Linux version 5.2.0-06234-g76bd729b2644 (kvaneesh@ltc-boston123) (gcc version 7.4.0
(Ubuntu 7.4.0-1ubuntu1~18.04.1)) #135 SMP Thu Jul 11 05:36:30 CDT 2019
    enter ? for help
    [link register   ] c0000000004bc834 __slab_alloc+0x64/0xc0
    [c0000000fac53400] c0000000fac53480 (unreliable)
    [c0000000fac53500] c0000000004bc818 __slab_alloc+0x48/0xc0
    [c0000000fac53560] c0000000004c30a0 __kmalloc_node_track_caller+0x3c0/0x6b0
    [c0000000fac535d0] c000000000cfafe4 devm_kmalloc+0x74/0xc0
    [c0000000fac53600] c000000000d69434 nd_region_activate+0x144/0x560
    [c0000000fac536d0] c000000000d6b19c nd_region_probe+0x17c/0x370
    [c0000000fac537b0] c000000000d6349c nvdimm_bus_probe+0x10c/0x230
    [c0000000fac53840] c000000000cf3cc4 really_probe+0x254/0x4e0
    [c0000000fac538d0] c000000000cf429c driver_probe_device+0x16c/0x1e0
    [c0000000fac53950] c000000000cf0b44 bus_for_each_drv+0x94/0x130
    [c0000000fac539b0] c000000000cf392c __device_attach+0xdc/0x200
    [c0000000fac53a50] c000000000cf231c bus_probe_device+0x4c/0xf0
    [c0000000fac53a90] c000000000ced268 device_add+0x528/0x810
    [c0000000fac53b60] c000000000d62a58 nd_async_device_register+0x28/0xa0
    [c0000000fac53bd0] c0000000001ccb8c async_run_entry_fn+0xcc/0x1f0
    [c0000000fac53c50] c0000000001bcd9c process_one_work+0x46c/0x860
    [c0000000fac53d20] c0000000001bd4f4 worker_thread+0x364/0x5f0
    [c0000000fac53db0] c0000000001c7260 kthread+0x1b0/0x1c0
    [c0000000fac53e20] c00000000000b954 ret_from_kernel_thread+0x5c/0x68

    The patch tries to fix this by picking the nearest online node as the SCM node.
    This does have a problem of us losing the information that SCM node is
    equidistant from two other online nodes. If applications need to understand these
    fine-grained details we should express then like x86 does via
    /sys/devices/system/node/nodeX/accessY/initiators/

    With the patch we get

     # numactl -H
```

```
     available: 2 nodes (0-1)
     node 0 cpus:
     node 0 size: 0 MB
     node 0 free: 0 MB
     node 1 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
 28 29 30 31
     node 1 size: 130865 MB
     node 1 free: 129130 MB
     node distances:
     node   0   1
       0:  10  20
       1:  20  10
      # cat /sys/bus/nd/devices/region0/numa_node
     0
      # dmesg | grep papr_scm
     [   91.332305] papr_scm ibm,persistent-memory:ibm,pmemory@44104001: Region registered
 with target node 2 and online node 0

     Signed-off-by: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
     Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
     Link: https://lore.kernel.org/r/20190729095128.23707-1-aneesh.kumar@linux.ibm.com
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit ec1da61b2224a5228187a35548a233ac2d7846f6
 Author: Kees Cook <keescook@chromium.org>
 Date:   Mon Jul 29 14:47:22 2019 -0700

     libata: zpodd: Fix small read overflow in zpodd_get_mech_type()

     [ Upstream commit 71d6c505b4d9e6f76586350450e785e3d452b346 ]

     Jeffrin reported a KASAN issue:

       BUG: KASAN: global-out-of-bounds in ata_exec_internal_sg+0x50f/0xc70
       Read of size 16 at addr ffffffff91f41f80 by task scsi_eh_1/149
       ...
       The buggy address belongs to the variable:
         cdb.48319+0x0/0x40

     Much like commit 18c9a99bce2a ("libata: zpodd: small read overflow in
     eject_tray()"), this fixes a cdb[] buffer length, this time in
     zpodd_get_mech_type():

     We read from the cdb[] buffer in ata_exec_internal_sg(). It has to be
     ATAPI_CDB_LEN (16) bytes long, but this buffer is only 12 bytes.

     Reported-by: Jeffrin Jose T <jeffrin@rajagiritech.edu.in>
     Fixes: afe759511808c ("libata: identify and init ZPODD devices")
     Link: https://lore.kernel.org/lkml/201907181423.E808958@keescook/
     Tested-by: Jeffrin Jose T <jeffrin@rajagiritech.edu.in>
     Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
     Signed-off-by: Kees Cook <keescook@chromium.org>
     Signed-off-by: Jens Axboe <axboe@kernel.dk>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 3829b274cda1075c9b80ecc3d9880c61ca89bee8
 Author: Yuki Tsunashima <ytsunashima@jp.adit-jv.com>
 Date:   Mon Jul 29 17:10:36 2019 +0200

     ALSA: pcm: fix lost wakeup event scenarios in snd_pcm_drain

     [ Upstream commit 37151a41df800493cfcbbef4f7208ffe04feb959 ]

     lost wakeup can occur after enabling irq, therefore put task
     into interruptible before enabling interrupts,
```

```
    without this change, task can be put to sleep and snd_pcm_drain
    will delay

    Fixes: f2b3614cefb6 ("ALSA: PCM - Don't check DMA time-out too shortly")
    Signed-off-by: Yuki Tsunashima <ytsunashima@jp.adit-jv.com>
    Signed-off-by: Suresh Udipi <sudipi@jp.adit-jv.com>
    [ported from 4.9]
    Signed-off-by: Adam Miartus <amiartus@de.adit-jv.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2fed94fdd0956a8de6130cad6b5dec41864583b8
Author: Michal Kalderon <michal.kalderon@marvell.com>
Date:   Sun Jul 28 14:13:38 2019 +0300

    RDMA/qedr: Fix the hca_type and hca_rev returned in device attributes

    [ Upstream commit 15fe6a8dcc3b48358c28e17b485fc837f9605ec4 ]

    There was a place holder for hca_type and vendor was returned
    in hca_rev. Fix the hca_rev to return the hw revision and fix
    the hca_type to return an informative string representing the
    hca.

    Signed-off-by: Michal Kalderon <michal.kalderon@marvell.com>
    Link: https://lore.kernel.org/r/20190728111338.21930-1-michal.kalderon@marvell.com
    Signed-off-by: Doug Ledford <dledford@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 99505ad90283ee094d64d811f04c108506ae1523
Author: Numfor Mbiziwo-Tiapo <nums@google.com>
Date:   Wed Jul 24 16:44:58 2019 -0700

    perf header: Fix use of unitialized value warning

    [ Upstream commit 20f9781f491360e7459c589705a2e4b1f136bee9 ]

    When building our local version of perf with MSAN (Memory Sanitizer) and
    running the perf record command, MSAN throws a use of uninitialized
    value warning in "tools/perf/util/util.c:333:6".

    This warning stems from the "buf" variable being passed into "write".
    It originated as the variable "ev" with the type union perf_event*
    defined in the "perf_event__synthesize_attr" function in
    "tools/perf/util/header.c".

    In the "perf_event__synthesize_attr" function they allocate space with a malloc
    call using ev, then go on to only assign some of the member variables before
    passing "ev" on as a parameter to the "process" function therefore "ev"
    contains uninitialized memory. Changing the malloc call to zalloc to initialize
    all the members of "ev" which gets rid of the warning.

    To reproduce this warning, build perf by running:
    make -C tools/perf CLANG=1 CC=clang EXTRA_CFLAGS="-fsanitize=memory\
     -fsanitize-memory-track-origins"

    (Additionally, llvm might have to be installed and clang might have to
    be specified as the compiler - export CC=/usr/bin/clang)

    then running:
    tools/perf/perf record -o - ls / | tools/perf/perf --no-pager annotate\
     -i - --stdio

    Please see the cover letter for why false positive warnings may be
```

```
      generated.

      Signed-off-by: Numfor Mbiziwo-Tiapo <nums@google.com>
      Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
      Cc: Ian Rogers <irogers@google.com>
      Cc: Jiri Olsa <jolsa@redhat.com>
      Cc: Mark Drayton <mbd@fb.com>
      Cc: Namhyung Kim <namhyung@kernel.org>
      Cc: Peter Zijlstra <peterz@infradead.org>
      Cc: Song Liu <songliubraving@fb.com>
      Cc: Stephane Eranian <eranian@google.com>
      Link: http://lkml.kernel.org/r/20190724234500.253358-2-nums@google.com
      Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 4b6da8b8ecccd1fec368ac5920b5139be0ce7e2f
  Author: Vince Weaver <vincent.weaver@maine.edu>
  Date:   Tue Jul 23 11:06:01 2019 -0400

      perf header: Fix divide by zero error if f_header.attr_size==0

      [ Upstream commit 7622236ceb167aa3857395f9bdaf871442aa467e ]

      So I have been having lots of trouble with hand-crafted perf.data files
      causing segfaults and the like, so I have started fuzzing the perf tool.

      First issue found:

      If f_header.attr_size is 0 in the perf.data file, then perf will crash
      with a divide-by-zero error.

      Committer note:

      Added a pr_err() to tell the user why the command failed.

      Signed-off-by: Vince Weaver <vincent.weaver@maine.edu>
      Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
      Cc: Jiri Olsa <jolsa@redhat.com>
      Cc: Namhyung Kim <namhyung@kernel.org>
      Cc: Peter Zijlstra <peterz@infradead.org>
      Link: http://lkml.kernel.org/r/alpine.DEB.2.21.1907231100440.14532@macbook-air
      Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 33c901020ab3583087ebb40d75f8d45ab93110c1
  Author: Arnaldo Carvalho de Melo <acme@redhat.com>
  Date:   Fri Jul 26 15:29:56 2019 -0300

      tools perf beauty: Fix usbdevfs_ioctl table generator to handle _IOC()

      [ Upstream commit 7ee526152db7a75d7b8713346dac76ffc3662b29 ]

      In addition to _IOW() and _IOR(), to handle this case:

        #define USBDEVFS_CONNINFO_EX(len)  _IOC(_IOC_READ, 'U', 32, len)

      That will happen in the next sync of this header file.

      Cc: Adrian Hunter <adrian.hunter@intel.com>
      Cc: Jiri Olsa <jolsa@kernel.org>
      Cc: Luis Cláudio Gonçalves <lclaudio@redhat.com>
      Cc: Namhyung Kim <namhyung@kernel.org>
      Link: https://lkml.kernel.org/n/tip-3br5e4t64e4lp0goo84che3s@git.kernel.org
      Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit db361cb406c137c915033ab3c43e313de8fe5eda
Author: Jaegeuk Kim <jaegeuk@kernel.org>
Date:   Wed Jul 17 18:31:53 2019 -0700

    f2fs: fix to read source block before invalidating it

    [ Upstream commit 543b8c468f55f27f3c0178a22a91a51aabbbc428 ]

    f2fs_allocate_data_block() invalidates old block address and enable new block
    address. Then, if we try to read old block by f2fs_submit_page_bio(), it will
    give WARN due to reading invalid blocks.

    Let's make the order sanely back.

    Reviewed-by: Chao Yu <yuchao0@huawei.com>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 62f9048d260eb5544a80a24b23407086c070b130
Author: Lucas Stach <l.stach@pengutronix.de>
Date:   Fri Jul 12 15:29:05 2019 +0200

    irqchip/irq-imx-gpcv2: Forward irq type to parent

    [ Upstream commit 9a446ef08f3bfc0c3deb9c6be840af2528ef8cf8 ]

    The GPCv2 is a stacked IRQ controller below the ARM GIC. It doesn't
    care about the IRQ type itself, but needs to forward the type to the
    parent IRQ controller, so this one can be configured correctly.

    Signed-off-by: Lucas Stach <l.stach@pengutronix.de>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6124def33c0400532dc71d8d97e8e3f434b1c58a
Author: Nianyao Tang <tangnianyao@huawei.com>
Date:   Fri Jul 26 17:32:57 2019 +0800

    irqchip/gic-v3-its: Free unused vpt_page when alloc vpe table fail

    [ Upstream commit 34f8eb92ca053cbba2887bb7e4dbf2b2cd6eb733 ]

    In its_vpe_init, when its_alloc_vpe_table fails, we should free
    vpt_page allocated just before, instead of vpe->vpt_page.
    Let's fix it.

    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Jason Cooper <jason@lakedaemon.net>
    Cc: Marc Zyngier <marc.zyngier@arm.com>
    Signed-off-by: Nianyao Tang <tangnianyao@huawei.com>
    Signed-off-by: Shaokun Zhang <zhangshaokun@hisilicon.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2996ba2512ddd35fca1ea4fa46c41fb39ad06e39
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Wed Jul 24 22:08:50 2019 +0800

    xen/pciback: remove set but not used variable 'old_state'

    [ Upstream commit 09e088a4903bd0dd911b4f1732b250130cdaffed ]

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

```
    drivers/xen/xen-pciback/conf_space_capability.c: In function pm_ctrl_write:
    drivers/xen/xen-pciback/conf_space_capability.c:119:25: warning:
     variable old_state set but not used [-Wunused-but-set-variable]

    It is never used so can be removed.

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

commit 424f6f05714de6462a1c8b1d7e44310e8d6aa0b1
Author: Christoph Hellwig <hch@lst.de>
Date:   Wed Jul 24 08:52:52 2019 +0200

    mm/hmm: always return EBUSY for invalid ranges in hmm_range_{fault,snapshot}

    [ Upstream commit 2bcbeaefde2f0384d6ad351c151b1a9fe7791a0a ]

    We should not have two different error codes for the same
    condition. EAGAIN must be reserved for the FAULT_FLAG_ALLOW_RETRY retry
    case and signals to the caller that the mmap_sem has been unlocked.

    Use EBUSY for the !valid case so that callers can get the locking right.

    Link: https://lore.kernel.org/r/20190724065258.16603-2-hch@lst.de
    Tested-by: Ralph Campbell <rcampbell@nvidia.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Ralph Campbell <rcampbell@nvidia.com>
    Reviewed-by: Jason Gunthorpe <jgg@mellanox.com>
    Reviewed-by: Felix Kuehling <Felix.Kuehling@amd.com>
    [jgg: elaborated commit message]
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d600580eee7ed1a3c2a7a38d17ecfdd583e7c796
Author: Rajneesh Bhardwaj <rajneesh.bhardwaj@linux.intel.com>
Date:   Fri Jun 14 13:39:40 2019 +0530

    platform/x86: intel_pmc_core: Add ICL-NNPI support to PMC Core

    [ Upstream commit 66013e8ec6850f9c62df6aea555fe7668e84dc3c ]

    Ice Lake Neural Network Processor for deep learning inference a.k.a.
    ICL-NNPI can re-use Ice Lake Mobile regmap to enable Intel PMC Core
    driver on it.

    Cc: Darren Hart <dvhart@infradead.org>
    Cc: Andy Shevchenko <andy@infradead.org>
    Cc: platform-driver-x86@vger.kernel.org
    Link: https://lkml.org/lkml/2019/6/5/1034
    Signed-off-by: Rajneesh Bhardwaj <rajneesh.bhardwaj@linux.intel.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 90a91551453c5478c292fc11db0f066ee2661a4e
Author: Jean Delvare <jdelvare@suse.de>
Date:   Sat Jun 29 11:41:36 2019 +0200

    platform/x86: pcengines-apuv2: Fix softdep statement

    [ Upstream commit edbfe83def34153a05439ecb3352ae0bb65024de ]

    Only first MODULE_SOFTDEP statement is handled per module.
```

```
    Multiple dependencies must be expressed in a single statement.

    Signed-off-by: Jean Delvare <jdelvare@suse.de>
    Cc: "Enrico Weigelt, metux IT consult" <info@metux.net>
    Cc: Darren Hart <dvhart@infradead.org>
    Cc: Andy Shevchenko <andy@infradead.org>
    [andy: massaged commit message]
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3d0ed0e4fafd9255fbb338c3cf22cdb123ef7645
Author: Christoph Hellwig <hch@lst.de>
Date:   Mon Jul 8 11:51:56 2019 -0700

    dma-mapping: check pfn validity in dma_common_{mmap,get_sgtable}

    [ Upstream commit 66d7780f18eae0232827fcffeaded39a6a168236 ]

    Check that the pfn returned from arch_dma_coherent_to_pfn refers to
    a valid page and reject the mmap / get_sgtable requests otherwise.

    Based on the arm implementation of the mmap and get_sgtable methods.

    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Tested-by: Vignesh Raghavendra <vigneshr@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1c9de345f7dafbd9c211407c1483d08fc564b09d
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Thu Jul 11 15:03:59 2019 +0200

    clk: renesas: cpg-mssr: Fix reset control race condition

    [ Upstream commit e1f1ae8002e4b06addc52443fcd975bbf554ae92 ]

    The module reset code in the Renesas CPG/MSSR driver uses
    read-modify-write (RMW) operations to write to a Software Reset Register
    (SRCRn), and simple writes to write to a Software Reset Clearing
    Register (SRSTCLRn), as was mandated by the R-Car Gen2 and Gen3 Hardware
    User's Manuals.

    However, this may cause a race condition when two devices are reset in
    parallel: if the reset for device A completes in the middle of the RMW
    operation for device B, device A may be reset again, causing subtle
    failures (e.g. i2c timeouts):

            thread A                        thread B
            --------                        --------

            val = SRCRn
            val |= bit A
            SRCRn = val

            delay

                                            val = SRCRn (bit A is set)

            SRSTCLRn = bit A
            (bit A in SRCRn is cleared)

                                            val |= bit B
                                            SRCRn = val (bit A and B are set)

    This can be reproduced on e.g. Salvator-XS using:
```

```
          $ while true; do i2cdump -f -y 4 0x6A b > /dev/null; done &
          $ while true; do i2cdump -f -y 2 0x10 b > /dev/null; done &

          i2c-rcar e6510000.i2c: error -110 : 40000002
          i2c-rcar e66d8000.i2c: error -110 : 40000002

      According to the R-Car Gen3 Hardware Manual Errata for Rev.
      0.80 of Feb 28, 2018, reflected in Rev. 1.00 of the R-Car Gen3 Hardware
      User's Manual, writes to SRCRn do not require read-modify-write cycles.

      Note that the R-Car Gen2 Hardware User's Manual has not been updated
      yet, and still says a read-modify-write sequence is required.  According
      to the hardware team, the reset hardware block is the same on both R-Car
      Gen2 and Gen3, though.

      Hence fix the issue by replacing the read-modify-write operations on
      SRCRn by simple writes.

      Reported-by: Yao Lihua <Lihua.Yao@desay-svautomotive.com>
      Fixes: 6197aa65c4905532 ("clk: renesas: cpg-mssr: Add support for reset control")
      Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
      Tested-by: Linh Phung <linh.phung.jy@renesas.com>
      Signed-off-by: Stephen Boyd <sboyd@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 98be3607eb756e5a0b6ef8467d4e417d73123032
  Author: Chunyan Zhang <chunyan.zhang@unisoc.com>
  Date:   Thu Jul 18 13:36:16 2019 +0800

      clk: sprd: Select REGMAP_MMIO to avoid compile errors

      [ Upstream commit c9a67cbb5189e966c70451562b2ca4c3876ab546 ]

      Make REGMAP_MMIO selected to avoid undefined reference to regmap symbols.

      Fixes: d41f59fd92f2 ("clk: sprd: Add common infrastructure")
      Signed-off-by: Chunyan Zhang <chunyan.zhang@unisoc.com>
      Signed-off-by: Stephen Boyd <sboyd@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 837471a3b48aa33442ae216991063599f049da35
  Author: Codrin Ciubotariu <codrin.ciubotariu@microchip.com>
  Date:   Tue Jun 25 12:10:02 2019 +0300

      clk: at91: generated: Truncate divisor to GENERATED_MAX_DIV + 1

      [ Upstream commit 1573eebeaa8055777eb753f9b4d1cbe653380c38 ]

      In clk_generated_determine_rate(), if the divisor is greater than
      GENERATED_MAX_DIV + 1, then the wrong best_rate will be returned.
      If clk_generated_set_rate() will be called later with this wrong
      rate, it will return -EINVAL, so the generated clock won't change
      its value. Do no let the divisor be greater than GENERATED_MAX_DIV + 1.

      Fixes: 8c7aa6328947 ("clk: at91: clk-generated: remove useless divisor loop")
      Signed-off-by: Codrin Ciubotariu <codrin.ciubotariu@microchip.com>
      Acked-by: Nicolas Ferre <nicolas.ferre@microchip.com>
      Acked-by: Ludovic Desroches <ludovic.desroches@microchip.com>
      Signed-off-by: Stephen Boyd <sboyd@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 42e213a431e1e95bbac51346c8a561ab66bcb645
  Author: Chuhong Yuan <hslester96@gmail.com>
  Date:   Wed Jul 17 16:21:01 2019 +0800
```

```
    IB/mlx5: Replace kfree with kvfree

    [ Upstream commit b7f406bb883ba7ac3222298f6b44cebc4cfe2dde ]

    Memory allocated by kvzalloc should not be freed by kfree(), use kvfree()
    instead.

    Fixes: 813e90b1aeaa ("IB/mlx5: Add advise_mr() support")
    Link: https://lore.kernel.org/r/20190717082101.14196-1-hslester96@gmail.com
    Signed-off-by: Chuhong Yuan <hslester96@gmail.com>
    Reviewed-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 521dc7e3c18c1e8e188cf120f01f2c2d07c98c02
Author: Xi Wang <wangxi11@huawei.com>
Date:   Thu Jul 11 09:32:17 2019 +0800

    RDMA/hns: Fix sg offset non-zero issue

    [ Upstream commit 60c3becfd1a138fdcfe48f2a5ef41ef0078d481e ]

    When run perftest in many times, the system will report a BUG as follows:

       BUG: Bad rss-counter state mm:(____ptrval____) idx:0 val:-1
       BUG: Bad rss-counter state mm:(____ptrval____) idx:1 val:1

    We tested with different kernel version and found it started from the the
    following commit:

    commit d10bcf947a3e ("RDMA/umem: Combine contiguous PAGE_SIZE regions in
    SGEs")

    In this commit, the sg->offset is always 0 when sg_set_page() is called in
    ib_umem_get() and the drivers are not allowed to change the sgl, otherwise
    it will get bad page descriptor when unfolding SGEs in __ib_umem_release()
    as sg_page_count() will get wrong result while sgl->offset is not 0.

    However, there is a weird sgl usage in the current hns driver, the driver
    modified sg->offset after calling ib_umem_get(), which caused we iterate
    past the wrong number of pages in for_each_sg_page iterator.

    This patch fixes it by correcting the non-standard sgl usage found in the
    hns_roce_db_map_user() function.

    Fixes: d10bcf947a3e ("RDMA/umem: Combine contiguous PAGE_SIZE regions in SGEs")
    Fixes: 0425e3e6e0c7 ("RDMA/hns: Support flush cqe for hip08 in kernel space")
    Link: https://lore.kernel.org/r/1562808737-45723-1-git-send-email-oulijun@huawei.com
    Signed-off-by: Xi Wang <wangxi11@huawei.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ab2fa8b52dc0cfb557515021ec2fa94f5b3826ba
Author: Aleix Roca Nonell <aleix.rocanonell@bsc.es>
Date:   Thu Aug 15 14:03:22 2019 +0200

    io_uring: fix manual setup of iov_iter for fixed buffers

    commit 99c79f6692ccdc42e04deea8a36e22bb48168a62 upstream.

    Commit bd11b3a391e3 ("io_uring: don't use iov_iter_advance() for fixed
    buffers") introduced an optimization to avoid using the slow
    iov_iter_advance by manually populating the iov_iter iterator in some
    cases.
```

```
    However, the computation of the iterator count field was erroneous: The
    first bvec was always accounted for an extent of page size even if the
    bvec length was smaller.

    In consequence, some I/O operations on fixed buffers were unable to
    operate on the full extent of the buffer, consistently skipping some
    bytes at the end of it.

    Fixes: bd11b3a391e3 ("io_uring: don't use iov_iter_advance() for fixed buffers")
    Cc: stable@vger.kernel.org
    Signed-off-by: Aleix Roca Nonell <aleix.rocanonell@bsc.es>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 63e2c0200e4f3d9dbbdefcd32f07f82518a41fda
Author: zhengbin <zhengbin13@huawei.com>
Date:   Mon Aug 12 20:36:55 2019 +0800

    blk-mq: move cancel of requeue_work to the front of blk_exit_queue

    commit e26cc08265dda37d2acc8394604f220ef412299d upstream.

    blk_exit_queue will free elevator_data, while blk_mq_requeue_work
    will access it. Move cancel of requeue_work to the front of
    blk_exit_queue to avoid use-after-free.

    blk_exit_queue                blk_mq_requeue_work
      __elevator_exit               blk_mq_run_hw_queues
        blk_mq_exit_sched             blk_mq_run_hw_queue
          dd_exit_queue                 blk_mq_hctx_has_pending
            kfree(elevator_data)          blk_mq_sched_has_work
                                            dd_has_work

    Fixes: fbc2a15e3433 ("blk-mq: move cancel of requeue_work into blk_mq_release")
    Cc: stable@vger.kernel.org
    Reviewed-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: zhengbin <zhengbin13@huawei.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2298d80121532d73b6216161c05cb4264ba854c1
Author: Fabio Estevam <festevam@gmail.com>
Date:   Thu Aug 8 18:01:36 2019 -0300

    Revert "i2c: imx: improve the error handling in i2c_imx_dma_request()"

    commit e8c220fac415d9f4a994b0c2871b835feac1eb4e upstream.

    Since commit e1ab9a468e3b ("i2c: imx: improve the error handling in
    i2c_imx_dma_request()") when booting with the DMA driver as module (such
    as CONFIG_FSL_EDMA=m) the following endless clk warnings are seen:

    [  153.077831] ------------[ cut here ]------------
    [  153.082528] WARNING: CPU: 0 PID: 15 at drivers/clk/clk.c:924
clk_core_disable_lock+0x18/0x24
    [  153.093077] i2c0 already disabled
    [  153.096416] Modules linked in:
    [  153.099521] CPU: 0 PID: 15 Comm: kworker/0:1 Tainted: G        W         5.2.0+ #321
    [  153.107290] Hardware name: Freescale Vybrid VF5xx/VF6xx (Device Tree)
    [  153.113772] Workqueue: events deferred_probe_work_func
    [  153.118979] [<c0019560>] (unwind_backtrace) from [<c0014734>] (show_stack+0x10/0x14)
    [  153.126778] [<c0014734>] (show_stack) from [<c083f8dc>] (dump_stack+0x9c/0xd4)
    [  153.134051] [<c083f8dc>] (dump_stack) from [<c0031154>] (__warn+0xf8/0x124)
    [  153.141056] [<c0031154>] (__warn) from [<c0031248>] (warn_slowpath_fmt+0x38/0x48)
    [  153.148580] [<c0031248>] (warn_slowpath_fmt) from [<c040fde0>]
```

```
  (clk_core_disable_lock+0x18/0x24)
      [  153.157413] [<c040fde0>] (clk_core_disable_lock) from [<c058f520>]
  (i2c_imx_probe+0x554/0x6ec)
      [  153.166076] [<c058f520>] (i2c_imx_probe) from [<c04b9178>]
  (platform_drv_probe+0x48/0x98)
      [  153.174297] [<c04b9178>] (platform_drv_probe) from [<c04b7298>]
  (really_probe+0x1d8/0x2c0)
      [  153.182605] [<c04b7298>] (really_probe) from [<c04b7554>]
  (driver_probe_device+0x5c/0x174)
      [  153.190909] [<c04b7554>] (driver_probe_device) from [<c04b58c8>]
  (bus_for_each_drv+0x44/0x8c)
      [  153.199480] [<c04b58c8>] (bus_for_each_drv) from [<c04b746c>]
  (__device_attach+0xa0/0x108)
      [  153.207782] [<c04b746c>] (__device_attach) from [<c04b65a4>]
  (bus_probe_device+0x88/0x90)
      [  153.215999] [<c04b65a4>] (bus_probe_device) from [<c04b6a04>]
  (deferred_probe_work_func+0x60/0x90)
      [  153.225003] [<c04b6a04>] (deferred_probe_work_func) from [<c004f190>]
  (process_one_work+0x204/0x634)
      [  153.234178] [<c004f190>] (process_one_work) from [<c004f618>]
  (worker_thread+0x20/0x484)
      [  153.242315] [<c004f618>] (worker_thread) from [<c0055c2c>] (kthread+0x118/0x150)
      [  153.249758] [<c0055c2c>] (kthread) from [<c00090b4>] (ret_from_fork+0x14/0x20)
      [  153.257006] Exception stack(0xdde43fb0 to 0xdde43ff8)
      [  153.262095] 3fa0:                                     00000000 00000000 00000000
  00000000
      [  153.270306] 3fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000
      [  153.278520] 3fe0: 00000000 00000000 00000000 00000000 00000013 00000000
      [  153.285159] irq event stamp: 3323022
      [  153.288787] hardirqs last  enabled at (3323021): [<c0861c4c>]
  _raw_spin_unlock_irq+0x24/0x2c
      [  153.297261] hardirqs last disabled at (3323022): [<c040d7a0>]
  clk_enable_lock+0x10/0x124
      [  153.305392] softirqs last  enabled at (3322092): [<c000a504>]
  __do_softirq+0x344/0x540
      [  153.313352] softirqs last disabled at (3322081): [<c00385c0>] irq_exit+0x10c/0x128
      [  153.320946] ---[ end trace a506731ccd9bd703 ]---

      This endless clk warnings behaviour is well explained by Andrey Smirnov:

      "Allocating DMA after registering I2C adapter can lead to infinite
      probing loop, for example, consider the following scenario:

          1. i2c_imx_probe() is called and successfully registers an I2C
             adapter via i2c_add_numbered_adapter()

          2. As a part of i2c_add_numbered_adapter() new I2C slave devices
             are added from DT which results in a call to
             driver_deferred_probe_trigger()

          3. i2c_imx_probe() continues and calls i2c_imx_dma_request() which
             due to lack of proper DMA driver returns -EPROBE_DEFER

          4. i2c_imx_probe() fails, removes I2C adapter and returns
             -EPROBE_DEFER, which places it into deferred probe list

          5. Deferred probe work triggered in #2 above kicks in and calls
             i2c_imx_probe() again thus bringing us to step #1"

      So revert commit e1ab9a468e3b ("i2c: imx: improve the error handling in
      i2c_imx_dma_request()") and restore the old behaviour, in order to
      avoid regressions on existing setups.

      Cc: <stable@vger.kernel.org>
```

```
    Reported-by: Andrey Smirnov <andrew.smirnov@gmail.com>
    Reported-by: Russell King <linux@armlinux.org.uk>
    Fixes: e1ab9a468e3b ("i2c: imx: improve the error handling in i2c_imx_dma_request()")
    Signed-off-by: Fabio Estevam <festevam@gmail.com>
    Signed-off-by: Wolfram Sang <wsa@the-dreams.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 10e58e99a520b91e1101403c4dd9c65aaa0aa2a8
Author: Vincent Chen <vincent.chen@sifive.com>
Date:   Wed Aug 14 16:23:53 2019 +0800

    riscv: Make __fstate_clean() work correctly.

    commit 69703eb9a8ae28a46cd5bce7d69ceeef6273a104 upstream.

    Make the __fstate_clean() function correctly set the
    state of sstatus.FS in pt_regs to SR_FS_CLEAN.

    Fixes: 7db91e57a0acd ("RISC-V: Task implementation")
    Cc: linux-stable <stable@vger.kernel.org>
    Signed-off-by: Vincent Chen <vincent.chen@sifive.com>
    Reviewed-by: Anup Patel <anup@brainfault.org>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    [paul.walmsley@sifive.com: expanded "Fixes" commit ID]
    Signed-off-by: Paul Walmsley <paul.walmsley@sifive.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 082ca8e35450fe007e9e3a3550fc2c0747008886
Author: Vincent Chen <vincent.chen@sifive.com>
Date:   Wed Aug 14 16:23:52 2019 +0800

    riscv: Correct the initialized flow of FP register

    commit 8ac71d7e46b94a4fc8ffc6f1c88004cdf24459e8 upstream.

      The following two reasons cause FP registers are sometimes not
    initialized before starting the user program.
    1. Currently, the FP context is initialized in flush_thread() function
       and we expect these initial values to be restored to FP register when
       doing FP context switch. However, the FP context switch only occurs in
       switch_to function. Hence, if this process does not be scheduled out
       and scheduled in before entering the user space, the FP registers
       have no chance to initialize.
    2. In flush_thread(), the state of reg->sstatus.FS inherits from the
       parent. Hence, the state of reg->sstatus.FS may be dirty. If this
       process is scheduled out during flush_thread() and initializing the
       FP register, the fstate_save() in switch_to will corrupt the FP context
       which has been initialized until flush_thread().

      To solve the 1st case, the initialization of the FP register will be
    completed in start_thread(). It makes sure all FP registers are initialized
    before starting the user program. For the 2nd case, the state of
    reg->sstatus.FS in start_thread will be set to SR_FS_OFF to prevent this
    process from corrupting FP context in doing context save. The FP state is
    set to SR_FS_INITIAL in start_trhead().

    Signed-off-by: Vincent Chen <vincent.chen@sifive.com>
    Reviewed-by: Anup Patel <anup@brainfault.org>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Fixes: 7db91e57a0acd ("RISC-V: Task implementation")
    Cc: stable@vger.kernel.org
    [paul.walmsley@sifive.com: fixed brace alignment issue reported by
     checkpatch]
    Signed-off-by: Paul Walmsley <paul.walmsley@sifive.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 48e73abb96864f838a1a53d594b8617f161369aa
Author: Florian Westphal <fw@strlen.de>
Date:   Mon Jul 29 17:58:10 2019 +0200

    netfilter: ebtables: also count base chain policies

    commit 3b48300d5cc7c7bed63fddb006c4046549ed4aec upstream.

    ebtables doesn't include the base chain policies in the rule count,
    so we need to add them manually when we call into the x_tables core
    to allocate space for the comapt offset table.

    This lead syzbot to trigger:
    WARNING: CPU: 1 PID: 9012 at net/netfilter/x_tables.c:649
    xt_compat_add_offset.cold+0x11/0x36 net/netfilter/x_tables.c:649

    Reported-by: syzbot+276ddebab3382bbf72db@syzkaller.appspotmail.com
    Fixes: 2035f3ff8eaa ("netfilter: ebtables: compat: un-break 32bit setsockopt when no
rules are present")
    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 27843db1180c7144393b9b2c182c081631a1a20a
Author: Eric Dumazet <edumazet@google.com>
Date:   Tue Jul 23 03:15:37 2019 -0700

    bpf: fix access to skb_shared_info->gso_segs

    commit 06a22d897d82f12776d44dbf0850f5895469cb2a upstream.

    It is possible we reach bpf_convert_ctx_access() with
    si->dst_reg == si->src_reg

    Therefore, we need to load BPF_REG_AX before eventually
    mangling si->src_reg.

    syzbot generated this x86 code :
       3:   55                      push   %rbp
       4:   48 89 e5                mov    %rsp,%rbp
       7:   48 81 ec 00 00 00 00    sub    $0x0,%rsp // Might be avoided ?
       e:   53                      push   %rbx
       f:   41 55                   push   %r13
      11:   41 56                   push   %r14
      13:   41 57                   push   %r15
      15:   6a 00                   pushq  $0x0
      17:   31 c0                   xor    %eax,%eax
      19:   48 8b bf c0 00 00 00    mov    0xc0(%rdi),%rdi
      20:   44 8b 97 bc 00 00 00    mov    0xbc(%rdi),%r10d
      27:   4c 01 d7                add    %r10,%rdi
      2a:   48 0f b7 7f 06          movzwq 0x6(%rdi),%rdi // Crash
      2f:   5b                      pop    %rbx
      30:   41 5f                   pop    %r15
      32:   41 5e                   pop    %r14
      34:   41 5d                   pop    %r13
      36:   5b                      pop    %rbx
      37:   c9                      leaveq
      38:   c3                      retq

    Fixes: d9ff286a0f59 ("bpf: allow BPF programs access skb_shared_info->gso_segs field")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit e0ad7a0c028b69dda98157e02986f17bd3c7ce13
Author: Denis Kirjanov <kda@linux-powerpc.org>
Date:   Tue Jul 30 15:13:57 2019 +0200

    net: usb: pegasus: fix improper read if get_registers() fail

    commit 224c04973db1125fcebefffd86115f99f50f8277 upstream.

    get_registers() may fail with -ENOMEM and in this
    case we can read a garbage from the status variable tmp.

    Reported-by: syzbot+3499a83b2d062ae409d4@syzkaller.appspotmail.com
    Signed-off-by: Denis Kirjanov <kda@linux-powerpc.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4a2fa005d274766cc0e61021acf996f0afea6137
Author: Oliver Neukum <oneukum@suse.com>
Date:   Tue Aug 6 09:05:55 2019 -0700

    Input: iforce - add sanity checks

    commit 849f5ae3a513c550cad741c68dd3d7eb2bcc2a2c upstream.

    The endpoint type should also be checked before a device
    is accepted.

    Reported-by: syzbot+5efc10c005014d061a74@syzkaller.appspotmail.com
    Signed-off-by: Oliver Neukum <oneukum@suse.com>
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 90343fa400b556b816ed7b723c632240891025cd
Author: Oliver Neukum <oneukum@suse.com>
Date:   Thu Aug 1 09:44:25 2019 -0700

    Input: kbtab - sanity check for endpoint type

    commit c88090dfc84254fa149174eb3e6a8458de1912c4 upstream.

    The driver should check whether the endpoint it uses has the correct
    type.

    Reported-by: syzbot+c7df50363aaff50aa363@syzkaller.appspotmail.com
    Signed-off-by: Oliver Neukum <oneukum@suse.com>
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 09f54291dcb4472952958c0143b050ac3e479428
Author: Hillf Danton <hdanton@sina.com>
Date:   Tue Aug 6 16:40:15 2019 +0800

    HID: hiddev: do cleanup in failure of opening a device

    commit 6d4472d7bec39917b54e4e80245784ea5d60ce49 upstream.

    Undo what we did for opening before releasing the memory slice.

    Reported-by: syzbot <syzbot+62a1e04fd3ec2abf099e@syzkaller.appspotmail.com>
    Cc: Andrey Konovalov <andreyknvl@google.com>
    Signed-off-by: Hillf Danton <hdanton@sina.com>
    Signed-off-by: Jiri Kosina <jkosina@suse.cz>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 2a5ef6d80e8ecf9311d9cf0f5bc67bdde4c91399
Author: Hillf Danton <hdanton@sina.com>
Date:   Tue Aug 6 16:38:58 2019 +0800

    HID: hiddev: avoid opening a disconnected device

    commit 9c09b214f30e3c11f9b0b03f89442df03643794d upstream.

    syzbot found the following crash on:

    HEAD commit:    e96407b4 usb-fuzzer: main usb gadget fuzzer driver
    git tree:       https://github.com/google/kasan.git usb-fuzzer
    console output: https://syzkaller.appspot.com/x/log.txt?x=147ac20c600000
    kernel config:  https://syzkaller.appspot.com/x/.config?x=792eb47789f57810
    dashboard link: https://syzkaller.appspot.com/bug?extid=62a1e04fd3ec2abf099e
    compiler:       gcc (GCC) 9.0.0 20181231 (experimental)

    ==================================================================
    BUG: KASAN: use-after-free in __lock_acquire+0x302a/0x3b50
    kernel/locking/lockdep.c:3753
    Read of size 8 at addr ffff8881cf591a08 by task syz-executor.1/26260

    CPU: 1 PID: 26260 Comm: syz-executor.1 Not tainted 5.3.0-rc2+ #24
    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+0xca/0x13e lib/dump_stack.c:113
      print_address_description+0x6a/0x32c mm/kasan/report.c:351
      __kasan_report.cold+0x1a/0x33 mm/kasan/report.c:482
      kasan_report+0xe/0x12 mm/kasan/common.c:612
      __lock_acquire+0x302a/0x3b50 kernel/locking/lockdep.c:3753
      lock_acquire+0x127/0x320 kernel/locking/lockdep.c:4412
      __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline]
      _raw_spin_lock_irqsave+0x32/0x50 kernel/locking/spinlock.c:159
      hiddev_release+0x82/0x520 drivers/hid/usbhid/hiddev.c:221
      __fput+0x2d7/0x840 fs/file_table.c:280
      task_work_run+0x13f/0x1c0 kernel/task_work.c:113
      exit_task_work include/linux/task_work.h:22 [inline]
      do_exit+0x8ef/0x2c50 kernel/exit.c:878
      do_group_exit+0x125/0x340 kernel/exit.c:982
      get_signal+0x466/0x23d0 kernel/signal.c:2728
      do_signal+0x88/0x14e0 arch/x86/kernel/signal.c:815
      exit_to_usermode_loop+0x1a2/0x200 arch/x86/entry/common.c:159
      prepare_exit_to_usermode arch/x86/entry/common.c:194 [inline]
      syscall_return_slowpath arch/x86/entry/common.c:274 [inline]
      do_syscall_64+0x45f/0x580 arch/x86/entry/common.c:299
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
    RIP: 0033:0x459829
    Code: fd b7 fb ff c3 66 2e 0f 1f 84 00 00 00 00 00 66 90 48 89 f8 48 89 f7
    48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff
    ff 0f 83 cb b7 fb ff c3 66 2e 0f 1f 84 00 00 00 00
    RSP: 002b:00007f75b2a6ccf8 EFLAGS: 00000246 ORIG_RAX: 00000000000000ca
    RAX: fffffffffffffe00 RBX: 000000000075c078 RCX: 0000000000459829
    RDX: 0000000000000000 RSI: 0000000000000080 RDI: 000000000075c078
    RBP: 000000000075c070 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000246 R12: 000000000075c07c
    R13: 00007ffcdfe1023f R14: 00007f75b2a6d9c0 R15: 000000000075c07c

    Allocated by task 104:
      save_stack+0x1b/0x80 mm/kasan/common.c:69
      set_track mm/kasan/common.c:77 [inline]
      __kasan_kmalloc mm/kasan/common.c:487 [inline]
      __kasan_kmalloc.constprop.0+0xbf/0xd0 mm/kasan/common.c:460
      kmalloc include/linux/slab.h:552 [inline]
```

```
    kzalloc include/linux/slab.h:748 [inline]
    hiddev_connect+0x242/0x5b0 drivers/hid/usbhid/hiddev.c:900
    hid_connect+0x239/0xbb0 drivers/hid/hid-core.c:1882
    hid_hw_start drivers/hid/hid-core.c:1981 [inline]
    hid_hw_start+0xa2/0x130 drivers/hid/hid-core.c:1972
    appleir_probe+0x13e/0x1a0 drivers/hid/hid-appleir.c:308
    hid_device_probe+0x2be/0x3f0 drivers/hid/hid-core.c:2209
    really_probe+0x281/0x650 drivers/base/dd.c:548
    driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
    __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
    bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
    __device_attach+0x217/0x360 drivers/base/dd.c:882
    bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
    device_add+0xae6/0x16f0 drivers/base/core.c:2114
    hid_add_device+0x33c/0x990 drivers/hid/hid-core.c:2365
    usbhid_probe+0xa81/0xfa0 drivers/hid/usbhid/hid-core.c:1386
    usb_probe_interface+0x305/0x7a0 drivers/usb/core/driver.c:361
    really_probe+0x281/0x650 drivers/base/dd.c:548
    driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
    __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
    bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
    __device_attach+0x217/0x360 drivers/base/dd.c:882
    bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
    device_add+0xae6/0x16f0 drivers/base/core.c:2114
    usb_set_configuration+0xdf6/0x1670 drivers/usb/core/message.c:2023
    generic_probe+0x9d/0xd5 drivers/usb/core/generic.c:210
    usb_probe_device+0x99/0x100 drivers/usb/core/driver.c:266
    really_probe+0x281/0x650 drivers/base/dd.c:548
    driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
    __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
    bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
    __device_attach+0x217/0x360 drivers/base/dd.c:882
    bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
    device_add+0xae6/0x16f0 drivers/base/core.c:2114
    usb_new_device.cold+0x6a4/0xe79 drivers/usb/core/hub.c:2536
    hub_port_connect drivers/usb/core/hub.c:5098 [inline]
    hub_port_connect_change drivers/usb/core/hub.c:5213 [inline]
    port_event drivers/usb/core/hub.c:5359 [inline]
    hub_event+0x1b5c/0x3640 drivers/usb/core/hub.c:5441
    process_one_work+0x92b/0x1530 kernel/workqueue.c:2269
    worker_thread+0x96/0xe20 kernel/workqueue.c:2415
    kthread+0x318/0x420 kernel/kthread.c:255
    ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:352

  Freed by task 104:
    save_stack+0x1b/0x80 mm/kasan/common.c:69
    set_track mm/kasan/common.c:77 [inline]
    __kasan_slab_free+0x130/0x180 mm/kasan/common.c:449
    slab_free_hook mm/slub.c:1423 [inline]
    slab_free_freelist_hook mm/slub.c:1470 [inline]
    slab_free mm/slub.c:3012 [inline]
    kfree+0xe4/0x2f0 mm/slub.c:3953
    hiddev_connect.cold+0x45/0x5c drivers/hid/usbhid/hiddev.c:914
    hid_connect+0x239/0xbb0 drivers/hid/hid-core.c:1882
    hid_hw_start drivers/hid/hid-core.c:1981 [inline]
    hid_hw_start+0xa2/0x130 drivers/hid/hid-core.c:1972
    appleir_probe+0x13e/0x1a0 drivers/hid/hid-appleir.c:308
    hid_device_probe+0x2be/0x3f0 drivers/hid/hid-core.c:2209
    really_probe+0x281/0x650 drivers/base/dd.c:548
    driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
    __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
    bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
    __device_attach+0x217/0x360 drivers/base/dd.c:882
    bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
    device_add+0xae6/0x16f0 drivers/base/core.c:2114
```

```
        hid_add_device+0x33c/0x990 drivers/hid/hid-core.c:2365
        usbhid_probe+0xa81/0xfa0 drivers/hid/usbhid/hid-core.c:1386
        usb_probe_interface+0x305/0x7a0 drivers/usb/core/driver.c:361
        really_probe+0x281/0x650 drivers/base/dd.c:548
        driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
        __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
        bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
        __device_attach+0x217/0x360 drivers/base/dd.c:882
        bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
        device_add+0xae6/0x16f0 drivers/base/core.c:2114
        usb_set_configuration+0xdf6/0x1670 drivers/usb/core/message.c:2023
        generic_probe+0x9d/0xd5 drivers/usb/core/generic.c:210
        usb_probe_device+0x99/0x100 drivers/usb/core/driver.c:266
        really_probe+0x281/0x650 drivers/base/dd.c:548
        driver_probe_device+0x101/0x1b0 drivers/base/dd.c:709
        __device_attach_driver+0x1c2/0x220 drivers/base/dd.c:816
        bus_for_each_drv+0x15c/0x1e0 drivers/base/bus.c:454
        __device_attach+0x217/0x360 drivers/base/dd.c:882
        bus_probe_device+0x1e4/0x290 drivers/base/bus.c:514
        device_add+0xae6/0x16f0 drivers/base/core.c:2114
        usb_new_device.cold+0x6a4/0xe79 drivers/usb/core/hub.c:2536
        hub_port_connect drivers/usb/core/hub.c:5098 [inline]
        hub_port_connect_change drivers/usb/core/hub.c:5213 [inline]
        port_event drivers/usb/core/hub.c:5359 [inline]
        hub_event+0x1b5c/0x3640 drivers/usb/core/hub.c:5441
        process_one_work+0x92b/0x1530 kernel/workqueue.c:2269
        worker_thread+0x96/0xe20 kernel/workqueue.c:2415
        kthread+0x318/0x420 kernel/kthread.c:255
        ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:352

      The buggy address belongs to the object at ffff8881cf591900
        which belongs to the cache kmalloc-512 of size 512
      The buggy address is located 264 bytes inside of
        512-byte region [ffff8881cf591900, ffff8881cf591b00)
      The buggy address belongs to the page:
      page:ffffea00073d6400 refcount:1 mapcount:0 mapping:ffff8881da002500
      index:0x0 compound_mapcount: 0
      flags: 0x200000000010200(slab|head)
      raw: 0200000000010200 0000000000000000 0000000100000001 ffff8881da002500
      raw: 0000000000000000 00000000000c000c 00000001ffffffff 0000000000000000
      page dumped because: kasan: bad access detected

      Memory state around the buggy address:
        ffff8881cf591900: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
        ffff8881cf591980: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
      > ffff8881cf591a00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
                          ^
        ffff8881cf591a80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
        ffff8881cf591b00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
      ==================================================================

      In order to avoid opening a disconnected device, we need to check exist
      again after acquiring the existance lock, and bail out if necessary.

      Reported-by: syzbot <syzbot+62a1e04fd3ec2abf099e@syzkaller.appspotmail.com>
      Cc: Andrey Konovalov <andreyknvl@google.com>
      Signed-off-by: Hillf Danton <hdanton@sina.com>
      Signed-off-by: Jiri Kosina <jkosina@suse.cz>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 0811cfe54896e892084abc1fe96c6108cac89e6d
  Author: Oliver Neukum <oneukum@suse.com>
  Date:   Thu Jul 25 15:13:33 2019 +0200

      HID: holtek: test for sanity of intfdata
```

```
    commit 01ec0a5f19c8c82960a07f6c7410fc9e01d7fb51 upstream.

    The ioctl handler uses the intfdata of a second interface,
    which may not be present in a broken or malicious device, hence
    the intfdata needs to be checked for NULL.

    [jkosina@suse.cz: fix newly added spurious space]
    Reported-by: syzbot+965152643a75a56737be@syzkaller.appspotmail.com
    Signed-off-by: Oliver Neukum <oneukum@suse.com>
    Signed-off-by: Jiri Kosina <jkosina@suse.cz>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1c56b8510dcf013ac4e1e9c3887bd57ad3b1ed37
Author: Hui Wang <hui.wang@canonical.com>
Date:   Wed Aug 14 12:09:07 2019 +0800

    ALSA: hda - Let all conexant codec enter D3 when rebooting

    commit 401714d9534aad8c24196b32600da683116bbe09 upstream.

    We have 3 new lenovo laptops which have conexant codec 0x14f11f86,
    these 3 laptops also have the noise issue when rebooting, after
    letting the codec enter D3 before rebooting or poweroff, the noise
    disappers.

    Instead of adding a new ID again in the reboot_notify(), let us make
    this function apply to all conexant codec. In theory make codec enter
    D3 before rebooting or poweroff is harmless, and I tested this change
    on a couple of other Lenovo laptops which have different conexant
    codecs, there is no side effect so far.

    Cc: stable@vger.kernel.org
    Signed-off-by: Hui Wang <hui.wang@canonical.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ed3fcb0215225647abef6f4ee98eeea17657e9c1
Author: Hui Wang <hui.wang@canonical.com>
Date:   Wed Aug 14 12:09:08 2019 +0800

    ALSA: hda - Add a generic reboot_notify

    commit 871b9066027702e6e6589da0e1edd3b7dede7205 upstream.

    Make codec enter D3 before rebooting or poweroff can fix the noise
    issue on some laptops. And in theory it is harmless for all codecs
    to enter D3 before rebooting or poweroff, let us add a generic
    reboot_notify, then realtek and conexant drivers can call this
    function.

    Cc: stable@vger.kernel.org
    Signed-off-by: Hui Wang <hui.wang@canonical.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a0afc19de1d0212bbb5be1c143fd7a40eb2ee922
Author: Wenwen Wang <wenwen@cs.uga.edu>
Date:   Fri Aug 9 23:29:48 2019 -0500

    ALSA: hda - Fix a memory leak bug

    commit cfef67f016e4c00a2f423256fc678a6967a9fc09 upstream.

    In snd_hda_parse_generic_codec(), 'spec' is allocated through kzalloc().
```

```
    Then, the pin widgets in 'codec' are parsed. However, if the parsing
    process fails, 'spec' is not deallocated, leading to a memory leak.

    To fix the above issue, free 'spec' before returning the error.

    Fixes: 352f7f914ebb ("ALSA: hda - Merge Realtek parser code to generic parser")
    Signed-off-by: Wenwen Wang <wenwen@cs.uga.edu>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5fe02a81b67971df49c87875ad2b0bfd10b940fb
Author: Takashi Iwai <tiwai@suse.de>
Date:   Fri Aug 9 11:23:00 2019 +0200

    ALSA: hda - Apply workaround for another AMD chip 1022:1487

    commit de768ce45466f3009809719eb7b1f6f5277d9373 upstream.

    MSI MPG X570 board is with another AMD HD-audio controller (PCI ID
    1022:1487) and it requires the same workaround applied for X370, etc
    (PCI ID 1022:1457).

    BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=195303
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f505b8cc6cc7cc6cee15920cfe1937c15f0ae35f
Author: Hui Peng <benquike@gmail.com>
Date:   Tue Aug 13 22:34:04 2019 -0400

    ALSA: usb-audio: Fix an OOB bug in parse_audio_mixer_unit

    commit daac07156b330b18eb5071aec4b3ddca1c377f2c upstream.

    The `uac_mixer_unit_descriptor` shown as below is read from the
    device side. In `parse_audio_mixer_unit`, `baSourceID` field is
    accessed from index 0 to `bNrInPins` - 1, the current implementation
    assumes that descriptor is always valid (the length  of descriptor
    is no shorter than 5 + `bNrInPins`). If a descriptor read from
    the device side is invalid, it may trigger out-of-bound memory
    access.

    ```
    struct uac_mixer_unit_descriptor {
            __u8 bLength;
            __u8 bDescriptorType;
            __u8 bDescriptorSubtype;
            __u8 bUnitID;
            __u8 bNrInPins;
            __u8 baSourceID[];
    }
    ```

    This patch fixes the bug by add a sanity check on the length of
    the descriptor.

    Reported-by: Hui Peng <benquike@gmail.com>
    Reported-by: Mathias Payer <mathias.payer@nebelwelt.net>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Hui Peng <benquike@gmail.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit cee2dfc6400ad8c6f7079da0d1985340cc4ae1b9
Author: Hui Peng <benquike@gmail.com>
Date:   Thu Aug 15 00:31:34 2019 -0400

    ALSA: usb-audio: Fix a stack buffer overflow bug in check_input_term

    commit 19bce474c45be69a284ecee660aa12d8f1e88f18 upstream.

    `check_input_term` recursively calls itself with input from
    device side (e.g., uac_input_terminal_descriptor.bCSourceID)
    as argument (id). In `check_input_term`, if `check_input_term`
    is called with the same `id` argument as the caller, it triggers
    endless recursive call, resulting kernel space stack overflow.

    This patch fixes the bug by adding a bitmap to `struct mixer_build`
    to keep track of the checked ids and stop the execution if some id
    has been checked (similar to how parse_audio_unit handles unitid
    argument).

    Reported-by: Hui Peng <benquike@gmail.com>
    Reported-by: Mathias Payer <mathias.payer@nebelwelt.net>
    Signed-off-by: Hui Peng <benquike@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a390784fa7b9c63bcf36c3df48ab5ff67a7fcc0b
Author: Takashi Iwai <tiwai@suse.de>
Date:   Tue Aug 13 17:39:56 2019 +0200

    ALSA: hda/realtek - Add quirk for HP Envy x360

    commit 190d03814eb3b49d4f87ff38fef26d36f3568a60 upstream.

    HP Envy x360 (AMD Ryzen-based model) with 103c:8497 needs the same
    quirk like HP Spectre x360 for enabling the mute LED over Mic3 pin.

    BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=204373
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d457d4adef2efb36e02bd25554543c904d4832c4
Author: Will Deacon <will@kernel.org>
Date:   Fri Aug 16 14:57:43 2019 +0100

    arm64: ftrace: Ensure module ftrace trampoline is coherent with I-side

    commit b6143d10d23ebb4a77af311e8b8b7f019d0163e6 upstream.

    The initial support for dynamic ftrace trampolines in modules made use
    of an indirect branch which loaded its target from the beginning of
    a special section (e71a4e1bebaf7 ("arm64: ftrace: add support for far
    branches to dynamic ftrace")). Since no instructions were being patched,
    no cache maintenance was needed. However, later in be0f272bfc83 ("arm64:
    ftrace: emit ftrace-mod.o contents through code") this code was reworked
    to output the trampoline instructions directly into the PLT entry but,
    unfortunately, the necessary cache maintenance was overlooked.

    Add a call to __flush_icache_range() after writing the new trampoline
    instructions but before patching in the branch to the trampoline.

    Cc: Ard Biesheuvel <ard.biesheuvel@linaro.org>
    Cc: James Morse <james.morse@arm.com>
    Cc: <stable@vger.kernel.org>
```

```
    Fixes: be0f272bfc83 ("arm64: ftrace: emit ftrace-mod.o contents through code")
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6a7307fb161eb6f59b61c29cdb5285179cd00c8d
Author: Max Filippov <jcmvbkbc@gmail.com>
Date:   Mon Aug 12 15:01:30 2019 -0700

    xtensa: add missing isync to the cpu_reset TLB code

    commit cd8869f4cb257f22b89495ca40f5281e58ba359c upstream.

    ITLB entry modifications must be followed by the isync instruction
    before the new entries are possibly used. cpu_reset lacks one isync
    between ITLB way 6 initialization and jump to the identity mapping.
    Add missing isync to xtensa cpu_reset.

    Cc: stable@vger.kernel.org
    Signed-off-by: Max Filippov <jcmvbkbc@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 31c6c99c12ff66d7ebf9475ada2f62bd5aa0d899
Author: Lyude Paul <lyude@redhat.com>
Date:   Thu Aug 8 20:53:05 2019 -0400

    drm/nouveau: Only recalculate PBN/VCPI on mode/connector changes

    commit db1231ddc04682f60d56ff42447f13099c6c4a4c upstream.

    I -thought- I had fixed this entirely, but it looks like that I didn't
    test this thoroughly enough as we apparently still make one big mistake
    with nv50_msto_atomic_check() - we don't handle the following scenario:

    * CRTC #1 has n VCPI allocated to it, is attached to connector DP-4
      which is attached to encoder #1. enabled=y active=n
    * CRTC #1 is changed from DP-4 to DP-5, causing:
      * DP-4 crtc=#1→NULL (VCPI n→0)
      * DP-5 crtc=NULL→#1
      * CRTC #1 steals encoder #1 back from DP-4 and gives it to DP-5
      * CRTC #1 maintains the same mode as before, just with a different
        connector
    * mode_changed=n connectors_changed=y
      (we _SHOULD_ do VCPI 0→n here, but don't)

    Once the above scenario is repeated once, we'll attempt freeing VCPI
    from the connector that we didn't allocate due to the connectors
    changing, but the mode staying the same. Sigh.

    Since nv50_msto_atomic_check() has broken a few times now, let's rethink
    things a bit to be more careful: limit both VCPI/PBN allocations to
    mode_changed || connectors_changed, since neither VCPI or PBN should
    ever need to change outside of routing and mode changes.

    Changes since v1:
    * Fix accidental reversal of clock and bpp arguments in
      drm_dp_calc_pbn_mode() - William Lewis

    Signed-off-by: Lyude Paul <lyude@redhat.com>
    Reported-by: Bohdan Milar <bmilar@redhat.com>
    Tested-by: Bohdan Milar <bmilar@redhat.com>
    Fixes: 232c9eec417a ("drm/nouveau: Use atomic VCPI helpers for MST")
    References: 412e85b60531 ("drm/nouveau: Only release VCPI slots on mode changes")
    Cc: Lyude Paul <lyude@redhat.com>
    Cc: Ben Skeggs <bskeggs@redhat.com>
```

```
    Cc: Daniel Vetter <daniel.vetter@ffwll.ch>
    Cc: David Airlie <airlied@redhat.com>
    Cc: Jerry Zuo <Jerry.Zuo@amd.com>
    Cc: Harry Wentland <harry.wentland@amd.com>
    Cc: Juston Li <juston.li@intel.com>
    Cc: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Cc: Karol Herbst <karolherbst@gmail.com>
    Cc: Ilia Mirkin <imirkin@alum.mit.edu>
    Cc: <stable@vger.kernel.org> # v5.1+
    Acked-by: Ben Skeggs <bskeggs@redhat.com>
    Signed-off-by: Dave Airlie <airlied@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190809005307.18391-1-
lyude@redhat.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5938e7b577901e6dace373951e9b2da0e3ba44c5
Author: Pierre-Eric Pelloux-Prayer <pierre-eric.pelloux-prayer@amd.com>
Date:   Tue Aug 6 18:27:26 2019 +0200

    drm/amdgpu: fix gfx9 soft recovery

    commit 17b6d2d528542bc60ad400add35728b2259b3cc1 upstream.

    The SOC15_REG_OFFSET() macro wasn't used, making the soft recovery fail.

    v2: use WREG32_SOC15 instead of WREG32 + SOC15_REG_OFFSET

    Signed-off-by: Pierre-Eric Pelloux-Prayer <pierre-eric.pelloux-prayer@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@amd.com>
    Reviewed-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4b837b792353f721493ef6468d1b88805d90c717
Author: Viresh Kumar <viresh.kumar@linaro.org>
Date:   Wed Aug 7 12:36:01 2019 +0530

    cpufreq: schedutil: Don't skip freq update when limits change

    commit 600f5badb78c316146d062cfd7af4a2cfb655baa upstream.

    To avoid reducing the frequency of a CPU prematurely, we skip reducing
    the frequency if the CPU had been busy recently.

    This should not be done when the limits of the policy are changed, for
    example due to thermal throttling. We should always get the frequency
    within the new limits as soon as possible.

    Trying to fix this by using only one flag, i.e. need_freq_update, can
    lead to a race condition where the flag gets cleared without forcing us
    to change the frequency at least once. And so this patch introduces
    another flag to avoid that race condition.

    Fixes: ecd288429126 ("cpufreq: schedutil: Don't set next_freq to UINT_MAX")
    Cc: v4.18+ <stable@vger.kernel.org> # v4.18+
    Reported-by: Doug Smythies <dsmythies@telus.net>
    Tested-by: Doug Smythies <dsmythies@telus.net>
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0a8ae1db1a3eceeef0f0eba40e83b4be727ab15d
Author: Mel Gorman <mgorman@techsingularity.net>
Date:   Tue Aug 13 15:37:57 2019 -0700
```

mm, vmscan: do not special-case slab reclaim when watermarks are boosted

commit 28360f398778d7623a5ff8a8e90958c0d925e120 upstream.

Dave Chinner reported a problem pointing a finger at commit 1c30844d2dfe
("mm: reclaim small amounts of memory when an external fragmentation
event occurs").

The report is extensive:

  https://lore.kernel.org/linux-mm/20190807091858.2857-1-david@fromorbit.com/

and it's worth recording the most relevant parts (colorful language and
typos included).

> When running a simple, steady state 4kB file creation test to
> simulate extracting tarballs larger than memory full of small
> files into the filesystem, I noticed that once memory fills up
> the cache balance goes to hell.
>
> The workload is creating one dirty cached inode for every dirty
> page, both of which should require a single IO each to clean and
> reclaim, and creation of inodes is throttled by the rate at which
> dirty writeback runs at (via balance dirty pages). Hence the ingest
> rate of new cached inodes and page cache pages is identical and
> steady. As a result, memory reclaim should quickly find a steady
> balance between page cache and inode caches.
>
> The moment memory fills, the page cache is reclaimed at a much
> faster rate than the inode cache, and evidence suggests that
> the inode cache shrinker is not being called when large batches
> of pages are being reclaimed. In roughly the same time period
> that it takes to fill memory with 50% pages and 50% slab caches,
> memory reclaim reduces the page cache down to just dirty pages
> and slab caches fill the entirety of memory.
>
> The LRU is largely full of dirty pages, and we're getting spikes
> of random writeback from memory reclaim so it's all going to shit.
> Behaviour never recovers, the page cache remains pinned at just
> dirty pages, and nothing I could tune would make any difference.
> vfs_cache_pressure makes no difference - I would set it so high
> it should trim the entire inode caches in a single pass, yet it
> didn't do anything. It was clear from tracing and live telemetry
> that the shrinkers were pretty much not running except when
> there was absolutely no memory free at all, and then they did
> the minimum necessary to free memory to make progress.
>
> So I went looking at the code, trying to find places where pages
> got reclaimed and the shrinkers weren't called. There's only one
> - kswapd doing boosted reclaim as per commit 1c30844d2dfe ("mm:
> reclaim small amounts of memory when an external fragmentation
> event occurs").

The watermark boosting introduced by the commit is triggered in response
to an allocation "fragmentation event". The boosting was not intended
to target THP specifically and triggers even if THP is disabled.
However, with Dave's perfectly reasonable workload, fragmentation events
can be very common given the ratio of slab to page cache allocations so
boosting remains active for long periods of time.

As high-order allocations might use compaction and compaction cannot
move slab pages the decision was made in the commit to special-case
kswapd when watermarks are boosted -- kswapd avoids reclaiming slab as
reclaiming slab does not directly help compaction.

As Dave notes, this decision means that slab can be artificially protected for long periods of time and messes up the balance with slab and page caches.

Removing the special casing can still indirectly help avoid fragmentation by avoiding fragmentation-causing events due to slab allocation as pages from a slab pageblock will have some slab objects freed. Furthermore, with the special casing, reclaim behaviour is unpredictable as kswapd sometimes examines slab and sometimes does not in a manner that is tricky to tune or analyse.

This patch removes the special casing. The downside is that this is not a universal performance win. Some benchmarks that depend on the residency of data when rereading metadata may see a regression when slab reclaim is restored to its original behaviour. Similarly, some benchmarks that only read-once or write-once may perform better when page reclaim is too aggressive. The primary upside is that slab shrinker is less surprising (arguably more sane but that's a matter of opinion), behaves consistently regardless of the fragmentation state of the system and properly obeys VM sysctls.

A fsmark benchmark configuration was constructed similar to what Dave reported and is codified by the mmtest configuration config-io-fsmark-small-file-stream. It was evaluated on a 1-socket machine to avoid dealing with NUMA-related issues and the timing of reclaim. The storage was an SSD Samsung Evo and a fresh trimmed XFS filesystem was used for the test data.

This is not an exact replication of Dave's setup. The configuration scales its parameters depending on the memory size of the SUT to behave similarly across machines. The parameters mean the first sample reported by fs_mark is using 50% of RAM which will barely be throttled and look like a big outlier. Dave used fake NUMA to have multiple kswapd instances which I didn't replicate. Finally, the number of iterations differ from Dave's test as the target disk was not large enough. While not identical, it should be representative.

```
  fsmark
                                     5.3.0-rc3             5.3.0-rc3
                                       vanilla          shrinker-v1r1
  Min       1-files/sec     4444.80 (   0.00%)     4765.60 (   7.22%)
  1st-qrtle 1-files/sec     5005.10 (   0.00%)     5091.70 (   1.73%)
  2nd-qrtle 1-files/sec     4917.80 (   0.00%)     4855.60 (  -1.26%)
  3rd-qrtle 1-files/sec     4667.40 (   0.00%)     4831.20 (   3.51%)
  Max-1     1-files/sec    11421.50 (   0.00%)     9999.30 ( -12.45%)
  Max-5     1-files/sec    11421.50 (   0.00%)     9999.30 ( -12.45%)
  Max-10    1-files/sec    11421.50 (   0.00%)     9999.30 ( -12.45%)
  Max-90    1-files/sec     4649.60 (   0.00%)     4780.70 (   2.82%)
  Max-95    1-files/sec     4491.00 (   0.00%)     4768.20 (   6.17%)
  Max-99    1-files/sec     4491.00 (   0.00%)     4768.20 (   6.17%)
  Max       1-files/sec    11421.50 (   0.00%)     9999.30 ( -12.45%)
  Hmean     1-files/sec     5004.75 (   0.00%)     5075.96 (   1.42%)
  Stddev    1-files/sec     1778.70 (   0.00%)     1369.66 (  23.00%)
  CoeffVar  1-files/sec       33.70 (   0.00%)       26.05 (  22.71%)
  BHmean-99 1-files/sec     5053.72 (   0.00%)     5101.52 (   0.95%)
  BHmean-95 1-files/sec     5053.72 (   0.00%)     5101.52 (   0.95%)
  BHmean-90 1-files/sec     5107.05 (   0.00%)     5131.41 (   0.48%)
  BHmean-75 1-files/sec     5208.45 (   0.00%)     5206.68 (  -0.03%)
  BHmean-50 1-files/sec     5405.53 (   0.00%)     5381.62 (  -0.44%)
  BHmean-25 1-files/sec     6179.75 (   0.00%)     6095.14 (  -1.37%)

                    5.3.0-rc3   5.3.0-rc3
                      vanillashrinker-v1r1
  Duration User         501.82      497.29
```

```
  Duration System      4401.44     4424.08
  Duration Elapsed     8124.76     8358.05
```

This is showing a slight skew for the max result representing a large outlier for the 1st, 2nd and 3rd quartile are similar indicating that the bulk of the results show little difference. Note that an earlier version of the fsmark configuration showed a regression but that included more samples taken while memory was still filling.

Note that the elapsed time is higher. Part of this is that the configuration included time to delete all the test files when the test completes -- the test automation handles the possibility of testing fsmark with multiple thread counts. Without the patch, many of these objects would be memory resident which is part of what the patch is addressing.

There are other important observations that justify the patch.

1. With the vanilla kernel, the number of dirty pages in the system is very low for much of the test. With this patch, dirty pages is generally kept at 10% which matches vm.dirty_background_ratio which is normal expected historical behaviour.

2. With the vanilla kernel, the ratio of Slab/Pagecache is close to 0.95 for much of the test i.e. Slab is being left alone and dominating memory consumption. With the patch applied, the ratio varies between 0.35 and 0.45 with the bulk of the measured ratios roughly half way between those values. This is a different balance to what Dave reported but it was at least consistent.

3. Slabs are scanned throughout the entire test with the patch applied. The vanille kernel has periods with no scan activity and then relatively massive spikes.

4. Without the patch, kswapd scan rates are very variable. With the patch, the scan rates remain quite steady.

4. Overall vmstats are closer to normal expectations

```
                                    5.3.0-rc3      5.3.0-rc3
                                      vanilla  shrinker-v1r1
Ops Direct pages scanned             99388.00      328410.00
Ops Kswapd pages scanned          45382917.00    33451026.00
Ops Kswapd pages reclaimed        30869570.00    25239655.00
Ops Direct pages reclaimed           74131.00        5830.00
Ops Kswapd efficiency %                 68.02          75.45
Ops Kswapd velocity                   5585.75        4002.25
Ops Page reclaim immediate         1179721.00      430927.00
Ops Slabs scanned                 62367361.00    73581394.00
Ops Direct inode steals               2103.00        1002.00
Ops Kswapd inode steals             570180.00     5183206.00
```

    o Vanilla kernel is hitting direct reclaim more frequently,
      not very much in absolute terms but the fact the patch
      reduces it is interesting
    o "Page reclaim immediate" in the vanilla kernel indicates
      dirty pages are being encountered at the tail of the LRU.
      This is generally bad and means in this case that the LRU
      is not long enough for dirty pages to be cleaned by the
      background flush in time. This is much reduced by the
      patch.
    o With the patch, kswapd is reclaiming 10 times more slab
      pages than with the vanilla kernel. This is indicative
      of the watermark boosting over-protecting slab

```
    A more complete set of tests were run that were part of the basis for
    introducing boosting and while there are some differences, they are well
    within tolerances.

    Bottom line, the special casing kswapd to avoid slab behaviour is
    unpredictable and can lead to abnormal results for normal workloads.

    This patch restores the expected behaviour that slab and page cache is
    balanced consistently for a workload with a steady allocation ratio of
    slab/pagecache pages.  It also means that if there are workloads that
    favour the preservation of slab over pagecache that it can be tuned via
    vm.vfs_cache_pressure where as the vanilla kernel effectively ignores
    the parameter when boosting is active.

    Link: http://lkml.kernel.org/r/20190808182946.GM2739@techsingularity.net
    Fixes: 1c30844d2dfe ("mm: reclaim small amounts of memory when an external fragmentation
event occurs")
    Signed-off-by: Mel Gorman <mgorman@techsingularity.net>
    Reviewed-by: Dave Chinner <dchinner@redhat.com>
    Acked-by: Vlastimil Babka <vbabka@suse.cz>
    Cc: Michal Hocko <mhocko@kernel.org>
    Cc: <stable@vger.kernel.org>    [5.0+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 04a6826b8edb9693bbb7aaf82376c5c950aba43a
Author: Isaac J. Manjarres <isaacm@codeaurora.org>
Date:   Tue Aug 13 15:37:37 2019 -0700

    mm/usercopy: use memory range to be accessed for wraparound check

    commit 951531691c4bcaa59f56a316e018bc2ff1ddf855 upstream.

    Currently, when checking to see if accessing n bytes starting at address
    "ptr" will cause a wraparound in the memory addresses, the check in
    check_bogus_address() adds an extra byte, which is incorrect, as the
    range of addresses that will be accessed is [ptr, ptr + (n - 1)].

    This can lead to incorrectly detecting a wraparound in the memory
    address, when trying to read 4 KB from memory that is mapped to the the
    last possible page in the virtual address space, when in fact, accessing
    that range of memory would not cause a wraparound to occur.

    Use the memory range that will actually be accessed when considering if
    accessing a certain amount of bytes will cause the memory address to
    wrap around.

    Link: http://lkml.kernel.org/r/1564509253-23287-1-git-send-email-isaacm@codeaurora.org
    Fixes: f5509cc18daa ("mm: Hardened usercopy")
    Signed-off-by: Prasad Sodagudi <psodagud@codeaurora.org>
    Signed-off-by: Isaac J. Manjarres <isaacm@codeaurora.org>
    Co-developed-by: Prasad Sodagudi <psodagud@codeaurora.org>
    Reviewed-by: William Kucharski <william.kucharski@oracle.com>
    Acked-by: Kees Cook <keescook@chromium.org>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Trilok Soni <tsoni@codeaurora.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5ae015cde4bdb744994605a9b733d81829492281
Author: Miles Chen <miles.chen@mediatek.com>
Date:   Tue Aug 13 15:37:28 2019 -0700
```

```
    mm/memcontrol.c: fix use after free in mem_cgroup_iter()

    commit 54a83d6bcbf8f4700013766b974bf9190d40b689 upstream.

    This patch is sent to report an use after free in mem_cgroup_iter()
    after merging commit be2657752e9e ("mm: memcg: fix use after free in
    mem_cgroup_iter()").

    I work with android kernel tree (4.9 & 4.14), and commit be2657752e9e
    ("mm: memcg: fix use after free in mem_cgroup_iter()") has been merged
    to the trees.  However, I can still observe use after free issues
    addressed in the commit be2657752e9e.  (on low-end devices, a few times
    this month)

    backtrace:
            css_tryget <- crash here
            mem_cgroup_iter
            shrink_node
            shrink_zones
            do_try_to_free_pages
            try_to_free_pages
            __perform_reclaim
            __alloc_pages_direct_reclaim
            __alloc_pages_slowpath
            __alloc_pages_nodemask

    To debug, I poisoned mem_cgroup before freeing it:

      static void __mem_cgroup_free(struct mem_cgroup *memcg)
            for_each_node(node)
            free_mem_cgroup_per_node_info(memcg, node);
            free_percpu(memcg->stat);
      +     /* poison memcg before freeing it */
      +     memset(memcg, 0x78, sizeof(struct mem_cgroup));
            kfree(memcg);
      }

    The coredump shows the position=0xdbbc2a00 is freed.

      (gdb) p/x ((struct mem_cgroup_per_node *)0xe5009e00)->iter[8]
      $13 = {position = 0xdbbc2a00, generation = 0x2efd}

      0xdbbc2a00:     0xdbbc2e00      0x00000000      0xdbbc2800      0x00000100
      0xdbbc2a10:     0x00000200      0x78787878      0x00026218      0x00000000
      0xdbbc2a20:     0xdcad6000      0x00000001      0x78787800      0x00000000
      0xdbbc2a30:     0x78780000      0x00000000      0x0068fb84      0x78787878
      0xdbbc2a40:     0x78787878      0x78787878      0x78787878      0xe3fa5cc0
      0xdbbc2a50:     0x78787878      0x78787878      0x00000000      0x00000000
      0xdbbc2a60:     0x00000000      0x00000000      0x00000000      0x00000000
      0xdbbc2a70:     0x00000000      0x00000000      0x00000000      0x00000000
      0xdbbc2a80:     0x00000000      0x00000000      0x00000000      0x00000000
      0xdbbc2a90:     0x00000001      0x00000000      0x00000000      0x00100000
      0xdbbc2aa0:     0x00000001      0xdbbc2ac8      0x00000000      0x00000000
      0xdbbc2ab0:     0x00000000      0x00000000      0x00000000      0x00000000
      0xdbbc2ac0:     0x00000000      0x00000000      0xe5b02618      0x00001000
      0xdbbc2ad0:     0x00000000      0x78787878      0x78787878      0x78787878
      0xdbbc2ae0:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2af0:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b00:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b10:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b20:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b30:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b40:     0x78787878      0x78787878      0x78787878      0x78787878
      0xdbbc2b50:     0x78787878      0x78787878      0x78787878      0x78787878
```

```
        0xdbbc2b60:     0x78787878      0x78787878      0x78787878      0x78787878
        0xdbbc2b70:     0x78787878      0x78787878      0x78787878      0x78787878
        0xdbbc2b80:     0x78787878      0x78787878      0x00000000      0x78787878
        0xdbbc2b90:     0x78787878      0x78787878      0x78787878      0x78787878
        0xdbbc2ba0:     0x78787878      0x78787878      0x78787878      0x78787878
```

In the reclaim path, try_to_free_pages() does not setup
sc.target_mem_cgroup and sc is passed to do_try_to_free_pages(), ...,
shrink_node().

In mem_cgroup_iter(), root is set to root_mem_cgroup because
sc->target_mem_cgroup is NULL. It is possible to assign a memcg to
root_mem_cgroup.nodeinfo.iter in mem_cgroup_iter().

```
        try_to_free_pages
                struct scan_control sc = {...}, target_mem_cgroup is 0x0;
        do_try_to_free_pages
        shrink_zones
        shrink_node
                 mem_cgroup *root = sc->target_mem_cgroup;
                 memcg = mem_cgroup_iter(root, NULL, &reclaim);
        mem_cgroup_iter()
                if (!root)
                        root = root_mem_cgroup;
                ...

                css = css_next_descendant_pre(css, &root->css);
                memcg = mem_cgroup_from_css(css);
                cmpxchg(&iter->position, pos, memcg);
```

My device uses memcg non-hierarchical mode. When we release a memcg:
invalidate_reclaim_iterators() reaches only dead_memcg and its parents.
If non-hierarchical mode is used, invalidate_reclaim_iterators() never
reaches root_mem_cgroup.

```
  static void invalidate_reclaim_iterators(struct mem_cgroup *dead_memcg)
  {
        struct mem_cgroup *memcg = dead_memcg;

        for (; memcg; memcg = parent_mem_cgroup(memcg)
        ...
  }
```

So the use after free scenario looks like:

```
  CPU1                                          CPU2

  try_to_free_pages
  do_try_to_free_pages
  shrink_zones
  shrink_node
  mem_cgroup_iter()
      if (!root)
        root = root_mem_cgroup;
      ...
      css = css_next_descendant_pre(css, &root->css);
      memcg = mem_cgroup_from_css(css);
      cmpxchg(&iter->position, pos, memcg);

                                        invalidate_reclaim_iterators(memcg);
                                        ...
                                        __mem_cgroup_free()
                                                kfree(memcg);

  try_to_free_pages
```

```
       do_try_to_free_pages
       shrink_zones
       shrink_node
       mem_cgroup_iter()
           if (!root)
             root = root_mem_cgroup;
           ...
           mz = mem_cgroup_nodeinfo(root, reclaim->pgdat->node_id);
           iter = &mz->iter[reclaim->priority];
           pos = READ_ONCE(iter->position);
           css_tryget(&pos->css) <- use after free

    To avoid this, we should also invalidate root_mem_cgroup.nodeinfo.iter
    in invalidate_reclaim_iterators().

    [cai@lca.pw: fix -Wparentheses compilation warning]
      Link: http://lkml.kernel.org/r/1564580753-17531-1-git-send-email-cai@lca.pw
    Link: http://lkml.kernel.org/r/20190730015729.4406-1-miles.chen@mediatek.com
    Fixes: 5ac8fb31ad2e ("mm: memcontrol: convert reclaim iterator to simple css
refcounting")
    Signed-off-by: Miles Chen <miles.chen@mediatek.com>
    Signed-off-by: Qian Cai <cai@lca.pw>
    Acked-by: Michal Hocko <mhocko@suse.com>
    Cc: Johannes Weiner <hannes@cmpxchg.org>
    Cc: Vladimir Davydov <vdavydov.dev@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a6b0004e741c4a42975fff5d773f2da6cf598b7c
Author: Henry Burns <henryburns@google.com>
Date:   Tue Aug 13 15:37:25 2019 -0700

    mm/z3fold.c: fix z3fold_destroy_pool() race condition

    commit b997052bc3ac444a0bceab1093aff7ae71ed419e upstream.

    The constraint from the zpool use of z3fold_destroy_pool() is there are
    no outstanding handles to memory (so no active allocations), but it is
    possible for there to be outstanding work on either of the two wqs in
    the pool.

    Calling z3fold_deregister_migration() before the workqueues are drained
    means that there can be allocated pages referencing a freed inode,
    causing any thread in compaction to be able to trip over the bad pointer
    in PageMovable().

    Link: http://lkml.kernel.org/r/20190726224810.79660-2-henryburns@google.com
    Fixes: 1f862989b04a ("mm/z3fold.c: support page migration")
    Signed-off-by: Henry Burns <henryburns@google.com>
    Reviewed-by: Shakeel Butt <shakeelb@google.com>
    Reviewed-by: Jonathan Adams <jwadams@google.com>
    Cc: Vitaly Vul <vitaly.vul@sony.com>
    Cc: Vitaly Wool <vitalywool@gmail.com>
    Cc: David Howells <dhowells@redhat.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Al Viro <viro@zeniv.linux.org.uk>
    Cc: Henry Burns <henrywolfeburns@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d87e9ae7f6059350188379fd28d813ca7776de0a
```

```
Author: Henry Burns <henryburns@google.com>
Date:   Tue Aug 13 15:37:21 2019 -0700

    mm/z3fold.c: fix z3fold_destroy_pool() ordering

    commit 6051d3bd3b91e96c59e62b8be2dba1cc2b19ee40 upstream.

    The constraint from the zpool use of z3fold_destroy_pool() is there are
    no outstanding handles to memory (so no active allocations), but it is
    possible for there to be outstanding work on either of the two wqs in
    the pool.

    If there is work queued on pool->compact_workqueue when it is called,
    z3fold_destroy_pool() will do:

       z3fold_destroy_pool()
         destroy_workqueue(pool->release_wq)
         destroy_workqueue(pool->compact_wq)
           drain_workqueue(pool->compact_wq)
             do_compact_page(zhdr)
               kref_put(&zhdr->refcount)
                 __release_z3fold_page(zhdr, ...)
                   queue_work_on(pool->release_wq, &pool->work) *BOOM*

    So compact_wq needs to be destroyed before release_wq.

    Link: http://lkml.kernel.org/r/20190726224810.79660-1-henryburns@google.com
    Fixes: 5d03a6613957 ("mm/z3fold.c: use kref to prevent page free/compact race")
    Signed-off-by: Henry Burns <henryburns@google.com>
    Reviewed-by: Shakeel Butt <shakeelb@google.com>
    Reviewed-by: Jonathan Adams <jwadams@google.com>
    Cc: Vitaly Vul <vitaly.vul@sony.com>
    Cc: Vitaly Wool <vitalywool@gmail.com>
    Cc: David Howells <dhowells@redhat.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Al Viro <viro@zeniv.linux.org.uk
    Cc: Henry Burns <henrywolfeburns@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5c0e391bfa8a30120260c2214472e8d6316dca0b
Author: Yang Shi <yang.shi@linux.alibaba.com>
Date:   Tue Aug 13 15:37:18 2019 -0700

    mm: mempolicy: handle vma with unmovable pages mapped correctly in mbind

    commit a53190a4aaa36494f4d7209fd1fcc6f2ee08e0e0 upstream.

    When running syzkaller internally, we ran into the below bug on 4.9.x
    kernel:

      kernel BUG at mm/huge_memory.c:2124!
      invalid opcode: 0000 [#1] SMP KASAN
      CPU: 0 PID: 1518 Comm: syz-executor107 Not tainted 4.9.168+ #2
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 0.5.1 01/01/2011
      task: ffff880067b34900 task.stack: ffff880068998000
      RIP: split_huge_page_to_list+0x8fb/0x1030 mm/huge_memory.c:2124
      Call Trace:
        split_huge_page include/linux/huge_mm.h:100 [inline]
        queue_pages_pte_range+0x7e1/0x1480 mm/mempolicy.c:538
        walk_pmd_range mm/pagewalk.c:50 [inline]
        walk_pud_range mm/pagewalk.c:90 [inline]
        walk_pgd_range mm/pagewalk.c:116 [inline]
```

```
          __walk_page_range+0x44a/0xdb0 mm/pagewalk.c:208
          walk_page_range+0x154/0x370 mm/pagewalk.c:285
          queue_pages_range+0x115/0x150 mm/mempolicy.c:694
          do_mbind mm/mempolicy.c:1241 [inline]
          SYSC_mbind+0x3c3/0x1030 mm/mempolicy.c:1370
          SyS_mbind+0x46/0x60 mm/mempolicy.c:1352
          do_syscall_64+0x1d2/0x600 arch/x86/entry/common.c:282
          entry_SYSCALL_64_after_swapgs+0x5d/0xdb
        Code: c7 80 1c 02 00 e8 26 0a 76 01 <0f> 0b 48 c7 c7 40 46 45 84 e8 4c
        RIP  [<ffffffff81895d6b>] split_huge_page_to_list+0x8fb/0x1030 mm/huge_memory.c:2124
         RSP <ffff88006899f980>
```

with the below test:

```
        uint64_t r[1] = {0xffffffffffffffff};

        int main(void)
        {
              syscall(__NR_mmap, 0x20000000, 0x1000000, 3, 0x32, -1, 0);
                                    intptr_t res = 0;
              res = syscall(__NR_socket, 0x11, 3, 0x300);
              if (res != -1)
                      r[0] = res;
              *(uint32_t*)0x20000040 = 0x10000;
              *(uint32_t*)0x20000044 = 1;
              *(uint32_t*)0x20000048 = 0xc520;
              *(uint32_t*)0x2000004c = 1;
              syscall(__NR_setsockopt, r[0], 0x107, 0xd, 0x20000040, 0x10);
              syscall(__NR_mmap, 0x20fed000, 0x10000, 0, 0x8811, r[0], 0);
              *(uint64_t*)0x20000340 = 2;
              syscall(__NR_mbind, 0x20ff9000, 0x4000, 0x4002, 0x20000340, 0x45d4, 3);
              return 0;
        }
```

Actually the test does:

```
        mmap(0x20000000, 16777216, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
 -1, 0) = 0x20000000
        socket(AF_PACKET, SOCK_RAW, 768)        = 3
        setsockopt(3, SOL_PACKET, PACKET_TX_RING, {block_size=65536, block_nr=1,
 frame_size=50464, frame_nr=1}, 16) = 0
        mmap(0x20fed000, 65536, PROT_NONE, MAP_SHARED|MAP_FIXED|MAP_POPULATE|MAP_DENYWRITE, 3,
 0) = 0x20fed000
        mbind(..., MPOL_MF_STRICT|MPOL_MF_MOVE) = 0
```

The setsockopt() would allocate compound pages (16 pages in this test)
for packet tx ring, then the mmap() would call packet_mmap() to map the
pages into the user address space specified by the mmap() call.

When calling mbind(), it would scan the vma to queue the pages for
migration to the new node. It would split any huge page since 4.9
doesn't support THP migration, however, the packet tx ring compound
pages are not THP and even not movable. So, the above bug is triggered.

However, the later kernel is not hit by this issue due to commit
d44d363f6578 ("mm: don't assume anonymous pages have SwapBacked flag"),
which just removes the PageSwapBacked check for a different reason.

But, there is a deeper issue. According to the semantic of mbind(), it
should return -EIO if MPOL_MF_MOVE or MPOL_MF_MOVE_ALL was specified and
MPOL_MF_STRICT was also specified, but the kernel was unable to move all
existing pages in the range. The tx ring of the packet socket is
definitely not movable, however, mbind() returns success for this case.

Although the most socket file associates with non-movable pages, but XDP

```
    may have movable pages from gup.  So, it sounds not fine to just check
    the underlying file type of vma in vma_migratable().

    Change migrate_page_add() to check if the page is movable or not, if it
    is unmovable, just return -EIO.  But do not abort pte walk immediately,
    since there may be pages off LRU temporarily.  We should migrate other
    pages if MPOL_MF_MOVE* is specified.  Set has_unmovable flag if some
    paged could not be not moved, then return -EIO for mbind() eventually.

    With this change the above test would return -EIO as expected.

    [yang.shi@linux.alibaba.com: fix review comments from Vlastimil]
      Link: http://lkml.kernel.org/r/1563556862-54056-3-git-send-email-
yang.shi@linux.alibaba.com
    Link: http://lkml.kernel.org/r/1561162809-59140-3-git-send-email-
yang.shi@linux.alibaba.com
    Signed-off-by: Yang Shi <yang.shi@linux.alibaba.com>
    Reviewed-by: Vlastimil Babka <vbabka@suse.cz>
    Cc: Michal Hocko <mhocko@suse.com>
    Cc: Mel Gorman <mgorman@techsingularity.net>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f796f8de30640e3c7583bae02b13d159bae94ac3
Author: Yang Shi <yang.shi@linux.alibaba.com>
Date:   Tue Aug 13 15:37:15 2019 -0700

    mm: mempolicy: make the behavior consistent when MPOL_MF_MOVE* and MPOL_MF_STRICT were
specified

    commit d883544515aae54842c21730b880172e7894fde9 upstream.

    When both MPOL_MF_MOVE* and MPOL_MF_STRICT was specified, mbind() should
    try best to migrate misplaced pages, if some of the pages could not be
    migrated, then return -EIO.

    There are three different sub-cases:
     1. vma is not migratable
     2. vma is migratable, but there are unmovable pages
     3. vma is migratable, pages are movable, but migrate_pages() fails

    If #1 happens, kernel would just abort immediately, then return -EIO,
    after a7f40cfe3b7a ("mm: mempolicy: make mbind() return -EIO when
    MPOL_MF_STRICT is specified").

    If #3 happens, kernel would set policy and migrate pages with
    best-effort, but won't rollback the migrated pages and reset the policy
    back.

    Before that commit, they behaves in the same way.  It'd better to keep
    their behavior consistent.  But, rolling back the migrated pages and
    resetting the policy back sounds not feasible, so just make #1 behave as
    same as #3.

    Userspace will know that not everything was successfully migrated (via
    -EIO), and can take whatever steps it deems necessary - attempt
    rollback, determine which exact page(s) are violating the policy, etc.

    Make queue_pages_range() return 1 to indicate there are unmovable pages
    or vma is not migratable.

    The #2 is not handled correctly in the current kernel, the following
    patch will fix it.
```

```
    [yang.shi@linux.alibaba.com: fix review comments from Vlastimil]
      Link: http://lkml.kernel.org/r/1563556862-54056-2-git-send-email-
 yang.shi@linux.alibaba.com
    Link: http://lkml.kernel.org/r/1561162809-59140-2-git-send-email-
 yang.shi@linux.alibaba.com
    Signed-off-by: Yang Shi <yang.shi@linux.alibaba.com>
    Reviewed-by: Vlastimil Babka <vbabka@suse.cz>
    Cc: Michal Hocko <mhocko@suse.com>
    Cc: Mel Gorman <mgorman@techsingularity.net>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit b65f418c824143b3da9dfd9e1d99a18bbb53b05f
 Author: Ralph Campbell <rcampbell@nvidia.com>
 Date:   Tue Aug 13 15:37:11 2019 -0700

    mm/hmm: fix bad subpage pointer in try_to_unmap_one

    commit 1de13ee59225dfc98d483f8cce7d83f97c0b31de upstream.

    When migrating an anonymous private page to a ZONE_DEVICE private page,
    the source page->mapping and page->index fields are copied to the
    destination ZONE_DEVICE struct page and the page_mapcount() is
    increased.  This is so rmap_walk() can be used to unmap and migrate the
    page back to system memory.

    However, try_to_unmap_one() computes the subpage pointer from a swap pte
    which computes an invalid page pointer and a kernel panic results such
    as:

      BUG: unable to handle page fault for address: ffffea1fffffffc8

    Currently, only single pages can be migrated to device private memory so
    no subpage computation is needed and it can be set to "page".

    [rcampbell@nvidia.com: add comment]
      Link: http://lkml.kernel.org/r/20190724232700.23327-4-rcampbell@nvidia.com
    Link: http://lkml.kernel.org/r/20190719192955.30462-4-rcampbell@nvidia.com
    Fixes: a5430dda8a3a1c ("mm/migrate: support un-addressable ZONE_DEVICE page in
 migration")
    Signed-off-by: Ralph Campbell <rcampbell@nvidia.com>
    Cc: "Jérôme Glisse" <jglisse@redhat.com>
    Cc: "Kirill A. Shutemov" <kirill.shutemov@linux.intel.com>
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: Jason Gunthorpe <jgg@mellanox.com>
    Cc: John Hubbard <jhubbard@nvidia.com>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: Andrey Ryabinin <aryabinin@virtuozzo.com>
    Cc: Christoph Lameter <cl@linux.com>
    Cc: Dan Williams <dan.j.williams@intel.com>
    Cc: Dave Hansen <dave.hansen@linux.intel.com>
    Cc: Ira Weiny <ira.weiny@intel.com>
    Cc: Jan Kara <jack@suse.cz>
    Cc: Lai Jiangshan <jiangshanlai@gmail.com>
    Cc: Logan Gunthorpe <logang@deltatee.com>
    Cc: Martin Schwidefsky <schwidefsky@de.ibm.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Mel Gorman <mgorman@techsingularity.net>
    Cc: Michal Hocko <mhocko@suse.com>
    Cc: Pekka Enberg <penberg@kernel.org>
    Cc: Randy Dunlap <rdunlap@infradead.org>
```

```
    Cc: Vlastimil Babka <vbabka@suse.cz>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f20eee1ae81a0129805091c240283c322745f54c
Author: NeilBrown <neilb@suse.com>
Date:   Tue Aug 13 15:37:44 2019 -0700

    seq_file: fix problem when seeking mid-record

    commit 6a2aeab59e97101b4001bac84388fc49a992f87e upstream.

    If you use lseek or similar (e.g.  pread) to access a location in a
    seq_file file that is within a record, rather than at a record boundary,
    then the first read will return the remainder of the record, and the
    second read will return the whole of that same record (instead of the
    next record).  When seeking to a record boundary, the next record is
    correctly returned.

    This bug was introduced by a recent patch (identified below).  Before
    that patch, seq_read() would increment m->index when the last of the
    buffer was returned (m->count == 0).  After that patch, we rely on
    ->next to increment m->index after filling the buffer - but there was
    one place where that didn't happen.

    Link: https://lkml.kernel.org/lkml/877e7xl029.fsf@notabene.neil.brown.name/
    Fixes: 1f4aace60b0e ("fs/seq_file.c: simplify seq_file iteration code and interface")
    Signed-off-by: NeilBrown <neilb@suse.com>
    Reported-by: Sergei Turchanov <turchanov@farpost.com>
    Tested-by: Sergei Turchanov <turchanov@farpost.com>
    Cc: Alexander Viro <viro@zeniv.linux.org.uk>
    Cc: Markus Elfring <Markus.Elfring@web.de>
    Cc: <stable@vger.kernel.org>    [4.19+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2a239cbf82044c0969180cb648a0762dd681657
Author: Gustavo A. R. Silva <gustavo@embeddedor.com>
Date:   Fri Aug 9 23:43:56 2019 -0500

    sh: kernel: hw_breakpoint: Fix missing break in switch statement

    commit 1ee1119d184bb06af921b48c3021d921bbd85bac upstream.

    Add missing break statement in order to prevent the code from falling
    through to case SH_BREAKPOINT_WRITE.

    Fixes: 09a072947791 ("sh: hw-breakpoints: Add preliminary support for SH-4A UBC.")
    Cc: stable@vger.kernel.org
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Gustavo A. R. Silva <gustavo@embeddedor.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f820ecf609cc38676071ec6c6d3e96b26c73b747
Author: Roberto Sassu <roberto.sassu@huawei.com>
Date:   Mon Aug 5 18:44:27 2019 +0200

    KEYS: trusted: allow module init if TPM is inactive or deactivated

    commit 2d6c25215ab26bb009de3575faab7b685f138e92 upstream.
```

Commit c78719203fc6 ("KEYS: trusted: allow trusted.ko to initialize w/o a
TPM") allows the trusted module to be loaded even if a TPM is not found, to
avoid module dependency problems.

However, trusted module initialization can still fail if the TPM is
inactive or deactivated. tpm_get_random() returns an error.

This patch removes the call to tpm_get_random() and instead extends the PCR
specified by the user with zeros. The security of this alternative is
equivalent to the previous one, as either option prevents with a PCR update
unsealing and misuse of sealed data by a user space process.

Even if a PCR is extended with zeros, instead of random data, it is still
computationally infeasible to find a value as input for a new PCR extend
operation, to obtain again the PCR value that would allow unsealing.

Cc: stable@vger.kernel.org
Fixes: 240730437deb ("KEYS: trusted: explicitly use tpm_chip structure...")
Signed-off-by: Roberto Sassu <roberto.sassu@huawei.com>
Reviewed-by: Tyler Hicks <tyhicks@canonical.com>
Suggested-by: Mimi Zohar <zohar@linux.ibm.com>
Reviewed-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
Signed-off-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>