```
commit 2ba0babe7865cd5f4fac3d76ad15d9b6131bd283
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Tue Sep 19 12:30:30 2023 +0200

    Linux 6.5.4

    Link: https://lore.kernel.org/r/20230917191051.639202302@linuxfoundation.org
    Tested-by: SeongJae Park <sj@kernel.org>
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Salvatore Bonaccorso <carnil@debian.org>
    Tested-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 09ea6d0f8cc1e64b97dffa060940827a6a0e9546
Author: Wesley Chalmers <wesley.chalmers@amd.com>
Date:   Wed Jun 21 19:13:26 2023 -0400

    drm/amd/display: Fix a bug when searching for insert_above_mpcc

    commit 3d028d5d60d516c536de1ddd3ebf3d55f3f8983b upstream.

    [WHY]
    Currently, when insert_plane is called with insert_above_mpcc
    parameter that is equal to tree->opp_list, the function returns NULL.

    [HOW]
    Instead, the function should insert the plane at the top of the tree.

    Cc: Mario Limonciello <mario.limonciello@amd.com>
    Cc: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Reviewed-by: Jun Lei <jun.lei@amd.com>
    Acked-by: Tom Chung <chiahsuan.chung@amd.com>
    Signed-off-by: Wesley Chalmers <wesley.chalmers@amd.com>
    Tested-by: Daniel Wheeler <daniel.wheeler@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 85746e2ab3fa8c392103507a2de765b2078a609f
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Sat Sep 16 12:31:42 2023 -0700

    vm: fix move_vma() memory accounting being off

    commit 3cec50490969afd4a76ccee441f747d869ccff77 upstream.

    Commit 408579cd627a ("mm: Update do_vmi_align_munmap() return
    semantics") seems to have updated one of the callers of do_vmi_munmap()
    incorrectly: it used to check for the error case (which didn't
    change: negative means error).

    That commit changed the check to the success case (which did change:
    before that commit, 0 was success, and 1 was "success and lock
    downgraded".  After the change, it's always 0 for success, and the lock
    will have been released if requested).

    This didn't change any actual VM behavior _except_ for memory accounting
```

```
    when 'VM_ACCOUNT' was set on the vma.  Which made the wrong return value
    test fairly subtle, since everything continues to work.

    Or rather - it continues to work but the "Committed memory" accounting
    goes all wonky (Committed_AS value in /proc/meminfo), and depending on
    settings that then causes problems much much later as the VM relies on
    bogus statistics for its heuristics.

    Revert that one line of the change back to the original logic.

    Fixes: 408579cd627a ("mm: Update do_vmi_align_munmap() return semantics")
    Reported-by: Christoph Biedl <linux-kernel.bfrz@manchmal.in-ulm.de>
    Reported-bisected-and-tested-by: Michael Labiuk <michael.labiuk@virtuozzo.com>
    Cc: Bagas Sanjaya <bagasdotme@gmail.com>
    Cc: Liam R. Howlett <Liam.Howlett@oracle.com>
    Link: https://lore.kernel.org/all/1694366957@msgid.manchmal.in-ulm.de/
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Helge Deller <deller@gmx.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 992b2ac783aad360b98ed9d4686e86176a20f6f1
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Sep 11 19:27:53 2023 -0700

    kcm: Fix error handling for SOCK_DGRAM in kcm_sendmsg().

    [ Upstream commit a22730b1b4bf437c6bbfdeff5feddf54be4aeada ]

    syzkaller found a memory leak in kcm_sendmsg(), and commit c821a88bd720
    ("kcm: Fix memory leak in error path of kcm_sendmsg()") suppressed it by
    updating kcm_tx_msg(head)->last_skb if partial data is copied so that the
    following sendmsg() will resume from the skb.

    However, we cannot know how many bytes were copied when we get the error.
    Thus, we could mess up the MSG_MORE queue.

    When kcm_sendmsg() fails for SOCK_DGRAM, we should purge the queue as we
    do so for UDP by udp_flush_pending_frames().

    Even without this change, when the error occurred, the following sendmsg()
    resumed from a wrong skb and the queue was messed up.  However, we have
    yet to get such a report, and only syzkaller stumbled on it.  So, this
    can be changed safely.

    Note this does not change SOCK_SEQPACKET behaviour.

    Fixes: c821a88bd720 ("kcm: Fix memory leak in error path of kcm_sendmsg()")
    Fixes: ab7ac4eb9832 ("kcm: Kernel Connection Multiplexor module")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230912022753.33327-1-kuniyu@amazon.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5e93c6044b92bc6bea2211992fdfa9eaa1702f67
Author: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
Date:   Tue Sep 12 10:49:35 2023 +0900

    net: renesas: rswitch: Fix unmasking irq condition

    [ Upstream commit e7b1ef29420fe52c2c1a273a9b4b36103a522625 ]

    Fix unmasking irq condition by using napi_complete_done(). Otherwise,
    redundant interrupts happen.

    Fixes: 3590918b5d07 ("net: ethernet: renesas: Add support for "Ethernet Switch"")
```

```
    Signed-off-by: Yoshihiro Shimoda <yoshihiro.shimoda.uh@renesas.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0e3ea7e82a06014b9baf1b84ba579c38cbff3558
Author: Corinna Vinschen <vinschen@redhat.com>
Date:   Mon Sep 11 13:28:49 2023 -0700

    igb: clean up in all error paths when enabling SR-IOV

    [ Upstream commit bc6ed2fa24b14e40e1005488bbe11268ce7108fa ]

    After commit 50f303496d92 ("igb: Enable SR-IOV after reinit"), removing
    the igb module could hang or crash (depending on the machine) when the
    module has been loaded with the max_vfs parameter set to some value != 0.

    In case of one test machine with a dual port 82580, this hang occurred:

    [  232.480687] igb 0000:41:00.1: removed PHC on enp65s0f1
    [  233.093257] igb 0000:41:00.1: IOV Disabled
    [  233.329969] pcieport 0000:40:01.0: AER: Multiple Uncorrected (Non-Fatal) err0
    [  233.340302] igb 0000:41:00.0: PCIe Bus Error: severity=Uncorrected (Non-Fata)
    [  233.352248] igb 0000:41:00.0:   device [8086:1516] error status/mask=00100000
    [  233.361088] igb 0000:41:00.0:    [20] UnsupReq               (First)
    [  233.368183] igb 0000:41:00.0: AER:   TLP Header: 40000001 0000040f cdbfc00c c
    [  233.376846] igb 0000:41:00.1: PCIe Bus Error: severity=Uncorrected (Non-Fata)
    [  233.388779] igb 0000:41:00.1:   device [8086:1516] error status/mask=00100000
    [  233.397629] igb 0000:41:00.1:    [20] UnsupReq               (First)
    [  233.404736] igb 0000:41:00.1: AER:   TLP Header: 40000001 0000040f cdbfc00c c
    [  233.538214] pci 0000:41:00.1: AER: can't recover (no error_detected callback)
    [  233.538401] igb 0000:41:00.0: removed PHC on enp65s0f0
    [  233.546197] pcieport 0000:40:01.0: AER: device recovery failed
    [  234.157244] igb 0000:41:00.0: IOV Disabled
    [  371.619705] INFO: task irq/35-aerdrv:257 blocked for more than 122 seconds.
    [  371.627489]       Not tainted 6.4.0-dirty #2
    [  371.632257] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this.
    [  371.641000] task:irq/35-aerdrv   state:D stack:0     pid:257   ppid:2      f0
    [  371.650330] Call Trace:
    [  371.653061]  <TASK>
    [  371.655407]  __schedule+0x20e/0x660
    [  371.659313]  schedule+0x5a/0xd0
    [  371.662824]  schedule_preempt_disabled+0x11/0x20
    [  371.667983]  __mutex_lock.constprop.0+0x372/0x6c0
    [  371.673237]  ? __pfx_aer_root_reset+0x10/0x10
    [  371.678105]  report_error_detected+0x25/0x1c0
    [  371.682974]  ? __pfx_report_normal_detected+0x10/0x10
    [  371.688618]  pci_walk_bus+0x72/0x90
    [  371.692519]  pcie_do_recovery+0xb2/0x330
    [  371.696899]  aer_process_err_devices+0x117/0x170
    [  371.702055]  aer_isr+0x1c0/0x1e0
    [  371.705661]  ? __set_cpus_allowed_ptr+0x54/0xa0
    [  371.710723]  ? __pfx_irq_thread_fn+0x10/0x10
    [  371.715496]  irq_thread_fn+0x20/0x60
    [  371.719491]  irq_thread+0xe6/0x1b0
    [  371.723291]  ? __pfx_irq_thread_dtor+0x10/0x10
    [  371.728255]  ? __pfx_irq_thread+0x10/0x10
    [  371.732731]  kthread+0xe2/0x110
    [  371.736243]  ? __pfx_kthread+0x10/0x10
    [  371.740430]  ret_from_fork+0x2c/0x50
    [  371.744428]  </TASK>

    The reproducer was a simple script:

      #!/bin/sh
```

```
       for i in `seq 1 5`; do
         modprobe -rv igb
         modprobe -v igb max_vfs=1
         sleep 1
         modprobe -rv igb
       done
```

    It turned out that this could only be reproduce on 82580 (quad and
    dual-port), but not on 82576, i350 and i210. Further debugging showed
    that igb_enable_sriov()'s call to pci_enable_sriov() is failing, because
    dev->is_physfn is 0 on 82580.

    Prior to commit 50f303496d92 ("igb: Enable SR-IOV after reinit"),
    igb_enable_sriov() jumped into the "err_out" cleanup branch. After this
    commit it only returned the error code.

    So the cleanup didn't take place, and the incorrect VF setup in the
    igb_adapter structure fooled the igb driver into assuming that VFs have
    been set up where no VF actually existed.

    Fix this problem by cleaning up again if pci_enable_sriov() fails.

    Fixes: 50f303496d92 ("igb: Enable SR-IOV after reinit")
    Signed-off-by: Corinna Vinschen <vinschen@redhat.com>
    Reviewed-by: Akihiko Odaki <akihiko.odaki@daynix.com>
    Tested-by: Rafal Romanowski <rafal.romanowski@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 849b9fb734577e996d32cc3b11d1af739ce09249
Author: Vadim Fedorenko <vadim.fedorenko@linux.dev>
Date: Mon Sep 11 13:28:14 2023 -0700

    ixgbe: fix timestamp configuration code

    [Upstream commit 3c44191dd76cf9c0cc49adaf34384cbd42ef8ad2]

    The commit in fixes introduced flags to control the status of hardware
    configuration while processing packets. At the same time another structure
    is used to provide configuration of timestamper to user-space applications.
    The way it was coded makes this structures go out of sync easily. The
    repro is easy for 82599 chips:

    [root@hostname ~]# hwstamp_ctl -i eth0 -r 12 -t 1
    current settings:
    tx_type 0
    rx_filter 0
    new settings:
    tx_type 1
    rx_filter 12

    The eth0 device is properly configured to timestamp any PTPv2 events.

    [root@hostname ~]# hwstamp_ctl -i eth0 -r 1 -t 1
    current settings:
    tx_type 1
    rx_filter 12
    SIOCSHWTSTAMP failed: Numerical result out of range
    The requested time stamping mode is not supported by the hardware.

    The error is properly returned because HW doesn't support all packets
    timestamping. But the adapter->flags is cleared of timestamp flags
    even though no HW configuration was done. From that point no RX timestamps
    are received by user-space application. But configuration shows good

```
    values:

    [root@hostname ~]# hwstamp_ctl -i eth0
    current settings:
    tx_type 1
    rx_filter 12

    Fix the issue by applying new flags only when the HW was actually
    configured.

    Fixes: a9763f3cb54c ("ixgbe: Update PTP to support X550EM_x devices")
    Signed-off-by: Vadim Fedorenko <vadim.fedorenko@linux.dev>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Tested-by: Pucha Himasekhar Reddy <himasekharx.reddy.pucha@intel.com> (A Contingent
worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 591d4ca76dd239ff53617ee1828d712adea85277
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Sep 11 11:36:58 2023 -0700

    selftest: tcp: Fix address length in bind_wildcard.c.

    [ Upstream commit 0071d15517b4a3d265abc00395beb1138e7236c7 ]

    The selftest passes the IPv6 address length for an IPv4 address.
    We should pass the correct length.

    Note inet_bind_sk() does not check if the size is larger than
    sizeof(struct sockaddr_in), so there is no real bug in this
    selftest.

    Fixes: 13715acf8ab5 ("selftest: Add test for bind() conflicts.")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 25c757e451c1fa957178508417282dc5bf35442c
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Sep 11 11:36:57 2023 -0700

    tcp: Fix bind() regression for v4-mapped-v6 non-wildcard address.

    [ Upstream commit c48ef9c4aed3632566b57ba66cec6ec78624d4cb ]

    Since bhash2 was introduced, the example below does not work as expected.
    These two bind() should conflict, but the 2nd bind() now succeeds.

      from socket import *

      s1 = socket(AF_INET6, SOCK_STREAM)
      s1.bind(('::ffff:127.0.0.1', 0))

      s2 = socket(AF_INET, SOCK_STREAM)
      s2.bind(('127.0.0.1', s1.getsockname()[1]))

    During the 2nd bind() in inet_csk_get_port(), inet_bind2_bucket_find()
    fails to find the 1st socket's tb2, so inet_bind2_bucket_create() allocates
    a new tb2 for the 2nd socket.  Then, we call inet_csk_bind_conflict() that
    checks conflicts in the new tb2 by inet_bhash2_conflict().  However, the
    new tb2 does not include the 1st socket, thus the bind() finally succeeds.

    In this case, inet_bind2_bucket_match() must check if AF_INET6 tb2 has
```

```
    the conflicting v4-mapped-v6 address so that inet_bind2_bucket_find()
    returns the 1st socket's tb2.

    Note that if we bind two sockets to 127.0.0.1 and then ::FFFF:127.0.0.1,
    the 2nd bind() fails properly for the same reason mentinoed in the previous
    commit.

    Fixes: 28044fc1d495 ("net: Add a bhash2 table hashed by port and address")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Acked-by: Andrei Vagin <avagin@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a548ad0e346806b0912ea6b3f66e443c168f426
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Sep 11 11:36:56 2023 -0700

    tcp: Fix bind() regression for v4-mapped-v6 wildcard address.

    [ Upstream commit aa99e5f87bd54db55dd37cb130bd5eb55933027f ]

    Andrei Vagin reported bind() regression with strace logs.

    If we bind() a TCPv6 socket to ::FFFF:0.0.0.0 and then bind() a TCPv4
    socket to 127.0.0.1, the 2nd bind() should fail but now succeeds.

      from socket import *

      s1 = socket(AF_INET6, SOCK_STREAM)
      s1.bind(('::ffff:0.0.0.0', 0))

      s2 = socket(AF_INET, SOCK_STREAM)
      s2.bind(('127.0.0.1', s1.getsockname()[1]))

    During the 2nd bind(), if tb->family is AF_INET6 and sk->sk_family is
    AF_INET in inet_bind2_bucket_match_addr_any(), we still need to check
    if tb has the v4-mapped-v6 wildcard address.

    The example above does not work after commit 5456262d2baa ("net: Fix
    incorrect address comparison when searching for a bind2 bucket"), but
    the blamed change is not the commit.

    Before the commit, the leading zeros of ::FFFF:0.0.0.0 were treated
    as 0.0.0.0, and the sequence above worked by chance.  Technically, this
    case has been broken since bhash2 was introduced.

    Note that if we bind() two sockets to 127.0.0.1 and then ::FFFF:0.0.0.0,
    the 2nd bind() fails properly because we fall back to using bhash to
    detect conflicts for the v4-mapped-v6 address.

    Fixes: 28044fc1d495 ("net: Add a bhash2 table hashed by port and address")
    Reported-by: Andrei Vagin <avagin@google.com>
    Closes: https://lore.kernel.org/netdev/ZPuYBOFC8zsK6r9T@google.com/
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1f97354c16efc94853f563e972a5b7dd334c71fa
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Sep 11 11:36:55 2023 -0700

    tcp: Factorise sk_family-independent comparison in inet_bind2_bucket_match(_addr_any).
```

```
    [ Upstream commit c6d277064b1da7f9015b575a562734de87a7e463 ]

    This is a prep patch to make the following patches cleaner that touch
    inet_bind2_bucket_match() and inet_bind2_bucket_match_addr_any().

    Both functions have duplicated comparison for netns, port, and l3mdev.
    Let's factorise them.

    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Stable-dep-of: aa99e5f87bd5 ("tcp: Fix bind() regression for v4-mapped-v6 wildcard
address.")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7a811588048450e022ffcd3589e03c0a7574bb88
Author: Eric Dumazet <edumazet@google.com>
Date:   Mon Sep 11 15:42:13 2023 +0000

    ipv6: fix ip6_sock_set_addr_preferences() typo

    [ Upstream commit 8cdd9f1aaedf823006449faa4e540026c692ac43 ]

    ip6_sock_set_addr_preferences() second argument should be an integer.

    SUNRPC attempts to set IPV6_PREFER_SRC_PUBLIC were
    translated to IPV6_PREFER_SRC_TMP

    Fixes: 18d5ad623275 ("ipv6: add ip6_sock_set_addr_preferences")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: Chuck Lever <chuck.lever@oracle.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Link: https://lore.kernel.org/r/20230911154213.713941-1-edumazet@google.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1cf87594991845b58db6dc5347bc233f6f7c7f2
Author: Toke Høiland-Jørgensen <toke@redhat.com>
Date:   Mon Sep 11 15:58:25 2023 +0200

    veth: Update XDP feature set when bringing up device

    [ Upstream commit 7a6102aa6df0d5d032b4cbc51935d1d4cda17254 ]

    There's an early return in veth_set_features() if the device is in a down
    state, which leads to the XDP feature flags not being updated when enabling
    GRO while the device is down. Which in turn leads to XDP_REDIRECT not
    working, because the redirect code now checks the flags.

    Fix this by updating the feature flags after bringing the device up.

    Before this patch:

    NETDEV_XDP_ACT_BASIC:           yes
    NETDEV_XDP_ACT_REDIRECT:        yes
    NETDEV_XDP_ACT_NDO_XMIT:        no
    NETDEV_XDP_ACT_XSK_ZEROCOPY:    no
    NETDEV_XDP_ACT_HW_OFFLOAD:      no
    NETDEV_XDP_ACT_RX_SG:           yes
    NETDEV_XDP_ACT_NDO_XMIT_SG:     no

    After this patch:

    NETDEV_XDP_ACT_BASIC:           yes
```

```
    NETDEV_XDP_ACT_REDIRECT:        yes
    NETDEV_XDP_ACT_NDO_XMIT:        yes
    NETDEV_XDP_ACT_XSK_ZEROCOPY:    no
    NETDEV_XDP_ACT_HW_OFFLOAD:      no
    NETDEV_XDP_ACT_RX_SG:           yes
    NETDEV_XDP_ACT_NDO_XMIT_SG:     yes

    Fixes: fccca038f300 ("veth: take into account device reconfiguration for xdp_features
flag")
    Fixes: 66c0e13ad236 ("drivers: net: turn on XDP features")
    Signed-off-by: Toke Høiland-Jørgensen <toke@redhat.com>
    Link: https://lore.kernel.org/r/20230911135826.722295-1-toke@redhat.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1252572fea95161528b932193c7ddd0ab123ffc6
Author: Sascha Hauer <s.hauer@pengutronix.de>
Date:   Fri Sep 8 13:29:13 2023 +0200

    net: macb: fix sleep inside spinlock

    [ Upstream commit 403f0e771457e2b8811dc280719d11b9bacf10f4 ]

    macb_set_tx_clk() is called under a spinlock but itself calls clk_set_rate()
    which can sleep. This results in:

    | BUG: sleeping function called from invalid context at kernel/locking/mutex.c:580
    | pps pps1: new PPS source ptp1
    | in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 40, name: kworker/u4:3
    | preempt_count: 1, expected: 0
    | RCU nest depth: 0, expected: 0
    | 4 locks held by kworker/u4:3/40:
    |  #0: ffff000003409148
    | macb ff0c0000.ethernet: gem-ptp-timer ptp clock registered.
    |  ((wq_completion)events_power_efficient){+.+.}-{0:0}, at: process_one_work+0x14c/0x51c
    |  #1: ffff8000833cbdd8 ((work_completion)(&pl->resolve)){+.+.}-{0:0}, at:
process_one_work+0x14c/0x51c
    |  #2: ffff000004f01578 (&pl->state_mutex){+.+.}-{4:4}, at: phylink_resolve+0x44/0x4e8
    |  #3: ffff000004f06f50 (&bp->lock){....}-{3:3}, at: macb_mac_link_up+0x40/0x2ac
    | irq event stamp: 113998
    | hardirqs last  enabled at (113997): [<ffff800080e8503c>]
_raw_spin_unlock_irq+0x30/0x64
    | hardirqs last disabled at (113998): [<ffff800080e84478>]
_raw_spin_lock_irqsave+0xac/0xc8
    | softirqs last  enabled at (113608): [<ffff800080010630>] __do_softirq+0x430/0x4e4
    | softirqs last disabled at (113597): [<ffff80008001614c>] ____do_softirq+0x10/0x1c
    | CPU: 0 PID: 40 Comm: kworker/u4:3 Not tainted 6.5.0-11717-g9355ce8b2f50-dirty #368
    | Hardware name: ... ZynqMP ... (DT)
    | Workqueue: events_power_efficient phylink_resolve
    | Call trace:
    |  dump_backtrace+0x98/0xf0
    |  show_stack+0x18/0x24
    |  dump_stack_lvl+0x60/0xac
    |  dump_stack+0x18/0x24
    |  __might_resched+0x144/0x24c
    |  __might_sleep+0x48/0x98
    |  __mutex_lock+0x58/0x7b0
    |  mutex_lock_nested+0x24/0x30
    |  clk_prepare_lock+0x4c/0xa8
    |  clk_set_rate+0x24/0x8c
    |  macb_mac_link_up+0x25c/0x2ac
    |  phylink_resolve+0x178/0x4e8
    |  process_one_work+0x1ec/0x51c
    |  worker_thread+0x1ec/0x3e4
    |  kthread+0x120/0x124
```

```
      |  ret_from_fork+0x10/0x20

      The obvious fix is to move the call to macb_set_tx_clk() out of the
      protected area. This seems safe as rx and tx are both disabled anyway at
      this point.
      It is however not entirely clear what the spinlock shall protect. It
      could be the read-modify-write access to the NCFGR register, but this
      is accessed in macb_set_rx_mode() and macb_set_rxcsum_feature() as well
      without holding the spinlock. It could also be the register accesses
      done in mog_init_rings() or macb_init_buffers(), but again these
      functions are called without holding the spinlock in macb_hresp_error_task().
      The locking seems fishy in this driver and it might deserve another look
      before this patch is applied.

      Fixes: 633e98a711ac0 ("net: macb: use resolved link config in mac_link_up()")
      Signed-off-by: Sascha Hauer <s.hauer@pengutronix.de>
      Link: https://lore.kernel.org/r/20230908112913.1701766-1-s.hauer@pengutronix.de
      Signed-off-by: Paolo Abeni <pabeni@redhat.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 74aecad5da19004ccf6321fd397d14b10756622a
  Author: Liu Jian <liujian56@huawei.com>
  Date:   Sat Sep 9 16:14:34 2023 +0800

      net/tls: do not free tls_rec on async operation in bpf_exec_tx_verdict()

      [ Upstream commit cfaa80c91f6f99b9342b6557f0f0e1143e434066 ]

      I got the below warning when do fuzzing test:
      BUG: KASAN: null-ptr-deref in scatterwalk_copychunks+0x320/0x470
      Read of size 4 at addr 0000000000000008 by task kworker/u8:1/9

      CPU: 0 PID: 9 Comm: kworker/u8:1 Tainted: G           OE
      Hardware name: linux,dummy-virt (DT)
      Workqueue: pencrypt_parallel padata_parallel_worker
      Call trace:
       dump_backtrace+0x0/0x420
       show_stack+0x34/0x44
       dump_stack+0x1d0/0x248
       __kasan_report+0x138/0x140
       kasan_report+0x44/0x6c
       __asan_load4+0x94/0xd0
       scatterwalk_copychunks+0x320/0x470
       skcipher_next_slow+0x14c/0x290
       skcipher_walk_next+0x2fc/0x480
       skcipher_walk_first+0x9c/0x110
       skcipher_walk_aead_common+0x380/0x440
       skcipher_walk_aead_encrypt+0x54/0x70
       ccm_encrypt+0x13c/0x4d0
       crypto_aead_encrypt+0x7c/0xfc
       pcrypt_aead_enc+0x28/0x84
       padata_parallel_worker+0xd0/0x2dc
       process_one_work+0x49c/0xbdc
       worker_thread+0x124/0x880
       kthread+0x210/0x260
       ret_from_fork+0x10/0x18

      This is because the value of rec_seq of tls_crypto_info configured by the
      user program is too large, for example, 0xffffffffffffff. In addition, TLS
      is asynchronously accelerated. When tls_do_encryption() returns
      -EINPROGRESS and sk->sk_err is set to EBADMSG due to rec_seq overflow,
      skmsg is released before the asynchronous encryption process ends. As a
      result, the UAF problem occurs during the asynchronous processing of the
      encryption module.
```

```
    If the operation is asynchronous and the encryption module returns
    EINPROGRESS, do not free the record information.

    Fixes: 635d93981786 ("net/tls: free record only on encryption error")
    Signed-off-by: Liu Jian <liujian56@huawei.com>
    Reviewed-by: Sabrina Dubroca <sd@queasysnail.net>
    Link: https://lore.kernel.org/r/20230909081434.2324940-1-liujian56@huawei.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 85ad4eede956b3182e079390930e54b507a659c3
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Mon Sep 4 14:00:35 2023 +0200

    platform/mellanox: NVSW_SN2201 should depend on ACPI

    [ Upstream commit 0a138f1670bd1af13ba6949c48ea86ddd4bf557e ]

    The only probing method supported by the Nvidia SN2201 platform driver
    is probing through an ACPI match table.  Hence add a dependency on
    ACPI, to prevent asking the user about this driver when configuring a
    kernel without ACPI support.

    Fixes: 662f24826f95 ("platform/mellanox: Add support for new SN2201 system")
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Acked-by: Vadim Pasternak <vadimp@nvidia.com>
    Acked-by: Andi Shyti <andi.shyti@kernel.org>
    Link:
https://lore.kernel.org/r/ec5a4071691ab08d58771b7732a9988e89779268.1693828363.git.geert+rene
sas@glider.be
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d65f117e2ed4f4062a2f9702de9dec161159f6ed
Author: Shravan Kumar Ramani <shravankr@nvidia.com>
Date:   Tue Sep 5 08:49:33 2023 -0400

    platform/mellanox: mlxbf-pmc: Fix reading of unprogrammed events

    [ Upstream commit 0f5969452e162efc50bdc98968fb62b424a9874b ]

    This fix involves 2 changes:
     - All event regs have a reset value of 0, which is not a valid
       event_number as per the event_list for most blocks and hence seen
       as an error. Add a "disable" event with event_number 0 for all blocks.

     - The enable bit for each counter need not be checked before
       reading the event info, and hence removed.

    Fixes: 1a218d312e65 ("platform/mellanox: mlxbf-pmc: Add Mellanox BlueField PMC driver")
    Signed-off-by: Shravan Kumar Ramani <shravankr@nvidia.com>
    Reviewed-by: Vadim Pasternak <vadimp@nvidia.com>
    Reviewed-by: David Thompson <davthompson@nvidia.com>
    Link:
https://lore.kernel.org/r/04d0213932d32681de1c716b54320ed894e52425.1693917738.git.shravankr@
nvidia.com
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8748f70981ac2924beb04e3c1442f851e0e24029
Author: Shravan Kumar Ramani <shravankr@nvidia.com>
Date:   Tue Sep 5 08:49:32 2023 -0400

    platform/mellanox: mlxbf-pmc: Fix potential buffer overflows
```

```
    [ Upstream commit 80ccd40568bcd3655b0fd0be1e9b3379fd6e1056 ]

    Replace sprintf with sysfs_emit where possible.
    Size check in mlxbf_pmc_event_list_show should account for "\0".

    Fixes: 1a218d312e65 ("platform/mellanox: mlxbf-pmc: Add Mellanox BlueField PMC driver")
    Signed-off-by: Shravan Kumar Ramani <shravankr@nvidia.com>
    Reviewed-by: Vadim Pasternak <vadimp@nvidia.com>
    Reviewed-by: David Thompson <davthompson@nvidia.com>
    Link:
https://lore.kernel.org/r/bef39ef32319a31b32f999065911f61b0d3b17c3.1693917738.git.shravankr@
nvidia.com
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 23a9a765ba5e5149454aaf73142b666afec1e920
Author: Liming Sun <limings@nvidia.com>
Date:   Tue Aug 29 13:43:00 2023 -0400

    platform/mellanox: mlxbf-tmfifo: Drop jumbo frames

    [ Upstream commit fc4c655821546239abb3cf4274d66b9747aa87dd ]

    This commit drops over-sized network packets to avoid tmfifo
    queue stuck.

    Fixes: 1357dfd7261f ("platform/mellanox: Add TmFifo driver for Mellanox BlueField Soc")
    Signed-off-by: Liming Sun <limings@nvidia.com>
    Reviewed-by: Vadim Pasternak <vadimp@nvidia.com>
    Reviewed-by: David Thompson <davthompson@nvidia.com>
    Link:
https://lore.kernel.org/r/9318936c2447f76db475c985ca6d91f057efcd41.1693322547.git.limings@nv
idia.com
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8efa4d5833b756946aff51d2c6eb3af609a2d9b1
Author: Liming Sun <limings@nvidia.com>
Date:   Tue Aug 29 13:42:59 2023 -0400

    platform/mellanox: mlxbf-tmfifo: Drop the Rx packet if no more descriptors

    [ Upstream commit 78034cbece79c2d730ad0770b3b7f23eedbbecf5 ]

    This commit fixes tmfifo console stuck issue when the virtual
    networking interface is in down state. In such case, the network
    Rx descriptors runs out and causes the Rx network packet staying
    in the head of the tmfifo thus blocking the console packets. The
    fix is to drop the Rx network packet when no more Rx descriptors.
    Function name mlxbf_tmfifo_release_pending_pkt() is also renamed
    to mlxbf_tmfifo_release_pkt() to be more approperiate.

    Fixes: 1357dfd7261f ("platform/mellanox: Add TmFifo driver for Mellanox BlueField Soc")
    Signed-off-by: Liming Sun <limings@nvidia.com>
    Reviewed-by: Vadim Pasternak <vadimp@nvidia.com>
    Reviewed-by: David Thompson <davthompson@nvidia.com>
    Link:
https://lore.kernel.org/r/8c0177dc938ae03f52ff7e0b62dbeee74b7bec09.1693322547.git.limings@nv
idia.com
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit af8085e0fc3207ecbf8b9e7a635c790e36d058c6
Author: Shigeru Yoshida <syoshida@redhat.com>
Date:   Sun Sep 10 02:03:10 2023 +0900
```

```
    kcm: Fix memory leak in error path of kcm_sendmsg()

    [ Upstream commit c821a88bd720b0046433173185fd841a100d44ad ]

    syzbot reported a memory leak like below:

    BUG: memory leak
    unreferenced object 0xffff88810b088c00 (size 240):
      comm "syz-executor186", pid 5012, jiffies 4294943306 (age 13.680s)
      hex dump (first 32 bytes):
        00 89 08 0b 81 88 ff ff 00 00 00 00 00 00 00 00  ................
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
      backtrace:
        [<ffffffff83e5d5ff>] __alloc_skb+0x1ef/0x230 net/core/skbuff.c:634
        [<ffffffff84606e59>] alloc_skb include/linux/skbuff.h:1289 [inline]
        [<ffffffff84606e59>] kcm_sendmsg+0x269/0x1050 net/kcm/kcmsock.c:815
        [<ffffffff83e479c6>] sock_sendmsg_nosec net/socket.c:725 [inline]
        [<ffffffff83e479c6>] sock_sendmsg+0x56/0xb0 net/socket.c:748
        [<ffffffff83e47f55>] ____sys_sendmsg+0x365/0x470 net/socket.c:2494
        [<ffffffff83e4c389>] ___sys_sendmsg+0xc9/0x130 net/socket.c:2548
        [<ffffffff83e4c536>] __sys_sendmsg+0xa6/0x120 net/socket.c:2577
        [<ffffffff84ad7bb8>] do_syscall_x64 arch/x86/entry/common.c:50 [inline]
        [<ffffffff84ad7bb8>] do_syscall_64+0x38/0xb0 arch/x86/entry/common.c:80
        [<ffffffff84c0008b>] entry_SYSCALL_64_after_hwframe+0x63/0xcd

    In kcm_sendmsg(), kcm_tx_msg(head)->last_skb is used as a cursor to append
    newly allocated skbs to 'head'. If some bytes are copied, an error occurred,
    and jumped to out_error label, 'last_skb' is left unmodified. A later
    kcm_sendmsg() will use an obsoleted 'last_skb' reference, corrupting the
    'head' frag_list and causing the leak.

    This patch fixes this issue by properly updating the last allocated skb in
    'last_skb'.

    Fixes: ab7ac4eb9832 ("kcm: Kernel Connection Multiplexor module")
    Reported-and-tested-by: syzbot+6f98de741f7dbbfc4ccb@syzkaller.appspotmail.com
    Closes: https://syzkaller.appspot.com/bug?extid=6f98de741f7dbbfc4ccb
    Signed-off-by: Shigeru Yoshida <syoshida@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e552933d024299a44d7afe01caa37ffd7cabaaab
Author: Hayes Wang <hayeswang@realtek.com>
Date:   Fri Sep 8 15:01:52 2023 +0800

    r8152: check budget for r8152_poll()

    [ Upstream commit a7b8d60b37237680009dd0b025fe8c067aba0ee3 ]

    According to the document of napi, there is no rx process when the
    budget is 0. Therefore, r8152_poll() has to return 0 directly when the
    budget is equal to 0.

    Fixes: d2187f8e4454 ("r8152: divide the tx and rx bottom functions")
    Signed-off-by: Hayes Wang <hayeswang@realtek.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 271e48ec78bcfed9d6c3f59f78ee51fb2ef42ee2
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Sep 8 16:33:52 2023 +0300

    net: dsa: sja1105: block FDB accesses that are concurrent with a switch reset
```

```
    [ Upstream commit 86899e9e1e29e854b5f6dcc24ba4f75f792c89aa ]

    Currently, when we add the first sja1105 port to a bridge with
    vlan_filtering 1, then we sometimes see this output:

    sja1105 spi2.2: port 4 failed to read back entry for be:79:b4:9e:9e:96 vid 3088: -ENOENT
    sja1105 spi2.2: Reset switch and programmed static config. Reason: VLAN filtering
    sja1105 spi2.2: port 0 failed to add be:79:b4:9e:9e:96 vid 0 to fdb: -2

    It is because sja1105_fdb_add() runs from the dsa_owq which is no longer
    serialized with switch resets since it dropped the rtnl_lock() in the
    blamed commit.

    Either performing the FDB accesses before the reset, or after the reset,
    is equally fine, because sja1105_static_fdb_change() backs up those
    changes in the static config, but FDB access during reset isn't ok.

    Make sja1105_static_config_reload() take the fdb_lock to fix that.

    Fixes: 0faf890fc519 ("net: dsa: drop rtnl_lock from dsa_slave_switchdev_event_work")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2e52d6f9c651be854fe53b641e45f78fc9d1fec7
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Sep 8 16:33:51 2023 +0300

    net: dsa: sja1105: serialize sja1105_port_mcast_flood() with other FDB accesses

    [ Upstream commit ea32690daf4fa525dc5a4d164bd00ed8c756e1c6 ]

    sja1105_fdb_add() runs from the dsa_owq, and sja1105_port_mcast_flood()
    runs from switchdev_deferred_process_work(). Prior to the blamed commit,
    they used to be indirectly serialized through the rtnl_lock(), which
    no longer holds true because dsa_owq dropped that.

    So, it is now possible that we traverse the static config BLK_IDX_L2_LOOKUP
    elements concurrently compared to when we change them, in
    sja1105_static_fdb_change(). That is not ideal, since it might result in
    data corruption.

    Introduce a mutex which serializes accesses to the hardware FDB and to
    the static config elements for the L2 Address Lookup table.

    I can't find a good reason to add locking around sja1105_fdb_dump().
    I'll add it later if needed.

    Fixes: 0faf890fc519 ("net: dsa: drop rtnl_lock from dsa_slave_switchdev_event_work")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae3f25142a0c8d7f30cb1b43b34b62442d993c13
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Sep 8 16:33:50 2023 +0300

    net: dsa: sja1105: fix multicast forwarding working only for last added mdb entry

    [ Upstream commit 7cef293b9a634a05fcce9e1df4aee3aeed023345 ]

    The commit cited in Fixes: did 2 things: it refactored the read-back
    polling from sja1105_dynamic_config_read() into a new function,
    sja1105_dynamic_config_wait_complete(), and it called that from
    sja1105_dynamic_config_write() too.
```

```
    What is problematic is the refactoring.

    The refactored code from sja1105_dynamic_config_poll_valid() works like
    the previous one, but the problem is that it uses another packed_buf[]
    SPI buffer, and there was code at the end of sja1105_dynamic_config_read()
    which was relying on the read-back packed_buf[]:

            /* Don't dereference possibly NULL pointer - maybe caller
             * only wanted to see whether the entry existed or not.
             */
            if (entry)
                    ops->entry_packing(packed_buf, entry, UNPACK);

    After the change, the packed_buf[] that this code sees is no longer the
    entry read back from hardware, but the original entry that the caller
    passed to the sja1105_dynamic_config_read(), packed into this buffer.

    This difference is the most notable with the SJA1105_SEARCH uses from
    sja1105pqrs_fdb_add() - used for both fdb and mdb. There, we have logic
    added by commit 728db843df88 ("net: dsa: sja1105: ignore the FDB entry
    for unknown multicast when adding a new address") to figure out whether
    the address we're trying to add matches on any existing hardware entry,
    with the exception of the catch-all multicast address.

    That logic was broken, because with sja1105_dynamic_config_read() not
    working properly, it doesn't return us the entry read back from
    hardware, but the entry that we passed to it. And, since for multicast,
    a match will always exist, it will tell us that any mdb entry already
    exists at index=0 L2 Address Lookup table. It is index=0 because the
    caller doesn't know the index - it wants to find it out, and
    sja1105_dynamic_config_read() does:

            if (index < 0) { // SJA1105_SEARCH
                    /* Avoid copying a signed negative number to an u64 */
                    cmd.index = 0; // <- this
                    cmd.search = true;
            } else {
                    cmd.index = index;
                    cmd.search = false;
            }

    So, to the caller of sja1105_dynamic_config_read(), the returned info
    looks entirely legit, and it will add all mdb entries to FDB index 0.
    There, they will always overwrite each other (not to mention,
    potentially they can also overwrite a pre-existing bridge fdb entry),
    and the user-visible impact will be that only the last mdb entry will be
    forwarded as it should. The others won't (will be flooded or dropped,
    depending on the egress flood settings).

    Fixing is a bit more complicated, and involves either passing the same
    packed_buf[] to sja1105_dynamic_config_wait_complete(), or moving all
    the extra processing on the packed_buf[] to
    sja1105_dynamic_config_wait_complete(). I've opted for the latter,
    because it makes sja1105_dynamic_config_wait_complete() a bit more
    self-contained.

    Fixes: df405910ab9f ("net: dsa: sja1105: wait for dynamic config command completion on
 writes too")
    Reported-by: Yanan Yang <yanan.yang@nxp.com>
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit ca8e7b32bf2658f534e59eebeab49e85c88ad33f
```

```
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Sep 8 16:33:49 2023 +0300

    net: dsa: sja1105: propagate exact error code from sja1105_dynamic_config_poll_valid()

    [ Upstream commit c956798062b5a308db96e75157747291197f0378 ]

    Currently, sja1105_dynamic_config_wait_complete() returns either 0 or
    -ETIMEDOUT, because it just looks at the read_poll_timeout() return code.

    There will be future changes which move some more checks to
    sja1105_dynamic_config_poll_valid(). It is important that we propagate
    their exact return code (-ENOENT, -EINVAL), because callers of
    sja1105_dynamic_config_read() depend on them.

    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Stable-dep-of: 7cef293b9a63 ("net: dsa: sja1105: fix multicast forwarding working only
for last added mdb entry")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 314b9eef8408a3b6c32f4de553d65582e28b0d4b
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Sep 8 16:33:48 2023 +0300

    net: dsa: sja1105: hide all multicast addresses from "bridge fdb show"

    [ Upstream commit 02c652f5465011126152bbd93b6a582a1d0c32f1 ]

    Commit 4d9423549501 ("net: dsa: sja1105: offload bridge port flags to
    device") has partially hidden some multicast entries from showing up in
    the "bridge fdb show" output, but it wasn't enough. Addresses which are
    added through "bridge mdb add" still show up. Hide them all.

    Fixes: 291d1e72b756 ("net: dsa: sja1105: Add support for FDB and MDB management")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 93546f928ab6768ec1d77f88e66af90322338f92
Author: Ciprian Regus <ciprian.regus@analog.com>
Date:   Fri Sep 8 15:58:08 2023 +0300

    net:ethernet:adi:adin1110: Fix forwarding offload

    [ Upstream commit 32530dba1bd48da4437d18d9a8dbc9d2826938a6 ]

    Currently, when a new fdb entry is added (with both ports of the
    ADIN2111 bridged), the driver configures the MAC filters for the wrong
    port, which results in the forwarding being done by the host, and not
    actually hardware offloaded.

    The ADIN2111 offloads the forwarding by setting filters on the
    destination MAC address of incoming frames. Based on these, they may be
    routed to the other port. Thus, if a frame has to be forwarded from port
    1 to port 2, the required configuration for the ADDR_FILT_UPRn register
    should set the APPLY2PORT1 bit (instead of APPLY2PORT2, as it's
    currently the case).

    Fixes: bc93e19d088b ("net: ethernet: adi: Add ADIN1110 support")
    Signed-off-by: Ciprian Regus <ciprian.regus@analog.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 499d86b62723207d2cef19e9c6796c1d0970c575
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Fri Aug 4 17:35:31 2023 +0800

    net: ethernet: adi: adin1110: use eth_broadcast_addr() to assign broadcast address

    [ Upstream commit 54024dbec95585243391caeb9f04a2620e630765 ]

    Use eth_broadcast_addr() to assign broadcast address instead
    of memset().

    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Stable-dep-of: 32530dba1bd4 ("net:ethernet:adi:adin1110: Fix forwarding offload")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ed7a0ba7e840dc5d54cdbd8466be27e6aedce1e5
Author: Ziyang Xuan <william.xuanziyang@huawei.com>
Date:   Fri Sep 8 18:17:52 2023 +0800

    hsr: Fix uninit-value access in fill_frame_info()

    [ Upstream commit 484b4833c604c0adcf19eac1ca14b60b757355b5 ]

    Syzbot reports the following uninit-value access problem.

    =====================================================
    BUG: KMSAN: uninit-value in fill_frame_info net/hsr/hsr_forward.c:601 [inline]
    BUG: KMSAN: uninit-value in hsr_forward_skb+0x9bd/0x30f0 net/hsr/hsr_forward.c:616
     fill_frame_info net/hsr/hsr_forward.c:601 [inline]
     hsr_forward_skb+0x9bd/0x30f0 net/hsr/hsr_forward.c:616
     hsr_dev_xmit+0x192/0x330 net/hsr/hsr_device.c:223
     __netdev_start_xmit include/linux/netdevice.h:4889 [inline]
     netdev_start_xmit include/linux/netdevice.h:4903 [inline]
     xmit_one net/core/dev.c:3544 [inline]
     dev_hard_start_xmit+0x247/0xa10 net/core/dev.c:3560
     __dev_queue_xmit+0x34d0/0x52a0 net/core/dev.c:4340
     dev_queue_xmit include/linux/netdevice.h:3082 [inline]
     packet_xmit+0x9c/0x6b0 net/packet/af_packet.c:276
     packet_snd net/packet/af_packet.c:3087 [inline]
     packet_sendmsg+0x8b1d/0x9f30 net/packet/af_packet.c:3119
     sock_sendmsg_nosec net/socket.c:730 [inline]
     sock_sendmsg net/socket.c:753 [inline]
     __sys_sendto+0x781/0xa30 net/socket.c:2176
     __do_sys_sendto net/socket.c:2188 [inline]
     __se_sys_sendto net/socket.c:2184 [inline]
     __ia32_sys_sendto+0x11f/0x1c0 net/socket.c:2184
     do_syscall_32_irqs_on arch/x86/entry/common.c:112 [inline]
     __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178
     do_fast_syscall_32+0x37/0x80 arch/x86/entry/common.c:203
     do_SYSENTER_32+0x1f/0x30 arch/x86/entry/common.c:246
     entry_SYSENTER_compat_after_hwframe+0x70/0x82

    Uninit was created at:
     slab_post_alloc_hook+0x12f/0xb70 mm/slab.h:767
     slab_alloc_node mm/slub.c:3478 [inline]
     kmem_cache_alloc_node+0x577/0xa80 mm/slub.c:3523
     kmalloc_reserve+0x148/0x470 net/core/skbuff.c:559
     __alloc_skb+0x318/0x740 net/core/skbuff.c:644
     alloc_skb include/linux/skbuff.h:1286 [inline]
     alloc_skb_with_frags+0xc8/0xbd0 net/core/skbuff.c:6299
     sock_alloc_send_pskb+0xa80/0xbf0 net/core/sock.c:2794
     packet_alloc_skb net/packet/af_packet.c:2936 [inline]
     packet_snd net/packet/af_packet.c:3030 [inline]
```

```
      packet_sendmsg+0x70e8/0x9f30 net/packet/af_packet.c:3119
      sock_sendmsg_nosec net/socket.c:730 [inline]
      sock_sendmsg net/socket.c:753 [inline]
      __sys_sendto+0x781/0xa30 net/socket.c:2176
      __do_sys_sendto net/socket.c:2188 [inline]
      __se_sys_sendto net/socket.c:2184 [inline]
      __ia32_sys_sendto+0x11f/0x1c0 net/socket.c:2184
      do_syscall_32_irqs_on arch/x86/entry/common.c:112 [inline]
      __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178
      do_fast_syscall_32+0x37/0x80 arch/x86/entry/common.c:203
      do_SYSENTER_32+0x1f/0x30 arch/x86/entry/common.c:246
      entry_SYSENTER_compat_after_hwframe+0x70/0x82

    It is because VLAN not yet supported in hsr driver. Return error
    when protocol is ETH_P_8021Q in fill_frame_info() now to fix it.

    Fixes: 451d8123f897 ("net: prp: add packet handling support")
    Reported-by: syzbot+bf7e6250c7ce248f3ec9@syzkaller.appspotmail.com
    Closes: https://syzkaller.appspot.com/bug?extid=bf7e6250c7ce248f3ec9
    Signed-off-by: Ziyang Xuan <william.xuanziyang@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fe0195fe48f85182bc7e7eabcad925bd3cbc10f5
Author: Hangyu Hua <hbh25y@gmail.com>
Date:   Fri Sep 8 14:19:50 2023 +0800

    net: ethernet: mtk_eth_soc: fix possible NULL pointer dereference in
mtk_hwlro_get_fdir_all()

    [ Upstream commit e4c79810755f66c9a933ca810da2724133b1165a ]

    rule_locs is allocated in ethtool_get_rxnfc and the size is determined by
    rule_cnt from user space. So rule_cnt needs to be check before using
    rule_locs to avoid NULL pointer dereference.

    Fixes: 7aab747e5563 ("net: ethernet: mediatek: add ethtool functions to configure RX
flows of HW LRO")
    Signed-off-by: Hangyu Hua <hbh25y@gmail.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 625b70d31dd4df4b96b3ddcbe251debb33bd67f5
Author: Hangyu Hua <hbh25y@gmail.com>
Date:   Fri Sep 8 14:19:49 2023 +0800

    net: ethernet: mvpp2_main: fix possible OOB write in mvpp2_ethtool_get_rxnfc()

    [ Upstream commit 51fe0a470543f345e3c62b6798929de3ddcedc1d ]

    rules is allocated in ethtool_get_rxnfc and the size is determined by
    rule_cnt from user space. So rule_cnt needs to be check before using
    rules to avoid OOB writing or NULL pointer dereference.

    Fixes: 90b509b39ac9 ("net: mvpp2: cls: Add Classification offload support")
    Signed-off-by: Hangyu Hua <hbh25y@gmail.com>
    Reviewed-by: Marcin Wojtas <mw@semihalf.com>
    Reviewed-by: Russell King (Oracle) <rmk+kernel@armlinux.org.uk>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit adfbdc712f75efcd6c508a359bfc15e50eea000e
Author: Vincent Whitchurch <vincent.whitchurch@axis.com>
Date:   Thu Sep 7 12:46:31 2023 +0200
```

```
    net: stmmac: fix handling of zero coalescing tx-usecs

    [ Upstream commit fa60b8163816f194786f3ee334c9a458da7699c6 ]

    Setting ethtool -C eth0 tx-usecs 0 is supposed to disable the use of the
    coalescing timer but currently it gets programmed with zero delay
    instead.

    Disable the use of the coalescing timer if tx-usecs is zero by
    preventing it from being restarted.  Note that to keep things simple we
    don't start/stop the timer when the coalescing settings are changed, but
    just let that happen on the next transmit or timer expiry.

    Fixes: 8fce33317023 ("net: stmmac: Rework coalesce timer and fix multi-queue races")
    Signed-off-by: Vincent Whitchurch <vincent.whitchurch@axis.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b717463610a27fc0b58484cfead7a623d5913e61
Author: Guangguan Wang <guangguan.wang@linux.alibaba.com>
Date:   Fri Sep 8 11:31:43 2023 +0800

    net/smc: use smc_lgr_list.lock to protect smc_lgr_list.list iterate in smcr_port_add

    [ Upstream commit f5146e3ef0a9eea405874b36178c19a4863b8989 ]

    While doing smcr_port_add, there maybe linkgroup add into or delete
    from smc_lgr_list.list at the same time, which may result kernel crash.
    So, use smc_lgr_list.lock to protect smc_lgr_list.list iterate in
    smcr_port_add.

    The crash calltrace show below:
    BUG: kernel NULL pointer dereference, address: 0000000000000000
    PGD 0 P4D 0
    Oops: 0000 [#1] SMP NOPTI
    CPU: 0 PID: 559726 Comm: kworker/0:92 Kdump: loaded Tainted: G
    Hardware name: Alibaba Cloud Alibaba Cloud ECS, BIOS 449e491 04/01/2014
    Workqueue: events smc_ib_port_event_work [smc]
    RIP: 0010:smcr_port_add+0xa6/0xf0 [smc]
    RSP: 0000:ffffa5a2c8f67de0 EFLAGS: 00010297
    RAX: 0000000000000001 RBX: ffff9935e0650000 RCX: 0000000000000000
    RDX: 0000000000000010 RSI: ffff9935e0654290 RDI: ffff9935c8560000
    RBP: 0000000000000000 R08: 0000000000000000 R09: ffff9934c0401918
    R10: 0000000000000000 R11: ffffffffb4a5c278 R12: ffff99364029aae4
    R13: ffff99364029aa00 R14: 00000000ffffffed R15: ffff99364029ab08
    FS:  0000000000000000(0000) GS:ffff994380600000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 0000000000000000 CR3: 0000000f06a10003 CR4: 0000000002770ef0
    PKRU: 55555554
    Call Trace:
     smc_ib_port_event_work+0x18f/0x380 [smc]
     process_one_work+0x19b/0x340
     worker_thread+0x30/0x370
     ? process_one_work+0x340/0x340
     kthread+0x114/0x130
     ? __kthread_cancel_work+0x50/0x50
     ret_from_fork+0x1f/0x30

    Fixes: 1f90a05d9ff9 ("net/smc: add smcr_port_add() and smcr_link_up() processing")
    Signed-off-by: Guangguan Wang <guangguan.wang@linux.alibaba.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 67920084b37cb45fd1b26666c4b1f053b7456db3
```

```
Author: Ratheesh Kannoth <rkannoth@marvell.com>
Date:   Fri Sep 8 08:23:09 2023 +0530

    octeontx2-pf: Fix page pool cache index corruption.

    [ Upstream commit 88e69af061f2e061a68751ef9cad47a674527a1b ]

    The access to page pool `cache' array and the `count' variable
    is not locked. Page pool cache access is fine as long as there
    is only one consumer per pool.

    octeontx2 driver fills in rx buffers from page pool in NAPI context.
    If system is stressed and could not allocate buffers, refiiling work
    will be delegated to a delayed workqueue. This means that there are
    two cosumers to the page pool cache.

    Either workqueue or IRQ/NAPI can be run on other CPU. This will lead
    to lock less access, hence corruption of cache pool indexes.

    To fix this issue, NAPI is rescheduled from workqueue context to refill
    rx buffers.

    Fixes: b2e3406a38f0 ("octeontx2-pf: Add support for page pool")
    Signed-off-by: Ratheesh Kannoth <rkannoth@marvell.com>
    Reported-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Reviewed-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a26ba60413b2c8f95daf0ee0152cf82abd7bfbe4
Author: Jinjie Ruan <ruanjinjie@huawei.com>
Date:   Thu Sep 7 22:03:58 2023 +0800

    net: microchip: vcap api: Fix possible memory leak for vcap_dup_rule()

    [ Upstream commit 281f65d29d6da1a9b6907fb0b145aaf34f4e4822 ]

    Inject fault When select CONFIG_VCAP_KUNIT_TEST, the below memory leak
    occurs. If kzalloc() for duprule succeeds, but the following
    kmemdup() fails, the duprule, ckf and caf memory will be leaked. So kfree
    them in the error path.

    unreferenced object 0xffff122744c50600 (size 192):
      comm "kunit_try_catch", pid 346, jiffies 4294896122 (age 911.812s)
      hex dump (first 32 bytes):
        10 27 00 00 04 00 00 00 1e 00 00 00 2c 01 00 00  .'..........,...
        00 00 00 00 00 00 00 00 18 06 c5 44 27 12 ff ff  ...........D'...
      backtrace:
        [<00000000394b0db8>] __kmem_cache_alloc_node+0x274/0x2f8
        [<0000000001bedc67>] kmalloc_trace+0x38/0x88
        [<00000000b0612f98>] vcap_dup_rule+0x50/0x460
        [<000000005d2d3aca>] vcap_add_rule+0x8cc/0x1038
        [<00000000eef9d0f8>] test_vcap_xn_rule_creator.constprop.0.isra.0+0x238/0x494
        [<00000000cbda607b>] vcap_api_rule_remove_in_front_test+0x1ac/0x698
        [<00000000c8766299>] kunit_try_run_case+0xe0/0x20c
        [<00000000c4fe9186>] kunit_generic_run_threadfn_adapter+0x50/0x94
        [<00000000f6864acf>] kthread+0x2e8/0x374
        [<0000000022e639b3>] ret_from_fork+0x10/0x20

    Fixes: 814e7693207f ("net: microchip: vcap api: Add a storage state to a VCAP rule")
    Signed-off-by: Jinjie Ruan <ruanjinjie@huawei.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 46f67054a8ec22cc7aa161e488c1928ed6b59512
Author: Naveen N Rao <naveen@kernel.org>
Date:   Wed Jun 14 14:40:46 2023 +0530

    selftests/ftrace: Fix dependencies for some of the synthetic event tests

    [ Upstream commit 145036f88d693d7ef3aa8537a4b1aa22f8764647 ]

    Commit b81a3a100cca1b ("tracing/histogram: Add simple tests for
    stacktrace usage of synthetic events") changed the output text in
    tracefs README, but missed updating some of the dependencies specified
    in selftests. This causes some of the tests to exit as unsupported.

    Fix this by changing the grep pattern. Since we want these tests to work
    on older kernels, match only against the common last part of the
    pattern.

    Link: https://lore.kernel.org/linux-trace-kernel/20230614091046.2178539-1-
naveen@kernel.org

    Cc: <linux-kselftest@vger.kernel.org>
    Cc: Masami Hiramatsu <mhiramat@kernel.org>
    Cc: Shuah Khan <shuah@kernel.org>
    Fixes: b81a3a100cca ("tracing/histogram: Add simple tests for stacktrace usage of
synthetic events")
    Signed-off-by: Naveen N Rao <naveen@kernel.org>
    Signed-off-by: Steven Rostedt (Google) <rostedt@goodmis.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 45c9367015ef79a2f3671323a8046ef9d300ae13
Author: Björn Töpel <bjorn@rivosinc.com>
Date:   Tue Aug 22 15:58:37 2023 +0200

    selftests: Keep symlinks, when possible

    [ Upstream commit 3f3f384139ed147c71e1d770accf610133d5309b ]

    When kselftest is built/installed with the 'gen_tar' target, rsync is
    used for the installation step to copy files. Extra care is needed for
    tests that have symlinks. Commit ae108c48b5d2 ("selftests: net: Fix
    cross-tree inclusion of scripts") added '-L' (transform symlink into
    referent file/dir) to rsync, to fix dangling links. However, that
    broke some tests where the symlink (being a symlink) is part of the
    test (e.g. exec:execveat).

    Use rsync's '--copy-unsafe-links' that does right thing.

    Fixes: ae108c48b5d2 ("selftests: net: Fix cross-tree inclusion of scripts")
    Signed-off-by: Björn Töpel <bjorn@rivosinc.com>
    Reviewed-by: Benjamin Poirier <bpoirier@nvidia.com>
    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9225ced01220b3db386b7dced28047048027cc4c
Author: Björn Töpel <bjorn@rivosinc.com>
Date:   Wed Jul 5 13:53:17 2023 +0200

    kselftest/runner.sh: Propagate SIGTERM to runner child

    [ Upstream commit 9616cb34b08ec86642b162eae75c5a7ca8debe3c ]

    Timeouts in kselftest are done using the "timeout" command with the
    "--foreground" option. Without the "foreground" option, it is not
    possible for a user to cancel the runner using SIGINT, because the
    signal is not propagated to timeout which is running in a different
```

```
    process group. The "forground" options places the timeout in the same
    process group as its parent, but only sends the SIGTERM (on timeout)
    signal to the forked process. Unfortunately, this does not play nice
    with all kselftests, e.g. "net:fcnal-test.sh", where the child
    processes will linger because timeout does not send SIGTERM to the
    group.

    Some users have noted these hangs [1].

    Fix this by nesting the timeout with an additional timeout without the
    foreground option.

    Link: https://lore.kernel.org/all/7650b2eb-0aee-a2b0-2e64-c9bc63210f67@alu.unizg.hr/ #
[1]
    Fixes: 651e0d881461 ("kselftest/runner: allow to properly deliver signals to tests")
    Signed-off-by: Björn Töpel <bjorn@rivosinc.com>
    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 42652af5360d30b43b06057c193739e7dfb18f42
Author: Liu Jian <liujian56@huawei.com>
Date:   Thu Sep 7 10:57:09 2023 +0800

    net: ipv4: fix one memleak in __inet_del_ifa()

    [ Upstream commit ac28b1ec6135649b5d78b028e47264cb3ebca5ea ]

    I got the below warning when do fuzzing test:
    unregister_netdevice: waiting for bond0 to become free. Usage count = 2

    It can be repoduced via:

    ip link add bond0 type bond
    sysctl -w net.ipv4.conf.bond0.promote_secondaries=1
    ip addr add 4.117.174.103/0 scope 0x40 dev bond0
    ip addr add 192.168.100.111/255.255.255.254 scope 0 dev bond0
    ip addr add 0.0.0.4/0 scope 0x40 secondary dev bond0
    ip addr del 4.117.174.103/0 scope 0x40 dev bond0
    ip link delete bond0 type bond

    In this reproduction test case, an incorrect 'last_prim' is found in
    __inet_del_ifa(), as a result, the secondary address(0.0.0.4/0 scope 0x40)
    is lost. The memory of the secondary address is leaked and the reference of
    in_device and net_device is leaked.

    Fix this problem:
    Look for 'last_prim' starting at location of the deleted IP and inserting
    the promoted IP into the location of 'last_prim'.

    Fixes: 0ff60a45678e ("[IPV4]: Fix secondary IP addresses after promotion")
    Signed-off-by: Liu Jian <liujian56@huawei.com>
    Signed-off-by: Julian Anastasov <ja@ssi.bg>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a9d39abd011f67c0abcbb8406941b2f0250386f5
Author: Jinjie Ruan <ruanjinjie@huawei.com>
Date:   Sun Sep 3 15:10:25 2023 +0800

    kunit: Fix wild-memory-access bug in kunit_free_suite_set()

    [ Upstream commit 2810c1e99867a811e631dd24e63e6c1e3b78a59d ]

    Inject fault while probing kunit-example-test.ko, if kstrdup()
    fails in mod_sysfs_setup() in load_module(), the mod->state will
```

```
    switch from MODULE_STATE_COMING to MODULE_STATE_GOING instead of
    from MODULE_STATE_LIVE to MODULE_STATE_GOING, so only
    kunit_module_exit() will be called without kunit_module_init(), and
    the mod->kunit_suites is no set correctly and the free in
    kunit_free_suite_set() will cause below wild-memory-access bug.

    The mod->state state machine when load_module() succeeds:

    MODULE_STATE_UNFORMED ---> MODULE_STATE_COMING ---> MODULE_STATE_LIVE
            ^                                                   |
            |                                                   | delete_module
            +---------------- MODULE_STATE_GOING <---------+

    The mod->state state machine when load_module() fails at
    mod_sysfs_setup():

    MODULE_STATE_UNFORMED ---> MODULE_STATE_COMING ---> MODULE_STATE_GOING
            ^                                                   |
            |                                                   |
            +---------------------------------------------------+

    Call kunit_module_init() at MODULE_STATE_COMING state to fix the issue
    because MODULE_STATE_LIVE is transformed from it.

     Unable to handle kernel paging request at virtual address ffffff341e942a88
     KASAN: maybe wild-memory-access in range [0x0003f9a0f4a15440-0x0003f9a0f4a15447]
     Mem abort info:
       ESR = 0x0000000096000004
       EC = 0x25: DABT (current EL), IL = 32 bits
       SET = 0, FnV = 0
       EA = 0, S1PTW = 0
       FSC = 0x04: level 0 translation fault
     Data abort info:
       ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000
       CM = 0, WnR = 0, TnD = 0, TagAccess = 0
       GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0
     swapper pgtable: 4k pages, 48-bit VAs, pgdp=0000000441ea000
     [ffffff341e942a88] pgd=0000000000000000, p4d=0000000000000000
     Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP
     Modules linked in: kunit_example_test(-) cfg80211 rfkill 8021q garp mrp stp llc ipv6
 [last unloaded: kunit_example_test]
     CPU: 3 PID: 2035 Comm: modprobe Tainted: G        W        N 6.5.0-next-20230828+ #136
     Hardware name: linux,dummy-virt (DT)
     pstate: a0000005 (NzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--)
     pc : kfree+0x2c/0x70
     lr : kunit_free_suite_set+0xcc/0x13c
     sp : ffff8000829b75b0
     x29: ffff8000829b75b0 x28: ffff8000829b7b90 x27: 0000000000000000
     x26: dfff800000000000 x25: ffffcd07c82a7280 x24: ffffcd07a50ab300
     x23: ffffcd07a50ab2e8 x22: 1ffff00010536ec0 x21: dfff800000000000
     x20: ffffcd07a50ab2f0 x19: ffffcd07a50ab2f0 x18: 0000000000000000
     x17: 0000000000000000 x16: 0000000000000000 x15: ffffcd07c24b6764
     x14: ffffcd07c24b63c0 x13: ffffcd07c4cebb94 x12: ffff700010536ec7
     x11: 1ffff00010536ec6 x10: ffff700010536ec6 x9 : dfff800000000000
     x8 : 00008fffefac913a x7 : 0000000041b58ab3 x6 : 0000000000000000
     x5 : 1ffff00010536ec5 x4 : ffff8000829b7628 x3 : dfff800000000000
     x2 : ffffff341e942a80 x1 : ffffcd07a50aa000 x0 : fffffc0000000000
     Call trace:
      kfree+0x2c/0x70
      kunit_free_suite_set+0xcc/0x13c
      kunit_module_notify+0xd8/0x360
      blocking_notifier_call_chain+0xc4/0x128
      load_module+0x382c/0x44a4
      init_module_from_file+0xd4/0x128
      idempotent_init_module+0x2c8/0x524
```

```
        __arm64_sys_finit_module+0xac/0x100
        invoke_syscall+0x6c/0x258
        el0_svc_common.constprop.0+0x160/0x22c
        do_el0_svc+0x44/0x5c
        el0_svc+0x38/0x78
        el0t_64_sync_handler+0x13c/0x158
        el0t_64_sync+0x190/0x194
       Code: aa0003e1 b25657e0 d34cfc42 8b021802 (f9400440)
       ---[ end trace 0000000000000000 ]---
       Kernel panic - not syncing: Oops: Fatal exception
       SMP: stopping secondary CPUs
       Kernel Offset: 0x4d0742200000 from 0xffff800080000000
       PHYS_OFFSET: 0xffffee43c0000000
       CPU features: 0x88000203,3c020000,1000421b
       Memory Limit: none
       Rebooting in 1 seconds..

    Fixes: 3d6e44623841 ("kunit: unify module and builtin suite definitions")
    Signed-off-by: Jinjie Ruan <ruanjinjie@huawei.com>
    Reviewed-by: Rae Moar <rmoar@google.com>
    Reviewed-by: David Gow <davidgow@google.com>
    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ff4f59a022150f4a1939d9fa89aea7b767255795
Author: Helge Deller <deller@gmx.de>
Date:   Wed Aug 30 07:56:04 2023 +0200

    parisc: sba_iommu: Fix build warning if procfs if disabled

    [ Upstream commit 6428bc7bd3f35e43c8cb7359cb89d83248d339d2 ]

    Clean up the code, e.g. make proc_mckinley_root static, drop the now
    empty mckinley header file and remove some unneeded ifdefs around procfs
    functions.

    Signed-off-by: Helge Deller <deller@gmx.de>
    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/oe-kbuild-all/202308300800.Jod4sHzM-lkp@intel.com/
    Fixes: 77e0ddf097d6 ("parisc: ccio-dma: Create private runway procfs root entry")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2bf2d2ac9e67184dc99275875a6452ca6e3027ff
Author: Biju Das <biju.das.jz@bp.renesas.com>
Date:   Wed Aug 16 14:55:49 2023 +0100

    regulator: raa215300: Fix resource leak in case of error

    [ Upstream commit e21ac64e669e960688e79bf5babeed63132dac8a ]

    The clk_register_clkdev() allocates memory by calling vclkdev_alloc() and
    this memory is not freed in the error path. Similarly, resources allocated
    by clk_register_fixed_rate() are not freed in the error path.

    Fix these issues by using devm_clk_hw_register_fixed_rate() and
    devm_clk_hw_register_clkdev().

    After this, the static variable clk is not needed. Replace it with
    local variable hw in probe() and drop calling clk_unregister_fixed_rate()
    from raa215300_rtc_unregister_device().

    Fixes: 7bce16630837 ("regulator: Add Renesas PMIC RAA215300 driver")
    Cc: stable@kernel.org
    Signed-off-by: Biju Das <biju.das.jz@bp.renesas.com>
    Link: https://lore.kernel.org/r/20230816135550.146657-2-biju.das.jz@bp.renesas.com
```

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

commit 749491b59097138040971348e31dc924c82d46cf
Author: Biju Das <biju.das.jz@bp.renesas.com>
Date:   Thu Jun 29 11:42:00 2023 +0100

    regulator: raa215300: Change the scope of the variables {clkin_name, xin_name}

    [ Upstream commit 42a95739c5bc4d7a6e93a43117e9283598ba2287 ]

    Change the scope of the variables {clkin_name, xin_name} from global->local
    to fix the below warning.

    drivers/regulator/raa215300.c:42:12: sparse: sparse: symbol 'xin_name' was
    not declared. Should it be static?

    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/oe-kbuild-all/202306250552.Fan9WTiN-lkp@intel.com/
    Signed-off-by: Biju Das <biju.das.jz@bp.renesas.com>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Link: https://lore.kernel.org/r/20230629104200.102663-1-biju.das.jz@bp.renesas.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Stable-dep-of: e21ac64e669e ("regulator: raa215300: Fix resource leak in case of error")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fdb3a30cb17ab0e57c3ea47ca79c4c4e904a2e0b
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Aug 1 13:13:58 2023 +0200

    bpf: fix bpf_probe_read_kernel prototype mismatch

    [ Upstream commit 6a5a148aaf14747570cc634f9cdfcb0393f5617f ]

    bpf_probe_read_kernel() has a __weak definition in core.c and another
    definition with an incompatible prototype in kernel/trace/bpf_trace.c,
    when CONFIG_BPF_EVENTS is enabled.

    Since the two are incompatible, there cannot be a shared declaration in
    a header file, but the lack of a prototype causes a W=1 warning:

    kernel/bpf/core.c:1638:12: error: no previous prototype for 'bpf_probe_read_kernel' [-
Werror=missing-prototypes]

    On 32-bit architectures, the local prototype

    u64 __weak bpf_probe_read_kernel(void *dst, u32 size, const void *unsafe_ptr)

    passes arguments in other registers as the one in bpf_trace.c

    BPF_CALL_3(bpf_probe_read_kernel, void *, dst, u32, size,
               const void *, unsafe_ptr)

    which uses 64-bit arguments in pairs of registers.

    As both versions of the function are fairly simple and only really
    differ in one line, just move them into a header file as an inline
    function that does not add any overhead for the bpf_trace.c callers
    and actually avoids a function call for the other one.

    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/all/ac25cb0f-b804-1649-3afb-1dc6138c2716@iogearbox.net/
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Yonghong Song <yonghong.song@linux.dev>
    Link: https://lore.kernel.org/r/20230801111449.185301-1-arnd@kernel.org
```

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

commit 5479e06e07bdc0d77ae9646e8830978fb48aaf6a
Author: Hamza Mahfooz <hamza.mahfooz@amd.com>
Date:   Tue Aug 15 09:13:37 2023 -0400

    drm/amdgpu: register a dirty framebuffer callback for fbcon

    commit 0a611560f53bfd489e33f4a718c915f1a6123d03 upstream.

    fbcon requires that we implement &drm_framebuffer_funcs.dirty.
    Otherwise, the framebuffer might take a while to flush (which would
    manifest as noticeable lag). However, we can't enable this callback for
    non-fbcon cases since it may cause too many atomic commits to be made at
    once. So, implement amdgpu_dirtyfb() and only enable it for fbcon
    framebuffers (we can use the "struct drm_file file" parameter in the
    callback to check for this since it is only NULL when called by fbcon,
    at least in the mainline kernel) on devices that support atomic KMS.

    Cc: Aurabindo Pillai <aurabindo.pillai@amd.com>
    Cc: Mario Limonciello <mario.limonciello@amd.com>
    Cc: stable@vger.kernel.org # 6.1+
    Link: https://gitlab.freedesktop.org/drm/amd/-/issues/2519
    Reviewed-by: Mario Limonciello <mario.limonciello@amd.com>
    Signed-off-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 399b73d6b7720a9eae68a333193b53ed4f432fe5
Author: Jay Cornwall <jay.cornwall@amd.com>
Date:   Fri Aug 25 12:18:41 2023 -0400

    drm/amdkfd: Add missing gfx11 MQD manager callbacks

    commit e9dca969b2426702a73719ab9207e43c6d80b581 upstream.

    mqd_stride function was introduced in commit 2f77b9a242a2
    ("drm/amdkfd: Update MQD management on multi XCC setup")
    but not assigned for gfx11. Fixes a NULL dereference in debugfs.

    Signed-off-by: Jay Cornwall <jay.cornwall@amd.com>
    Signed-off-by: Harish Kasiviswanathan <Harish.Kasiviswanathan@amd.com>
    Acked-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org # 6.5.x
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aeeb30ac4448569af4da967753faf818fb796d67
Author: Gabe Teeger <gabe.teeger@amd.com>
Date:   Mon Aug 14 16:06:18 2023 -0400

    drm/amd/display: Remove wait while locked

    commit 5a3ccb1400339268c5e3dc1fa044a7f6c7f59a02 upstream.

    [Why]
    We wait for mpc idle while in a locked state, leading to potential
    deadlock.

    [What]
    Move the wait_for_idle call to outside of HW lock. This and a
    call to wait_drr_doublebuffer_pending_clear are moved added to a new
    static helper function called wait_for_outstanding_hw_updates, to make
    the interface clearer.
```

```
    Cc: stable@vger.kernel.org
    Fixes: 8f0d304d21b3 ("drm/amd/display: Do not commit pipe when updating DRR")
    Reviewed-by: Jun Lei <jun.lei@amd.com>
    Acked-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Gabe Teeger <gabe.teeger@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9bebf189f65f538411e67fcb475c56856462396e
Author: Wenjing Liu <wenjing.liu@amd.com>
Date:   Tue Aug 15 10:47:52 2023 -0400

    drm/amd/display: always switch off ODM before committing more streams

    commit 49a30c3d1a2258fc93cfe6eea8e4951dabadc824 upstream.

    ODM power optimization is only supported with single stream. When ODM
    power optimization is enabled, we might not have enough free pipes for
    enabling other stream. So when we are committing more than 1 stream we
    should first switch off ODM power optimization to make room for new
    stream and then allocating pipe resource for the new stream.

    Cc: stable@vger.kernel.org
    Fixes: 59de751e3845 ("drm/amd/display: add ODM case when looking for first split pipe")
    Reviewed-by: Dillon Varone <dillon.varone@amd.com>
    Acked-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Wenjing Liu <wenjing.liu@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 55a941a021fbda31c72227331a48ee297b01413b
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Mon Jul 31 02:49:33 2023 -0700

    perf hists browser: Fix the number of entries for 'e' key

    commit f6b8436bede3e80226e8b2100279c4450c73806a upstream.

    The 'e' key is to toggle expand/collapse the selected entry only.  But
    the current code has a bug that it only increases the number of entries
    by 1 in the hierarchy mode so users cannot move under the current entry
    after the key stroke.  This is due to a wrong assumption in the
    hist_entry__set_folding().

    The commit b33f922651011eff ("perf hists browser: Put hist_entry folding
    logic into single function") factored out the code, but actually it
    should be handled separately.  The hist_browser__set_folding() is to
    update fold state for each entry so it needs to traverse all (child)
    entries regardless of the current fold state.  So it increases the
    number of entries by 1.

    But the hist_entry__set_folding() only cares the currently selected
    entry and its all children.  So it should count all unfolded child
    entries.  This code is implemented in hist_browser__toggle_fold()
    already so we can just call it.

    Fixes: b33f922651011eff ("perf hists browser: Put hist_entry folding logic into single
function")
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
```

```
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230731094934.1616495-2-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cd668bcafa4cc9bc8374ae64da6316fc04c73529
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Thu Jul 27 19:24:47 2023 -0700

    perf build: Include generated header files properly

    commit c7e97f215a4ad634b746804679f5937d25f77e29 upstream.

    The flex and bison generate header files from the source.  When user
    specified a build directory with O= option, it'd generate files under
    the directory.  The build command has -I option to specify the header
    include directory.

    But the -I option only affects the files included like <...>.  Let's
    change the flex and bison headers to use it instead of "...".

    Fixes: 80eeb67fe577aa76 ("perf jevents: Program to convert JSON file")
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Andi Kleen <ak@linux.intel.com>
    Cc: Anup Sharma <anupnewsmail@gmail.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230728022447.1323563-2-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 41cb5ac3b4f3d0c419bd9151d93fb92e871c37c6
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Fri Aug 25 08:25:49 2023 -0700

    perf tools: Handle old data in PERF_RECORD_ATTR

    commit 9bf63282ea77a531ea58acb42fb3f40d2d1e4497 upstream.

    The PERF_RECORD_ATTR is used for a pipe mode to describe an event with
    attribute and IDs.  The ID table comes after the attr and it calculate
    size of the table using the total record size and the attr size.

      n_ids = (total_record_size - end_of_the_attr_field) / sizeof(u64)

    This is fine for most use cases, but sometimes it saves the pipe output
    in a file and then process it later.  And it becomes a problem if there
    is a change in attr size between the record and report.

      $ perf record -o- > perf-pipe.data  # old version
      $ perf report -i- < perf-pipe.data  # new version

    For example, if the attr size is 128 and it has 4 IDs, then it would
    save them in 168 byte like below:

       8 byte: perf event header { .type = PERF_RECORD_ATTR, .size = 168 },
     128 byte: perf event attr { .size = 128, ... },
      32 byte: event IDs [] = { 1234, 1235, 1236, 1237 },

    But when report later, it thinks the attr size is 136 then it only read
```

```
    the last 3 entries as ID.

       8 byte: perf event header { .type = PERF_RECORD_ATTR, .size = 168 },
     136 byte: perf event attr { .size = 136, ... },
      24 byte: event IDs [] = { 1235, 1236, 1237 },  // 1234 is missing

    So it should use the recorded version of the attr.  The attr has the
    size field already then it should honor the size when reading data.

    Fixes: 2c46dbb517a10b18 ("perf: Convert perf header attrs into attr events")
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Tom Zanussi <zanussi@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230825152552.112913-1-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9952bbd484f52ae92966954390d9af05ea46ba0e
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Fri Aug 25 09:41:51 2023 -0700

    perf test shell stat_bpf_counters: Fix test on Intel

    commit 68ca249c964f520af7f8763e22f12bd26b57b870 upstream.

    As of now, bpf counters (bperf) don't support event groups.  But the
    default perf stat includes topdown metrics if supported (on recent Intel
    machines) which require groups.  That makes perf stat exiting.

      $ sudo perf stat --bpf-counter true
      bpf managed perf events do not yet support groups.

    Actually the test explicitly uses cycles event only, but it missed to
    pass the option when it checks the availability of the command.

    Fixes: 2c0cb9f56020d2ea ("perf test: Add a shell test for 'perf stat --bpf-counters' new
option")
    Reviewed-by: Song Liu <song@kernel.org>
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230825164152.165610-2-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0753b9d0bfe67f26d33ad4ddb9d4b087d0b0f471
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Thu Jul 27 19:24:46 2023 -0700

    perf build: Update build rule for generated files

    commit 7822a8913f4c51c7d1aff793b525d60c3384fb5b upstream.

    The bison and flex generate C files from the source (.y and .l)
    files.  When O= option is used, they are saved in a separate directory
```

```
    but the default build rule assumes the .C files are in the source
    directory.  So it might read invalid file if there are generated files
    from an old version.  The same is true for the pmu-events files.

    For example, the following command would cause a build failure:

      $ git checkout v6.3
      $ make -C tools/perf  # build in the same directory

      $ git checkout v6.5-rc2
      $ mkdir build  # create a build directory
      $ make -C tools/perf O=build  # build in a different directory but it
                                    # refers files in the source directory

    Let's update the build rule to specify those cases explicitly to depend
    on the files in the output directory.

    Note that it's not a complete fix and it needs the next patch for the
    include path too.

    Fixes: 80eeb67fe577aa76 ("perf jevents: Program to convert JSON file")
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Andi Kleen <ak@linux.intel.com>
    Cc: Anup Sharma <anupnewsmail@gmail.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230728022447.1323563-1-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8a5b35fe024d25f9c2bfbe6df24f9188746fafce
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Mon Jul 31 02:49:32 2023 -0700

    perf hists browser: Fix hierarchy mode header

    commit e2cabf2a44791f01c21f8d5189b946926e34142e upstream.

    The commit ef9ff6017e3c4593 ("perf ui browser: Move the extra title
    lines from the hists browser") introduced ui_browser__gotorc_title() to
    help moving non-title lines easily.  But it missed to update the title
    for the hierarchy mode so it won't print the header line on TUI at all.

      $ perf report --hierarchy

    Fixes: ef9ff6017e3c4593 ("perf ui browser: Move the extra title lines from the hists
browser")
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230731094934.1616495-1-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ca874c5ffad26a70b2dfd5f1fcbeaecfed129520
Author: Maciej W. Rozycki <macro@orcam.me.uk>
```

```
Date:   Tue Jul 18 15:37:18 2023 +0100

    MIPS: Fix CONFIG_CPU_DADDI_WORKAROUNDS `modules_install' regression

    commit a79a404e6c2241ebc528b9ebf4c0832457b498c3 upstream.

    Remove a build-time check for the presence of the GCC `-msym32' option.
    This option has been there since GCC 4.1.0, which is below the minimum
    required as at commit 805b2e1d427a ("kbuild: include Makefile.compiler
    only when compiler is needed"), when an error message:

    arch/mips/Makefile:306: *** CONFIG_CPU_DADDI_WORKAROUNDS unsupported without -msym32.
Stop.

    started to trigger for the `modules_install' target with configurations
    such as `decstation_64_defconfig' that set CONFIG_CPU_DADDI_WORKAROUNDS,
    because said commit has made `cc-option-yn' an undefined function for
    non-build targets.

    Reported-by: Jan-Benedict Glaw <jbglaw@lug-owl.de>
    Signed-off-by: Maciej W. Rozycki <macro@orcam.me.uk>
    Fixes: 805b2e1d427a ("kbuild: include Makefile.compiler only when compiler is needed")
    Cc: stable@vger.kernel.org # v5.13+
    Reviewed-by: Philippe Mathieu-Daudé <philmd@linaro.org>
    Signed-off-by: Thomas Bogendoerfer <tsbogend@alpha.franken.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 20f8c9ae46a5dba4f6317d990becd5cc81f2a53c
Author: Maciej W. Rozycki <macro@orcam.me.uk>
Date:   Tue Jul 18 15:37:23 2023 +0100

    MIPS: Only fiddle with CHECKFLAGS if `need-compiler'

    commit 4fe4a6374c4db9ae2b849b61e84b58685dca565a upstream.

    We have originally guarded fiddling with CHECKFLAGS in our arch Makefile
    by checking for the CONFIG_MIPS variable, not set for targets such as
    `distclean', etc. that neither include `.config' nor use the compiler.

    Starting from commit 805b2e1d427a ("kbuild: include Makefile.compiler
    only when compiler is needed") we have had a generic `need-compiler'
    variable explicitly telling us if the compiler will be used and thus its
    capabilities need to be checked and expressed in the form of compilation
    flags.  If this variable is not set, then `make' functions such as
    `cc-option' are undefined, causing all kinds of weirdness to happen if
    we expect specific results to be returned, most recently:

    cc1: error: '-mloongson-mmi' must be used with '-mhard-float'

    messages with configurations such as `fuloong2e_defconfig' and the
    `modules_install' target, which does include `.config' and yet does not
    use the compiler.

    Replace the check for CONFIG_MIPS with one for `need-compiler' instead,
    so as to prevent the compiler from being ever called for CHECKFLAGS when
    not needed.

    Reported-by: Guillaume Tucker <guillaume.tucker@collabora.com>
    Closes: https://lore.kernel.org/r/85031c0c-d981-031e-8a50-bc4fad2ddcd8@collabora.com/
    Signed-off-by: Maciej W. Rozycki <macro@orcam.me.uk>
    Fixes: 805b2e1d427a ("kbuild: include Makefile.compiler only when compiler is needed")
    Cc: stable@vger.kernel.org # v5.13+
    Reported-by: "kernelci.org bot" <bot@kernelci.org>
    Signed-off-by: Thomas Bogendoerfer <tsbogend@alpha.franken.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 61c97e3ee78b3244e66bd1e4c4e0a10d3b49c2dd
Author: Sean Christopherson <seanjc@google.com>
Date:   Thu Aug 24 19:23:57 2023 -0700

    KVM: SVM: Skip VMSA init in sev_es_init_vmcb() if pointer is NULL

    commit 1952e74da96fb3e48b72a2d0ece78c688a5848c1 upstream.

    Skip initializing the VMSA physical address in the VMCB if the VMSA is
    NULL, which occurs during intrahost migration as KVM initializes the VMCB
    before copying over state from the source to the destination (including
    the VMSA and its physical address).

    In normal builds, __pa() is just math, so the bug isn't fatal, but with
    CONFIG_DEBUG_VIRTUAL=y, the validity of the virtual address is verified
    and passing in NULL will make the kernel unhappy.

    Fixes: 6defa24d3b12 ("KVM: SEV: Init target VMCBs in sev_migrate_from")
    Cc: stable@vger.kernel.org
    Cc: Peter Gonda <pgonda@google.com>
    Reviewed-by: Peter Gonda <pgonda@google.com>
    Reviewed-by: Pankaj Gupta <pankaj.gupta@amd.com>
    Link: https://lore.kernel.org/r/20230825022357.2852133-3-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e9898d822940ac28ef38ebd21e852fba4fb68fd9
Author: Sean Christopherson <seanjc@google.com>
Date:   Tue Aug 8 16:31:32 2023 -0700

    KVM: SVM: Set target pCPU during IRTE update if target vCPU is running

    commit f3cebc75e7425d6949d726bb8e937095b0aef025 upstream.

    Update the target pCPU for IOMMU doorbells when updating IRTE routing if
    KVM is actively running the associated vCPU.  KVM currently only updates
    the pCPU when loading the vCPU (via avic_vcpu_load()), and so doorbell
    events will be delayed until the vCPU goes through a put+load cycle (which
    might very well "never" happen for the lifetime of the VM).

    To avoid inserting a stale pCPU, e.g. due to racing between updating IRTE
    routing and vCPU load/put, get the pCPU information from the vCPU's
    Physical APIC ID table entry (a.k.a. avic_physical_id_cache in KVM) and
    update the IRTE while holding ir_list_lock.  Add comments with --verbose
    enabled to explain exactly what is and isn't protected by ir_list_lock.

    Fixes: 411b44ba80ab ("svm: Implements update_pi_irte hook to setup posted interrupt")
    Reported-by: dengqiao.joey <dengqiao.joey@bytedance.com>
    Cc: stable@vger.kernel.org
    Cc: Alejandro Jimenez <alejandro.j.jimenez@oracle.com>
    Cc: Joao Martins <joao.m.martins@oracle.com>
    Cc: Maxim Levitsky <mlevitsk@redhat.com>
    Cc: Suravee Suthikulpanit <suravee.suthikulpanit@amd.com>
    Tested-by: Alejandro Jimenez <alejandro.j.jimenez@oracle.com>
    Reviewed-by: Joao Martins <joao.m.martins@oracle.com>
    Link: https://lore.kernel.org/r/20230808233132.2499764-3-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e91c07f6cf7060d2acb3aeee31a6baebe3773d3f
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 28 18:15:49 2023 -0700

    KVM: nSVM: Load L1's TSC multiplier based on L1 state, not L2 state
```

```
    commit 0c94e2468491cbf0754f49a5136ab51294a96b69 upstream.

    When emulating nested VM-Exit, load L1's TSC multiplier if L1's desired
    ratio doesn't match the current ratio, not if the ratio L1 is using for
    L2 diverges from the default.  Functionally, the end result is the same
    as KVM will run L2 with L1's multiplier if L2's multiplier is the default,
    i.e. checking that L1's multiplier is loaded is equivalent to checking if
    L2 has a non-default multiplier.

    However, the assertion that TSC scaling is exposed to L1 is flawed, as
    userspace can trigger the WARN at will by writing the MSR and then
    updating guest CPUID to hide the feature (modifying guest CPUID is
    allowed anytime before KVM_RUN).  E.g. hacking KVM's state_test
    selftest to do

                    vcpu_set_msr(vcpu, MSR_AMD64_TSC_RATIO, 0);
                    vcpu_clear_cpuid_feature(vcpu, X86_FEATURE_TSCRATEMSR);

    after restoring state in a new VM+vCPU yields an endless supply of:

      ------------[ cut here ]------------
      WARNING: CPU: 10 PID: 206939 at arch/x86/kvm/svm/nested.c:1105
               nested_svm_vmexit+0x6af/0x720 [kvm_amd]
      Call Trace:
       nested_svm_exit_handled+0x102/0x1f0 [kvm_amd]
       svm_handle_exit+0xb9/0x180 [kvm_amd]
       kvm_arch_vcpu_ioctl_run+0x1eab/0x2570 [kvm]
       kvm_vcpu_ioctl+0x4c9/0x5b0 [kvm]
       ? trace_hardirqs_off+0x4d/0xa0
       __se_sys_ioctl+0x7a/0xc0
       __x64_sys_ioctl+0x21/0x30
       do_syscall_64+0x41/0x90
       entry_SYSCALL_64_after_hwframe+0x63/0xcd

    Unlike the nested VMRUN path, hoisting the svm->tsc_scaling_enabled check
    into the if-statement is wrong as KVM needs to ensure L1's multiplier is
    loaded in the above scenario.   Alternatively, the WARN_ON() could simply
    be deleted, but that would make KVM's behavior even more subtle, e.g. it's
    not immediately obvious why it's safe to write MSR_AMD64_TSC_RATIO when
    checking only tsc_ratio_msr.

    Fixes: 5228eb96a487 ("KVM: x86: nSVM: implement nested TSC scaling")
    Cc: Maxim Levitsky <mlevitsk@redhat.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230729011608.1065019-3-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 02b24270568f65dd607c4a848512dc8055b4491b
  Author: Sean Christopherson <seanjc@google.com>
  Date:   Fri Jul 28 18:15:48 2023 -0700

    KVM: nSVM: Check instead of asserting on nested TSC scaling support

    commit 7cafe9b8e22bb3d77f130c461aedf6868c4aaf58 upstream.

    Check for nested TSC scaling support on nested SVM VMRUN instead of
    asserting that TSC scaling is exposed to L1 if L1's MSR_AMD64_TSC_RATIO
    has diverged from KVM's default.  Userspace can trigger the WARN at will
    by writing the MSR and then updating guest CPUID to hide the feature
    (modifying guest CPUID is allowed anytime before KVM_RUN).  E.g. hacking
    KVM's state_test selftest to do

                    vcpu_set_msr(vcpu, MSR_AMD64_TSC_RATIO, 0);
```

```
                    vcpu_clear_cpuid_feature(vcpu, X86_FEATURE_TSCRATEMSR);
```

after restoring state in a new VM+vCPU yields an endless supply of:

```
  ------------[ cut here ]------------
  WARNING: CPU: 164 PID: 62565 at arch/x86/kvm/svm/nested.c:699
           nested_vmcb02_prepare_control+0x3d6/0x3f0 [kvm_amd]
  Call Trace:
   <TASK>
   enter_svm_guest_mode+0x114/0x560 [kvm_amd]
   nested_svm_vmrun+0x260/0x330 [kvm_amd]
   vmrun_interception+0x29/0x30 [kvm_amd]
   svm_invoke_exit_handler+0x35/0x100 [kvm_amd]
   svm_handle_exit+0xe7/0x180 [kvm_amd]
   kvm_arch_vcpu_ioctl_run+0x1eab/0x2570 [kvm]
   kvm_vcpu_ioctl+0x4c9/0x5b0 [kvm]
   __se_sys_ioctl+0x7a/0xc0
   __x64_sys_ioctl+0x21/0x30
   do_syscall_64+0x41/0x90
   entry_SYSCALL_64_after_hwframe+0x63/0xcd
  RIP: 0033:0x45ca1b
```

Note, the nested #VMEXIT path has the same flaw, but needs a different
fix and will be handled separately.

Fixes: 5228eb96a487 ("KVM: x86: nSVM: implement nested TSC scaling")
Cc: Maxim Levitsky <mlevitsk@redhat.com>
Cc: stable@vger.kernel.org
Link: https://lore.kernel.org/r/20230729011608.1065019-2-seanjc@google.com
Signed-off-by: Sean Christopherson <seanjc@google.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2ee4b180d51b12a45bdd3264629719ef6a572a73
Author: Sean Christopherson <seanjc@google.com>
Date: Thu Aug 24 19:23:56 2023 -0700

KVM: SVM: Get source vCPUs from source VM for SEV-ES intrahost migration

commit f1187ef24eb8f36e8ad8106d22615ceddeea6097 upstream.

Fix a goof where KVM tries to grab source vCPUs from the destination VM
when doing intrahost migration. Grabbing the wrong vCPU not only hoses
the guest, it also crashes the host due to the VMSA pointer being left
NULL.

```
  BUG: unable to handle page fault for address: ffffe38687000000
  #PF: supervisor read access in kernel mode
  #PF: error_code(0x0000) - not-present page
  PGD 0 P4D 0
  Oops: 0000 [#1] SMP NOPTI
  CPU: 39 PID: 17143 Comm: sev_migrate_tes Tainted: GO       6.5.0-smp--fff2e47e6c3b-
next #151
  Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 34.28.0 07/10/2023
  RIP: 0010:__free_pages+0x15/0xd0
  RSP: 0018:ffff923fcf6e3c78 EFLAGS: 00010246
  RAX: 0000000000000000 RBX: ffffe38687000000 RCX: 0000000000000100
  RDX: 0000000000000100 RSI: 0000000000000000 RDI: ffffe38687000000
  RBP: ffff923fcf6e3c88 R08: ffff923fcafb0000 R09: 0000000000000000
  R10: 0000000000000000 R11: ffffffff83619b90 R12: ffff923fa9540000
  R13: 0000000000080007 R14: ffff923f6d35d000 R15: 0000000000000000
  FS:  0000000000000000(0000) GS:ffff929d0d7c0000(0000) knlGS:0000000000000000
  CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
  CR2: ffffe38687000000 CR3: 0000005224c34005 CR4: 0000000000770ee0
  PKRU: 55555554
  Call Trace:
```

```
        <TASK>
        sev_free_vcpu+0xcb/0x110 [kvm_amd]
        svm_vcpu_free+0x75/0xf0 [kvm_amd]
        kvm_arch_vcpu_destroy+0x36/0x140 [kvm]
        kvm_destroy_vcpus+0x67/0x100 [kvm]
        kvm_arch_destroy_vm+0x161/0x1d0 [kvm]
        kvm_put_kvm+0x276/0x560 [kvm]
        kvm_vm_release+0x25/0x30 [kvm]
        __fput+0x106/0x280
        ____fput+0x12/0x20
        task_work_run+0x86/0xb0
        do_exit+0x2e3/0x9c0
        do_group_exit+0xb1/0xc0
        __x64_sys_exit_group+0x1b/0x20
        do_syscall_64+0x41/0x90
        entry_SYSCALL_64_after_hwframe+0x63/0xcd
        </TASK>
       CR2: ffffe38687000000
```

    Fixes: 6defa24d3b12 ("KVM: SEV: Init target VMCBs in sev_migrate_from")
    Cc: stable@vger.kernel.org
    Cc: Peter Gonda <pgonda@google.com>
    Reviewed-by: Peter Gonda <pgonda@google.com>
    Reviewed-by: Pankaj Gupta <pankaj.gupta@amd.com>
    Link: https://lore.kernel.org/r/20230825022357.2852133-2-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 713a3ab70439babbc1e44894179086ff88ae0ae4
Author: Sean Christopherson <seanjc@google.com>
Date: Thu Aug 24 18:36:18 2023 -0700

    KVM: SVM: Don't inject #UD if KVM attempts to skip SEV guest insn

    commit cb49631ad111570f1bad37702c11c2ae07fa2e3c upstream.

    Don't inject a #UD if KVM attempts to "emulate" to skip an instruction
    for an SEV guest, and instead resume the guest and hope that it can make
    forward progress. When commit 04c40f344def ("KVM: SVM: Inject #UD on
    attempted emulation for SEV guest w/o insn buffer") added the completely
    arbitrary #UD behavior, there were no known scenarios where a well-behaved
    guest would induce a VM-Exit that triggered emulation, i.e. it was thought
    that injecting #UD would be helpful.

    However, now that KVM (correctly) attempts to re-inject INT3/INTO, e.g. if
    a #NPF is encountered when attempting to deliver the INT3/INTO, an SEV
    guest can trigger emulation without a buffer, through no fault of its own.
    Resuming the guest and retrying the INT3/INTO is architecturally wrong,
    e.g. the vCPU will incorrectly re-hit code #DBs, but for SEV guests there
    is literally no other option that has a chance of making forward progress.

    Drop the #UD injection for all "skip" emulation, not just those related to
    INT3/INTO, even though that means that the guest will likely end up in an
    infinite loop instead of getting a #UD (the vCPU may also crash, e.g. if
    KVM emulated everything about an instruction except for advancing RIP).
    There's no evidence that suggests that an unexpected #UD is actually
    better than hanging the vCPU, e.g. a soft-hung vCPU can still respond to
    IRQs and NMIs to generate a backtrace.

    Reported-by: Wu Zongyo <wuzongyo@mail.ustc.edu.cn>
    Closes: https://lore.kernel.org/all/8eb933fd-2cf3-d7a9-32fe-
2a1d82eac42a@mail.ustc.edu.cn
    Fixes: 6ef88d6e36c2 ("KVM: SVM: Re-inject INT3/INTO instead of retrying the
instruction")
    Cc: stable@vger.kernel.org

```
    Cc: Tom Lendacky <thomas.lendacky@amd.com>
    Link: https://lore.kernel.org/r/20230825013621.2845700-2-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 133962a91462758200c6f48280fd87d919fa5f24
Author: Sean Christopherson <seanjc@google.com>
Date:   Tue Aug 8 16:31:31 2023 -0700

    KVM: SVM: Take and hold ir_list_lock when updating vCPU's Physical ID entry

    commit 4c08e737f056fec930b416a2bd37ed266d724f95 upstream.

    Hoist the acquisition of ir_list_lock from avic_update_iommu_vcpu_affinity()
    to its two callers, avic_vcpu_load() and avic_vcpu_put(), specifically to
    encapsulate the write to the vCPU's entry in the AVIC Physical ID table.
    This will allow a future fix to pull information from the Physical ID entry
    when updating the IRTE, without potentially consuming stale information,
    i.e. without racing with the vCPU being (un)loaded.

    Add a comment to call out that ir_list_lock does NOT protect against
    multiple writers, specifically that reading the Physical ID entry in
    avic_vcpu_put() outside of the lock is safe.

    To preserve some semblance of independence from ir_list_lock, keep the
    READ_ONCE() in avic_vcpu_load() even though acuiring the spinlock
    effectively ensures the load(s) will be generated after acquiring the
    lock.

    Cc: stable@vger.kernel.org
    Tested-by: Alejandro Jimenez <alejandro.j.jimenez@oracle.com>
    Reviewed-by: Joao Martins <joao.m.martins@oracle.com>
    Link: https://lore.kernel.org/r/20230808233132.2499764-2-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c6b4258ab85d4ab5d4353ff6a590cd32edbaec48
Author: Sean Christopherson <seanjc@google.com>
Date:   Thu Aug 24 18:45:32 2023 -0700

    KVM: VMX: Refresh available regs and IDT vectoring info before NMI handling

    commit 50011c2a245792993f2756e5b5b571512bfa409e upstream.

    Reset the mask of available "registers" and refresh the IDT vectoring
    info snapshot in vmx_vcpu_enter_exit(), before KVM potentially handles a
    an NMI VM-Exit.  One of the "registers" that KVM VMX lazily loads is the
    vmcs.VM_EXIT_INTR_INFO field, which is holds the vector+type on "exception
    or NMI" VM-Exits, i.e. is needed to identify NMIs.  Clearing the available
    registers bitmask after handling NMIs results in KVM querying info from
    the last VM-Exit that read vmcs.VM_EXIT_INTR_INFO, and leads to both
    missed NMIs and spurious NMIs in the host.

    Opportunistically grab vmcs.IDT_VECTORING_INFO_FIELD early in the VM-Exit
    path too, e.g. to guard against similar consumption of stale data.  The
    field is read on every "normal" VM-Exit, and there's no point in delaying
    the inevitable.

    Reported-by: Like Xu <like.xu.linux@gmail.com>
    Fixes: 11df586d774f ("KVM: VMX: Handle NMI VM-Exits in noinstr region")
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230825014532.2846714-1-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 240713e695abba407df9c917b16f9818bd42bbc2
Author: Hamza Mahfooz <hamza.mahfooz@amd.com>
Date:   Tue Sep 5 13:27:22 2023 -0400

    drm/amd/display: prevent potential division by zero errors

    commit 07e388aab042774f284a2ad75a70a194517cdad4 upstream.

    There are two places in apply_below_the_range() where it's possible for
    a divide by zero error to occur. So, to fix this make sure the divisor
    is non-zero before attempting the computation in both cases.

    Cc: stable@vger.kernel.org
    Link: https://gitlab.freedesktop.org/drm/amd/-/issues/2637
    Fixes: a463b263032f ("drm/amd/display: Fix frames_to_insert math")
    Fixes: ded6119e825a ("drm/amd/display: Reinstate LFC optimization")
    Reviewed-by: Aurabindo Pillai <aurabindo.pillai@amd.com>
    Signed-off-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f5da0ee56e5f103c91cb3601fa945d52c82dd715
Author: Hamza Mahfooz <hamza.mahfooz@amd.com>
Date:   Thu Aug 31 15:22:35 2023 -0400

    drm/amd/display: limit the v_startup workaround to ASICs older than DCN3.1

    commit 47428f4b638d3b3264a2efa1a567b0bbddbb6107 upstream.

    Since, calling dcn20_adjust_freesync_v_startup() on DCN3.1+ ASICs
    can cause the display to flicker and underflow to occur, we shouldn't
    call it for them. So, ensure that the DCN version is less than
    DCN_VERSION_3_1 before calling dcn20_adjust_freesync_v_startup().

    Cc: stable@vger.kernel.org
    Reviewed-by: Fangzhi Zuo <jerry.zuo@amd.com>
    Signed-off-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit df1092760dc2d6a81487f3aa437fe4cbb3370ccd
Author: Melissa Wen <mwen@igalia.com>
Date:   Thu Aug 31 15:12:28 2023 -0100

    drm/amd/display: enable cursor degamma for DCN3+ DRM legacy gamma

    commit 57a943ebfcdb4a97fbb409640234bdb44bfa1953 upstream.

    For DRM legacy gamma, AMD display manager applies implicit sRGB degamma
    using a pre-defined sRGB transfer function. It works fine for DCN2
    family where degamma ROM and custom curves go to the same color block.
    But, on DCN3+, degamma is split into two blocks: degamma ROM for
    pre-defined TFs and `gamma correction` for user/custom curves and
    degamma ROM settings doesn't apply to cursor plane. To get DRM legacy
    gamma working as expected, enable cursor degamma ROM for implict sRGB
    degamma on HW with this configuration.

    Cc: stable@vger.kernel.org
    Link: https://gitlab.freedesktop.org/drm/amd/-/issues/2803
    Fixes: 96b020e2163f ("drm/amd/display: check attr flag before set cursor degamma on
DCN3+")
    Signed-off-by: Melissa Wen <mwen@igalia.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit cc502fe191074c208cb9fe32ce5bda29c720a7b1
Author: Hamza Mahfooz <hamza.mahfooz@amd.com>
Date:   Thu Aug 31 15:17:14 2023 -0400

    Revert "drm/amd/display: Remove v_startup workaround for dcn3+"

    commit a81de4a22bbe3183b7f0d6f13f592b8f5b5a3c18 upstream.

    This reverts commit 3a31e8b89b7240d9a17ace8a1ed050bdcb560f9e.

    We still need to call dcn20_adjust_freesync_v_startup() for older DCN3+
    ASICs. Otherwise, it can cause DP to HDMI 2.1 PCONs to fail to light up.

    Cc: stable@vger.kernel.org
    Link: https://gitlab.freedesktop.org/drm/amd/-/issues/2809
    Reviewed-by: Fangzhi Zuo <jerry.zuo@amd.com>
    Reviewed-by: Harry Wentland <harry.wentland@amd.com>
    Signed-off-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4a9928c8c489964a0223a84a09b7fc25426ccdc2
Author: William Zhang <william.zhang@broadcom.com>
Date:   Thu Jul 6 11:29:05 2023 -0700

    mtd: rawnand: brcmnand: Fix ECC level field setting for v7.2 controller

    commit 2ec2839a9062db8a592525a3fdabd42dcd9a3a9b upstream.

    v7.2 controller has different ECC level field size and shift in the acc
    control register than its predecessor and successor controller. It needs
    to be set specifically.

    Fixes: decba6d47869 ("mtd: brcmnand: Add v7.2 controller support")
    Signed-off-by: William Zhang <william.zhang@broadcom.com>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230706182909.79151-2-
william.zhang@broadcom.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 15c066155d9f9be3c1fdd1ead7385ff106af08f8
Author: William Zhang <william.zhang@broadcom.com>
Date:   Thu Jul 6 11:29:06 2023 -0700

    mtd: rawnand: brcmnand: Fix potential false time out warning

    commit 9cc0a598b944816f2968baf2631757f22721b996 upstream.

    If system is busy during the command status polling function, the driver
    may not get the chance to poll the status register till the end of time
    out and return the premature status.  Do a final check after time out
    happens to ensure reading the correct status.

    Fixes: 9d2ee0a60b8b ("mtd: nand: brcmnand: Check flash #WP pin status before nand
erase/program")
    Signed-off-by: William Zhang <william.zhang@broadcom.com>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230706182909.79151-3-
william.zhang@broadcom.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 17584a3d9b814c61044795036e50e285946c111a
Author: Linus Walleij <linus.walleij@linaro.org>
Date:   Tue Jul 18 13:56:11 2023 +0200

    mtd: spi-nor: Correct flags for Winbond w25q128

    commit 83e824a4a595132f9bd7ac4f5afff857bfc5991e upstream.

    The Winbond "w25q128" (actual vendor name W25Q128JV) has
    exactly the same flags as the sibling device "w25q128jv".
    The devices both require unlocking to enable write access.

    The actual product naming between devices vs the Linux
    strings in winbond.c:

    0xef4018: "w25q128"   W25Q128JV-IN/IQ/JQ
    0xef7018: "w25q128jv" W25Q128JV-IM/JM

    The latter device, "w25q128jv" supports features named DTQ
    and QPI, otherwise it is the same.

    Not having the right flags has the annoying side effect
    that write access does not work.

    After this patch I can write to the flash on the Inteno
    XG6846 router.

    The flash memory also supports dual and quad SPI modes.
    This does not currently manifest, but by turning on SFDP
    parsing, the right SPI modes are emitted in
    /sys/kernel/debug/spi-nor/spi1.0/capabilities
    for this chip, so we also turn on this.

    Since we now have determined that SFDP parsing works on
    the device, we also detect the geometry using SFDP.

    After this dmesg and sysfs says:
    [    1.062401] spi-nor spi1.0: w25q128 (16384 Kbytes)
    cat erasesize
    65536
    (16384*1024)/65536 = 256 sectors

    spi-nor sysfs:
    cat jedec_id
    ef4018
    cat manufacturer
    winbond
    cat partname
    w25q128
    hexdump -v -C sfdp
    00000000  53 46 44 50 05 01 00 ff  00 05 01 10 80 00 00 ff
    00000010  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000020  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000030  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000040  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000050  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000060  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000070  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
    00000080  e5 20 f9 ff ff ff ff 07  44 eb 08 6b 08 3b 42 bb
    00000090  fe ff ff ff ff ff 00 00  ff ff 40 eb 0c 20 0f 52
    000000a0  10 d8 00 00 36 02 a6 00  82 ea 14 c9 e9 63 76 33
    000000b0  7a 75 7a 75 f7 a2 d5 5c  19 f7 4d ff e9 30 f8 80

    Cc: stable@vger.kernel.org
    Suggested-by: Michael Walle <michael@walle.cc>
```

```
    Reviewed-by: Michael Walle <michael@walle.cc>
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Link: https://lore.kernel.org/r/20230718-spi-nor-winbond-w25q128-v5-1-
a73653ee46c3@linaro.org
    Signed-off-by: Tudor Ambarus <tudor.ambarus@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 648d1150a688698e37f7aaf302860180901cb30e
Author: William Zhang <william.zhang@broadcom.com>
Date:   Thu Jul 6 11:29:08 2023 -0700

    mtd: rawnand: brcmnand: Fix potential out-of-bounds access in oob write

    commit 5d53244186c9ac58cb88d76a0958ca55b83a15cd upstream.

    When the oob buffer length is not in multiple of words, the oob write
    function does out-of-bounds read on the oob source buffer at the last
    iteration. Fix that by always checking length limit on the oob buffer
    read and fill with 0xff when reaching the end of the buffer to the oob
    registers.

    Fixes: 27c5b17cd1b1 ("mtd: nand: add NAND driver "library" for Broadcom STB NAND
controller")
    Signed-off-by: William Zhang <william.zhang@broadcom.com>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230706182909.79151-5-
william.zhang@broadcom.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dc4085bef6bdcff4d0836fd772c7da4d2bb0bc04
Author: William Zhang <william.zhang@broadcom.com>
Date:   Thu Jul 6 11:29:07 2023 -0700

    mtd: rawnand: brcmnand: Fix crash during the panic_write

    commit e66dd317194daae0475fe9e5577c80aa97f16cb9 upstream.

    When executing a NAND command within the panic write path, wait for any
    pending command instead of calling BUG_ON to avoid crashing while
    already crashing.

    Fixes: 27c5b17cd1b1 ("mtd: nand: add NAND driver "library" for Broadcom STB NAND
controller")
    Signed-off-by: William Zhang <william.zhang@broadcom.com>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Reviewed-by: Kursad Oney <kursad.oney@broadcom.com>
    Reviewed-by: Kamal Dasu <kamal.dasu@broadcom.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230706182909.79151-4-
william.zhang@broadcom.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0f98de0a11d29821d9448114178ddc1b1fe32a18
Author: Liu Ying <victor.liu@nxp.com>
Date:   Mon Jun 12 17:23:59 2023 +0800

    drm/mxsfb: Disable overlay plane in mxsfb_plane_overlay_atomic_disable()

    commit aa656d48e871a1b062e1bbf9474d8b831c35074c upstream.

    When disabling overlay plane in mxsfb_plane_overlay_atomic_update(),
    overlay plane's framebuffer pointer is NULL.  So, dereferencing it would
```

```
    cause a kernel Oops(NULL pointer dereferencing).  Fix the issue by
    disabling overlay plane in mxsfb_plane_overlay_atomic_disable() instead.

    Fixes: cb285a5348e7 ("drm: mxsfb: Replace mxsfb_get_fb_paddr() with
drm_fb_cma_get_gem_addr()")
    Cc: stable@vger.kernel.org # 5.19+
    Signed-off-by: Liu Ying <victor.liu@nxp.com>
    Reviewed-by: Marek Vasut <marex@denx.de>
    Signed-off-by: Marek Vasut <marex@denx.de>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230612092359.784115-1-
victor.liu@nxp.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ce585c9b1cd700324fecedae130a9c35261fec30
Author: Qu Wenruo <wqu@suse.com>
Date:   Thu Aug 3 14:33:31 2023 +0800

    btrfs: scrub: fix grouping of read IO

    commit ae76d8e3e1351aa1ba09cc68dab6866d356f2e17 upstream.

    [REGRESSION]
    There are several regression reports about the scrub performance with
    v6.4 kernel.

    On a PCIe 3.0 device, the old v6.3 kernel can go 3GB/s scrub speed, but
    v6.4 can only go 1GB/s, an obvious 66% performance drop.

    [CAUSE]
    Iostat shows a very different behavior between v6.3 and v6.4 kernel:

      Device         r/s      rkB/s   rrqm/s  %rrqm r_await rareq-sz aqu-sz  %util
      nvme0n1p3  9731.00 3425544.00 17237.00  63.92    2.18   352.02  21.18 100.00
      nvme0n1p3 15578.00  993616.00     5.00   0.03    0.09    63.78   1.32 100.00

    The upper one is v6.3 while the lower one is v6.4.

    There are several obvious differences:

    - Very few read merges
      This turns out to be a behavior change that we no longer do bio
      plug/unplug.

    - Very low aqu-sz
      This is due to the submit-and-wait behavior of flush_scrub_stripes(),
      and extra extent/csum tree search.

    Both behaviors are not that obvious on SATA SSDs, as SATA SSDs have NCQ
    to merge the reads, while SATA SSDs can not handle high queue depth well
    either.

    [FIX]
    For now this patch focuses on the read speed fix. Dev-replace replace
    speed needs more work.

    For the read part, we go two directions to fix the problems:

    - Re-introduce blk plug/unplug to merge read requests
      This is pretty simple, and the behavior is pretty easy to observe.

      This would enlarge the average read request size to 512K.

    - Introduce multi-group reads and no longer wait for each group
      Instead of the old behavior, which submits 8 stripes and waits for
      them, here we would enlarge the total number of stripes to 16 * 8.
```

```
        Which is 8M per device, the same limit as the old scrub in-flight
        bios size limit.

        Now every time we fill a group (8 stripes), we submit them and
        continue to next stripes.

        Only when the full 16 * 8 stripes are all filled, we submit the
        remaining ones (the last group), and wait for all groups to finish.
        Then submit the repair writes and dev-replace writes.

        This should enlarge the queue depth.

    This would greatly improve the merge rate (thus read block size) and
    queue depth:

    Before (with regression, and cached extent/csum path):

     Device         r/s      rkB/s   rrqm/s  %rrqm r_await rareq-sz aqu-sz  %util
     nvme0n1p3 20666.00 1318240.00    10.00   0.05    0.08    63.79   1.63 100.00

    After (with all patches applied):

     nvme0n1p3  5165.00 2278304.00 30557.00  85.54    0.55   441.10   2.81 100.00

    i.e. 1287 to 2224 MB/s.

    CC: stable@vger.kernel.org # 6.4+
    Signed-off-by: Qu Wenruo <wqu@suse.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6a1a0e43b1eed39c0dbfd4591238d74991dd861d
Author: Qu Wenruo <wqu@suse.com>
Date:   Thu Aug 3 14:33:30 2023 +0800

    btrfs: scrub: avoid unnecessary csum tree search preparing stripes

    commit 3c771c194402ffe20d4de68d9fc21e703179a9ce upstream.

    One of the bottleneck of the new scrub code is the extra csum tree
    search.

    The old code would only do the csum tree search for each scrub bio,
    which can be as large as 512KiB, thus they can afford to allocate a new
    path each time.

    But the new scrub code is doing csum tree search for each stripe, which
    is only 64KiB, this means we'd better re-use the same csum path during
    each search.

    This patch would introduce a per-sctx path for csum tree search, as we
    don't need to re-allocate the path every time we need to do a csum tree
    search.

    With this change we can further improve the queue depth and improve the
    scrub read performance:

    Before (with regression and cached extent tree path):

     Device         r/s      rkB/s   rrqm/s  %rrqm r_await rareq-sz aqu-sz  %util
     nvme0n1p3 15875.00 1013328.00    12.00   0.08    0.08    63.83   1.35 100.00

    After (with both cached extent/csum tree path):
```

```
      nvme0n1p3 17759.00 1133280.00    10.00   0.06    0.08    63.81   1.50 100.00

    Fixes: e02ee89baa66 ("btrfs: scrub: switch scrub_simple_mirror() to scrub_stripe
infrastructure")
    CC: stable@vger.kernel.org # 6.4+
    Signed-off-by: Qu Wenruo <wqu@suse.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5a5b4a234bfa7ac60cf60991dd85ac16e72a89fe
Author: Qu Wenruo <wqu@suse.com>
Date:   Thu Aug 3 14:33:29 2023 +0800

    btrfs: scrub: avoid unnecessary extent tree search preparing stripes

    commit 1dc4888e725dc748b82858984f2a5bd41efc5201 upstream.

    Since commit e02ee89baa66 ("btrfs: scrub: switch scrub_simple_mirror()
    to scrub_stripe infrastructure"), scrub no longer re-use the same path
    for extent tree search.

    This can lead to unnecessary extent tree search, especially for the new
    stripe based scrub, as we have way more stripes to prepare.

    This patch would re-introduce a shared path for extent tree search, and
    properly release it when the block group is scrubbed.

    This change alone can improve scrub performance slightly by reducing the
    time spend preparing the stripe thus improving the queue depth.

    Before (with regression):

     Device         r/s      rkB/s   rrqm/s  %rrqm r_await rareq-sz aqu-sz  %util
     nvme0n1p3 15578.00  993616.00     5.00   0.03    0.09    63.78   1.32 100.00

    After (with this patch):

     nvme0n1p3 15875.00 1013328.00    12.00   0.08    0.08    63.83   1.35 100.00

    Fixes: e02ee89baa66 ("btrfs: scrub: switch scrub_simple_mirror() to scrub_stripe
infrastructure")
    CC: stable@vger.kernel.org # 6.4+
    Signed-off-by: Qu Wenruo <wqu@suse.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 06a03d77043364fe5181504e88ecf594ddf633f9
Author: Anand Jain <anand.jain@oracle.com>
Date:   Mon Jul 31 19:16:34 2023 +0800

    btrfs: use the correct superblock to compare fsid in btrfs_validate_super

    commit d167aa76dc0683828588c25767da07fb549e4f48 upstream.

    The function btrfs_validate_super() should verify the fsid in the provided
    superblock argument. Because, all its callers expect it to do that.

    Such as in the following stack:

       write_all_supers()
           sb = fs_info->super_for_commit;
           btrfs_validate_write_super(.., sb)
             btrfs_validate_super(.., sb, ..)
```

```
        scrub_one_super()
             btrfs_validate_super(.., sb, ..)

    And
        check_dev_super()
             btrfs_validate_super(.., sb, ..)

    However, it currently verifies the fs_info::super_copy::fsid instead,
    which is not correct.  Fix this using the correct fsid in the superblock
    argument.

    CC: stable@vger.kernel.org # 5.4+
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Tested-by: Guilherme G. Piccoli <gpiccoli@igalia.com>
    Signed-off-by: Anand Jain <anand.jain@oracle.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8dd2239e46f2dcba1d561899da7d146698dd37c
Author: Naohiro Aota <naohiro.aota@wdc.com>
Date:   Tue Aug 8 01:12:40 2023 +0900

    btrfs: zoned: re-enable metadata over-commit for zoned mode

    commit 5b135b382a360f4c87cf8896d1465b0b07f10cb0 upstream.

    Now that, we can re-enable metadata over-commit. As we moved the activation
    from the reservation time to the write time, we no longer need to ensure
    all the reserved bytes is properly activated.

    Without the metadata over-commit, it suffers from lower performance because
    it needs to flush the delalloc items more often and allocate more block
    groups. Re-enabling metadata over-commit will solve the issue.

    Fixes: 79417d040f4f ("btrfs: zoned: disable metadata overcommit for zoned")
    CC: stable@vger.kernel.org # 6.1+
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Signed-off-by: Naohiro Aota <naohiro.aota@wdc.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9d1e020ed9649cf140fcfafd052cfdcce9e9d67d
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Mon Jul 31 11:13:00 2023 -0400

    btrfs: set page extent mapped after read_folio in relocate_one_page

    commit e7f1326cc24e22b38afc3acd328480a1183f9e79 upstream.

    One of the CI runs triggered the following panic

      assertion failed: PagePrivate(page) && page->private, in fs/btrfs/subpage.c:229
      ------------[ cut here ]------------
      kernel BUG at fs/btrfs/subpage.c:229!
      Internal error: Oops - BUG: 00000000f2000800 [#1] SMP
      CPU: 0 PID: 923660 Comm: btrfs Not tainted 6.5.0-rc3+ #1
      pstate: 61400005 (nZCv daif +PAN -UAO -TCO +DIT -SSBS BTYPE=--)
      pc : btrfs_subpage_assert+0xbc/0xf0
      lr : btrfs_subpage_assert+0xbc/0xf0
      sp : ffff800093213720
      x29: ffff800093213720 x28: ffff8000932138b4 x27: 000000000c280000
      x26: 00000001b5d00000 x25: 000000000c281000 x24: 000000000c281fff
      x23: 0000000000001000 x22: 0000000000000000 x21: ffffff42b95bf880
```

```
        x20: ffff42b9528e0000 x19: 0000000000001000 x18: ffffffffffffffff
        x17: 667274622f736620 x16: 6e69202c65746176 x15: 0000000000000028
        x14: 0000000000000003 x13: 00000000002672d7 x12: 0000000000000000
        x11: ffffcd3f0ccd9204 x10: ffffcd3f0554ae50 x9 : ffffcd3f0379528c
        x8 : ffff800093213428 x7 : 0000000000000000 x6 : ffffcd3f091771e8
        x5 : ffff42b97f333948 x4 : 0000000000000000 x3 : 0000000000000000
        x2 : 0000000000000000 x1 : ffff42b9556cde80 x0 : 000000000000004f
        Call trace:
         btrfs_subpage_assert+0xbc/0xf0
         btrfs_subpage_set_dirty+0x38/0xa0
         btrfs_page_set_dirty+0x58/0x88
         relocate_one_page+0x204/0x5f0
         relocate_file_extent_cluster+0x11c/0x180
         relocate_data_extent+0xd0/0xf8
         relocate_block_group+0x3d0/0x4e8
         btrfs_relocate_block_group+0x2d8/0x490
         btrfs_relocate_chunk+0x54/0x1a8
         btrfs_balance+0x7f4/0x1150
         btrfs_ioctl+0x10f0/0x20b8
         __arm64_sys_ioctl+0x120/0x11d8
         invoke_syscall.constprop.0+0x80/0xd8
         do_el0_svc+0x6c/0x158
         el0_svc+0x50/0x1b0
         el0t_64_sync_handler+0x120/0x130
         el0t_64_sync+0x194/0x198
        Code: 91098021 b0007fa0 91346000 97e9c6d2 (d4210000)

    This is the same problem outlined in 17b17fcd6d44 ("btrfs:
    set_page_extent_mapped after read_folio in btrfs_cont_expand") , and the
    fix is the same.  I originally looked for the same pattern elsewhere in
    our code, but mistakenly skipped over this code because I saw the page
    cache readahead before we set_page_extent_mapped, not realizing that
    this was only in the !page case, that we can still end up with a
    !uptodate page and then do the btrfs_read_folio further down.

    The fix here is the same as the above mentioned patch, move the
    set_page_extent_mapped call to after the btrfs_read_folio() block to
    make sure that we have the subpage blocksize stuff setup properly before
    using the page.

    CC: stable@vger.kernel.org # 6.1+
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 70fa3d4a856d52ba72619b49d8181a053fa47d50
Author: Filipe Manana <fdmanana@suse.com>
Date:   Wed Jul 26 16:56:57 2023 +0100

    btrfs: don't start transaction when joining with TRANS_JOIN_NOSTART

    commit 4490e803e1fe9fab8db5025e44e23b55df54078b upstream.

    When joining a transaction with TRANS_JOIN_NOSTART, if we don't find a
    running transaction we end up creating one. This goes against the purpose
    of TRANS_JOIN_NOSTART which is to join a running transaction if its state
    is at or below the state TRANS_STATE_COMMIT_START, otherwise return an
    -ENOENT error and don't start a new transaction. So fix this to not create
    a new transaction if there's no running transaction at or below that
    state.

    CC: stable@vger.kernel.org # 4.14+
    Fixes: a6d155d2e363 ("Btrfs: fix deadlock between fiemap and transaction commits")
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
```

```
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b5cd2c8a07c61040ef7418f2188dbced4d206f7b
Author: Boris Burkov <boris@bur.io>
Date:   Fri Jul 21 09:02:06 2023 -0700

    btrfs: free qgroup rsv on io failure

    commit e28b02118b94e42be3355458a2406c6861e2dd32 upstream.

    If we do a write whose bio suffers an error, we will never reclaim the
    qgroup reserved space for it. We allocate the space in the write_iter
    codepath, then release the reservation as we allocate the ordered
    extent, but we only create a delayed ref if the ordered extent finishes.
    If it has an error, we simply leak the rsv. This is apparent in running
    any error injecting (dmerror) fstests like btrfs/146 or btrfs/160. Such
    tests fail due to dmesg on umount complaining about the leaked qgroup
    data space.

    When we clean up other aspects of space on failed ordered_extents, also
    free the qgroup rsv.

    Reviewed-by: Josef Bacik <josef@toxicpanda.com>
    CC: stable@vger.kernel.org # 5.10+
    Signed-off-by: Boris Burkov <boris@bur.io>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6fd4d27daccbcd0e79d9f7eb58a4d21dbbd3158a
Author: Boris Burkov <boris@bur.io>
Date:   Fri Jul 21 09:02:07 2023 -0700

    btrfs: fix start transaction qgroup rsv double free

    commit a6496849671a5bc9218ecec25a983253b34351b1 upstream.

    btrfs_start_transaction reserves metadata space of the PERTRANS type
    before it identifies a transaction to start/join. This allows flushing
    when reserving that space without a deadlock. However, it results in a
    race which temporarily breaks qgroup rsv accounting.

    T1                                              T2
    start_transaction
    do_stuff
                                                start_transaction
                                                    qgroup_reserve_meta_pertrans
    commit_transaction
        qgroup_free_meta_all_pertrans
                                                hit an error starting txn
                                                goto reserve_fail
                                                qgroup_free_meta_pertrans (already freed!)

    The basic issue is that there is nothing preventing another commit from
    committing before start_transaction finishes (in fact sometimes we
    intentionally wait for it) so any error path that frees the reserve is
    at risk of this race.

    While this exact space was getting freed anyway, and it's not a huge
    deal to double free it (just a warning, the free code catches this), it
    can result in incorrectly freeing some other pertrans reservation in
    this same reservation, which could then lead to spuriously granting
    reservations we might not have the space for. Therefore, I do believe it
    is worth fixing.
```

```
    To fix it, use the existing prealloc->pertrans conversion mechanism.
    When we first reserve the space, we reserve prealloc space and only when
    we are sure we have a transaction do we convert it to pertrans. This way
    any racing commits do not blow away our reservation, but we still get a
    pertrans reservation that is freed when _this_ transaction gets committed.

    This issue can be reproduced by running generic/269 with either qgroups
    or squotas enabled via mkfs on the scratch device.

    Reviewed-by: Josef Bacik <josef@toxicpanda.com>
    CC: stable@vger.kernel.org # 5.10+
    Signed-off-by: Boris Burkov <boris@bur.io>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7b2d588572e75489341ea60ff809c6d6c4b746f9
Author: Naohiro Aota <naohiro.aota@wdc.com>
Date:   Fri Jul 21 16:42:14 2023 +0900

    btrfs: zoned: do not zone finish data relocation block group

    commit 332581bde2a419d5f12a93a1cdc2856af649a3cc upstream.

    When multiple writes happen at once, we may need to sacrifice a currently
    active block group to be zone finished for a new allocation. We choose a
    block group with the least free space left, and zone finish it.

    To do the finishing, we need to send IOs for already allocated region
    and wait for them and on-going IOs. Otherwise, these IOs fail because the
    zone is already finished at the time the IO reach a device.

    However, if a block group dedicated to the data relocation is zone
    finished, there is a chance that finishing it before an ongoing write IO
    reaches the device. That is because there is timing gap between an
    allocation is done (block_group->reservations == 0, as pre-allocation is
    done) and an ordered extent is created when the relocation IO starts.
    Thus, if we finish the zone between them, we can fail the IOs.

    We cannot simply use "fs_info->data_reloc_bg == block_group->start" to
    avoid the zone finishing. Because, the data_reloc_bg may already switch to
    a new block group, while there are still ongoing write IOs to the old
    data_reloc_bg.

    So, this patch reworks the BLOCK_GROUP_FLAG_ZONED_DATA_RELOC bit to
    indicate there is a data relocation allocation and/or ongoing write to the
    block group. The bit is set on allocation and cleared in end_io function of
    the last IO for the currently allocated region.

    To change the timing of the bit setting also solves the issue that the bit
    being left even after there is no IO going on. With the current code, if
    the data_reloc_bg switches after the last IO to the current data_reloc_bg,
    the bit is set at this timing and there is no one clearing that bit. As a
    result, that block group is kept unallocatable for anything.

    Fixes: 343d8a30851c ("btrfs: zoned: prevent allocation from previous data relocation
BG")
    Fixes: 74e91b12b115 ("btrfs: zoned: zone finish unused block group")
    CC: stable@vger.kernel.org # 6.1+
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Signed-off-by: Naohiro Aota <naohiro.aota@wdc.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b638e83f6781d0ddcb4954d5726e296ca827472f
```

```
Author: ruanmeisi <ruan.meisi@zte.com.cn>
Date:   Tue Apr 25 19:13:54 2023 +0800

    fuse: nlookup missing decrement in fuse_direntplus_link

    commit b8bd342d50cbf606666488488f9fea374aceb2d5 upstream.

    During our debugging of glusterfs, we found an Assertion failed error:
    inode_lookup >= nlookup, which was caused by the nlookup value in the
    kernel being greater than that in the FUSE file system.

    The issue was introduced by fuse_direntplus_link, where in the function,
    fuse_iget increments nlookup, and if d_splice_alias returns failure,
    fuse_direntplus_link returns failure without decrementing nlookup
    https://github.com/gluster/glusterfs/pull/4081

    Signed-off-by: ruanmeisi <ruan.meisi@zte.com.cn>
    Fixes: 0b05b18381ee ("fuse: implement NFS-like readdirplus support")
    Cc: <stable@vger.kernel.org> # v3.9
    Signed-off-by: Miklos Szeredi <mszeredi@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3583d05ab60f100eac5fc7ec1e7ab19ba5eb4a98
Author: Damien Le Moal <dlemoal@kernel.org>
Date:   Thu Aug 24 07:41:59 2023 +0900

    ata: pata_ftide010: Add missing MODULE_DESCRIPTION

    commit 7274eef5729037300f29d14edeb334a47a098f65 upstream.

    Add the missing MODULE_DESCRIPTION() to avoid warnings such as:

    WARNING: modpost: missing MODULE_DESCRIPTION() in drivers/ata/pata_ftide010.o

    when compiling with W=1.

    Fixes: be4e456ed3a5 ("ata: Add driver for Faraday Technology FTIDE010")
    Cc: stable@vger.kernel.org
    Signed-off-by: Damien Le Moal <dlemoal@kernel.org>
    Reviewed-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 45139e7f21ea767795fe100ced1bb8e78d8d8d23
Author: Damien Le Moal <dlemoal@kernel.org>
Date:   Thu Aug 24 07:43:18 2023 +0900

    ata: sata_gemini: Add missing MODULE_DESCRIPTION

    commit 8566572bf3b4d6e416a4bf2110dbb4817d11ba59 upstream.

    Add the missing MODULE_DESCRIPTION() to avoid warnings such as:

    WARNING: modpost: missing MODULE_DESCRIPTION() in drivers/ata/sata_gemini.o

    when compiling with W=1.

    Fixes: be4e456ed3a5 ("ata: Add driver for Faraday Technology FTIDE010")
    Cc: stable@vger.kernel.org
    Signed-off-by: Damien Le Moal <dlemoal@kernel.org>
    Reviewed-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a9ddf95ac5a2810fdc2920c634865773614fee6d
Author: Michael Schmitz <schmitzmic@gmail.com>
Date:   Sun Aug 27 16:13:47 2023 +1200
```

```
    ata: pata_falcon: fix IO base selection for Q40

    commit 8a1f00b753ecfdb117dc1a07e68c46d80e7923ea upstream.

    With commit 44b1fbc0f5f3 ("m68k/q40: Replace q40ide driver
    with pata_falcon and falconide"), the Q40 IDE driver was
    replaced by pata_falcon.c.

    Both IO and memory resources were defined for the Q40 IDE
    platform device, but definition of the IDE register addresses
    was modeled after the Falcon case, both in use of the memory
    resources and in including register shift and byte vs. word
    offset in the address.

    This was correct for the Falcon case, which does not apply
    any address translation to the register addresses. In the
    Q40 case, all of device base address, byte access offset
    and register shift is included in the platform specific
    ISA access translation (in asm/mm_io.h).

    As a consequence, such address translation gets applied
    twice, and register addresses are mangled.

    Use the device base address from the platform IO resource
    for Q40 (the IO address translation will then add the correct
    ISA window base address and byte access offset), with register
    shift 1. Use MMIO base address and register shift 2 as before
    for Falcon.

    Encode PIO_OFFSET into IO port addresses for all registers
    for Q40 except the data transfer register. Encode the MMIO
    offset there (pata_falcon_data_xfer() directly uses raw IO
    with no address translation).

    Reported-by: William R Sowerbutts <will@sowerbutts.com>
    Closes:
https://lore.kernel.org/r/CAMuHMdUU62jjunJh9cqSqHT87B0H0A4udOOPs=WN7WZKpcagVA@mail.gmail.com
    Link:
https://lore.kernel.org/r/CAMuHMdUU62jjunJh9cqSqHT87B0H0A4udOOPs=WN7WZKpcagVA@mail.gmail.com
    Fixes: 44b1fbc0f5f3 ("m68k/q40: Replace q40ide driver with pata_falcon and falconide")
    Cc: stable@vger.kernel.org
    Cc: Finn Thain <fthain@linux-m68k.org>
    Cc: Geert Uytterhoeven <geert@linux-m68k.org>
    Tested-by: William R Sowerbutts <will@sowerbutts.com>
    Signed-off-by: Michael Schmitz <schmitzmic@gmail.com>
    Reviewed-by: Sergey Shtylyov <s.shtylyov@omp.ru>
    Reviewed-by: Geert Uytterhoeven <geert@linux-m68k.org>
    Signed-off-by: Damien Le Moal <dlemoal@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 247d2f1f5a3cdb56b0486be23016586ca3341ff7
Author: Werner Fischer <devlists@wefi.net>
Date:   Tue Aug 29 13:33:58 2023 +0200

    ata: ahci: Add Elkhart Lake AHCI controller

    commit 2a2df98ec592667927b5c1351afa6493ea125c9f upstream.

    Elkhart Lake is the successor of Apollo Lake and Gemini Lake. These
    CPUs and their PCHs are used in mobile and embedded environments.

    With this patch I suggest that Elkhart Lake SATA controllers [1] should
    use the default LPM policy for mobile chipsets.
    The disadvantage of missing hot-plug support with this setting should
```

```
    not be an issue, as those CPUs are used in embedded environments and
    not in servers with hot-plug backplanes.

    We discovered that the Elkhart Lake SATA controllers have been missing
    in ahci.c after a customer reported the throttling of his SATA SSD
    after a short period of higher I/O. We determined the high temperature
    of the SSD controller in idle mode as the root cause for that.

    Depending on the used SSD, we have seen up to 1.8 Watt lower system
    idle power usage and up to 30°C lower SSD controller temperatures in
    our tests, when we set med_power_with_dipm manually. I have provided a
    table showing seven different SATA SSDs from ATP, Intel/Solidigm and
    Samsung [2].

    Intel lists a total of 3 SATA controller IDs (4B60, 4B62, 4B63) in [1]
    for those mobile PCHs.
    This commit just adds 0x4b63 as I do not have test systems with 0x4b60
    and 0x4b62 SATA controllers.
    I have tested this patch with a system which uses 0x4b63 as SATA
    controller.

    [1] https://sata-io.org/product/8803
    [2] https://www.thomas-krenn.com/en/wiki/SATA_Link_Power_Management#Example_LES_v4

    Signed-off-by: Werner Fischer <devlists@wefi.net>
    Cc: stable@vger.kernel.org
    Signed-off-by: Damien Le Moal <dlemoal@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b9d30c38ee859d833a51131b5b4b864c7a6219d0
Author: Johannes Weiner <hannes@cmpxchg.org>
Date:   Wed Aug 23 15:54:30 2023 -0700

    memcontrol: ensure memcg acquired by id is properly set up

    commit 6f0df8e16eb543167f2929cb756e695709a3551d upstream.

    In the eviction recency check, we attempt to retrieve the memcg to which
    the folio belonged when it was evicted, by the memcg id stored in the
    shadow entry.  However, there is a chance that the retrieved memcg is not
    the original memcg that has been killed, but a new one which happens to
    have the same id.

    This is a somewhat unfortunate, but acceptable and rare inaccuracy in the
    heuristics.  However, if we retrieve this new memcg between its allocation
    and when it is properly attached to the memcg hierarchy, we could run into
    the following NULL pointer exception during the memcg hierarchy traversal
    done in mem_cgroup_get_nr_swap_pages():

    [ 155757.793456] BUG: kernel NULL pointer dereference, address: 00000000000000c0
    [ 155757.807568] #PF: supervisor read access in kernel mode
    [ 155757.818024] #PF: error_code(0x0000) - not-present page
    [ 155757.828482] PGD 401f77067 P4D 401f77067 PUD 401f76067 PMD 0
    [ 155757.839985] Oops: 0000 [#1] SMP
    [ 155757.887870] RIP: 0010:mem_cgroup_get_nr_swap_pages+0x3d/0xb0
    [ 155757.899377] Code: 29 19 4a 02 48 39 f9 74 63 48 8b 97 c0 00 00 00 48 8b b7 58 02 00
00 48 2b b7 c0 01 00 00 48 39 f0 48 0f 4d c6 48 39 d1 74 42 <48> 8b b2 c0 00 00 00 48 8b ba
58 02 00 00 48 2b ba c0 01 00 00 48
    [ 155757.937125] RSP: 0018:ffffc9002ecdfbc8 EFLAGS: 00010286
    [ 155757.947755] RAX: 00000000003a3b1c RBX: 000007fffffffff RCX: ffff888280183000
    [ 155757.962202] RDX: 0000000000000000 RSI: 0007ffffffffffff RDI: ffff888bbc2d1000
    [ 155757.976648] RBP: 0000000000000001 R08: 000000000000000b R09: ffff888ad9cedba0
    [ 155757.991094] R10: ffffea0039c07900 R11: 0000000000000010 R12: ffff888b23a7b000
    [ 155758.005540] R13: 0000000000000000 R14: ffff888bbc2d1000 R15: 000007ffffc71354
    [ 155758.019991] FS:  00007f6234c68640(0000) GS:ffff88903f9c0000(0000)
```

```
 knlGS:0000000000000000
     [ 155758.036356] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     [ 155758.048023] CR2: 00000000000000c0 CR3: 0000000a83eb8004 CR4: 00000000007706e0
     [ 155758.062473] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     [ 155758.076924] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     [ 155758.091376] PKRU: 55555554
     [ 155758.096957] Call Trace:
     [ 155758.102016]  <TASK>
     [ 155758.106502]  ? __die+0x78/0xc0
     [ 155758.112793]  ? page_fault_oops+0x286/0x380
     [ 155758.121175]  ? exc_page_fault+0x5d/0x110
     [ 155758.129209]  ? asm_exc_page_fault+0x22/0x30
     [ 155758.137763]  ? mem_cgroup_get_nr_swap_pages+0x3d/0xb0
     [ 155758.148060]  workingset_test_recent+0xda/0x1b0
     [ 155758.157133]  workingset_refault+0xca/0x1e0
     [ 155758.165508]  filemap_add_folio+0x4d/0x70
     [ 155758.173538]  page_cache_ra_unbounded+0xed/0x190
     [ 155758.182919]  page_cache_sync_ra+0xd6/0x1e0
     [ 155758.191738]  filemap_read+0x68d/0xdf0
     [ 155758.199495]  ? mlx5e_napi_poll+0x123/0x940
     [ 155758.207981]  ? __napi_schedule+0x55/0x90
     [ 155758.216095]  __x64_sys_pread64+0x1d6/0x2c0
     [ 155758.224601]  do_syscall_64+0x3d/0x80
     [ 155758.232058]  entry_SYSCALL_64_after_hwframe+0x46/0xb0
     [ 155758.242473] RIP: 0033:0x7f62c29153b5
     [ 155758.249938] Code: e8 48 89 75 f0 89 7d f8 48 89 4d e0 e8 b4 e6 f7 ff 41 89 c0 4c 8b
 55 e0 48 8b 55 e8 48 8b 75 f0 8b 7d f8 b8 11 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 33 44 89
 c7 48 89 45 f8 e8 e7 e6 f7 ff 48 8b
     [ 155758.288005] RSP: 002b:00007f6234c5ffd0 EFLAGS: 00000293 ORIG_RAX: 0000000000000011
     [ 155758.303474] RAX: ffffffffffffffda RBX: 00007f628c4e70c0 RCX: 00007f62c29153b5
     [ 155758.318075] RDX: 000000000003c041 RSI: 00007f61d2986000 RDI: 0000000000000076
     [ 155758.332678] RBP: 00007f6234c5fff0 R08: 0000000000000000 R09: 0000000064d5230c
     [ 155758.347452] R10: 000000000027d450 R11: 0000000000000293 R12: 000000000003c041
     [ 155758.362044] R13: 00007f61d2986000 R14: 00007f629e11b060 R15: 000000000027d450
     [ 155758.376661]  </TASK>

     This patch fixes the issue by moving the memcg's id publication from the
     alloc stage to online stage, ensuring that any memcg acquired via id must
     be connected to the memcg tree.

     Link: https://lkml.kernel.org/r/20230823225430.166925-1-nphamcs@gmail.com
     Fixes: f78dfc7b77d5 ("workingset: fix confusion around eviction vs refault container")
     Signed-off-by: Johannes Weiner <hannes@cmpxchg.org>
     Co-developed-by: Nhat Pham <nphamcs@gmail.com>
     Signed-off-by: Nhat Pham <nphamcs@gmail.com>
     Acked-by: Shakeel Butt <shakeelb@google.com>
     Cc: Yosry Ahmed <yosryahmed@google.com>
     Cc: Michal Hocko <mhocko@suse.com>
     Cc: Roman Gushchin <roman.gushchin@linux.dev>
     Cc: Muchun Song <songmuchun@bytedance.com>
     Cc: <stable@vger.kernel.org>
     Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 6ef09eca632f869efdd44ff39cb357f324fc7be9
 Author: Christian Marangi <ansuelsmth@gmail.com>
 Date:   Sun Jul 16 04:28:04 2023 +0200

     hwspinlock: qcom: add missing regmap config for SFPB MMIO implementation

     commit 23316be8a9d450f33a21f1efe7d89570becbec58 upstream.

     Commit 5d4753f741d8 ("hwspinlock: qcom: add support for MMIO on older
     SoCs") introduced and made regmap_config mandatory in the of_data struct
     but didn't add the regmap_config for sfpb based devices.
```

```
    SFPB based devices can both use the legacy syscon way to probe or the
    new MMIO way and currently device that use the MMIO way are broken as
    they lack the definition of the now required regmap_config and always
    return -EINVAL (and indirectly makes fail probing everything that
    depends on it, smem, nandc with smem-parser...)

    Fix this by correctly adding the missing regmap_config and restore
    function of hwspinlock on SFPB based devices with MMIO implementation.

    Cc: stable@vger.kernel.org
    Fixes: 5d4753f741d8 ("hwspinlock: qcom: add support for MMIO on older SoCs")
    Signed-off-by: Christian Marangi <ansuelsmth@gmail.com>
    Link: https://lore.kernel.org/r/20230716022804.21239-1-ansuelsmth@gmail.com
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4be83c312df6413b2d4191fdcaf16a20f6e496d6
Author: Nathan Chancellor <nathan@kernel.org>
Date:   Mon Aug 7 08:36:28 2023 -0700

    lib: test_scanf: Add explicit type cast to result initialization in test_number_prefix()

    commit 92382d744176f230101d54f5c017bccd62770f01 upstream.

    A recent change in clang allows it to consider more expressions as
    compile time constants, which causes it to point out an implicit
    conversion in the scanf tests:

      lib/test_scanf.c:661:2: warning: implicit conversion from 'int' to 'unsigned char'
changes value from -168 to 88 [-Wconstant-conversion]
        661 |         test_number_prefix(unsigned char,       "0xA7", "%2hhx%hhx", 0, 0xa7,
2, check_uchar);
            |
^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      lib/test_scanf.c:609:29: note: expanded from macro 'test_number_prefix'
        609 |         T result[2] = {~expect[0], ~expect[1]};
\
            |                           ~           ^~~~~~~~~
      1 warning generated.

    The result of the bitwise negation is the type of the operand after
    going through the integer promotion rules, so this truncation is
    expected but harmless, as the initial values in the result array get
    overwritten by _test() anyways. Add an explicit cast to the expected
    type in test_number_prefix() to silence the warning. There is no
    functional change, as all the tests still pass with GCC 13.1.0 and clang
    18.0.0.

    Cc: stable@vger.kernel.org
    Link: https://github.com/ClangBuiltLinux/linuxq/issues/1899
    Link: https://github.com/llvm/llvm-
project/commit/610ec954e1f81c0e8fcadedcd25afe643f5a094e
    Suggested-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Nathan Chancellor <nathan@kernel.org>
    Reviewed-by: Petr Mladek <pmladek@suse.com>
    Signed-off-by: Petr Mladek <pmladek@suse.com>
    Link: https://lore.kernel.org/r/20230807-test_scanf-wconstant-conversion-v2-1-
839ca39083e1@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 88f10fb3170d3f8f8d7f7212f5cec03c42ef6773
Author: Jaegeuk Kim <jaegeuk@kernel.org>
Date:   Fri Aug 18 11:34:32 2023 -0700
```

```
    f2fs: avoid false alarm of circular locking

    commit 5c13e2388bf3426fd69a89eb46e50469e9624e56 upstream.

    =====================================================
    WARNING: possible circular locking dependency detected
    6.5.0-rc5-syzkaller-00353-gae545c3283dc #0 Not tainted
    -----------------------------------------------------
    syz-executor273/5027 is trying to acquire lock:
    ffff888077fe1fb0 (&fi->i_sem){+.+.}-{3:3}, at: f2fs_down_write fs/f2fs/f2fs.h:2133
 [inline]
    ffff888077fe1fb0 (&fi->i_sem){+.+.}-{3:3}, at: f2fs_add_inline_entry+0x300/0x6f0
 fs/f2fs/inline.c:644

    but task is already holding lock:
    ffff888077fe07c8 (&fi->i_xattr_sem){.+.+}-{3:3}, at: f2fs_down_read fs/f2fs/f2fs.h:2108
 [inline]
    ffff888077fe07c8 (&fi->i_xattr_sem){.+.+}-{3:3}, at: f2fs_add_dentry+0x92/0x230
 fs/f2fs/dir.c:783

    which lock already depends on the new lock.

    the existing dependency chain (in reverse order) is:

    -> #1 (&fi->i_xattr_sem){.+.+}-{3:3}:
           down_read+0x9c/0x470 kernel/locking/rwsem.c:1520
           f2fs_down_read fs/f2fs/f2fs.h:2108 [inline]
           f2fs_getxattr+0xb1e/0x12c0 fs/f2fs/xattr.c:532
           __f2fs_get_acl+0x5a/0x900 fs/f2fs/acl.c:179
           f2fs_acl_create fs/f2fs/acl.c:377 [inline]
           f2fs_init_acl+0x15c/0xb30 fs/f2fs/acl.c:420
           f2fs_init_inode_metadata+0x159/0x1290 fs/f2fs/dir.c:558
           f2fs_add_regular_entry+0x79e/0xb90 fs/f2fs/dir.c:740
           f2fs_add_dentry+0x1de/0x230 fs/f2fs/dir.c:788
           f2fs_do_add_link+0x190/0x280 fs/f2fs/dir.c:827
           f2fs_add_link fs/f2fs/f2fs.h:3554 [inline]
           f2fs_mkdir+0x377/0x620 fs/f2fs/namei.c:781
           vfs_mkdir+0x532/0x7e0 fs/namei.c:4117
           do_mkdirat+0x2a9/0x330 fs/namei.c:4140
           __do_sys_mkdir fs/namei.c:4160 [inline]
           __se_sys_mkdir fs/namei.c:4158 [inline]
           __x64_sys_mkdir+0xf2/0x140 fs/namei.c:4158
           do_syscall_x64 arch/x86/entry/common.c:50 [inline]
           do_syscall_64+0x38/0xb0 arch/x86/entry/common.c:80
           entry_SYSCALL_64_after_hwframe+0x63/0xcd

    -> #0 (&fi->i_sem){+.+.}-{3:3}:
           check_prev_add kernel/locking/lockdep.c:3142 [inline]
           check_prevs_add kernel/locking/lockdep.c:3261 [inline]
           validate_chain kernel/locking/lockdep.c:3876 [inline]
           __lock_acquire+0x2e3d/0x5de0 kernel/locking/lockdep.c:5144
           lock_acquire kernel/locking/lockdep.c:5761 [inline]
           lock_acquire+0x1ae/0x510 kernel/locking/lockdep.c:5726
           down_write+0x93/0x200 kernel/locking/rwsem.c:1573
           f2fs_down_write fs/f2fs/f2fs.h:2133 [inline]
           f2fs_add_inline_entry+0x300/0x6f0 fs/f2fs/inline.c:644
           f2fs_add_dentry+0xa6/0x230 fs/f2fs/dir.c:784
           f2fs_do_add_link+0x190/0x280 fs/f2fs/dir.c:827
           f2fs_add_link fs/f2fs/f2fs.h:3554 [inline]
           f2fs_mkdir+0x377/0x620 fs/f2fs/namei.c:781
           vfs_mkdir+0x532/0x7e0 fs/namei.c:4117
           ovl_do_mkdir fs/overlayfs/overlayfs.h:196 [inline]
           ovl_mkdir_real+0xb5/0x370 fs/overlayfs/dir.c:146
           ovl_workdir_create+0x3de/0x820 fs/overlayfs/super.c:309
           ovl_make_workdir fs/overlayfs/super.c:711 [inline]
```

```
            ovl_get_workdir fs/overlayfs/super.c:864 [inline]
            ovl_fill_super+0xdab/0x6180 fs/overlayfs/super.c:1400
            vfs_get_super+0xf9/0x290 fs/super.c:1152
            vfs_get_tree+0x88/0x350 fs/super.c:1519
            do_new_mount fs/namespace.c:3335 [inline]
            path_mount+0x1492/0x1ed0 fs/namespace.c:3662
            do_mount fs/namespace.c:3675 [inline]
            __do_sys_mount fs/namespace.c:3884 [inline]
            __se_sys_mount fs/namespace.c:3861 [inline]
            __x64_sys_mount+0x293/0x310 fs/namespace.c:3861
            do_syscall_x64 arch/x86/entry/common.c:50 [inline]
            do_syscall_64+0x38/0xb0 arch/x86/entry/common.c:80
            entry_SYSCALL_64_after_hwframe+0x63/0xcd

     other info that might help us debug this:

      Possible unsafe locking scenario:

            CPU0                    CPU1
            ----                    ----
       rlock(&fi->i_xattr_sem);
                                    lock(&fi->i_sem);
                                    lock(&fi->i_xattr_sem);
       lock(&fi->i_sem);

     Cc: <stable@vger.kernel.org>
     Reported-and-tested-by: syzbot+e5600587fa9cbf8e3826@syzkaller.appspotmail.com
     Fixes: 5eda1ad1aaff "f2fs: fix deadlock in i_xattr_sem and inode page lock"
     Tested-by: Guenter Roeck <linux@roeck-us.net>
     Reviewed-by: Chao Yu <chao@kernel.org>
     Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit bfa7853bb47fee0c17030b377c98cf4ede47ba33
  Author: Jaegeuk Kim <jaegeuk@kernel.org>
  Date:   Fri Jul 7 07:03:13 2023 -0700

     f2fs: flush inode if atomic file is aborted

     commit a3ab55746612247ce3dcaac6de66f5ffc055b9df upstream.

     Let's flush the inode being aborted atomic operation to avoid stale dirty
     inode during eviction in this call stack:

       f2fs_mark_inode_dirty_sync+0x22/0x40 [f2fs]
       f2fs_abort_atomic_write+0xc4/0xf0 [f2fs]
       f2fs_evict_inode+0x3f/0x690 [f2fs]
       ? sugov_start+0x140/0x140
       evict+0xc3/0x1c0
       evict_inodes+0x17b/0x210
       generic_shutdown_super+0x32/0x120
       kill_block_super+0x21/0x50
       deactivate_locked_super+0x31/0x90
       cleanup_mnt+0x100/0x160
       task_work_run+0x59/0x90
       do_exit+0x33b/0xa50
       do_group_exit+0x2d/0x80
       __x64_sys_exit_group+0x14/0x20
       do_syscall_64+0x3b/0x90
       entry_SYSCALL_64_after_hwframe+0x63/0xcd

     This triggers f2fs_bug_on() in f2fs_evict_inode:
      f2fs_bug_on(sbi, is_inode_flag_set(inode, FI_DIRTY_INODE));

     This fixes the syzbot report:
```

```
    loop0: detected capacity change from 0 to 131072
    F2FS-fs (loop0): invalid crc value
    F2FS-fs (loop0): Found nat_bits in checkpoint
    F2FS-fs (loop0): Mounted with checkpoint version = 48b305e4
    ------------[ cut here ]------------
    kernel BUG at fs/f2fs/inode.c:869!
    invalid opcode: 0000 [#1] PREEMPT SMP KASAN
    CPU: 0 PID: 5014 Comm: syz-executor220 Not tainted 6.4.0-syzkaller-11479-g6cd06ab12d1a
#0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
05/27/2023
    RIP: 0010:f2fs_evict_inode+0x172d/0x1e00 fs/f2fs/inode.c:869
    Code: ff df 48 c1 ea 03 80 3c 02 00 0f 85 6a 06 00 00 8b 75 40 ba 01 00 00 00 4c 89 e7
e8 6d ce 06 00 e9 aa fc ff ff e8 63 22 e2 fd <0f> 0b e8 5c 22 e2 fd 48 c7 c0 a8 3a 18 8d 48
ba 00 00 00 00 00 fc
    RSP: 0018:ffffc90003a6fa00 EFLAGS: 00010293
    RAX: 0000000000000000 RBX: 0000000000000001 RCX: 0000000000000000
    RDX: ffff8880273b8000 RSI: ffffffff83a2bd0d RDI: 0000000000000007
    RBP: ffff888077db91b0 R08: 0000000000000007 R09: 0000000000000000
    R10: 0000000000000001 R11: 0000000000000001 R12: ffff888029a3c000
    R13: ffff888077db9660 R14: ffff888029a3c0b8 R15: ffff888077db9c50
    FS:  0000000000000000(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007f1909bb9000 CR3: 00000000276a9000 CR4: 0000000000350ef0
    Call Trace:
     <TASK>
     evict+0x2ed/0x6b0 fs/inode.c:665
     dispose_list+0x117/0x1e0 fs/inode.c:698
     evict_inodes+0x345/0x440 fs/inode.c:748
     generic_shutdown_super+0xaf/0x480 fs/super.c:478
     kill_block_super+0x64/0xb0 fs/super.c:1417
     kill_f2fs_super+0x2af/0x3c0 fs/f2fs/super.c:4704
     deactivate_locked_super+0x98/0x160 fs/super.c:330
     deactivate_super+0xb1/0xd0 fs/super.c:361
     cleanup_mnt+0x2ae/0x3d0 fs/namespace.c:1254
     task_work_run+0x16f/0x270 kernel/task_work.c:179
     exit_task_work include/linux/task_work.h:38 [inline]
     do_exit+0xa9a/0x29a0 kernel/exit.c:874
     do_group_exit+0xd4/0x2a0 kernel/exit.c:1024
     __do_sys_exit_group kernel/exit.c:1035 [inline]
     __se_sys_exit_group kernel/exit.c:1033 [inline]
     __x64_sys_exit_group+0x3e/0x50 kernel/exit.c:1033
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x63/0xcd
    RIP: 0033:0x7f309be71a09
    Code: Unable to access opcode bytes at 0x7f309be719df.
    RSP: 002b:00007fff171df518 EFLAGS: 00000246 ORIG_RAX: 00000000000000e7
    RAX: ffffffffffffffda RBX: 00007f309bef7330 RCX: 00007f309be71a09
    RDX: 000000000000003c RSI: 00000000000000e7 RDI: 0000000000000001
    RBP: 0000000000000001 R08: fffffffffffffff0 R09: 00007f309bef1e40
    R10: 0000000000010600 R11: 0000000000000246 R12: 00007f309bef7330
    R13: 0000000000000001 R14: 0000000000000000 R15: 0000000000000001
     </TASK>
    Modules linked in:
    ---[ end trace 0000000000000000 ]---
    RIP: 0010:f2fs_evict_inode+0x172d/0x1e00 fs/f2fs/inode.c:869
    Code: ff df 48 c1 ea 03 80 3c 02 00 0f 85 6a 06 00 00 8b 75 40 ba 01 00 00 00 4c 89 e7
e8 6d ce 06 00 e9 aa fc ff ff e8 63 22 e2 fd <0f> 0b e8 5c 22 e2 fd 48 c7 c0 a8 3a 18 8d 48
ba 00 00 00 00 00 fc
    RSP: 0018:ffffc90003a6fa00 EFLAGS: 00010293
    RAX: 0000000000000000 RBX: 0000000000000001 RCX: 0000000000000000
    RDX: ffff8880273b8000 RSI: ffffffff83a2bd0d RDI: 0000000000000007
    RBP: ffff888077db91b0 R08: 0000000000000007 R09: 0000000000000000
```

```
    R10: 0000000000000001 R11: 0000000000000001 R12: ffff888029a3c000
    R13: ffff888077db9660 R14: ffff888029a3c0b8 R15: ffff888077db9c50
    FS:  0000000000000000(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007f1909bb9000 CR3: 00000000276a9000 CR4: 0000000000350ef0

    Cc: <stable@vger.kernel.org>
    Reported-and-tested-by: syzbot+e1246909d526a9d470fa@syzkaller.appspotmail.com
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6cba5457aa9d5c571da06a54c9c316166af2e1a8
Author: Jaegeuk Kim <jaegeuk@kernel.org>
Date:   Thu Jan 19 10:47:00 2023 -0800

    f2fs: get out of a repeat loop when getting a locked data page

    commit d2d9bb3b6d2fbccb5b33d3a85a2830971625a4ea upstream.

    https://bugzilla.kernel.org/show_bug.cgi?id=216050

    Somehow we're getting a page which has a different mapping.
    Let's avoid the infinite loop.

    Cc: <stable@vger.kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 697b223d3226ae3de464b15d4a689182e08abd24
Author: Brian Foster <bfoster@redhat.com>
Date:   Thu Aug 10 12:55:59 2023 -0400

    ext4: drop dio overwrite only flag and associated warning

    commit 194505b55dd7899da114a4d47825204eefc0fff5 upstream.

    The commit referenced below opened up concurrent unaligned dio under
    shared locking for pure overwrites. In doing so, it enabled use of
    the IOMAP_DIO_OVERWRITE_ONLY flag and added a warning on unexpected
    -EAGAIN returns as an extra precaution, since ext4 does not retry
    writes in such cases. The flag itself is advisory in this case since
    ext4 checks for unaligned I/Os and uses appropriate locking up
    front, rather than on a retry in response to -EAGAIN.

    As it turns out, the warning check is susceptible to false positives
    because there are scenarios where -EAGAIN can be expected from lower
    layers without necessarily having IOCB_NOWAIT set on the iocb. For
    example, one instance of the warning has been seen where io_uring
    sets IOCB_HIPRI, which in turn results in REQ_POLLED|REQ_NOWAIT on
    the bio. This results in -EAGAIN if the block layer is unable to
    allocate a request, etc. [Note that there is an outstanding patch to
    untangle REQ_POLLED and REQ_NOWAIT such that the latter relies on
    IOCB_NOWAIT, which would also address this instance of the warning.]

    Another instance of the warning has been reproduced by syzbot. A dio
    write is interrupted down in __get_user_pages_locked() waiting on
    the mm lock and returns -EAGAIN up the stack. If the iomap dio
    iteration layer has made no progress on the write to this point,
    -EAGAIN returns up to the filesystem and triggers the warning.

    This use of the overwrite flag in ext4 is precautionary and
    half-baked. I.e., ext4 doesn't actually implement overwrite checking
    in the iomap callbacks when the flag is set, so the only extra
    verification it provides are i_size checks in the generic iomap dio
```

```
    layer. Combined with the tendency for false positives, the added
    verification is not worth the extra trouble. Remove the flag,
    associated warning, and update the comments to document when
    concurrent unaligned dio writes are allowed and why said flag is not
    used.

    Cc: stable@kernel.org
    Reported-by: syzbot+5050ad0fb47527b1808a@syzkaller.appspotmail.com
    Reported-by: Pengfei Xu <pengfei.xu@intel.com>
    Fixes: 310ee0902b8d ("ext4: allow concurrent unaligned dio overwrites")
    Signed-off-by: Brian Foster <bfoster@redhat.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230810165559.946222-1-bfoster@redhat.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 36daf050be3f6f067631dc52054de2d3b7cc849f
Author: Luís Henriques <lhenriques@suse.de>
Date:   Thu Aug 3 10:17:13 2023 +0100

    ext4: fix memory leaks in ext4_fname_{setup_filename,prepare_lookup}

    commit 7ca4b085f430f3774c3838b3da569ceccd6a0177 upstream.

    If the filename casefolding fails, we'll be leaking memory from the
    fscrypt_name struct, namely from the 'crypto_buf.name' member.

    Make sure we free it in the error path on both ext4_fname_setup_filename()
    and ext4_fname_prepare_lookup() functions.

    Cc: stable@kernel.org
    Fixes: 1ae98e295fa2 ("ext4: optimize match for casefolded encrypted dirs")
    Signed-off-by: Luís Henriques <lhenriques@suse.de>
    Reviewed-by: Eric Biggers <ebiggers@google.com>
    Link: https://lore.kernel.org/r/20230803091713.13239-1-lhenriques@suse.de
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8903b3c300bd6b5c43d21002d9ada3382409759a
Author: Wang Jianjian <wangjianjian0@foxmail.com>
Date:   Thu Aug 3 00:28:39 2023 +0800

    ext4: add correct group descriptors and reserved GDT blocks to system zone

    commit 68228da51c9a436872a4ef4b5a7692e29f7e5bc7 upstream.

    When setup_system_zone, flex_bg is not initialized so it is always 1.
    Use a new helper function, ext4_num_base_meta_blocks() which does not
    depend on sbi->s_log_groups_per_flex being initialized.

    [ Squashed two patches in the Link URL's below together into a single
      commit, which is simpler to review/understand.  Also fix checkpatch
      warnings. --TYT ]

    Cc: stable@kernel.org
    Signed-off-by: Wang Jianjian <wangjianjian0@foxmail.com>
    Link: https://lore.kernel.org/r/tencent_21AF0D446A9916ED5C51492CC6C9A0A77B05@qq.com
    Link: https://lore.kernel.org/r/tencent_D744D1450CC169AEA77FCF0A64719909ED05@qq.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c15bf3330a9e3c01b23e59899a6a02432a62ddc3
Author: Baokun Li <libaokun1@huawei.com>
Date:   Tue Aug 15 15:08:08 2023 +0800
```

ext4: fix slab-use-after-free in ext4_es_insert_extent()

commit 768d612f79822d30a1e7d132a4d4b05337ce42ec upstream.

Yikebaer reported an issue:

```
==================================================================
BUG: KASAN: slab-use-after-free in ext4_es_insert_extent+0xc68/0xcb0
fs/ext4/extents_status.c:894
Read of size 4 at addr ffff888112ecc1a4 by task syz-executor/8438

CPU: 1 PID: 8438 Comm: syz-executor Not tainted 6.5.0-rc5 #1
Call Trace:
 [...]
 kasan_report+0xba/0xf0 mm/kasan/report.c:588
 ext4_es_insert_extent+0xc68/0xcb0 fs/ext4/extents_status.c:894
 ext4_map_blocks+0x92a/0x16f0 fs/ext4/inode.c:680
 ext4_alloc_file_blocks.isra.0+0x2df/0xb70 fs/ext4/extents.c:4462
 ext4_zero_range fs/ext4/extents.c:4622 [inline]
 ext4_fallocate+0x251c/0x3ce0 fs/ext4/extents.c:4721
 [...]

Allocated by task 8438:
 [...]
 kmem_cache_zalloc include/linux/slab.h:693 [inline]
 __es_alloc_extent fs/ext4/extents_status.c:469 [inline]
 ext4_es_insert_extent+0x672/0xcb0 fs/ext4/extents_status.c:873
 ext4_map_blocks+0x92a/0x16f0 fs/ext4/inode.c:680
 ext4_alloc_file_blocks.isra.0+0x2df/0xb70 fs/ext4/extents.c:4462
 ext4_zero_range fs/ext4/extents.c:4622 [inline]
 ext4_fallocate+0x251c/0x3ce0 fs/ext4/extents.c:4721
 [...]

Freed by task 8438:
 [...]
 kmem_cache_free+0xec/0x490 mm/slub.c:3823
 ext4_es_try_to_merge_right fs/ext4/extents_status.c:593 [inline]
 __es_insert_extent+0x9f4/0x1440 fs/ext4/extents_status.c:802
 ext4_es_insert_extent+0x2ca/0xcb0 fs/ext4/extents_status.c:882
 ext4_map_blocks+0x92a/0x16f0 fs/ext4/inode.c:680
 ext4_alloc_file_blocks.isra.0+0x2df/0xb70 fs/ext4/extents.c:4462
 ext4_zero_range fs/ext4/extents.c:4622 [inline]
 ext4_fallocate+0x251c/0x3ce0 fs/ext4/extents.c:4721
 [...]
==================================================================
```

The flow of issue triggering is as follows:

1. remove es

```
      raw es               es  removed  es1
|------------------| -> |----|.......|------|
```

2. insert es

```
  es   insert   es1      merge with es  es1     merge with es and free es1
|----|.......|------| -> |------------|------| -> |-------------------|
```

es merges with newes, then merges with es1, frees es1, then determines
if es1->es_len is 0 and triggers a UAF.

The code flow is as follows:

```
ext4_es_insert_extent
  es1 = __es_alloc_extent(true);
  es2 = __es_alloc_extent(true);
  __es_remove_extent(inode, lblk, end, NULL, es1)
    __es_insert_extent(inode, &newes, es1) ---> insert es1 to es tree
  __es_insert_extent(inode, &newes, es2)
    ext4_es_try_to_merge_right
```

```
           ext4_es_free_extent(inode, es1) --->  es1 is freed
       if (es1 && !es1->es_len)
         // Trigger UAF by determining if es1 is used.
```

    We determine whether es1 or es2 is used immediately after calling
    __es_remove_extent() or __es_insert_extent() to avoid triggering a
    UAF if es1 or es2 is freed.

    Reported-by: Yikebaer Aizezi <yikebaer61@gmail.com>
    Closes:
https://lore.kernel.org/lkml/CALcu4raD4h9coiyEBL4Bm0zjDwxC2CyPiTwsP3zFuhot6y9Beg@mail.gmail.
com
    Fixes: 2a69c450083d ("ext4: using nofail preallocation in ext4_es_insert_extent()")
    Cc: stable@kernel.org
    Signed-off-by: Baokun Li <libaokun1@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230815070808.3377171-1-libaokun1@huawei.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 05874c69576c8cf0d67efb5e2bd329e0c7536f23
Author: Zhang Yi <yi.zhang@huawei.com>
Date: Mon Jun 26 15:33:22 2023 +0800

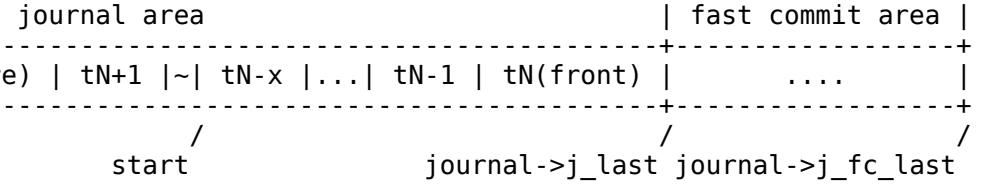
    jbd2: correct the end of the journal recovery scan range

    commit 2dfba3bb40ad8536b9fa802364f2d40da31aa88e upstream.

    We got a filesystem inconsistency issue below while running generic/475
    I/O failure pressure test with fast_commit feature enabled.

     Symlink /p3/d3/d1c/d6c/dd6/dce/l101 (inode #132605) is invalid.

    If fast_commit feature is enabled, a special fast_commit journal area is
    appended to the end of the normal journal area. The journal->j_last
    point to the first unused block behind the normal journal area instead
    of the whole log area, and the journal->j_fc_last point to the first
    unused block behind the fast_commit journal area. While doing journal
    recovery, do_one_pass(PASS_SCAN) should first scan the normal journal
    area and turn around to the first block once it meet journal->j_last,
    but the wrap() macro misuse the journal->j_fc_last, so the recovering
    could not read the next magic block (commit block perhaps) and would end
    early mistakenly and missing tN and every transaction after it in the
    following example. Finally, it could lead to filesystem inconsistency.



    This patch fix it by use the correct ending journal->j_last.

    Fixes: 5b849b5f96b4 ("jbd2: fast commit recovery path")
    Cc: stable@kernel.org
    Reported-by: Theodore Ts'o <tytso@mit.edu>
    Link: https://lore.kernel.org/linux-ext4/20230613043120.GB1584772@mit.edu/
    Signed-off-by: Zhang Yi <yi.zhang@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230626073322.3956567-1-yi.zhang@huaweicloud.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2298f2589903a8bc03061b54b31fd97985ab6529

```
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Fri Jul 14 10:55:27 2023 +0800

    jbd2: check 'jh->b_transaction' before removing it from checkpoint

    commit 590a809ff743e7bd890ba5fb36bc38e20a36de53 upstream.

    Following process will corrupt ext4 image:
    Step 1:
    jbd2_journal_commit_transaction
     __jbd2_journal_insert_checkpoint(jh, commit_transaction)
     // Put jh into trans1->t_checkpoint_list
     journal->j_checkpoint_transactions = commit_transaction
     // Put trans1 into journal->j_checkpoint_transactions

    Step 2:
    do_get_write_access
     test_clear_buffer_dirty(bh) // clear buffer dirty，set jbd dirty
     __jbd2_journal_file_buffer(jh, transaction) // jh belongs to trans2

    Step 3:
    drop_cache
     journal_shrink_one_cp_list
      jbd2_journal_try_remove_checkpoint
       if (!trylock_buffer(bh))  // lock bh, true
       if (buffer_dirty(bh))     // buffer is not dirty
       __jbd2_journal_remove_checkpoint(jh)
       // remove jh from trans1->t_checkpoint_list

    Step 4:
    jbd2_log_do_checkpoint
     trans1 = journal->j_checkpoint_transactions
     // jh is not in trans1->t_checkpoint_list
     jbd2_cleanup_journal_tail(journal)  // trans1 is done

    Step 5: Power cut, trans2 is not committed, jh is lost in next mounting.

    Fix it by checking 'jh->b_transaction' before remove it from checkpoint.

    Cc: stable@kernel.org
    Fixes: 46f881b5b175 ("jbd2: fix a race when checking checkpoint buffer busy")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Zhang Yi <yi.zhang@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230714025528.564988-3-yi.zhang@huaweicloud.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8313c4a44631c8f1dfb9d3ad5e1127120287c68b
Author: Zhang Yi <yi.zhang@huawei.com>
Date:   Fri Jul 14 10:55:26 2023 +0800

    jbd2: fix checkpoint cleanup performance regression

    commit 373ac521799d9e97061515aca6ec6621789036bb upstream.

    journal_clean_one_cp_list() has been merged into
    journal_shrink_one_cp_list(), but do chekpoint buffer cleanup from the
    committing process is just a best effort, it should stop scan once it
    meet a busy buffer, or else it will cause a lot of invalid buffer scan
    and checks. We catch a performance regression when doing fs_mark tests
    below.

    Test cmd:
     ./fs_mark  -d  scratch  -s  1024  -n  10000  -t  1  -D  100  -N  100
```

```
    Before merging checkpoint buffer cleanup:
     FSUse%        Count         Size    Files/sec     App Overhead
         95        10000         1024       8304.9            49033

    After merging checkpoint buffer cleanup:
     FSUse%        Count         Size    Files/sec     App Overhead
         95        10000         1024       7649.0            50012
     FSUse%        Count         Size    Files/sec     App Overhead
         95        10000         1024       2107.1            50871

    After merging checkpoint buffer cleanup, the total loop count in
    journal_shrink_one_cp_list() could be up to 6,261,600+ (50,000+ ~
    100,000+ in general), most of them are invalid. This patch fix it
    through passing 'shrink_type' into journal_shrink_one_cp_list() and add
    a new 'SHRINK_BUSY_STOP' to indicate it should stop once meet a busy
    buffer. After fix, the loop count descending back to 10,000+.

    After this fix:
     FSUse%        Count         Size    Files/sec     App Overhead
         95        10000         1024       8558.4            49109

    Cc: stable@kernel.org
    Fixes: b98dba273a0e ("jbd2: remove journal_clean_one_cp_list()")
    Signed-off-by: Zhang Yi <yi.zhang@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230714025528.564988-2-yi.zhang@huaweicloud.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d8061ff566c22671a75b3be38fd568d97bcc9c1d
Author: Ekansh Gupta <quic_ekangupt@quicinc.com>
Date:   Fri Aug 11 12:56:42 2023 +0100

    misc: fastrpc: Fix incorrect DMA mapping unmap request

    commit a2cb9cd6a3949a3804ad9fd7da234892ce6719ec upstream.

    Scatterlist table is obtained during map create request and the same
    table is used for DMA mapping unmap. In case there is any failure
    while getting the sg_table, ERR_PTR is returned instead of sg_table.

    When the map is getting freed, there is only a non-NULL check of
    sg_table which will also be true in case failure was returned instead
    of sg_table. This would result in improper unmap request. Add proper
    check before setting map table to avoid bad unmap request.

    Fixes: c68cfb718c8f ("misc: fastrpc: Add support for context Invoke method")
    Cc: stable <stable@kernel.org>
    Signed-off-by: Ekansh Gupta <quic_ekangupt@quicinc.com>
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Link: https://lore.kernel.org/r/20230811115643.38578-3-srinivas.kandagatla@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d559064318e766d4c66775688fd6e4389d605eb
Author: Ekansh Gupta <quic_ekangupt@quicinc.com>
Date:   Fri Aug 11 12:56:41 2023 +0100

    misc: fastrpc: Fix remote heap allocation request

    commit ada6c2d99aedd1eac2f633d03c652e070bc2ea74 upstream.

    Remote heap is used by DSP audioPD on need basis. This memory is
    allocated from reserved CMA memory region and is then shared with
    audioPD to use it for it's functionality.
```

```
    Current implementation of remote heap is not allocating the memory
    from CMA region, instead it is allocating the memory from SMMU
    context bank. The arguments passed to scm call for the reassignment
    of ownership is also not correct. Added changes to allocate CMA
    memory and have a proper ownership reassignment.

    Fixes: 532ad70c6d44 ("misc: fastrpc: Add mmap request assigning for static PD pool")
    Cc: stable <stable@kernel.org>
    Tested-by: Ekansh Gupta <quic_ekangupt@quicinc.com>
    Signed-off-by: Ekansh Gupta <quic_ekangupt@quicinc.com>
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Link: https://lore.kernel.org/r/20230811115643.38578-2-srinivas.kandagatla@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f75ef413b99c201d74dfbca5168c9f81c138908f
Author: Hien Huynh <hien.huynh.px@renesas.com>
Date:   Thu Jul 6 12:21:50 2023 +0100

    dmaengine: sh: rz-dmac: Fix destination and source data size setting

    commit c6ec8c83a29fb3aec3efa6fabbf5344498f57c7f upstream.

    Before setting DDS and SDS values, we need to clear its value first
    otherwise, we get incorrect results when we change/update the DMA bus
    width several times due to the 'OR' expression.

    Fixes: 5000d37042a6 ("dmaengine: sh: Add DMAC driver for RZ/G2L SoC")
    Cc: stable@kernel.org
    Signed-off-by: Hien Huynh <hien.huynh.px@renesas.com>
    Signed-off-by: Biju Das <biju.das.jz@bp.renesas.com>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Link: https://lore.kernel.org/r/20230706112150.198941-3-biju.das.jz@bp.renesas.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit da86cf1974b8926e46e89e07f18c7338f4ad783c
Author: Walter Chang <walter.chang@mediatek.com>
Date:   Mon Jul 17 17:07:34 2023 +0800

    clocksource/drivers/arm_arch_timer: Disable timer before programming CVAL

    commit e7d65e40ab5a5940785c5922f317602d0268caaf upstream.

    Due to the fact that the use of `writeq_relaxed()` to program CVAL is
    not guaranteed to be atomic, it is necessary to disable the timer before
    programming CVAL.

    However, if the MMIO timer is already enabled and has not yet expired,
    there is a possibility of unexpected behavior occurring: when the CPU
    enters the idle state during this period, and if the CPU's local event
    is earlier than the broadcast event, the following process occurs:

    tick_broadcast_enter()
      tick_broadcast_oneshot_control(TICK_BROADCAST_ENTER)
        __tick_broadcast_oneshot_control()
          ___tick_broadcast_oneshot_control()
            tick_broadcast_set_event()
              clockevents_program_event()
                set_next_event_mem()

    During this process, the MMIO timer remains enabled while programming
    CVAL. To prevent such behavior, disable timer explicitly prior to
    programming CVAL.
```

```
    Fixes: 8b82c4f883a7 ("clocksource/drivers/arm_arch_timer: Move MMIO timer programming
over to CVAL")
    Cc: stable@vger.kernel.org
    Signed-off-by: Walter Chang <walter.chang@mediatek.com>
    Acked-by: Marc Zyngier <maz@kernel.org>
    Reviewed-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Signed-off-by: Daniel Lezcano <daniel.lezcano@linaro.org>
    Link: https://lore.kernel.org/r/20230717090735.19370-1-walter.chang@mediatek.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ba5f2f952e6e45e4c31b12fee463324890537ff5
Author: Pavel Kozlov <pavel.kozlov@synopsys.com>
Date:   Tue Aug 15 19:11:36 2023 +0400

    ARC: atomics: Add compiler barrier to atomic operations...

    commit 42f51fb24fd39cc547c086ab3d8a314cc603a91c upstream.

    ... to avoid unwanted gcc optimizations

    SMP kernels fail to boot with commit 596ff4a09b89
    ("cpumask: re-introduce constant-sized cpumask optimizations").

    |
    | percpu: BUG: failure at mm/percpu.c:2981/pcpu_build_alloc_info()!
    |

    The write operation performed by the SCOND instruction in the atomic
    inline asm code is not properly passed to the compiler. The compiler
    cannot correctly optimize a nested loop that runs through the cpumask
    in the pcpu_build_alloc_info() function.

    Fix this by add a compiler barrier (memory clobber in inline asm).

    Apparently atomic ops used to have memory clobber implicitly via
    surrounding smp_mb(). However commit b64be6836993c431e
    ("ARC: atomics: implement relaxed variants") removed the smp_mb() for
    the relaxed variants, but failed to add the explicit compiler barrier.

    Link: https://github.com/foss-for-synopsys-dwc-arc-processors/linux/issues/135
    Cc: <stable@vger.kernel.org> # v6.3+
    Fixes: b64be6836993c43 ("ARC: atomics: implement relaxed variants")
    Signed-off-by: Pavel Kozlov <pavel.kozlov@synopsys.com>
    Signed-off-by: Vineet Gupta <vgupta@kernel.org>
    [vgupta: tweaked the changelog and added Fixes tag]
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 97ea0ac092cdfbf52b20e1bdac9065680ca015e3
Author: Fangzhi Zuo <jerry.zuo@amd.com>
Date:   Thu Jul 20 12:04:39 2023 -0400

    drm/amd/display: Temporary Disable MST DP Colorspace Property

    commit 69a959610229ec31b534eaa5f6ec75965f321bed upstream.

    Create MST colorsapce property for downstream device would trigger
    warning message "RIP: 0010:drm_mode_object_add+0x8e/0xa0 [drm]"

    After driver is loaded and drm device is registered, create
    dp colorspace property triggers warning storm at
    WARN_ON(!dev->driver->load && dev->registered && !obj_free_cb);

    Temporary disabling MST dp colorspace property for now.

    Reviewed-by: Aurabindo Pillai <aurabindo.pillai@amd.com>
```

```
    Acked-by: Aurabindo Pillai <aurabindo.pillai@amd.com>
    Signed-off-by: Fangzhi Zuo <jerry.zuo@amd.com>
    Tested-by: Daniel Wheeler <daniel.wheeler@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: Mario Limonciello <mario.limonciello@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ef035adf6c1f096ea0976380cc7388d78cb2ebb1
Author: Florent CARLI <fcarli@gmail.com>
Date:   Fri Jul 21 10:13:47 2023 +0200

    watchdog: advantech_ec_wdt: fix Kconfig dependencies

    commit 6eb28a38f6478a650c7e76b2d6910669615d8a62 upstream.

    This driver uses the WATCHDOG_CORE framework and ISA_BUS_API.
    This commit has these dependencies correctly selected.

    Signed-off-by: Florent CARLI <fcarli@gmail.com>
    Co-authored-by: Yoann Congal <yoann.congal@smile.fr>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20230721081347.52069-1-fcarli@gmail.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Cc: Yoann Congal <yoann.congal@smile.fr>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 53092c767eef50dd42c0a74abcf77ac466589b3f
Author: Masahiro Yamada <masahiroy@kernel.org>
Date:   Wed Sep 6 03:46:57 2023 +0900

    linux/export: fix reference to exported functions for parisc64

    commit 08700ec705043eb0cee01b35cf5b9d63f0230d12 upstream.

    John David Anglin reported parisc has been broken since commit
    ddb5cdbafaaa ("kbuild: generate KSYMTAB entries by modpost").

    Like ia64, parisc64 uses a function descriptor. The function
    references must be prefixed with P%.

    Also, symbols prefixed $$ from the library have the symbol type
    STT_LOPROC instead of STT_FUNC. They should be handled as functions
    too.

    Fixes: ddb5cdbafaaa ("kbuild: generate KSYMTAB entries by modpost")
    Reported-by: John David Anglin <dave.anglin@bell.net>
    Tested-by: John David Anglin <dave.anglin@bell.net>
    Tested-by: Helge Deller <deller@gmx.de>
    Closes: https://lore.kernel.org/linux-parisc/1901598a-e11d-f7dd-a5d9-
9a69d06e6b6e@bell.net/T/#u
    Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
    Signed-off-by: Helge Deller <deller@gmx.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 610dbd8ac271aa36080aac50b928d700ee3fe4de
Author: Duoming Zhou <duoming@zju.edu.cn>
Date:   Wed Aug 2 11:37:37 2023 +0800

    sh: push-switch: Reorder cleanup operations to avoid use-after-free bug

    [ Upstream commit 246f80a0b17f8f582b2c0996db02998239057c65 ]

    The original code puts flush_work() before timer_shutdown_sync()
    in switch_drv_remove(). Although we use flush_work() to stop
```

```
    the worker, it could be rescheduled in switch_timer(). As a result,
    a use-after-free bug can occur. The details are shown below:

          (cpu 0)                    |      (cpu 1)
    switch_drv_remove()              |
     flush_work()                    |
      ...                            |  switch_timer // timer
                                     |   schedule_work(&psw->work)
     timer_shutdown_sync()           |
     ...                             |  switch_work_handler // worker
     kfree(psw) // free              |
                                     |   psw->state = 0 // use

    This patch puts timer_shutdown_sync() before flush_work() to
    mitigate the bugs. As a result, the worker and timer will be
    stopped safely before the deallocate operations.

    Fixes: 9f5e8eee5cfe ("sh: generic push-switch framework.")
    Signed-off-by: Duoming Zhou <duoming@zju.edu.cn>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
    Link: https://lore.kernel.org/r/20230802033737.9738-1-duoming@zju.edu.cn
    Signed-off-by: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e7c5d573d05fdf0cdc3b423aa06a10acc37d30d6
Author: Petr Tesarik <petr.tesarik.ext@huawei.com>
Date:   Mon Jul 24 14:07:42 2023 +0200

    sh: boards: Fix CEU buffer size passed to dma_declare_coherent_memory()

    [ Upstream commit fb60211f377b69acffead3147578f86d0092a7a5 ]

    In all these cases, the last argument to dma_declare_coherent_memory() is
    the buffer end address, but the expected value should be the size of the
    reserved region.

    Fixes: 39fb993038e1 ("media: arch: sh: ap325rxa: Use new renesas-ceu camera driver")
    Fixes: c2f9b05fd5c1 ("media: arch: sh: ecovec: Use new renesas-ceu camera driver")
    Fixes: f3590dc32974 ("media: arch: sh: kfr2r09: Use new renesas-ceu camera driver")
    Fixes: 186c446f4b84 ("media: arch: sh: migor: Use new renesas-ceu camera driver")
    Fixes: 1a3c230b4151 ("media: arch: sh: ms7724se: Use new renesas-ceu camera driver")
    Signed-off-by: Petr Tesarik <petr.tesarik.ext@huawei.com>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Jacopo Mondi <jacopo.mondi@ideasonboard.com>
    Reviewed-by: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
    Reviewed-by: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
    Link: https://lore.kernel.org/r/20230724120742.2187-1-petrtesarik@huaweicloud.com
    Signed-off-by: John Paul Adrian Glaubitz <glaubitz@physik.fu-berlin.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a59e2ea0139f31c215a78fa232eed7f8b3f99bc9
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Wed Sep 6 17:16:09 2023 +0300

    net: enetc: distinguish error from valid pointers in enetc_fixup_clear_rss_rfs()

    [ Upstream commit 1b36955cc048c8ff6ba448dbf4be0e52f59f2963 ]

    enetc_psi_create() returns an ERR_PTR() or a valid station interface
    pointer, but checking for the non-NULL quality of the return code blurs
    that difference away. So if enetc_psi_create() fails, we call
    enetc_psi_destroy() when we shouldn't. This will likely result in
    crashes, since enetc_psi_create() cleans up everything after itself when
    it returns an ERR_PTR().
```

```
    Fixes: f0168042a212 ("net: enetc: reimplement RFS/RSS memory clearing as PCI quirk")
    Reported-by: Dan Carpenter <dan.carpenter@linaro.org>
    Closes: https://lore.kernel.org/netdev/582183ef-e03b-402b-8e2d-
6d9bb3c83bd9@moroto.mountain/
    Suggested-by: Dan Carpenter <dan.carpenter@linaro.org>
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Link: https://lore.kernel.org/r/20230906141609.247579-1-vladimir.oltean@nxp.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9397be66f278dc5557665c9c7cf07d0548b10472
Author: Jie Wang <wangjie125@huawei.com>
Date:   Wed Sep 6 15:20:18 2023 +0800

    net: hns3: remove GSO partial feature bit

    [ Upstream commit 60326634f6c54528778de18bfef1e8a7a93b3771 ]

    HNS3 NIC does not support GSO partial packets segmentation. Actually tunnel
    packets for example NvGRE packets segment offload and checksum offload is
    already supported. There is no need to keep gso partial feature bit. So
    this patch removes it.

    Fixes: 76ad4f0ee747 ("net: hns3: Add support of HNS3 Ethernet Driver for hip08 SoC")
    Signed-off-by: Jie Wang <wangjie125@huawei.com>
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a483057efc600f3b18351c4ed3c1e02558aea57
Author: Yisen Zhuang <yisen.zhuang@huawei.com>
Date:   Wed Sep 6 15:20:17 2023 +0800

    net: hns3: fix the port information display when sfp is absent

    [ Upstream commit 674d9591a32d01df75d6b5fffed4ef942a294376 ]

    When sfp is absent or unidentified, the port type should be
    displayed as PORT_OTHERS, rather than PORT_FIBRE.

    Fixes: 88d10bd6f730 ("net: hns3: add support for multiple media type")
    Signed-off-by: Yisen Zhuang <yisen.zhuang@huawei.com>
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7627e29561d4de15c66beb8e2400418e38a36f02
Author: Jijie Shao <shaojijie@huawei.com>
Date:   Wed Sep 6 15:20:16 2023 +0800

    net: hns3: fix invalid mutex between tc qdisc and dcb ets command issue

    [ Upstream commit fa5564945f7d15ae2390b00c08b6abaef0165cda ]

    We hope that tc qdisc and dcb ets commands can not be used crosswise.
    If we want to use any of the commands to configure tc,
    We must use the other command to clear the existing configuration.

    However, when we configure a single tc with tc qdisc,
    we can still configure it with dcb ets.
    Because we use mqprio_active as the tag of tc qdisc configuration,
    but with dcb ets, we do not check mqprio_active.
```

```
    This patch fix this issue by check mqprio_active before
    executing the dcb ets command. and add dcb_ets_active to
    replace HCLGE_FLAG_DCB_ENABLE and HCLGE_FLAG_MQPRIO_ENABLE
    at the hclge layer,

    Fixes: cacde272dd00 ("net: hns3: Add hclge_dcb module for the support of DCB feature")
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 435abe129e7504380ea539eae3fd24635e7e966e
Author: Hao Chen <chenhao418@huawei.com>
Date:   Wed Sep 6 15:20:15 2023 +0800

    net: hns3: fix debugfs concurrency issue between kfree buffer and read

    [ Upstream commit c295160b1d95e885f1af4586a221cb221d232d10 ]

    Now in hns3_dbg_uninit(), there may be concurrency between
    kfree buffer and read, it may result in memory error.

    Moving debugfs_remove_recursive() in front of kfree buffer to ensure
    they don't happen at the same time.

    Fixes: 5e69ea7ee2a6 ("net: hns3: refactor the debugfs process")
    Signed-off-by: Hao Chen <chenhao418@huawei.com>
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 83210404caa947746f8d6a20a64eb7a2a7fa647b
Author: Hao Chen <chenhao418@huawei.com>
Date:   Wed Sep 6 15:20:14 2023 +0800

    net: hns3: fix byte order conversion issue in hclge_dbg_fd_tcam_read()

    [ Upstream commit efccf655e99b6907ca07a466924e91805892e7d3 ]

    req1->tcam_data is defined as "u8 tcam_data[8]", and we convert it as
    (u32 *) without considerring byte order conversion,
    it may result in printing wrong data for tcam_data.

    Convert tcam_data to (__le32 *) first to fix it.

    Fixes: b5a0b70d77b9 ("net: hns3: refactor dump fd tcam of debugfs")
    Signed-off-by: Hao Chen <chenhao418@huawei.com>
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e61362ee8419d0edcada0b1298e065eea876127b
Author: Jian Shen <shenjian15@huawei.com>
Date:   Wed Sep 6 15:20:12 2023 +0800

    net: hns3: fix tx timeout issue

    [ Upstream commit 61a1deacc3d4fd3d57d7fda4d935f7f7503e8440 ]

    Currently, the driver knocks the ring doorbell before updating
    the ring->last_to_use in tx flow. if the hardware transmiting
    packet and napi poll scheduling are fast enough, it may get
    the old ring->last_to_use in drivers' napi poll.
    In this case, the driver will think the tx is not completed, and
    return directly without clear the flag __QUEUE_STATE_STACK_XOFF,
    which may cause tx timeout.
```

```
    Fixes: 20d06ca2679c ("net: hns3: optimize the tx clean process")
    Signed-off-by: Jian Shen <shenjian15@huawei.com>
    Signed-off-by: Jijie Shao <shaojijie@huawei.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 68cec6dcfda69a95564ccd0408dfa6642983e046
Author: Lukasz Majewski <lukma@denx.de>
Date:   Tue Sep 5 11:33:15 2023 +0200

    net: phy: Provide Module 4 KSZ9477 errata (DS80000754C)

    [ Upstream commit 08c6d8bae48c2c28f7017d7b61b5d5a1518ceb39 ]

    The KSZ9477 errata points out (in 'Module 4') the link up/down problems
    when EEE (Energy Efficient Ethernet) is enabled in the device to which
    the KSZ9477 tries to auto negotiate.

    The suggested workaround is to clear advertisement of EEE for PHYs in
    this chip driver.

    To avoid regressions with other switch ICs the new MICREL_NO_EEE flag
    has been introduced.

    Moreover, the in-register disablement of MMD_DEVICE_ID_EEE_ADV.MMD_EEE_ADV
    MMD register is removed, as this code is both; now executed too late
    (after previous rework of the PHY and DSA for KSZ switches) and not
    required as setting all members of eee_broken_modes bit field prevents
    the KSZ9477 from advertising EEE.

    Fixes: 69d3b36ca045 ("net: dsa: microchip: enable EEE support") # for KSZ9477
    Signed-off-by: Lukasz Majewski <lukma@denx.de>
    Tested-by: Oleksij Rempel <o.rempel@pengutronix.de> # Confirmed disabled EEE with
oscilloscope.
    Reviewed-by: Oleksij Rempel <o.rempel@pengutronix.de>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Link: https://lore.kernel.org/r/20230905093315.784052-1-lukma@denx.de
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ef81e4557d051cbd5de7a95bf3c91a3e9649feaf
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Wed Sep 6 11:42:02 2023 +0200

    netfilter: nf_tables: Unbreak audit log reset

    [ Upstream commit 9b5ba5c9c5109bf89dc64a3f4734bd125d1ce52e ]

    Deliver audit log from __nf_tables_dump_rules(), table dereference at
    the end of the table list loop might point to the list head, leading to
    this crash.

    [ 4137.407349] BUG: unable to handle page fault for address: 00000000001f3c50
    [ 4137.407357] #PF: supervisor read access in kernel mode
    [ 4137.407359] #PF: error_code(0x0000) - not-present page
    [ 4137.407360] PGD 0 P4D 0
    [ 4137.407363] Oops: 0000 [#1] PREEMPT SMP PTI
    [ 4137.407365] CPU: 4 PID: 500177 Comm: nft Not tainted 6.5.0+ #277
    [ 4137.407369] RIP: 0010:string+0x49/0xd0
    [ 4137.407374] Code: ff 77 36 45 89 d1 31 f6 49 01 f9 66 45 85 d2 75 19 eb 1e 49 39 f8
76 02 88 07 48 83 c7 01 83 c6 01 48 83 c2 01 4c 39 cf 74 07 <0f> b6 02 84 c0 75 e2 4c 89 c2
e9 58 e5 ff ff 48 c7 c0 0e b2 ff 81
    [ 4137.407377] RSP: 0018:ffff8881179737f0 EFLAGS: 00010286
    [ 4137.407379] RAX: 00000000001f2c50 RBX: ffff888117973848 RCX: ffff0a00ffffff04
```

```
    [ 4137.407380] RDX: 00000000001f3c50 RSI: 0000000000000000 RDI: 0000000000000000
    [ 4137.407381] RBP: 0000000000000000 R08: 0000000000000000 R09: 00000000ffffffff
    [ 4137.407383] R10: ffffffffffffffff R11: ffff88813584d200 R12: 0000000000000000
    [ 4137.407384] R13: ffffffffa15cf709 R14: 0000000000000000 R15: ffffffffa15cf709
    [ 4137.407385] FS:  00007fcfc18bb580(0000) GS:ffff88840e700000(0000)
knlGS:0000000000000000
    [ 4137.407387] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [ 4137.407388] CR2: 00000000001f3c50 CR3: 00000001055b2001 CR4: 00000000001706e0
    [ 4137.407390] Call Trace:
    [ 4137.407392]  <TASK>
    [ 4137.407393]  ? __die+0x1b/0x60
    [ 4137.407397]  ? page_fault_oops+0x6b/0xa0
    [ 4137.407399]  ? exc_page_fault+0x60/0x120
    [ 4137.407403]  ? asm_exc_page_fault+0x22/0x30
    [ 4137.407408]  ? string+0x49/0xd0
    [ 4137.407410]  vsnprintf+0x257/0x4f0
    [ 4137.407414]  kvasprintf+0x3e/0xb0
    [ 4137.407417]  kasprintf+0x3e/0x50
    [ 4137.407419]  nf_tables_dump_rules+0x1c0/0x360 [nf_tables]
    [ 4137.407439]  ? __alloc_skb+0xc3/0x170
    [ 4137.407442]  netlink_dump+0x170/0x330
    [ 4137.407447]  __netlink_dump_start+0x227/0x300
    [ 4137.407449]  nf_tables_getrule+0x205/0x390 [nf_tables]

    Deliver audit log only once at the end of the rule dump+reset for
    consistency with the set dump+reset.

    Ensure audit reset access to table under rcu read side lock. The table
    list iteration holds rcu read lock side, but recent audit code
    dereferences table object out of the rcu read lock side.

    Fixes: ea078ae9108e ("netfilter: nf_tables: Audit log rule reset")
    Fixes: 7e9be1124dbe ("netfilter: nf_tables: Audit log setelem reset")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Acked-by: Phil Sutter <phil@nwl.cc>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e3213ff99a355cda811b41e8dbb3472d13167a3a
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Mon Sep 4 02:14:36 2023 +0200

    netfilter: nft_set_rbtree: skip sync GC for new elements in this transaction

    [ Upstream commit 2ee52ae94baabf7ee09cf2a8d854b990dac5d0e4 ]

    New elements in this transaction might expired before such transaction
    ends. Skip sync GC for such elements otherwise commit path might walk
    over an already released object. Once transaction is finished, async GC
    will collect such expired element.

    Fixes: f6c383b8c31a ("netfilter: nf_tables: adapt set backend to use GC transaction
API")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3d0f898b80ac9b049e590b3ee6391716002da17
Author: Wander Lairson Costa <wander@redhat.com>
Date:   Fri Sep 1 10:50:20 2023 -0300

    netfilter: nfnetlink_osf: avoid OOB read

    [ Upstream commit f4f8a7803119005e87b716874bec07c751efafec ]
```

```
    The opt_num field is controlled by user mode and is not currently
    validated inside the kernel. An attacker can take advantage of this to
    trigger an OOB read and potentially leak information.

    BUG: KASAN: slab-out-of-bounds in nf_osf_match_one+0xbed/0xd10
 net/netfilter/nfnetlink_osf.c:88
    Read of size 2 at addr ffff88804bc64272 by task poc/6431

    CPU: 1 PID: 6431 Comm: poc Not tainted 6.0.0-rc4 #1
    Call Trace:
     nf_osf_match_one+0xbed/0xd10 net/netfilter/nfnetlink_osf.c:88
     nf_osf_find+0x186/0x2f0 net/netfilter/nfnetlink_osf.c:281
     nft_osf_eval+0x37f/0x590 net/netfilter/nft_osf.c:47
     expr_call_ops_eval net/netfilter/nf_tables_core.c:214
     nft_do_chain+0x2b0/0x1490 net/netfilter/nf_tables_core.c:264
     nft_do_chain_ipv4+0x17c/0x1f0 net/netfilter/nft_chain_filter.c:23
     [..]

    Also add validation to genre, subtype and version fields.

    Fixes: 11eeef41d5f6 ("netfilter: passive OS fingerprint xtables match")
    Reported-by: Lucas Leong <wmliang@infosec.exchange>
    Signed-off-by: Wander Lairson Costa <wander@redhat.com>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c8f292322ff16b9a2272a67de396c09a50e09dce
Author: Florian Westphal <fw@strlen.de>
Date:   Tue Sep 5 23:13:56 2023 +0200

    netfilter: nftables: exthdr: fix 4-byte stack OOB write

    [ Upstream commit fd94d9dadee58e09b49075240fe83423eb1dcd36 ]

    If priv->len is a multiple of 4, then dst[len / 4] can write past
    the destination array which leads to stack corruption.

    This construct is necessary to clean the remainder of the register
    in case ->len is NOT a multiple of the register size, so make it
    conditional just like nft_payload.c does.

    The bug was added in 4.1 cycle and then copied/inherited when
    tcp/sctp and ip option support was added.

    Bug reported by Zero Day Initiative project (ZDI-CAN-21950,
    ZDI-CAN-21951, ZDI-CAN-21961).

    Fixes: 49499c3e6e18 ("netfilter: nf_tables: switch registers to 32 bit addressing")
    Fixes: 935b7f643018 ("netfilter: nft_exthdr: add TCP option matching")
    Fixes: 133dc203d77d ("netfilter: nft_exthdr: Support SCTP chunks")
    Fixes: dbb5281a1f84 ("netfilter: nf_tables: add support for matching IPv4 options")
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 262d8c0dacfdefbc26714c6a6b43a4b32dc673a8
Author: Martin KaFai Lau <martin.lau@kernel.org>
Date:   Fri Sep 1 16:11:28 2023 -0700

    bpf: bpf_sk_storage: Fix the missing uncharge in sk_omem_alloc

    [ Upstream commit 55d49f750b1cb1f177fb1b00ae02cba4613bcfb7 ]

    The commit c83597fa5dc6 ("bpf: Refactor some inode/task/sk storage functions
    for reuse"), refactored the bpf_{sk,task,inode}_storage_free() into
    bpf_local_storage_unlink_nolock() which then later renamed to
```

```
    bpf_local_storage_destroy(). The commit accidentally passed the
    "bool uncharge_mem = false" argument to bpf_selem_unlink_storage_nolock()
    which then stopped the uncharge from happening to the sk->sk_omem_alloc.

    This missing uncharge only happens when the sk is going away (during
    __sk_destruct).

    This patch fixes it by always passing "uncharge_mem = true". It is a
    noop to the task/inode/cgroup storage because they do not have the
    map_local_storage_(un)charge enabled in the map_ops. A followup patch
    will be done in bpf-next to remove the uncharge_mem argument.

    A selftest is added in the next patch.

    Fixes: c83597fa5dc6 ("bpf: Refactor some inode/task/sk storage functions for reuse")
    Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Link: https://lore.kernel.org/bpf/20230901231129.578493-3-martin.lau@linux.dev
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 300415caa373a07782fcbc2f8d9429bc2dc27a47
Author: Martin KaFai Lau <martin.lau@kernel.org>
Date:   Fri Sep 1 16:11:27 2023 -0700

    bpf: bpf_sk_storage: Fix invalid wait context lockdep report

    [ Upstream commit a96a44aba556c42b432929d37d60158aca21ad4c ]

    './test_progs -t test_local_storage' reported a splat:

    [   27.137569] =============================
    [   27.138122] [ BUG: Invalid wait context ]
    [   27.138650] 6.5.0-03980-gd11ae1b16b0a #247 Tainted: G          O
    [   27.139542] -----------------------------
    [   27.140106] test_progs/1729 is trying to lock:
    [   27.140713] ffff8883ef047b88 (stock_lock){-.-.}-{3:3}, at:
local_lock_acquire+0x9/0x130
    [   27.141834] other info that might help us debug this:
    [   27.142437] context-{5:5}
    [   27.142856] 2 locks held by test_progs/1729:
    [   27.143352]  #0: ffffffff84bcd9c0 (rcu_read_lock){....}-{1:3}, at:
rcu_lock_acquire+0x4/0x40
    [   27.144492]  #1: ffff888107deb2c0 (&storage->lock){..-.}-{2:2}, at:
bpf_local_storage_update+0x39e/0x8e0
    [   27.145855] stack backtrace:
    [   27.146274] CPU: 0 PID: 1729 Comm: test_progs Tainted: G          O       6.5.0-
03980-gd11ae1b16b0a #247
    [   27.147550] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-
g155821a1990b-prebuilt.qemu.org 04/01/2014
    [   27.149127] Call Trace:
    [   27.149490]  <TASK>
    [   27.149867]  dump_stack_lvl+0x130/0x1d0
    [   27.152609]  dump_stack+0x14/0x20
    [   27.153131]  __lock_acquire+0x1657/0x2220
    [   27.153677]  lock_acquire+0x1b8/0x510
    [   27.157908]  local_lock_acquire+0x29/0x130
    [   27.159048]  obj_cgroup_charge+0xf4/0x3c0
    [   27.160794]  slab_pre_alloc_hook+0x28e/0x2b0
    [   27.161931]  __kmem_cache_alloc_node+0x51/0x210
    [   27.163557]  __kmalloc+0xaa/0x210
    [   27.164593]  bpf_map_kzalloc+0xbc/0x170
    [   27.165147]  bpf_selem_alloc+0x130/0x510
    [   27.166295]  bpf_local_storage_update+0x5aa/0x8e0
    [   27.167042]  bpf_fd_sk_storage_update_elem+0xdb/0x1a0
    [   27.169199]  bpf_map_update_value+0x415/0x4f0
```

```
     [   27.169871]  map_update_elem+0x413/0x550
     [   27.170330]  __sys_bpf+0x5e9/0x640
     [   27.174065]  __x64_sys_bpf+0x80/0x90
     [   27.174568]  do_syscall_64+0x48/0xa0
     [   27.175201]  entry_SYSCALL_64_after_hwframe+0x6e/0xd8
     [   27.175932] RIP: 0033:0x7effb40e41ad
     [   27.176357] Code: ff c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa 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 73 01 c3 48
  8b 0d8
     [   27.179028] RSP: 002b:00007ffe64c21fc8 EFLAGS: 00000202 ORIG_RAX: 0000000000000141
     [   27.180088] RAX: ffffffffffffffda RBX: 00007ffe64c22768 RCX: 00007effb40e41ad
     [   27.181082] RDX: 0000000000000020 RSI: 00007ffe64c22008 RDI: 0000000000000002
     [   27.182030] RBP: 00007ffe64c21ff0 R08: 0000000000000000 R09: 00007ffe64c22788
     [   27.183038] R10: 0000000000000064 R11: 0000000000000202 R12: 0000000000000000
     [   27.184006] R13: 00007ffe64c22788 R14: 00007effb42a1000 R15: 0000000000000000
     [   27.184958]  </TASK>

     It complains about acquiring a local_lock while holding a raw_spin_lock.
     It means it should not allocate memory while holding a raw_spin_lock
     since it is not safe for RT.

     raw_spin_lock is needed because bpf_local_storage supports tracing
     context. In particular for task local storage, it is easy to
     get a "current" task PTR_TO_BTF_ID in tracing bpf prog.
     However, task (and cgroup) local storage has already been moved to
     bpf mem allocator which can be used after raw_spin_lock.

     The splat is for the sk storage. For sk (and inode) storage,
     it has not been moved to bpf mem allocator. Using raw_spin_lock or not,
     kzalloc(GFP_ATOMIC) could theoretically be unsafe in tracing context.
     However, the local storage helper requires a verifier accepted
     sk pointer (PTR_TO_BTF_ID), it is hypothetical if that (mean running
     a bpf prog in a kzalloc unsafe context and also able to hold a verifier
     accepted sk pointer) could happen.

     This patch avoids kzalloc after raw_spin_lock to silent the splat.
     There is an existing kzalloc before the raw_spin_lock. At that point,
     a kzalloc is very likely required because a lookup has just been done
     before. Thus, this patch always does the kzalloc before acquiring
     the raw_spin_lock and remove the later kzalloc usage after the
     raw_spin_lock. After this change, it will have a charge and then
     uncharge during the syscall bpf_map_update_elem() code path.
     This patch opts for simplicity and not continue the old
     optimization to save one charge and uncharge.

     This issue is dated back to the very first commit of bpf_sk_storage
     which had been refactored multiple times to create task, inode, and
     cgroup storage. This patch uses a Fixes tag with a more recent
     commit that should be easier to do backport.

     Fixes: b00fa38a9c1c ("bpf: Enable non-atomic allocations in local storage")
     Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
     Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
     Link: https://lore.kernel.org/bpf/20230901231129.578493-2-martin.lau@linux.dev
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 6decfb1b5ae3f69e3a8a87eb4d44cb730af814d3
  Author: Ilya Leoshkevich <iii@linux.ibm.com>
  Date:   Wed Sep 6 02:44:19 2023 +0200

     s390/bpf: Pass through tail call counter in trampolines

     [ Upstream commit a192103a11465e9d517975c50f9944dc80e44d61 ]

     s390x eBPF programs use the following extension to the s390x calling
```

```
    convention: tail call counter is passed on stack at offset
    STK_OFF_TCCNT, which callees otherwise use as scratch space.

    Currently trampoline does not respect this and clobbers tail call
    counter. This breaks enforcing tail call limits in eBPF programs, which
    have trampolines attached to them.

    Fix by forwarding a copy of the tail call counter to the original eBPF
    program in the trampoline (for fexit), and by restoring it at the end
    of the trampoline (for fentry).

    Fixes: 528eb2cb87bc ("s390/bpf: Implement arch_prepare_bpf_trampoline()")
    Reported-by: Leon Hwang <hffilwlqm@gmail.com>
    Signed-off-by: Ilya Leoshkevich <iii@linux.ibm.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Link: https://lore.kernel.org/bpf/20230906004448.111674-1-iii@linux.ibm.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f66a0738c76c2ae31a8de9b29e365297fe318edd
Author: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
Date:   Wed Aug 30 10:04:05 2023 +0200

    bpf: Assign bpf_tramp_run_ctx::saved_run_ctx before recursion check.

    [ Upstream commit 6764e767f4af1e35f87f3497e1182d945de37f93 ]

    __bpf_prog_enter_recur() assigns bpf_tramp_run_ctx::saved_run_ctx before
    performing the recursion check which means in case of a recursion
    __bpf_prog_exit_recur() uses the previously set bpf_tramp_run_ctx::saved_run_ctx
    value.

    __bpf_prog_enter_sleepable_recur() assigns bpf_tramp_run_ctx::saved_run_ctx
    after the recursion check which means in case of a recursion
    __bpf_prog_exit_sleepable_recur() uses an uninitialized value. This does not
    look right. If I read the entry trampoline code right, then bpf_tramp_run_ctx
    isn't initialized upfront.

    Align __bpf_prog_enter_sleepable_recur() with __bpf_prog_enter_recur() and
    set bpf_tramp_run_ctx::saved_run_ctx before the recursion check is made.
    Remove the assignment of saved_run_ctx in kern_sys_bpf() since it happens
    a few cycles later.

    Fixes: e384c7b7b46d0 ("bpf, x86: Create bpf_tramp_run_ctx on the caller thread's stack")
    Signed-off-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Jiri Olsa <jolsa@kernel.org>
    Link: https://lore.kernel.org/bpf/20230830080405.251926-3-bigeasy@linutronix.de
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5dfee42998e873df3e57772247f28d2632d52734
Author: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
Date:   Wed Aug 30 10:04:04 2023 +0200

    bpf: Invoke __bpf_prog_exit_sleepable_recur() on recursion in kern_sys_bpf().

    [ Upstream commit 7645629f7dc88cd777f98970134bf1a54c8d77e3 ]

    If __bpf_prog_enter_sleepable_recur() detects recursion then it returns
    0 without undoing rcu_read_lock_trace(), migrate_disable() or
    decrementing the recursion counter. This is fine in the JIT case because
    the JIT code will jump in the 0 case to the end and invoke the matching
    exit trampoline (__bpf_prog_exit_sleepable_recur()).

    This is not the case in kern_sys_bpf() which returns directly to the
    caller with an error code.
```

```
    Add __bpf_prog_exit_sleepable_recur() as clean up in the recursion case.

    Fixes: b1d18a7574d0d ("bpf: Extend sys_bpf commands for bpf_syscall programs.")
    Signed-off-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Jiri Olsa <jolsa@kernel.org>
    Link: https://lore.kernel.org/bpf/20230830080405.251926-2-bigeasy@linutronix.de
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f491a7984e8d38a53c6d8c0e0a97e7750ac7105d
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Tue Sep 5 16:42:02 2023 -0700

    net: phylink: fix sphinx complaint about invalid literal

    [ Upstream commit 1a961e74d5abbea049588a3d74b759955b4ed9d5 ]

    sphinx complains about the use of "%PHYLINK_PCS_NEG_*":

    Documentation/networking/kapi:144: ./include/linux/phylink.h:601: WARNING: Inline
literal start-string without end-string.
    Documentation/networking/kapi:144: ./include/linux/phylink.h:633: WARNING: Inline
literal start-string without end-string.

    These are not valid symbols so drop the '%' prefix.

    Alternatively we could use %PHYLINK_PCS_NEG_\* (escape the *)
    or use normal literal ``PHYLINK_PCS_NEG_*`` but there is already
    a handful of un-adorned DEFINE_* in this file.

    Fixes: f99d471afa03 ("net: phylink: add PCS negotiation mode")
    Reported-by: Stephen Rothwell <sfr@canb.auug.org.au>
    Link: https://lore.kernel.org/all/20230626162908.2f149f98@canb.auug.org.au/
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Reviewed-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b8fc445b5d896cacc1406a84f217636dfb48482f
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Wed Sep 6 00:53:38 2023 +0300

    net: dsa: sja1105: complete tc-cbs offload support on SJA1110

    [ Upstream commit 180a7419fe4adc8d9c8e0ef0fd17bcdd0cf78acd ]

    The blamed commit left this delta behind:

      struct sja1105_cbs_entry {
     -      u64 port;
     -      u64 prio;
     +      u64 port; /* Not used for SJA1110 */
     +      u64 prio; /* Not used for SJA1110 */
            u64 credit_hi;
            u64 credit_lo;
            u64 send_slope;
            u64 idle_slope;
      };

    but did not actually implement tc-cbs offload fully for the new switch.
    The offload is accepted, but it doesn't work.

    The difference compared to earlier switch generations is that now, the
    table of CBS shapers is sparse, because there are many more shapers, so
```

```
    the mapping between a {port, prio} and a table index is static, rather
    than requiring us to store the port and prio into the sja1105_cbs_entry.

    So, the problem is that the code programs the CBS shaper parameters at a
    dynamic table index which is incorrect.

    All that needs to be done for SJA1110 CBS shapers to work is to bypass
    the logic which allocates shapers in a dense manner, as for SJA1105, and
    use the fixed mapping instead.

    Fixes: 3e77e59bf8cf ("net: dsa: sja1105: add support for the SJA1110 switch family")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ddf9bc72b8c5e2f09d6d64a3deaa7654317a2811
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Wed Sep 6 00:53:37 2023 +0300

    net: dsa: sja1105: fix -ENOSPC when replacing the same tc-cbs too many times

    [ Upstream commit 894cafc5c62ccced758077bd4e970dc714c42637 ]

    After running command [2] too many times in a row:

    [1] $ tc qdisc add dev sw2p0 root handle 1: mqprio num_tc 8 \
            map 0 1 2 3 4 5 6 7 queues 1@0 1@1 1@2 1@3 1@4 1@5 1@6 1@7 hw 0
    [2] $ tc qdisc replace dev sw2p0 parent 1:1 cbs offload 1 \
            idleslope 120000 sendslope -880000 locredit -1320 hicredit 180

    (aka more than priv->info->num_cbs_shapers times)

    we start seeing the following error message:

    Error: Specified device failed to setup cbs hardware offload.

    This comes from the fact that ndo_setup_tc(TC_SETUP_QDISC_CBS) presents
    the same API for the qdisc create and replace cases, and the sja1105
    driver fails to distinguish between the 2. Thus, it always thinks that
    it must allocate the same shaper for a {port, queue} pair, when it may
    instead have to replace an existing one.

    Fixes: 4d7525085a9b ("net: dsa: sja1105: offload the Credit-Based Shaper qdisc")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9d09ea4ec01b332415fa023fe3d3ea663eee3f32
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Wed Sep 6 00:53:36 2023 +0300

    net: dsa: sja1105: fix bandwidth discrepancy between tc-cbs software and offload

    [ Upstream commit 954ad9bf13c4f95a4958b5f8433301f2ab99e1f5 ]

    More careful measurement of the tc-cbs bandwidth shows that the stream
    bandwidth (effectively idleslope) increases, there is a larger and
    larger discrepancy between the rate limit obtained by the software
    Qdisc, and the rate limit obtained by its offloaded counterpart.

    The discrepancy becomes so large, that e.g. at an idleslope of 40000
    (40Mbps), the offloaded cbs does not actually rate limit anything, and
    traffic will pass at line rate through a 100 Mbps port.

    The reason for the discrepancy is that the hardware documentation I've
```

been following is incorrect. UM11040.pdf (for SJA1105P/Q/R/S) states about IDLE_SLOPE that it is "the rate (in unit of bytes/sec) at which the credit counter is increased".

Cross-checking with UM10944.pdf (for SJA1105E/T) and UM11107.pdf (for SJA1110), the wording is different: "This field specifies the value, in bytes per second times link speed, by which the credit counter is increased".

So there's an extra scaling for link speed that the driver is currently not accounting for, and apparently (empirically), that link speed is expressed in Kbps.

I've pondered whether to pollute the sja1105_mac_link_up() implementation with CBS shaper reprogramming, but I don't think it is worth it. IMO, the UAPI exposed by tc-cbs requires user space to recalculate the sendslope anyway, since the formula for that depends on port_transmit_rate (see man tc-cbs), which is not an invariant from tc's perspective.

So we use the offload->sendslope and offload->idleslope to deduce the original port_transmit_rate from the CBS formula, and use that value to scale the offload->sendslope and offload->idleslope to values that the hardware understands.

Some numerical data points:

```
 40Mbps stream, max interfering frame size 1500, port speed 100M
 ---------------------------------------------------------------

 tc-cbs parameters:
 idleslope 40000 sendslope -60000 locredit -900 hicredit 600

 which result in hardware values:

 Before (doesn't work)          After (works)
 credit_hi    600               600
 credit_lo    900               900
 send_slope   7500000           75
 idle_slope   5000000           50

 40Mbps stream, max interfering frame size 1500, port speed 1G
 -------------------------------------------------------------

 tc-cbs parameters:
 idleslope 40000 sendslope -960000 locredit -1440 hicredit 60

 which result in hardware values:

 Before (doesn't work)          After (works)
 credit_hi    60                60
 credit_lo    1440              1440
 send_slope   120000000         120
 idle_slope   5000000           5

 5.12Mbps stream, max interfering frame size 1522, port speed 100M
 -----------------------------------------------------------------

 tc-cbs parameters:
 idleslope 5120 sendslope -94880 locredit -1444 hicredit 77

 which result in hardware values:

 Before (doesn't work)          After (works)
 credit_hi    77                77
```

```
      credit_lo    1444              1444
      send_slope   11860000          118
      idle_slope   640000            6

    Tested on SJA1105T, SJA1105S and SJA1110A, at 1Gbps and 100Mbps.

    Fixes: 4d7525085a9b ("net: dsa: sja1105: offload the Credit-Based Shaper qdisc")
    Reported-by: Yanan Yang <yanan.yang@nxp.com>
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4b912846fd6877e825092a0862195bb99052e39a
Author: Bodong Wang <bodong@nvidia.com>
Date:   Tue Sep 5 10:48:46 2023 -0700

    mlx5/core: E-Switch, Create ACL FT for eswitch manager in switchdev mode

    [ Upstream commit 344134609a564f28b3cc81ca6650319ccd5d8961 ]

    ACL flow table is required in switchdev mode when metadata is enabled,
    driver creates such table when loading each vport. However, not every
    vport is loaded in switchdev mode. Such as ECPF if it's the eswitch manager.
    In this case, ACL flow table is still needed.

    To make it modularized, create ACL flow table for eswitch manager as
    default and skip such operations when loading manager vport.

    Also, there is no need to load the eswitch manager vport in switchdev mode.
    This means there is no need to load it on regular connect-x HCAs where
    the PF is the eswitch manager. This will avoid creating duplicate ACL
    flow table for host PF vport.

    Fixes: 29bcb6e4fe70 ("net/mlx5e: E-Switch, Use metadata for vport matching in send-to-
vport rules")
    Fixes: eb8e9fae0a22 ("mlx5/core: E-Switch, Allocate ECPF vport if it's an eswitch
manager")
    Fixes: 5019833d661f ("net/mlx5: E-switch, Introduce helper function to enable/disable
vports")
    Signed-off-by: Bodong Wang <bodong@nvidia.com>
    Reviewed-by: Mark Bloch <mbloch@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d5683ec3899da43fb309393e55f2b740524477c7
Author: Jiri Pirko <jiri@resnulli.us>
Date:   Thu May 25 10:01:02 2023 +0200

    net/mlx5: Push devlink port PF/VF init/cleanup calls out of
devlink_port_register/unregister()

    [ Upstream commit d9833bcfe840fff5d368b1c7c68e05c95be8d19c ]

    In order to prepare for
    mlx5_esw_offloads_devlink_port_register/unregister() to be used
    for SFs as well, push out the PF/VF specific init/cleanup calls outside.
    Introduce mlx5_eswitch_load/unload_pf_vf_vport() and call them from
    there. Use these new helpers of PF/VF loading and make
    mlx5_eswitch_local/unload_vport() reusable for SFs.

    Signed-off-by: Jiri Pirko <jiri@nvidia.com>
    Reviewed-by: Shay Drory <shayd@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Stable-dep-of: 344134609a56 ("mlx5/core: E-Switch, Create ACL FT for eswitch manager in
```

```
 switchdev mode")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 5e3e61f7eaac804a110db0002675ccc8f571740c
 Author: Jiri Pirko <jiri@resnulli.us>
 Date:   Wed May 24 17:46:47 2023 +0200

    net/mlx5: Rework devlink port alloc/free into init/cleanup

    [ Upstream commit 4c0dac1ef8abc6295a91197884f5ceb5d11c2bd9 ]

    In order to prepare the devlink port registration function to be common
    for PFs/VFs and SFs, change the existing devlink port allocation and
    free functions into PF/VF init and cleanup, so similar helpers could be
    later on introduced for SFs. Make the init/cleanup helpers responsible
    for setting/clearing the vport->dl_port pointer.

    Signed-off-by: Jiri Pirko <jiri@nvidia.com>
    Reviewed-by: Shay Drory <shayd@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Stable-dep-of: 344134609a56 ("mlx5/core: E-Switch, Create ACL FT for eswitch manager in
 switchdev mode")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit c4a9d6a5c68172c76b58a9afa48f419f4ab19b2c
 Author: Jiri Pirko <jiri@resnulli.us>
 Date:   Thu May 25 09:42:09 2023 +0200

    net/mlx5: Give esw_offloads_load/unload_rep() "mlx5_" prefix

    [ Upstream commit 9eca8bb8da4385b02bd02b6876af8d4225bf4713 ]

    As esw_offloads_load/unload_rep() are used outside eswitch.c it is nicer
    for them to have "mlx5_" prefix. Add it.

    Signed-off-by: Jiri Pirko <jiri@nvidia.com>
    Reviewed-by: Shay Drory <shayd@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Stable-dep-of: 344134609a56 ("mlx5/core: E-Switch, Create ACL FT for eswitch manager in
 switchdev mode")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit c1aece30ed884cb30c73ce12ee49c715831b391b
 Author: Jianbo Liu <jianbol@nvidia.com>
 Date:   Tue Sep 5 10:48:45 2023 -0700

    net/mlx5e: Clear mirred devices array if the rule is split

    [ Upstream commit b7558a77529fef60e7992f40fb5353fed8be0cf8 ]

    In the cited commit, the mirred devices are recorded and checked while
    parsing the actions. In order to avoid system crash, the duplicate
    action in a single rule is not allowed.

    But the rule is actually break down into several FTEs in different
    tables, for either mirroring, or the specified types of actions which
    use post action infrastructure.

    It will reject certain action list by mistake, for example:
        actions:enp8s0f0_1,set(ipv4(ttl=63)),enp8s0f0_0,enp8s0f0_1.
    Here the rule is split to two FTEs because of pedit action.

    To fix this issue, when parsing the rule actions, reset if_count to
    clear the mirred devices array if the rule is split to multiple
    FTEs, and then the duplicate checking is restarted.
```

```
    Fixes: 554fe75c1b3f ("net/mlx5e: Avoid duplicating rule destinations")
    Signed-off-by: Jianbo Liu <jianbol@nvidia.com>
    Reviewed-by: Vlad Buslov <vladbu@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b9c7805994fa706cd56b1a0ee9e701bf33b298e8
Author: Eric Dumazet <edumazet@google.com>
Date:   Tue Sep 5 13:40:46 2023 +0000

    ip_tunnels: use DEV_STATS_INC()

    [ Upstream commit 9b271ebaf9a2c5c566a54bc6cd915962e8241130 ]

    syzbot/KCSAN reported data-races in iptunnel_xmit_stats() [1]

    This can run from multiple cpus without mutual exclusion.

    Adopt SMP safe DEV_STATS_INC() to update dev->stats fields.

    [1]
    BUG: KCSAN: data-race in iptunnel_xmit / iptunnel_xmit

    read-write to 0xffff8881353df170 of 8 bytes by task 30263 on cpu 1:
    iptunnel_xmit_stats include/net/ip_tunnels.h:493 [inline]
    iptunnel_xmit+0x432/0x4a0 net/ipv4/ip_tunnel_core.c:87
    ip_tunnel_xmit+0x1477/0x1750 net/ipv4/ip_tunnel.c:831
    __gre_xmit net/ipv4/ip_gre.c:469 [inline]
    ipgre_xmit+0x516/0x570 net/ipv4/ip_gre.c:662
    __netdev_start_xmit include/linux/netdevice.h:4889 [inline]
    netdev_start_xmit include/linux/netdevice.h:4903 [inline]
    xmit_one net/core/dev.c:3544 [inline]
    dev_hard_start_xmit+0x11b/0x3f0 net/core/dev.c:3560
    __dev_queue_xmit+0xeee/0x1de0 net/core/dev.c:4340
    dev_queue_xmit include/linux/netdevice.h:3082 [inline]
    __bpf_tx_skb net/core/filter.c:2129 [inline]
    __bpf_redirect_no_mac net/core/filter.c:2159 [inline]
    __bpf_redirect+0x723/0x9c0 net/core/filter.c:2182
    ____bpf_clone_redirect net/core/filter.c:2453 [inline]
    bpf_clone_redirect+0x16c/0x1d0 net/core/filter.c:2425
    ___bpf_prog_run+0xd7d/0x41e0 kernel/bpf/core.c:1954
    __bpf_prog_run512+0x74/0xa0 kernel/bpf/core.c:2195
    bpf_dispatcher_nop_func include/linux/bpf.h:1181 [inline]
    __bpf_prog_run include/linux/filter.h:609 [inline]
    bpf_prog_run include/linux/filter.h:616 [inline]
    bpf_test_run+0x15d/0x3d0 net/bpf/test_run.c:423
    bpf_prog_test_run_skb+0x77b/0xa00 net/bpf/test_run.c:1045
    bpf_prog_test_run+0x265/0x3d0 kernel/bpf/syscall.c:3996
    __sys_bpf+0x3af/0x780 kernel/bpf/syscall.c:5353
    __do_sys_bpf kernel/bpf/syscall.c:5439 [inline]
    __se_sys_bpf kernel/bpf/syscall.c:5437 [inline]
    __x64_sys_bpf+0x43/0x50 kernel/bpf/syscall.c:5437
    do_syscall_x64 arch/x86/entry/common.c:50 [inline]
    do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80
    entry_SYSCALL_64_after_hwframe+0x63/0xcd

    read-write to 0xffff8881353df170 of 8 bytes by task 30249 on cpu 0:
    iptunnel_xmit_stats include/net/ip_tunnels.h:493 [inline]
    iptunnel_xmit+0x432/0x4a0 net/ipv4/ip_tunnel_core.c:87
    ip_tunnel_xmit+0x1477/0x1750 net/ipv4/ip_tunnel.c:831
    __gre_xmit net/ipv4/ip_gre.c:469 [inline]
    ipgre_xmit+0x516/0x570 net/ipv4/ip_gre.c:662
    __netdev_start_xmit include/linux/netdevice.h:4889 [inline]
```

```
     netdev_start_xmit include/linux/netdevice.h:4903 [inline]
     xmit_one net/core/dev.c:3544 [inline]
     dev_hard_start_xmit+0x11b/0x3f0 net/core/dev.c:3560
     __dev_queue_xmit+0xeee/0x1de0 net/core/dev.c:4340
     dev_queue_xmit include/linux/netdevice.h:3082 [inline]
     __bpf_tx_skb net/core/filter.c:2129 [inline]
     __bpf_redirect_no_mac net/core/filter.c:2159 [inline]
     __bpf_redirect+0x723/0x9c0 net/core/filter.c:2182
     ____bpf_clone_redirect net/core/filter.c:2453 [inline]
     bpf_clone_redirect+0x16c/0x1d0 net/core/filter.c:2425
     ___bpf_prog_run+0xd7d/0x41e0 kernel/bpf/core.c:1954
     __bpf_prog_run512+0x74/0xa0 kernel/bpf/core.c:2195
     bpf_dispatcher_nop_func include/linux/bpf.h:1181 [inline]
     __bpf_prog_run include/linux/filter.h:609 [inline]
     bpf_prog_run include/linux/filter.h:616 [inline]
     bpf_test_run+0x15d/0x3d0 net/bpf/test_run.c:423
     bpf_prog_test_run_skb+0x77b/0xa00 net/bpf/test_run.c:1045
     bpf_prog_test_run+0x265/0x3d0 kernel/bpf/syscall.c:3996
     __sys_bpf+0x3af/0x780 kernel/bpf/syscall.c:5353
     __do_sys_bpf kernel/bpf/syscall.c:5439 [inline]
     __se_sys_bpf kernel/bpf/syscall.c:5437 [inline]
     __x64_sys_bpf+0x43/0x50 kernel/bpf/syscall.c:5437
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x63/0xcd

     value changed: 0x0000000000018830 -> 0x0000000000018831

     Reported by Kernel Concurrency Sanitizer on:
     CPU: 0 PID: 30249 Comm: syz-executor.4 Not tainted 6.5.0-syzkaller-11704-g3f86ed6ec0b3
  #0

     Fixes: 039f50629b7f ("ip_tunnel: Move stats update to iptunnel_xmit()")
     Reported-by: syzbot <syzkaller@googlegroups.com>
     Signed-off-by: Eric Dumazet <edumazet@google.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit ed66a54bce05cba0cde30773630c193430facee9
  Author: Ariel Marcovitch <arielmarcovitch@gmail.com>
  Date:   Sat Aug 26 20:33:17 2023 +0300

     idr: fix param name in idr_alloc_cyclic() doc

     [ Upstream commit 2a15de80dd0f7e04a823291aa9eb49c5294f56af ]

     The relevant parameter is 'start' and not 'nextid'

     Fixes: 460488c58ca8 ("idr: Remove idr_alloc_ext")
     Signed-off-by: Ariel Marcovitch <arielmarcovitch@gmail.com>
     Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit e52c1698088ccbc36c5984fce6b29e3813d697b9
  Author: Jerome Neanne <jneanne@baylibre.com>
  Date:   Tue Sep 5 16:07:34 2023 +0200

     regulator: tps6594-regulator: Fix random kernel crash

     [ Upstream commit ca0e36e3e39a4e8b5a4b647dff8c5938ca6ccbec ]

     Random kernel crash detected in TI CICD when regulator driver is added.
     This is root caused to irq index increment being done twice causing
     irq_data being allocated outside of the range.
```

```
    - Rework tps6594_request_reg_irqs with correct index increment
    - Adjust irq_data kmalloc size to the exact size needed for the device

    This has been reported on TI mainline. No public bug report associated.

    Reported-by: Udit Kumar <u-kumar1@ti.com>
    Fixes: f17ccc5deb4d ("regulator: tps6594-regulator: Add driver for TI TPS6594
regulators")
    Signed-off-by: Jerome Neanne <jneanne@baylibre.com>
    Link: https://lore.kernel.org/r/20230828-tps6594_random_boot_crash_fix-v1-1-
f29cbf9ddb37@baylibre.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 174f11ef1615ec3ab1e2189685864433c0d855a2
Author: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
Date:   Thu Aug 31 13:59:59 2023 +0300

    s390/zcrypt: don't leak memory if dev_set_name() fails

    [ Upstream commit 6252f47b78031979ad919f971dc8468b893488bd ]

    When dev_set_name() fails, zcdn_create() doesn't free the newly
    allocated resources. Do it.

    Fixes: 00fab2350e6b ("s390/zcrypt: multiple zcrypt device nodes support")
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Link: https://lore.kernel.org/r/20230831110000.24279-1-andriy.shevchenko@linux.intel.com
    Signed-off-by: Harald Freudenberger <freude@linux.ibm.com>
    Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2a86ad78abe053a0109561676ea37ff6f7c0a1ca
Author: Olga Zaborska <olga.zaborska@intel.com>
Date:   Tue Jul 25 10:10:58 2023 +0200

    igb: Change IGB_MIN to allow set rx/tx value between 64 and 80

    [ Upstream commit 6319685bdc8ad5310890add907b7c42f89302886 ]

    Change the minimum value of RX/TX descriptors to 64 to enable setting the rx/tx
    value between 64 and 80. All igb devices can use as low as 64 descriptors.
    This change will unify igb with other drivers.
    Based on commit 7b1be1987c1e ("e1000e: lower ring minimum size to 64")

    Fixes: 9d5c824399de ("igb: PCI-Express 82575 Gigabit Ethernet driver")
    Signed-off-by: Olga Zaborska <olga.zaborska@intel.com>
    Tested-by: Pucha Himasekhar Reddy <himasekharx.reddy.pucha@intel.com> (A Contingent
worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a47535904c5bc22904c209f4684ca94b405af94e
Author: Olga Zaborska <olga.zaborska@intel.com>
Date:   Tue Jul 25 10:10:57 2023 +0200

    igbvf: Change IGBVF_MIN to allow set rx/tx value between 64 and 80

    [ Upstream commit 8360717524a24a421c36ef8eb512406dbd42160a ]

    Change the minimum value of RX/TX descriptors to 64 to enable setting the rx/tx
    value between 64 and 80. All igbvf devices can use as low as 64 descriptors.
    This change will unify igbvf with other drivers.
    Based on commit 7b1be1987c1e ("e1000e: lower ring minimum size to 64")
```

```
    Fixes: d4e0fe01a38a ("igbvf: add new driver to support 82576 virtual functions")
    Signed-off-by: Olga Zaborska <olga.zaborska@intel.com>
    Tested-by: Rafal Romanowski <rafal.romanowski@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3ce31d747265cbfbff972850e8a96ec81f1d218b
Author: Olga Zaborska <olga.zaborska@intel.com>
Date:   Tue Jul 25 10:10:56 2023 +0200

    igc: Change IGC_MIN to allow set rx/tx value between 64 and 80

    [ Upstream commit 5aa48279712e1f134aac908acde4df798955a955 ]

    Change the minimum value of RX/TX descriptors to 64 to enable setting the rx/tx
    value between 64 and 80. All igc devices can use as low as 64 descriptors.
    This change will unify igc with other drivers.
    Based on commit 7b1be1987c1e ("e1000e: lower ring minimum size to 64")

    Fixes: 0507ef8a0372 ("igc: Add transmit and receive fastpath and interrupt handlers")
    Signed-off-by: Olga Zaborska <olga.zaborska@intel.com>
    Tested-by: Naama Meir <naamax.meir@linux.intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7167f68d0b0e87ade7bf176f3885f5d304699d8d
Author: Geetha sowjanya <gakula@marvell.com>
Date:   Tue Sep 5 12:18:16 2023 +0530

    octeontx2-af: Fix truncation of smq in CN10K NIX AQ enqueue mbox handler

    [ Upstream commit 29fe7a1b62717d58f033009874554d99d71f7d37 ]

    The smq value used in the CN10K NIX AQ instruction enqueue mailbox
    handler was truncated to 9-bit value from 10-bit value because of
    typecasting the CN10K mbox request structure to the CN9K structure.
    Though this hasn't caused any problems when programming the NIX SQ
    context to the HW because the context structure is the same size.
    However, this causes a problem when accessing the structure parameters.
    This patch reads the right smq value for each platform.

    Fixes: 30077d210c83 ("octeontx2-af: cn10k: Update NIX/NPA context structure")
    Signed-off-by: Geetha sowjanya <gakula@marvell.com>
    Signed-off-by: Sunil Kovvuri Goutham <sgoutham@marvell.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 703e5f708360b1400ff269a14c6837c42ef3365e
Author: Shigeru Yoshida <syoshida@redhat.com>
Date:   Sun Sep 3 02:07:08 2023 +0900

    kcm: Destroy mutex in kcm_exit_net()

    [ Upstream commit 6ad40b36cd3b04209e2d6c89d252c873d8082a59 ]

    kcm_exit_net() should call mutex_destroy() on knet->mutex. This is especially
    needed if CONFIG_DEBUG_MUTEXES is enabled.

    Fixes: ab7ac4eb9832 ("kcm: Kernel Connection Multiplexor module")
    Signed-off-by: Shigeru Yoshida <syoshida@redhat.com>
    Link: https://lore.kernel.org/r/20230902170708.1727999-1-syoshida@redhat.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e5471b82c36396e809817cb988dfc4bce0a688cb
```

```
Author: valis <sec@valis.email>
Date:   Fri Sep 1 12:22:37 2023 -0400

    net: sched: sch_qfq: Fix UAF in qfq_dequeue()

    [ Upstream commit 8fc134fee27f2263988ae38920bc03da416b03d8 ]

    When the plug qdisc is used as a class of the qfq qdisc it could trigger a
    UAF. This issue can be reproduced with following commands:

      tc qdisc add dev lo root handle 1: qfq
      tc class add dev lo parent 1: classid 1:1 qfq weight 1 maxpkt 512
      tc qdisc add dev lo parent 1:1 handle 2: plug
      tc filter add dev lo parent 1: basic classid 1:1
      ping -c1 127.0.0.1

    and boom:

    [  285.353793] BUG: KASAN: slab-use-after-free in qfq_dequeue+0xa7/0x7f0
    [  285.354910] Read of size 4 at addr ffff8880bad312a8 by task ping/144
    [  285.355903]
    [  285.356165] CPU: 1 PID: 144 Comm: ping Not tainted 6.5.0-rc3+ #4
    [  285.357112] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-2
04/01/2014
    [  285.358376] Call Trace:
    [  285.358773]  <IRQ>
    [  285.359109]  dump_stack_lvl+0x44/0x60
    [  285.359708]  print_address_description.constprop.0+0x2c/0x3c0
    [  285.360611]  kasan_report+0x10c/0x120
    [  285.361195]  ? qfq_dequeue+0xa7/0x7f0
    [  285.361780]  qfq_dequeue+0xa7/0x7f0
    [  285.362342]  __qdisc_run+0xf1/0x970
    [  285.362903]  net_tx_action+0x28e/0x460
    [  285.363502]  __do_softirq+0x11b/0x3de
    [  285.364097]  do_softirq.part.0+0x72/0x90
    [  285.364721]  </IRQ>
    [  285.365072]  <TASK>
    [  285.365422]  __local_bh_enable_ip+0x77/0x90
    [  285.366079]  __dev_queue_xmit+0x95f/0x1550
    [  285.366732]  ? __pfx_csum_and_copy_from_iter+0x10/0x10
    [  285.367526]  ? __pfx___dev_queue_xmit+0x10/0x10
    [  285.368259]  ? __build_skb_around+0x129/0x190
    [  285.368960]  ? ip_generic_getfrag+0x12c/0x170
    [  285.369653]  ? __pfx_ip_generic_getfrag+0x10/0x10
    [  285.370390]  ? csum_partial+0x8/0x20
    [  285.370961]  ? raw_getfrag+0xe5/0x140
    [  285.371559]  ip_finish_output2+0x539/0xa40
    [  285.372222]  ? __pfx_ip_finish_output2+0x10/0x10
    [  285.372954]  ip_output+0x113/0x1e0
    [  285.373512]  ? __pfx_ip_output+0x10/0x10
    [  285.374130]  ? icmp_out_count+0x49/0x60
    [  285.374739]  ? __pfx_ip_finish_output+0x10/0x10
    [  285.375457]  ip_push_pending_frames+0xf3/0x100
    [  285.376173]  raw_sendmsg+0xef5/0x12d0
    [  285.376760]  ? do_syscall_64+0x40/0x90
    [  285.377359]  ? __static_call_text_end+0x136578/0x136578
    [  285.378173]  ? do_syscall_64+0x40/0x90
    [  285.378772]  ? kasan_enable_current+0x11/0x20
    [  285.379469]  ? __pfx_raw_sendmsg+0x10/0x10
    [  285.380137]  ? __sock_create+0x13e/0x270
    [  285.380673]  ? __sys_socket+0xf3/0x180
    [  285.381174]  ? __x64_sys_socket+0x3d/0x50
    [  285.381725]  ? entry_SYSCALL_64_after_hwframe+0x6e/0xd8
    [  285.382425]  ? __rcu_read_unlock+0x48/0x70
    [  285.382975]  ? ip4_datagram_release_cb+0xd8/0x380
```

```
[  285.383608]  ? __pfx_ip4_datagram_release_cb+0x10/0x10
[  285.384295]  ? preempt_count_sub+0x14/0xc0
[  285.384844]  ? __list_del_entry_valid+0x76/0x140
[  285.385467]  ? _raw_spin_lock_bh+0x87/0xe0
[  285.386014]  ? __pfx__raw_spin_lock_bh+0x10/0x10
[  285.386645]  ? release_sock+0xa0/0xd0
[  285.387148]  ? preempt_count_sub+0x14/0xc0
[  285.387712]  ? freeze_secondary_cpus+0x348/0x3c0
[  285.388341]  ? aa_sk_perm+0x177/0x390
[  285.388856]  ? __pfx_aa_sk_perm+0x10/0x10
[  285.389441]  ? check_stack_object+0x22/0x70
[  285.390032]  ? inet_send_prepare+0x2f/0x120
[  285.390603]  ? __pfx_inet_sendmsg+0x10/0x10
[  285.391172]  sock_sendmsg+0xcc/0xe0
[  285.391667]  __sys_sendto+0x190/0x230
[  285.392168]  ? __pfx___sys_sendto+0x10/0x10
[  285.392727]  ? kvm_clock_get_cycles+0x14/0x30
[  285.393328]  ? set_normalized_timespec64+0x57/0x70
[  285.393980]  ? _raw_spin_unlock_irq+0x1b/0x40
[  285.394578]  ? __x64_sys_clock_gettime+0x11c/0x160
[  285.395225]  ? __pfx___x64_sys_clock_gettime+0x10/0x10
[  285.395908]  ? _copy_to_user+0x3e/0x60
[  285.396432]  ? exit_to_user_mode_prepare+0x1a/0x120
[  285.397086]  ? syscall_exit_to_user_mode+0x22/0x50
[  285.397734]  ? do_syscall_64+0x71/0x90
[  285.398258]  __x64_sys_sendto+0x74/0x90
[  285.398786]  do_syscall_64+0x64/0x90
[  285.399273]  ? exit_to_user_mode_prepare+0x1a/0x120
[  285.399949]  ? syscall_exit_to_user_mode+0x22/0x50
[  285.400605]  ? do_syscall_64+0x71/0x90
[  285.401124]  entry_SYSCALL_64_after_hwframe+0x6e/0xd8
[  285.401807] RIP: 0033:0x495726
[  285.402233] Code: ff ff ff f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 41 89
 ca 64 8b 04 25 18 00 00 00 85 c0 75 11 b8 2c 00 00 00 0f 09
[  285.404683] RSP: 002b:00007ffcc25fb618 EFLAGS: 00000246 ORIG_RAX: 000000000000002c
[  285.405677] RAX: ffffffffffffffda RBX: 0000000000000040 RCX: 0000000000495726
[  285.406628] RDX: 0000000000000040 RSI: 0000000002518750 RDI: 0000000000000000
[  285.407565] RBP: 00000000005205ef R08: 00000000005f8838 R09: 000000000000001c
[  285.408523] R10: 0000000000000000 R11: 0000000000000246 R12: 0000000002517634
[  285.409460] R13: 00007ffcc25fb6f0 R14: 0000000000000003 R15: 0000000000000000
[  285.410403]  </TASK>
[  285.410704]
[  285.410929] Allocated by task 144:
[  285.411402]  kasan_save_stack+0x1e/0x40
[  285.411926]  kasan_set_track+0x21/0x30
[  285.412442]  __kasan_slab_alloc+0x55/0x70
[  285.412973]  kmem_cache_alloc_node+0x187/0x3d0
[  285.413567]  __alloc_skb+0x1b4/0x230
[  285.414060]  __ip_append_data+0x17f7/0x1b60
[  285.414633]  ip_append_data+0x97/0xf0
[  285.415144]  raw_sendmsg+0x5a8/0x12d0
[  285.415640]  sock_sendmsg+0xcc/0xe0
[  285.416117]  __sys_sendto+0x190/0x230
[  285.416626]  __x64_sys_sendto+0x74/0x90
[  285.417145]  do_syscall_64+0x64/0x90
[  285.417624]  entry_SYSCALL_64_after_hwframe+0x6e/0xd8
[  285.418306]
[  285.418531] Freed by task 144:
[  285.418960]  kasan_save_stack+0x1e/0x40
[  285.419469]  kasan_set_track+0x21/0x30
[  285.419988]  kasan_save_free_info+0x27/0x40
[  285.420556]  ____kasan_slab_free+0x109/0x1a0
[  285.421146]  kmem_cache_free+0x1c2/0x450
[  285.421680]  __netif_receive_skb_core+0x2ce/0x1870
[  285.422333]  __netif_receive_skb_one_core+0x97/0x140
```

```
     [  285.423003]  process_backlog+0x100/0x2f0
     [  285.423537]  __napi_poll+0x5c/0x2d0
     [  285.424023]  net_rx_action+0x2be/0x560
     [  285.424510]  __do_softirq+0x11b/0x3de
     [  285.425034]
     [  285.425254] The buggy address belongs to the object at ffff8880bad31280
     [  285.425254]  which belongs to the cache skbuff_head_cache of size 224
     [  285.426993] The buggy address is located 40 bytes inside of
     [  285.426993]  freed 224-byte region [ffff8880bad31280, ffff8880bad31360)
     [  285.428572]
     [  285.428798] The buggy address belongs to the physical page:
     [  285.429540] page:00000000f4b77674 refcount:1 mapcount:0 mapping:0000000000000000
 index:0x0 pfn:0xbad31
     [  285.430758] flags: 0x100000000000200(slab|node=0|zone=1)
     [  285.431447] page_type: 0xffffffff()
     [  285.431934] raw: 0100000000000200 ffff88810094a8c0 dead000000000122 0000000000000000
     [  285.432757] raw: 0000000000000000 00000000800c000c 00000001ffffffff 0000000000000000
     [  285.433562] page dumped because: kasan: bad access detected
     [  285.434144]
     [  285.434320] Memory state around the buggy address:
     [  285.434828]  ffff8880bad31180: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [  285.435580]  ffff8880bad31200: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [  285.436264] >ffff8880bad31280: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     [  285.436777]                                     ^
     [  285.437106]  ffff8880bad31300: fb fb fb fb fb fb fb fb fb fb fb fb fc fc fc fc
     [  285.437616]  ffff8880bad31380: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
     [  285.438126] ==================================================================
     [  285.438662] Disabling lock debugging due to kernel taint

     Fix this by:
     1. Changing sch_plug's .peek handler to qdisc_peek_dequeued(), a
     function compatible with non-work-conserving qdiscs
     2. Checking the return value of qdisc_dequeue_peeked() in sch_qfq.

     Fixes: 462dbc9101ac ("pkt_sched: QFQ Plus: fair-queueing service at DRR cost")
     Reported-by: valis <sec@valis.email>
     Signed-off-by: valis <sec@valis.email>
     Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
     Link: https://lore.kernel.org/r/20230901162237.11525-1-jhs@mojatatu.com
     Signed-off-by: Paolo Abeni <pabeni@redhat.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit a226df7e00dd0c5154813c52ae38282bbb533851
 Author: Kuniyuki Iwashima <kuniyu@amazon.com>
 Date:   Fri Sep 1 17:27:08 2023 -0700

     af_unix: Fix data race around sk->sk_err.

     [ Upstream commit b192812905e4b134f7b7994b079eb647e9d2d37e ]

     As with sk->sk_shutdown shown in the previous patch, sk->sk_err can be
     read locklessly by unix_dgram_sendmsg().

     Let's use READ_ONCE() for sk_err as well.

     Note that the writer side is marked by commit cc04410af7de ("af_unix:
     annotate lockless accesses to sk->sk_err").

     Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
     Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
     Reviewed-by: Eric Dumazet <edumazet@google.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit d883243048aca57b4299ec4f332be205898c8d17
```

```
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Sep 1 17:27:07 2023 -0700

    af_unix: Fix data-races around sk->sk_shutdown.

    [ Upstream commit afe8764f76346ba838d4f162883e23d2fcfaa90e ]

    sk->sk_shutdown is changed under unix_state_lock(sk), but
    unix_dgram_sendmsg() calls two functions to read sk_shutdown locklessly.

      sock_alloc_send_pskb
      `- sock_wait_for_wmem

    Let's use READ_ONCE() there.

    Note that the writer side was marked by commit e1d09c2c2f57 ("af_unix:
    Fix data races around sk->sk_shutdown.").

    BUG: KCSAN: data-race in sock_alloc_send_pskb / unix_release_sock

    write (marked) to 0xffff8880069af12c of 1 bytes by task 1 on cpu 1:
     unix_release_sock+0x75c/0x910 net/unix/af_unix.c:631
     unix_release+0x59/0x80 net/unix/af_unix.c:1053
     __sock_release+0x7d/0x170 net/socket.c:654
     sock_close+0x19/0x30 net/socket.c:1386
     __fput+0x2a3/0x680 fs/file_table.c:384
     ____fput+0x15/0x20 fs/file_table.c:412
     task_work_run+0x116/0x1a0 kernel/task_work.c:179
     resume_user_mode_work include/linux/resume_user_mode.h:49 [inline]
     exit_to_user_mode_loop kernel/entry/common.c:171 [inline]
     exit_to_user_mode_prepare+0x174/0x180 kernel/entry/common.c:204
     __syscall_exit_to_user_mode_work kernel/entry/common.c:286 [inline]
     syscall_exit_to_user_mode+0x1a/0x30 kernel/entry/common.c:297
     do_syscall_64+0x4b/0x90 arch/x86/entry/common.c:86
     entry_SYSCALL_64_after_hwframe+0x6e/0xd8

    read to 0xffff8880069af12c of 1 bytes by task 28650 on cpu 0:
     sock_alloc_send_pskb+0xd2/0x620 net/core/sock.c:2767
     unix_dgram_sendmsg+0x2f8/0x14f0 net/unix/af_unix.c:1944
     unix_seqpacket_sendmsg net/unix/af_unix.c:2308 [inline]
     unix_seqpacket_sendmsg+0xba/0x130 net/unix/af_unix.c:2292
     sock_sendmsg_nosec net/socket.c:725 [inline]
     sock_sendmsg+0x148/0x160 net/socket.c:748
     ____sys_sendmsg+0x4e4/0x610 net/socket.c:2494
     ___sys_sendmsg+0xc6/0x140 net/socket.c:2548
     __sys_sendmsg+0x94/0x140 net/socket.c:2577
     __do_sys_sendmsg net/socket.c:2586 [inline]
     __se_sys_sendmsg net/socket.c:2584 [inline]
     __x64_sys_sendmsg+0x45/0x50 net/socket.c:2584
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x6e/0xd8

    value changed: 0x00 -> 0x03

    Reported by Kernel Concurrency Sanitizer on:
    CPU: 0 PID: 28650 Comm: systemd-coredum Not tainted 6.4.0-11989-g6843306689af #6
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-
prebuilt.qemu.org 04/01/2014

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Reported-by: syzkaller <syzkaller@googlegroups.com>
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
```

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

commit afc284a4a781defbb12b2a40427fae34c3d20e17
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Sep 1 17:27:06 2023 -0700

    af_unix: Fix data-race around unix_tot_inflight.

    [ Upstream commit ade32bd8a738d7497ffe9743c46728db26740f78 ]

    unix_tot_inflight is changed under spin_lock(unix_gc_lock), but
    unix_release_sock() reads it locklessly.

    Let's use READ_ONCE() for unix_tot_inflight.

    Note that the writer side was marked by commit 9d6d7f1cb67c ("af_unix:
    annote lockless accesses to unix_tot_inflight & gc_in_progress")

    BUG: KCSAN: data-race in unix_inflight / unix_release_sock

    write (marked) to 0xffffffff871852b8 of 4 bytes by task 123 on cpu 1:
     unix_inflight+0x130/0x180 net/unix/scm.c:64
     unix_attach_fds+0x137/0x1b0 net/unix/scm.c:123
     unix_scm_to_skb net/unix/af_unix.c:1832 [inline]
     unix_dgram_sendmsg+0x46a/0x14f0 net/unix/af_unix.c:1955
     sock_sendmsg_nosec net/socket.c:724 [inline]
     sock_sendmsg+0x148/0x160 net/socket.c:747
     ____sys_sendmsg+0x4e4/0x610 net/socket.c:2493
     ___sys_sendmsg+0xc6/0x140 net/socket.c:2547
     __sys_sendmsg+0x94/0x140 net/socket.c:2576
     __do_sys_sendmsg net/socket.c:2585 [inline]
     __se_sys_sendmsg net/socket.c:2583 [inline]
     __x64_sys_sendmsg+0x45/0x50 net/socket.c:2583
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x72/0xdc

    read to 0xffffffff871852b8 of 4 bytes by task 4891 on cpu 0:
     unix_release_sock+0x608/0x910 net/unix/af_unix.c:671
     unix_release+0x59/0x80 net/unix/af_unix.c:1058
     __sock_release+0x7d/0x170 net/socket.c:653
     sock_close+0x19/0x30 net/socket.c:1385
     __fput+0x179/0x5e0 fs/file_table.c:321
     ____fput+0x15/0x20 fs/file_table.c:349
     task_work_run+0x116/0x1a0 kernel/task_work.c:179
     resume_user_mode_work include/linux/resume_user_mode.h:49 [inline]
     exit_to_user_mode_loop kernel/entry/common.c:171 [inline]
     exit_to_user_mode_prepare+0x174/0x180 kernel/entry/common.c:204
     __syscall_exit_to_user_mode_work kernel/entry/common.c:286 [inline]
     syscall_exit_to_user_mode+0x1a/0x30 kernel/entry/common.c:297
     do_syscall_64+0x4b/0x90 arch/x86/entry/common.c:86
     entry_SYSCALL_64_after_hwframe+0x72/0xdc

    value changed: 0x00000000 -> 0x00000001

    Reported by Kernel Concurrency Sanitizer on:
    CPU: 0 PID: 4891 Comm: systemd-coredum Not tainted 6.4.0-rc5-01219-gfa0e21fa4443 #5
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-
prebuilt.qemu.org 04/01/2014

    Fixes: 9305cfa4443d ("[AF_UNIX]: Make unix_tot_inflight counter non-atomic")
    Reported-by: syzkaller <syzkaller@googlegroups.com>
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
```

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

commit ac92f239a079678a035c0faad9089354a874aede
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Sep 1 17:27:05 2023 -0700

    af_unix: Fix data-races around user->unix_inflight.

    [ Upstream commit 0bc36c0650b21df36fbec8136add83936eaf0607 ]

    user->unix_inflight is changed under spin_lock(unix_gc_lock),
    but too_many_unix_fds() reads it locklessly.

    Let's annotate the write/read accesses to user->unix_inflight.

    BUG: KCSAN: data-race in unix_attach_fds / unix_inflight

    write to 0xffffffff8546f2d0 of 8 bytes by task 44798 on cpu 1:
     unix_inflight+0x157/0x180 net/unix/scm.c:66
     unix_attach_fds+0x147/0x1e0 net/unix/scm.c:123
     unix_scm_to_skb net/unix/af_unix.c:1827 [inline]
     unix_dgram_sendmsg+0x46a/0x14f0 net/unix/af_unix.c:1950
     unix_seqpacket_sendmsg net/unix/af_unix.c:2308 [inline]
     unix_seqpacket_sendmsg+0xba/0x130 net/unix/af_unix.c:2292
     sock_sendmsg_nosec net/socket.c:725 [inline]
     sock_sendmsg+0x148/0x160 net/socket.c:748
     ____sys_sendmsg+0x4e4/0x610 net/socket.c:2494
     ___sys_sendmsg+0xc6/0x140 net/socket.c:2548
     __sys_sendmsg+0x94/0x140 net/socket.c:2577
     __do_sys_sendmsg net/socket.c:2586 [inline]
     __se_sys_sendmsg net/socket.c:2584 [inline]
     __x64_sys_sendmsg+0x45/0x50 net/socket.c:2584
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x6e/0xd8

    read to 0xffffffff8546f2d0 of 8 bytes by task 44814 on cpu 0:
     too_many_unix_fds net/unix/scm.c:101 [inline]
     unix_attach_fds+0x54/0x1e0 net/unix/scm.c:110
     unix_scm_to_skb net/unix/af_unix.c:1827 [inline]
     unix_dgram_sendmsg+0x46a/0x14f0 net/unix/af_unix.c:1950
     unix_seqpacket_sendmsg net/unix/af_unix.c:2308 [inline]
     unix_seqpacket_sendmsg+0xba/0x130 net/unix/af_unix.c:2292
     sock_sendmsg_nosec net/socket.c:725 [inline]
     sock_sendmsg+0x148/0x160 net/socket.c:748
     ____sys_sendmsg+0x4e4/0x610 net/socket.c:2494
     ___sys_sendmsg+0xc6/0x140 net/socket.c:2548
     __sys_sendmsg+0x94/0x140 net/socket.c:2577
     __do_sys_sendmsg net/socket.c:2586 [inline]
     __se_sys_sendmsg net/socket.c:2584 [inline]
     __x64_sys_sendmsg+0x45/0x50 net/socket.c:2584
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x6e/0xd8

    value changed: 0x000000000000000c -> 0x000000000000000d

    Reported by Kernel Concurrency Sanitizer on:
    CPU: 0 PID: 44814 Comm: systemd-coredum Not tainted 6.4.0-11989-g6843306689af #6
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-
prebuilt.qemu.org 04/01/2014

    Fixes: 712f4aad406b ("unix: properly account for FDs passed over unix sockets")
    Reported-by: syzkaller <syzkaller@googlegroups.com>
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
```

```
    Acked-by: Willy Tarreau <w@1wt.eu>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 51bd1196ea4d0a2e63b49f93553bb5231a9f48cb
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Sep 1 16:46:04 2023 -0700

    af_unix: Fix msg_controllen test in scm_pidfd_recv() for MSG_CMSG_COMPAT.

    [ Upstream commit 718e6b51298e0f254baca0d40ab52a00e004e014 ]

    Heiko Carstens reported that SCM_PIDFD does not work with MSG_CMSG_COMPAT
    because scm_pidfd_recv() always checks msg_controllen against sizeof(struct
    cmsghdr).

    We need to use sizeof(struct compat_cmsghdr) for the compat case.

    Fixes: 5e2ff6704a27 ("scm: add SO_PASSPIDFD and SCM_PIDFD")
    Reported-by: Heiko Carstens <hca@linux.ibm.com>
    Closes: https://lore.kernel.org/netdev/20230901200517.8742-A-hca@linux.ibm.com/
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Tested-by: Heiko Carstens <hca@linux.ibm.com>
    Reviewed-by: Alexander Mikhalitsyn <aleksandr.mikhalitsyn@canonical.com>
    Reviewed-by: Michal Swiatkowski <michal.swiatkowski@linux.intel.com>
    Acked-by: Christian Brauner <brauner@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e6b5e47adb9166e732cdf7e6e034946e3f89f36d
Author: John Fastabend <john.fastabend@gmail.com>
Date:   Fri Sep 1 13:21:37 2023 -0700

    bpf, sockmap: Fix skb refcnt race after locking changes

    [ Upstream commit a454d84ee20baf7bd7be90721b9821f73c7d23d9 ]

    There is a race where skb's from the sk_psock_backlog can be referenced
    after userspace side has already skb_consumed() the sk_buff and its refcnt
    dropped to zer0 causing use after free.

    The flow is the following:

      while ((skb = skb_peek(&psock->ingress_skb))
        sk_psock_handle_Skb(psock, skb, ..., ingress)
        if (!ingress) ...
        sk_psock_skb_ingress
           sk_psock_skb_ingress_enqueue(skb)
              msg->skb = skb
              sk_psock_queue_msg(psock, msg)
        skb_dequeue(&psock->ingress_skb)

    The sk_psock_queue_msg() puts the msg on the ingress_msg queue. This is
    what the application reads when recvmsg() is called. An application can
    read this anytime after the msg is placed on the queue. The recvmsg hook
    will also read msg->skb and then after user space reads the msg will call
    consume_skb(skb) on it effectively free'ing it.

    But, the race is in above where backlog queue still has a reference to
    the skb and calls skb_dequeue(). If the skb_dequeue happens after the
    user reads and free's the skb we have a use after free.

    The !ingress case does not suffer from this problem because it uses
    sendmsg_*(sk, msg) which does not pass the sk_buff further down the
```

```
    stack.

    The following splat was observed with 'test_progs -t sockmap_listen':

      [ 1022.710250][ T2556] general protection fault, ...
      [...]
      [ 1022.712830][ T2556] Workqueue: events sk_psock_backlog
      [ 1022.713262][ T2556] RIP: 0010:skb_dequeue+0x4c/0x80
      [ 1022.713653][ T2556] Code: ...
      [...]
      [ 1022.720699][ T2556] Call Trace:
      [ 1022.720984][ T2556]  <TASK>
      [ 1022.721254][ T2556]  ? die_addr+0x32/0x80^M
      [ 1022.721589][ T2556]  ? exc_general_protection+0x25a/0x4b0
      [ 1022.722026][ T2556]  ? asm_exc_general_protection+0x22/0x30
      [ 1022.722489][ T2556]  ? skb_dequeue+0x4c/0x80
      [ 1022.722854][ T2556]  sk_psock_backlog+0x27a/0x300
      [ 1022.723243][ T2556]  process_one_work+0x2a7/0x5b0
      [ 1022.723633][ T2556]  worker_thread+0x4f/0x3a0
      [ 1022.723998][ T2556]  ? __pfx_worker_thread+0x10/0x10
      [ 1022.724386][ T2556]  kthread+0xfd/0x130
      [ 1022.724709][ T2556]  ? __pfx_kthread+0x10/0x10
      [ 1022.725066][ T2556]  ret_from_fork+0x2d/0x50
      [ 1022.725409][ T2556]  ? __pfx_kthread+0x10/0x10
      [ 1022.725799][ T2556]  ret_from_fork_asm+0x1b/0x30
      [ 1022.726201][ T2556]  </TASK>

    To fix we add an skb_get() before passing the skb to be enqueued in the
    engress queue. This bumps the skb->users refcnt so that consume_skb()
    and kfree_skb will not immediately free the sk_buff. With this we can
    be sure the skb is still around when we do the dequeue. Then we just
    need to decrement the refcnt or free the skb in the backlog case which
    we do by calling kfree_skb() on the ingress case as well as the sendmsg
    case.

    Before locking change from fixes tag we had the sock locked so we
    couldn't race with user and there was no issue here.

    Fixes: 799aa7f98d53e ("skmsg: Avoid lock_sock() in sk_psock_backlog()")
    Reported-by: Jiri Olsa  <jolsa@kernel.org>
    Signed-off-by: John Fastabend <john.fastabend@gmail.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Tested-by: Xu Kuohai <xukuohai@huawei.com>
    Tested-by: Jiri Olsa <jolsa@kernel.org>
    Link: https://lore.kernel.org/bpf/20230901202137.214666-1-john.fastabend@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3db68536ce76ebc19a9ed0d7e1b1c246be5ac9ef
Author: Oleksij Rempel <linux@rempel-privat.de>
Date:   Fri Sep 1 06:53:23 2023 +0200

    net: phy: micrel: Correct bit assignments for phy_device flags

    [ Upstream commit 719c5e37e99d2fd588d1c994284d17650a66354c ]

    Previously, the defines for phy_device flags in the Micrel driver were
    ambiguous in their representation. They were intended to be bit masks
    but were mistakenly defined as bit positions. This led to the following
    issues:

    - MICREL_KSZ8_P1_ERRATA, designated for KSZ88xx switches, overlapped
      with MICREL_PHY_FXEN and MICREL_PHY_50MHZ_CLK.
    - Due to this overlap, the code path for MICREL_PHY_FXEN, tailored for
      the KSZ8041 PHY, was not executed for KSZ88xx PHYs.
    - Similarly, the code associated with MICREL_PHY_50MHZ_CLK wasn't
```

```
       triggered for KSZ88xx.

    To rectify this, all three flags have now been explicitly converted to
    use the `BIT()` macro, ensuring they are defined as bit masks and
    preventing potential overlaps in the future.

    Fixes: 49011e0c1555 ("net: phy: micrel: ksz886x/ksz8081: add cabletest support")
    Signed-off-by: Oleksij Rempel <o.rempel@pengutronix.de>
    Reviewed-by: Russell King (Oracle) <rmk+kernel@armlinux.org.uk>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fbdfbc80362b431c0f12427b07e78d6512b7a715
Author: Alex Henrie <alexhenrie24@gmail.com>
Date:   Thu Aug 31 22:41:27 2023 -0600

    net: ipv6/addrconf: avoid integer underflow in ipv6_create_tempaddr

    [ Upstream commit f31867d0d9d82af757c1e0178b659438f4c1ea3c ]

    The existing code incorrectly casted a negative value (the result of a
    subtraction) to an unsigned value without checking. For example, if
    /proc/sys/net/ipv6/conf/*/temp_prefered_lft was set to 1, the preferred
    lifetime would jump to 4 billion seconds. On my machine and network the
    shortest lifetime that avoided underflow was 3 seconds.

    Fixes: 76506a986dc3 ("IPv6: fix DESYNC_FACTOR")
    Signed-off-by: Alex Henrie <alexhenrie24@gmail.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cad8c215be8c36720105de7e9fb67cc1c8fd6006
Author: Liang Chen <liangchen.linux@gmail.com>
Date:   Fri Sep 1 12:09:21 2023 +0800

    veth: Fixing transmit return status for dropped packets

    [ Upstream commit 151e887d8ff97e2e42110ffa1fb1e6a2128fb364 ]

    The veth_xmit function returns NETDEV_TX_OK even when packets are dropped.
    This behavior leads to incorrect calculations of statistics counts, as
    well as things like txq->trans_start updates.

    Fixes: e314dbdc1c0d ("[NET]: Virtual ethernet device driver.")
    Signed-off-by: Liang Chen <liangchen.linux@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 605f45b418e0882f2a5cf836c2a26cbadb4c3e81
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 21:38:12 2023 +0000

    gve: fix frag_list chaining

    [ Upstream commit 817c7cd2043a83a3d8147f40eea1505ac7300b62 ]

    gve_rx_append_frags() is able to build skbs chained with frag_list,
    like GRO engine.

    Problem is that shinfo->frag_list should only be used
    for the head of the chain.

    All other links should use skb->next pointer.
```

```
    Otherwise, built skbs are not valid and can cause crashes.

    Equivalent code in GRO (skb_gro_receive()) is:

        if (NAPI_GRO_CB(p)->last == p)
            skb_shinfo(p)->frag_list = skb;
        else
            NAPI_GRO_CB(p)->last->next = skb;
        NAPI_GRO_CB(p)->last = skb;

    Fixes: 9b8dd5e5ea48 ("gve: DQO: Add RX path")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Bailey Forrest <bcf@google.com>
    Cc: Willem de Bruijn <willemb@google.com>
    Cc: Catherine Sullivan <csully@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 39acd66ada7136dd596f254a88eccd5d23a4ce7b
Author: Corinna Vinschen <vinschen@redhat.com>
Date:   Thu Aug 31 14:19:13 2023 +0200

    igb: disable virtualization features on 82580

    [ Upstream commit fa09bc40b21a33937872c4c4cf0f266ec9fa4869 ]

    Disable virtualization features on 82580 just as on i210/i211.
    This avoids that virt functions are acidentally called on 82850.

    Fixes: 55cac248caa4 ("igb: Add full support for 82580 devices")
    Signed-off-by: Corinna Vinschen <vinschen@redhat.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae211fd518d700e96b1a8c4e61aea2520e92df65
Author: Xu Kuohai <xukuohai@huawei.com>
Date:   Fri Sep 1 11:10:37 2023 +0800

    selftests/bpf: Fix a CI failure caused by vsock write

    [ Upstream commit c1970e26bdc1209974bb5cf31cc23f2b7ad6ce50 ]

    While commit 90f0074cd9f9 ("selftests/bpf: fix a CI failure caused by vsock sockmap
test")
    fixes a receive failure of vsock sockmap test, there is still a write failure:

    Error: #211/79 sockmap_listen/sockmap VSOCK test_vsock_redir
    Error: #211/79 sockmap_listen/sockmap VSOCK test_vsock_redir
      ./test_progs:vsock_unix_redir_connectible:1501: egress: write: Transport endpoint is
not connected
      vsock_unix_redir_connectible:FAIL:1501
      ./test_progs:vsock_unix_redir_connectible:1501: ingress: write: Transport endpoint is
not connected
      vsock_unix_redir_connectible:FAIL:1501
      ./test_progs:vsock_unix_redir_connectible:1501: egress: write: Transport endpoint is
not connected
      vsock_unix_redir_connectible:FAIL:1501

    The reason is that the vsock connection in the test is set to ESTABLISHED state
    by function virtio_transport_recv_pkt, which is executed in a workqueue thread,
    so when the user space test thread runs before the workqueue thread, this
    problem occurs.
```

```
    To fix it, before writing the connection, wait for it to be connected.

    Fixes: d61bd8c1fd02 ("selftests/bpf: add a test case for vsock sockmap")
    Signed-off-by: Xu Kuohai <xukuohai@huawei.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Link: https://lore.kernel.org/bpf/20230901031037.3314007-1-xukuohai@huaweicloud.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit be5879bade482a2bb605dc7b21e2c211720a8ff4
Author: Sriram Yagnaraman <sriram.yagnaraman@est.tech>
Date:   Thu Aug 31 10:03:31 2023 +0200

    ipv6: ignore dst hint for multipath routes

    [ Upstream commit 8423be8926aa82cd2e28bba5cc96ccb72c7ce6be ]

    Route hints when the nexthop is part of a multipath group causes packets
    in the same receive batch to be sent to the same nexthop irrespective of
    the multipath hash of the packet. So, do not extract route hint for
    packets whose destination is part of a multipath group.

    A new SKB flag IP6SKB_MULTIPATH is introduced for this purpose, set the
    flag when route is looked up in fib6_select_path() and use it in
    ip6_can_use_hint() to check for the existence of the flag.

    Fixes: 197dbf24e360 ("ipv6: introduce and uses route look hints for list input.")
    Signed-off-by: Sriram Yagnaraman <sriram.yagnaraman@est.tech>
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 292a742fe33b6a3a57bac4a8befd59a8776b2811
Author: Sriram Yagnaraman <sriram.yagnaraman@est.tech>
Date:   Thu Aug 31 10:03:30 2023 +0200

    ipv4: ignore dst hint for multipath routes

    [ Upstream commit 6ac66cb03ae306c2e288a9be18226310529f5b25 ]

    Route hints when the nexthop is part of a multipath group causes packets
    in the same receive batch to be sent to the same nexthop irrespective of
    the multipath hash of the packet. So, do not extract route hint for
    packets whose destination is part of a multipath group.

    A new SKB flag IPSKB_MULTIPATH is introduced for this purpose, set the
    flag when route is looked up in ip_mkroute_input() and use it in
    ip_extract_route_hint() to check for the existence of the flag.

    Fixes: 02b24941619f ("ipv4: use dst hint for ipv4 list receive")
    Signed-off-by: Sriram Yagnaraman <sriram.yagnaraman@est.tech>
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5f4836eca4f9c5fb6e3e49d9b09e1545cf472684
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 13:52:12 2023 +0000

    net: annotate data-races around sk->sk_bind_phc

    [ Upstream commit 251cd405a9e6e70b92fe5afbdd17fd5caf9d3266 ]
```

```
    sk->sk_bind_phc is read locklessly. Add corresponding annotations.

    Fixes: d463126e23f1 ("net: sock: extend SO_TIMESTAMPING for PHC binding")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Yangbo Lu <yangbo.lu@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b352717b3cc265c97ad817696c24c173cd7f9294
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 13:52:11 2023 +0000

    net: annotate data-races around sk->sk_tsflags

    [ Upstream commit e3390b30a5dfb112e8e802a59c0f68f947b638b2 ]

    sk->sk_tsflags can be read locklessly, add corresponding annotations.

    Fixes: b9f40e21ef42 ("net-timestamp: move timestamp flags out of sk_flags")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Willem de Bruijn <willemb@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ca7fe4b1131de42954a68c5bf898f86ebef20f57
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 13:52:10 2023 +0000

    mptcp: annotate data-races around msk->rmem_fwd_alloc

    [ Upstream commit 9531e4a83febc3fb47ac77e24cfb5ea97e50034d ]

    msk->rmem_fwd_alloc can be read locklessly.

    Add mptcp_rmem_fwd_alloc_add(), similar to sk_forward_alloc_add(),
    and appropriate READ_ONCE()/WRITE_ONCE() annotations.

    Fixes: 6511882cdd82 ("mptcp: allocate fwd memory separately on the rx and tx path")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 12f7eda26d690986621fbba0d7cbf118a192d826
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 13:52:09 2023 +0000

    net: annotate data-races around sk->sk_forward_alloc

    [ Upstream commit 5e6300e7b3a4ab5b72a82079753868e91fbf9efc ]

    Every time sk->sk_forward_alloc is read locklessly,
    add a READ_ONCE().

    Add sk_forward_alloc_add() helper to centralize updates,
    to reduce number of WRITE_ONCE().

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

commit 3d34ec2ba1fb28e4f8fb70fddfb072aac221c043
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 13:52:08 2023 +0000
```

```
    net: use sk_forward_alloc_get() in sk_get_meminfo()

    [ Upstream commit 66d58f046c9d3a8f996b7138d02e965fd0617de0 ]

    inet_sk_diag_fill() has been changed to use sk_forward_alloc_get(),
    but sk_get_meminfo() was forgotten.

    Fixes: 292e6077b040 ("net: introduce sk_forward_alloc_get()")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 93d69f18edcca282351394c5870bec24cc99d745
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Aug 31 08:45:09 2023 +0000

    net/handshake: fix null-ptr-deref in handshake_nl_done_doit()

    [ Upstream commit 82ba0ff7bf0483d962e592017bef659ae022d754 ]

    We should not call trace_handshake_cmd_done_err() if socket lookup has failed.

    Also we should call trace_handshake_cmd_done_err() before releasing the file,
    otherwise dereferencing sock->sk can return garbage.

    This also reverts 7afc6d0a107f ("net/handshake: Fix uninitialized local variable")

    Unable to handle kernel paging request at virtual address dfff800000000003
    KASAN: null-ptr-deref in range [0x0000000000000018-0x000000000000001f]
    Mem abort info:
    ESR = 0x0000000096000005
    EC = 0x25: DABT (current EL), IL = 32 bits
    SET = 0, FnV = 0
    EA = 0, S1PTW = 0
    FSC = 0x05: level 1 translation fault
    Data abort info:
    ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000
    CM = 0, WnR = 0, TnD = 0, TagAccess = 0
    GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0
    [dfff800000000003] address between user and kernel address ranges
    Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP
    Modules linked in:
    CPU: 1 PID: 5986 Comm: syz-executor292 Not tainted 6.5.0-rc7-syzkaller-gfe4469582053 #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
07/26/2023
    pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--)
    pc : handshake_nl_done_doit+0x198/0x9c8 net/handshake/netlink.c:193
    lr : handshake_nl_done_doit+0x180/0x9c8
    sp : ffff800096e37180
    x29: ffff800096e37200 x28: 1ffff00012dc6e34 x27: dfff800000000000
    x26: ffff800096e373d0 x25: 0000000000000000 x24: 00000000ffffffa8
    x23: ffff800096e373f0 x22: 1ffff00012dc6e38 x21: 0000000000000000
    x20: ffff800096e371c0 x19: 0000000000000018 x18: 0000000000000000
    x17: 0000000000000000 x16: ffff800080516cc4 x15: 0000000000000001
    x14: 1fffe0001b14aa3b x13: 0000000000000000 x12: 0000000000000000
    x11: 0000000000000000 x10: 0000000000000000 x9 : 0000000000000003
    x8 : 0000000000000003 x7 : ffff800080afe47c x6 : 0000000000000000
    x5 : 0000000000000000 x4 : 0000000000000000 x3 : ffff800080a88078
    x2 : 0000000000000001 x1 : 00000000ffffffa8 x0 : 0000000000000000
    Call trace:
    handshake_nl_done_doit+0x198/0x9c8 net/handshake/netlink.c:193
    genl_family_rcv_msg_doit net/netlink/genetlink.c:970 [inline]
    genl_family_rcv_msg net/netlink/genetlink.c:1050 [inline]
    genl_rcv_msg+0x96c/0xc50 net/netlink/genetlink.c:1067
```

```
    netlink_rcv_skb+0x214/0x3c4 net/netlink/af_netlink.c:2549
    genl_rcv+0x38/0x50 net/netlink/genetlink.c:1078
    netlink_unicast_kernel net/netlink/af_netlink.c:1339 [inline]
    netlink_unicast+0x660/0x8d4 net/netlink/af_netlink.c:1365
    netlink_sendmsg+0x834/0xb18 net/netlink/af_netlink.c:1914
    sock_sendmsg_nosec net/socket.c:725 [inline]
    sock_sendmsg net/socket.c:748 [inline]
    ____sys_sendmsg+0x56c/0x840 net/socket.c:2494
    ___sys_sendmsg net/socket.c:2548 [inline]
    __sys_sendmsg+0x26c/0x33c net/socket.c:2577
    __do_sys_sendmsg net/socket.c:2586 [inline]
    __se_sys_sendmsg net/socket.c:2584 [inline]
    __arm64_sys_sendmsg+0x80/0x94 net/socket.c:2584
    __invoke_syscall arch/arm64/kernel/syscall.c:37 [inline]
    invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:51
    el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:136
    do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:155
    el0_svc+0x58/0x16c arch/arm64/kernel/entry-common.c:678
    el0t_64_sync_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:696
    el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:591
    Code: 12800108 b90043e8 910062b3 d343fe68 (387b6908)

    Fixes: 3b3009ea8abb ("net/handshake: Create a NETLINK service for handling handshake
requests")
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Chuck Lever <chuck.lever@oracle.com>
    Reviewed-by: Michal Kubiak <michal.kubiak@intel.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 69b49b24f29dc7eadf77616216e33e20c805ff5d
Author: Hamza Mahfooz <hamza.mahfooz@amd.com>
Date:   Fri Aug 18 09:11:11 2023 -0400

    drm/amd/display: fix mode scaling (RMX_.*)

    [ Upstream commit ea7971af7a911a7a388b4c47db2a231a6b8dcc29 ]

    As made mention of in commit 4a2df0d1f28e ("drm/amd/display: Fixed
    non-native modes not lighting up"), we shouldn't call
    drm_mode_set_crtcinfo() once the crtc timings have been decided. Since,
    it can cause settings to be unintentionally overwritten. So, since
    dm_state is never NULL now, we can use old_stream to determine if we
    should call drm_mode_set_crtcinfo() because we only need to set the crtc
    timing parameters for entirely new streams.

    Cc: Harry Wentland <harry.wentland@amd.com>
    Cc: Rodrigo Siqueira <rodrigo.siqueira@amd.com>
    Fixes: bd49f19039c1 ("drm/amd/display: Always set crtcinfo from create_stream_for_sink")
    Reviewed-by: Harry Wentland <harry.wentland@amd.com>
    Signed-off-by: Hamza Mahfooz <hamza.mahfooz@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ed059a6675da157b19f67d6f406739a9b1833477
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 28 18:35:16 2023 -0700

    drm/i915/gvt: Drop unused helper intel_vgpu_reset_gtt()

    [ Upstream commit a90c367e5af63880008e21dd199dac839e0e9e0f ]

    Drop intel_vgpu_reset_gtt() as it no longer has any callers.  In addition
    to eliminating dead code, this eliminates the last possible scenario where
```

```
    __kvmgt_protect_table_find() can be reached without holding vgpu_lock.
    Requiring vgpu_lock to be held when calling __kvmgt_protect_table_find()
    will allow a protecting the gfn hash with vgpu_lock without too much fuss.

    No functional change intended.

    Fixes: ba25d977571e ("drm/i915/gvt: Do not destroy ppgtt_mm during vGPU D3->D0.")
    Reviewed-by: Yan Zhao <yan.y.zhao@intel.com>
    Tested-by: Yongwei Ma <yongwei.ma@intel.com>
    Reviewed-by: Zhi Wang <zhi.a.wang@intel.com>
    Link: https://lore.kernel.org/r/20230729013535.1070024-11-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6d2357b2c8e84ac53e154536d6c80027eb1b8761
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 28 18:35:11 2023 -0700

    drm/i915/gvt: Put the page reference obtained by KVM's gfn_to_pfn()

    [ Upstream commit 708e49583d7da863898b25dafe4bcd799c414278 ]

    Put the struct page reference acquired by gfn_to_pfn(), KVM's API is that
    the caller is ultimately responsible for dropping any reference.

    Note, kvm_release_pfn_clean() ensures the pfn is actually a refcounted
    struct page before trying to put any references.

    Fixes: b901b252b6cf ("drm/i915/gvt: Add 2M huge gtt support")
    Reviewed-by: Yan Zhao <yan.y.zhao@intel.com>
    Tested-by: Yongwei Ma <yongwei.ma@intel.com>
    Reviewed-by: Zhi Wang <zhi.a.wang@intel.com>
    Link: https://lore.kernel.org/r/20230729013535.1070024-6-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dcb7c713f0112dcfba8405379aa33a9f0c36f170
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 28 18:35:07 2023 -0700

    drm/i915/gvt: Verify pfn is "valid" before dereferencing "struct page"

    [ Upstream commit f046923af79158361295ed4f0a588c80b9fdcc1d ]

    Check that the pfn found by gfn_to_pfn() is actually backed by "struct
    page" memory prior to retrieving and dereferencing the page.  KVM
    supports backing guest memory with VM_PFNMAP, VM_IO, etc., and so
    there is no guarantee the pfn returned by gfn_to_pfn() has an associated
    "struct page".

    Fixes: b901b252b6cf ("drm/i915/gvt: Add 2M huge gtt support")
    Reviewed-by: Yan Zhao <yan.y.zhao@intel.com>
    Tested-by: Yongwei Ma <yongwei.ma@intel.com>
    Reviewed-by: Zhi Wang <zhi.a.wang@intel.com>
    Link: https://lore.kernel.org/r/20230729013535.1070024-2-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 595931912357fa3507e522a7f8a0a76e423c23e4
Author: Magnus Karlsson <magnus.karlsson@intel.com>
Date:   Thu Aug 31 12:01:17 2023 +0200
```

```
    xsk: Fix xsk_diag use-after-free error during socket cleanup

    [ Upstream commit 3e019d8a05a38abb5c85d4f1e85fda964610aa14 ]

    Fix a use-after-free error that is possible if the xsk_diag interface
    is used after the socket has been unbound from the device. This can
    happen either due to the socket being closed or the device
    disappearing. In the early days of AF_XDP, the way we tested that a
    socket was not bound to a device was to simply check if the netdevice
    pointer in the xsk socket structure was NULL. Later, a better system
    was introduced by having an explicit state variable in the xsk socket
    struct. For example, the state of a socket that is on the way to being
    closed and has been unbound from the device is XSK_UNBOUND.

    The commit in the Fixes tag below deleted the old way of signalling
    that a socket is unbound, setting dev to NULL. This in the belief that
    all code using the old way had been exterminated. That was
    unfortunately not true as the xsk diagnostics code was still using the
    old way and thus does not work as intended when a socket is going
    down. Fix this by introducing a test against the state variable. If
    the socket is in the state XSK_UNBOUND, simply abort the diagnostic's
    netlink operation.

    Fixes: 18b1ab7aa76b ("xsk: Fix race at socket teardown")
    Reported-by: syzbot+822d1359297e2694f873@syzkaller.appspotmail.com
    Signed-off-by: Magnus Karlsson <magnus.karlsson@intel.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Tested-by: syzbot+822d1359297e2694f873@syzkaller.appspotmail.com
    Tested-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Reviewed-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Link: https://lore.kernel.org/bpf/20230831100119.17408-1-magnus.karlsson@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2c13f83db659f0b9be9cfd4674eb308db9e0ee0
Author: Florian Westphal <fw@strlen.de>
Date:   Wed Aug 30 13:00:37 2023 +0200

    net: fib: avoid warn splat in flow dissector

    [ Upstream commit 8aae7625ff3f0bd5484d01f1b8d5af82e44bec2d ]

    New skbs allocated via nf_send_reset() have skb->dev == NULL.

    fib*_rules_early_flow_dissect helpers already have a 'struct net'
    argument but its not passed down to the flow dissector core, which
    will then WARN as it can't derive a net namespace to use:

     WARNING: CPU: 0 PID: 0 at net/core/flow_dissector.c:1016
__skb_flow_dissect+0xa91/0x1cd0
     [..]
      ip_route_me_harder+0x143/0x330
      nf_send_reset+0x17c/0x2d0 [nf_reject_ipv4]
      nft_reject_inet_eval+0xa9/0xf2 [nft_reject_inet]
      nft_do_chain+0x198/0x5d0 [nf_tables]
      nft_do_chain_inet+0xa4/0x110 [nf_tables]
      nf_hook_slow+0x41/0xc0
      ip_local_deliver+0xce/0x110
      ..

    Cc: Stanislav Fomichev <sdf@google.com>
    Cc: David Ahern <dsahern@kernel.org>
    Cc: Ido Schimmel <idosch@nvidia.com>
    Fixes: 812fa71f0d96 ("netfilter: Dissect flow after packet mangling")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=217826
    Signed-off-by: Florian Westphal <fw@strlen.de>
```

```
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Link: https://lore.kernel.org/r/20230830110043.30497-1-fw@strlen.de
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b1f5b890b89cb38a6c0bac91984d56cd69808e8c
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 30 10:12:44 2023 +0000

    net: read sk->sk_family once in sk_mc_loop()

    [ Upstream commit a3e0fdf71bbe031de845e8e08ed7fba49f9c702c ]

    syzbot is playing with IPV6_ADDRFORM quite a lot these days,
    and managed to hit the WARN_ON_ONCE(1) in sk_mc_loop()

    We have many more similar issues to fix.

    WARNING: CPU: 1 PID: 1593 at net/core/sock.c:782 sk_mc_loop+0x165/0x260
    Modules linked in:
    CPU: 1 PID: 1593 Comm: kworker/1:3 Not tainted 6.1.40-syzkaller #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
07/26/2023
    Workqueue: events_power_efficient gc_worker
    RIP: 0010:sk_mc_loop+0x165/0x260 net/core/sock.c:782
    Code: 34 1b fd 49 81 c7 18 05 00 00 4c 89 f8 48 c1 e8 03 42 80 3c 20 00 74 08 4c 89 ff
e8 25 36 6d fd 4d 8b 37 eb 13 e8 db 33 1b fd <0f> 0b b3 01 eb 34 e8 d0 33 1b fd 45 31 f6 49
83 c6 38 4c 89 f0 48
    RSP: 0018:ffffc90000388530 EFLAGS: 00010246
    RAX: ffffffff846d9b55 RBX: 0000000000000011 RCX: ffff88814f884980
    RDX: 0000000000000102 RSI: ffffffff87ae5160 RDI: 0000000000000011
    RBP: ffffc90000388550 R08: 0000000000000003 R09: ffffffff846d9a65
    R10: 0000000000000002 R11: ffff88814f884980 R12: dfffffc0000000000
    R13: ffff88810dbee000 R14: 0000000000000010 R15: ffff888150084000
    FS: 0000000000000000(0000) GS:ffff8881f6b00000(0000) knlGS:0000000000000000
    CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 0000000020000180 CR3: 000000014ee5b000 CR4: 00000000003506e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
    <IRQ>
    [<ffffffff8507734f>] ip6_finish_output2+0x33f/0x1ae0 net/ipv6/ip6_output.c:83
    [<ffffffff85062766>] __ip6_finish_output net/ipv6/ip6_output.c:200 [inline]
    [<ffffffff85062766>] ip6_finish_output+0x6c6/0xb10 net/ipv6/ip6_output.c:211
    [<ffffffff85061f8c>] NF_HOOK_COND include/linux/netfilter.h:298 [inline]
    [<ffffffff85061f8c>] ip6_output+0x2bc/0x3d0 net/ipv6/ip6_output.c:232
    [<ffffffff852071cf>] dst_output include/net/dst.h:444 [inline]
    [<ffffffff852071cf>] ip6_local_out+0x10f/0x140 net/ipv6/output_core.c:161
    [<ffffffff83618fb4>] ipvlan_process_v6_outbound drivers/net/ipvlan/ipvlan_core.c:483
[inline]
    [<ffffffff83618fb4>] ipvlan_process_outbound drivers/net/ipvlan/ipvlan_core.c:529
[inline]
    [<ffffffff83618fb4>] ipvlan_xmit_mode_l3 drivers/net/ipvlan/ipvlan_core.c:602 [inline]
    [<ffffffff83618fb4>] ipvlan_queue_xmit+0x1174/0x1be0
drivers/net/ipvlan/ipvlan_core.c:677
    [<ffffffff8361ddd9>] ipvlan_start_xmit+0x49/0x100 drivers/net/ipvlan/ipvlan_main.c:229
    [<ffffffff84763fc0>] netdev_start_xmit include/linux/netdevice.h:4925 [inline]
    [<ffffffff84763fc0>] xmit_one net/core/dev.c:3644 [inline]
    [<ffffffff84763fc0>] dev_hard_start_xmit+0x320/0x980 net/core/dev.c:3660
    [<ffffffff8494c650>] sch_direct_xmit+0x2a0/0x9c0 net/sched/sch_generic.c:342
    [<ffffffff8494d883>] qdisc_restart net/sched/sch_generic.c:407 [inline]
    [<ffffffff8494d883>] __qdisc_run+0xb13/0x1e70 net/sched/sch_generic.c:415
    [<ffffffff8478c426>] qdisc_run+0xd6/0x260 include/net/pkt_sched.h:125
    [<ffffffff84796eac>] net_tx_action+0x7ac/0x940 net/core/dev.c:5247
```

```
    [<ffffffff858002bd>] __do_softirq+0x2bd/0x9bd kernel/softirq.c:599
    [<ffffffff814c3fe8>] invoke_softirq kernel/softirq.c:430 [inline]
    [<ffffffff814c3fe8>] __irq_exit_rcu+0xc8/0x170 kernel/softirq.c:683
    [<ffffffff814c3f09>] irq_exit_rcu+0x9/0x20 kernel/softirq.c:695

    Fixes: 7ad6848c7e81 ("ip: fix mc_loop checks for tunnels with multicast outer
addresses")
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230830101244.1146934-1-edumazet@google.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a558ff4ae5238d19cba11677614b78b859ed8c7c
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 30 09:55:20 2023 +0000

    ipv4: annotate data-races around fi->fib_dead

    [ Upstream commit fce92af1c29d90184dfec638b5738831097d66e9 ]

    syzbot complained about a data-race in fib_table_lookup() [1]

    Add appropriate annotations to document it.

    [1]
    BUG: KCSAN: data-race in fib_release_info / fib_table_lookup

    write to 0xffff888150f31744 of 1 bytes by task 1189 on cpu 0:
    fib_release_info+0x3a0/0x460 net/ipv4/fib_semantics.c:281
    fib_table_delete+0x8d2/0x900 net/ipv4/fib_trie.c:1777
    fib_magic+0x1c1/0x1f0 net/ipv4/fib_frontend.c:1106
    fib_del_ifaddr+0x8cf/0xa60 net/ipv4/fib_frontend.c:1317
    fib_inetaddr_event+0x77/0x200 net/ipv4/fib_frontend.c:1448
    notifier_call_chain kernel/notifier.c:93 [inline]
    blocking_notifier_call_chain+0x90/0x200 kernel/notifier.c:388
    __inet_del_ifa+0x4df/0x800 net/ipv4/devinet.c:432
    inet_del_ifa net/ipv4/devinet.c:469 [inline]
    inetdev_destroy net/ipv4/devinet.c:322 [inline]
    inetdev_event+0x553/0xaf0 net/ipv4/devinet.c:1606
    notifier_call_chain kernel/notifier.c:93 [inline]
    raw_notifier_call_chain+0x6b/0x1c0 kernel/notifier.c:461
    call_netdevice_notifiers_info net/core/dev.c:1962 [inline]
    call_netdevice_notifiers_mtu+0xd2/0x130 net/core/dev.c:2037
    dev_set_mtu_ext+0x30b/0x3e0 net/core/dev.c:8673
    do_setlink+0x5be/0x2430 net/core/rtnetlink.c:2837
    rtnl_setlink+0x255/0x300 net/core/rtnetlink.c:3177
    rtnetlink_rcv_msg+0x807/0x8c0 net/core/rtnetlink.c:6445
    netlink_rcv_skb+0x126/0x220 net/netlink/af_netlink.c:2549
    rtnetlink_rcv+0x1c/0x20 net/core/rtnetlink.c:6463
    netlink_unicast_kernel net/netlink/af_netlink.c:1339 [inline]
    netlink_unicast+0x56f/0x640 net/netlink/af_netlink.c:1365
    netlink_sendmsg+0x665/0x770 net/netlink/af_netlink.c:1914
    sock_sendmsg_nosec net/socket.c:725 [inline]
    sock_sendmsg net/socket.c:748 [inline]
    sock_write_iter+0x1aa/0x230 net/socket.c:1129
    do_iter_write+0x4b4/0x7b0 fs/read_write.c:860
    vfs_writev+0x1a8/0x320 fs/read_write.c:933
    do_writev+0xf8/0x220 fs/read_write.c:976
    __do_sys_writev fs/read_write.c:1049 [inline]
    __se_sys_writev fs/read_write.c:1046 [inline]
    __x64_sys_writev+0x45/0x50 fs/read_write.c:1046
    do_syscall_x64 arch/x86/entry/common.c:50 [inline]
    do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80
```

```
    entry_SYSCALL_64_after_hwframe+0x63/0xcd

    read to 0xffff888150f31744 of 1 bytes by task 21839 on cpu 1:
    fib_table_lookup+0x2bf/0xd50 net/ipv4/fib_trie.c:1585
    fib_lookup include/net/ip_fib.h:383 [inline]
    ip_route_output_key_hash_rcu+0x38c/0x12c0 net/ipv4/route.c:2751
    ip_route_output_key_hash net/ipv4/route.c:2641 [inline]
    __ip_route_output_key include/net/route.h:134 [inline]
    ip_route_output_flow+0xa6/0x150 net/ipv4/route.c:2869
    send4+0x1e7/0x500 drivers/net/wireguard/socket.c:61
    wg_socket_send_skb_to_peer+0x94/0x130 drivers/net/wireguard/socket.c:175
    wg_socket_send_buffer_to_peer+0xd6/0x100 drivers/net/wireguard/socket.c:200
    wg_packet_send_handshake_initiation drivers/net/wireguard/send.c:40 [inline]
    wg_packet_handshake_send_worker+0x10c/0x150 drivers/net/wireguard/send.c:51
    process_one_work+0x434/0x860 kernel/workqueue.c:2600
    worker_thread+0x5f2/0xa10 kernel/workqueue.c:2751
    kthread+0x1d7/0x210 kernel/kthread.c:389
    ret_from_fork+0x2e/0x40 arch/x86/kernel/process.c:145
    ret_from_fork_asm+0x11/0x20 arch/x86/entry/entry_64.S:304

    value changed: 0x00 -> 0x01

    Reported by Kernel Concurrency Sanitizer on:
    CPU: 1 PID: 21839 Comm: kworker/u4:18 Tainted: G W 6.5.0-syzkaller #0

    Fixes: dccd9ecc3744 ("ipv4: Do not use dead fib_info entries.")
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Link: https://lore.kernel.org/r/20230830095520.1046984-1-edumazet@google.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 59dcfbcc51512583a734525c4989f49f5044dc33
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 30 09:45:19 2023 +0000

    sctp: annotate data-races around sk->sk_wmem_queued

    [ Upstream commit dc9511dd6f37fe803f6b15b61b030728d7057417 ]

    sk->sk_wmem_queued can be read locklessly from sctp_poll()

    Use sk_wmem_queued_add() when the field is changed,
    and add READ_ONCE() annotations in sctp_writeable()
    and sctp_assocs_seq_show()

    syzbot reported:

    BUG: KCSAN: data-race in sctp_poll / sctp_wfree

    read-write to 0xffff888149d77810 of 4 bytes by interrupt on cpu 0:
    sctp_wfree+0x170/0x4a0 net/sctp/socket.c:9147
    skb_release_head_state+0xb7/0x1a0 net/core/skbuff.c:988
    skb_release_all net/core/skbuff.c:1000 [inline]
    __kfree_skb+0x16/0x140 net/core/skbuff.c:1016
    consume_skb+0x57/0x180 net/core/skbuff.c:1232
    sctp_chunk_destroy net/sctp/sm_make_chunk.c:1503 [inline]
    sctp_chunk_put+0xcd/0x130 net/sctp/sm_make_chunk.c:1530
    sctp_datamsg_put+0x29a/0x300 net/sctp/chunk.c:128
    sctp_chunk_free+0x34/0x50 net/sctp/sm_make_chunk.c:1515
    sctp_outq_sack+0xafa/0xd70 net/sctp/outqueue.c:1381
    sctp_cmd_process_sack net/sctp/sm_sideeffect.c:834 [inline]
    sctp_cmd_interpreter net/sctp/sm_sideeffect.c:1366 [inline]
    sctp_side_effects net/sctp/sm_sideeffect.c:1198 [inline]
```

```
     sctp_do_sm+0x12c7/0x31b0 net/sctp/sm_sideeffect.c:1169
     sctp_assoc_bh_rcv+0x2b2/0x430 net/sctp/associola.c:1051
     sctp_inq_push+0x108/0x120 net/sctp/inqueue.c:80
     sctp_rcv+0x116e/0x1340 net/sctp/input.c:243
     sctp6_rcv+0x25/0x40 net/sctp/ipv6.c:1120
     ip6_protocol_deliver_rcu+0x92f/0xf30 net/ipv6/ip6_input.c:437
     ip6_input_finish net/ipv6/ip6_input.c:482 [inline]
     NF_HOOK include/linux/netfilter.h:303 [inline]
     ip6_input+0xbd/0x1b0 net/ipv6/ip6_input.c:491
     dst_input include/net/dst.h:468 [inline]
     ip6_rcv_finish+0x1e2/0x2e0 net/ipv6/ip6_input.c:79
     NF_HOOK include/linux/netfilter.h:303 [inline]
     ipv6_rcv+0x74/0x150 net/ipv6/ip6_input.c:309
     __netif_receive_skb_one_core net/core/dev.c:5452 [inline]
     __netif_receive_skb+0x90/0x1b0 net/core/dev.c:5566
     process_backlog+0x21f/0x380 net/core/dev.c:5894
     __napi_poll+0x60/0x3b0 net/core/dev.c:6460
     napi_poll net/core/dev.c:6527 [inline]
     net_rx_action+0x32b/0x750 net/core/dev.c:6660
     __do_softirq+0xc1/0x265 kernel/softirq.c:553
     run_ksoftirqd+0x17/0x20 kernel/softirq.c:921
     smpboot_thread_fn+0x30a/0x4a0 kernel/smpboot.c:164
     kthread+0x1d7/0x210 kernel/kthread.c:389
     ret_from_fork+0x2e/0x40 arch/x86/kernel/process.c:145
     ret_from_fork_asm+0x11/0x20 arch/x86/entry/entry_64.S:304

     read to 0xffff888149d77810 of 4 bytes by task 17828 on cpu 1:
     sctp_writeable net/sctp/socket.c:9304 [inline]
     sctp_poll+0x265/0x410 net/sctp/socket.c:8671
     sock_poll+0x253/0x270 net/socket.c:1374
     vfs_poll include/linux/poll.h:88 [inline]
     do_pollfd fs/select.c:873 [inline]
     do_poll fs/select.c:921 [inline]
     do_sys_poll+0x636/0xc00 fs/select.c:1015
     __do_sys_ppoll fs/select.c:1121 [inline]
     __se_sys_ppoll+0x1af/0x1f0 fs/select.c:1101
     __x64_sys_ppoll+0x67/0x80 fs/select.c:1101
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x63/0xcd

     value changed: 0x00019e80 -> 0x0000cc80

     Reported by Kernel Concurrency Sanitizer on:
     CPU: 1 PID: 17828 Comm: syz-executor.1 Not tainted 6.5.0-rc7-syzkaller-00185-
  g28f20a19294d #0
     Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
  07/26/2023

     Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
     Reported-by: syzbot <syzkaller@googlegroups.com>
     Signed-off-by: Eric Dumazet <edumazet@google.com>
     Cc: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
     Acked-by: Xin Long <lucien.xin@gmail.com>
     Link: https://lore.kernel.org/r/20230830094519.950007-1-edumazet@google.com
     Signed-off-by: Paolo Abeni <pabeni@redhat.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit e093000e7d13569c9cb07d7500acd5142c3c43cb
  Author: Eric Dumazet <edumazet@google.com>
  Date:   Tue Aug 29 12:35:41 2023 +0000

     net/sched: fq_pie: avoid stalls in fq_pie_timer()

     [ Upstream commit 8c21ab1bae945686c602c5bfa4e3f3352c2452c5 ]
```

```
    When setting a high number of flows (limit being 65536),
    fq_pie_timer() is currently using too much time as syzbot reported.

    Add logic to yield the cpu every 2048 flows (less than 150 usec
    on debug kernels).
    It should also help by not blocking qdisc fast paths for too long.
    Worst case (65536 flows) would need 31 jiffies for a complete scan.

    Relevant extract from syzbot report:

    rcu: INFO: rcu_preempt detected expedited stalls on CPUs/tasks: { 0-.... } 2663 jiffies
s: 873 root: 0x1/.
    rcu: blocking rcu_node structures (internal RCU debug):
    Sending NMI from CPU 1 to CPUs 0:
    NMI backtrace for cpu 0
    CPU: 0 PID: 5177 Comm: syz-executor273 Not tainted 6.5.0-syzkaller-00453-g727dbda16b83
#0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
07/26/2023
    RIP: 0010:check_kcov_mode kernel/kcov.c:173 [inline]
    RIP: 0010:write_comp_data+0x21/0x90 kernel/kcov.c:236
    Code: 2e 0f 1f 84 00 00 00 00 00 65 8b 05 01 b2 7d 7e 49 89 f1 89 c6 49 89 d2 81 e6 00
01 00 00 49 89 f8 65 48 8b 14 25 80 b9 03 00 <a9> 00 01 ff 00 74 0e 85 f6 74 59 8b 82 04 16
00 00 85 c0 74 4f 8b
    RSP: 0018:ffffc90000007bb8 EFLAGS: 00000206
    RAX: 0000000000000101 RBX: ffffc9000dc0d140 RCX: ffffffff885893b0
    RDX: ffff88807c075940 RSI: 0000000000000100 RDI: 0000000000000001
    RBP: 0000000000000000 R08: 0000000000000001 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000000 R12: ffffc9000dc0d178
    R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
    FS:  0000555555d54380(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007f6b442f6130 CR3: 000000006fe1c000 CR4: 00000000003506f0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     <NMI>
     </NMI>
     <IRQ>
     pie_calculate_probability+0x480/0x850 net/sched/sch_pie.c:415
     fq_pie_timer+0x1da/0x4f0 net/sched/sch_fq_pie.c:387
     call_timer_fn+0x1a0/0x580 kernel/time/timer.c:1700

    Fixes: ec97ecf1ebe4 ("net: sched: add Flow Queue PIE packet scheduler")
    Link: https://lore.kernel.org/lkml/00000000000017ad3f06040bf394@google.com/
    Reported-by: syzbot+e46fbd5289363464bc13@syzkaller.appspotmail.com
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: Michal Kubiak <michal.kubiak@intel.com>
    Reviewed-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Link: https://lore.kernel.org/r/20230829123541.3745013-1-edumazet@google.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cdbb5e973faa4b6cc3e7f0c26ad893207880301f
Author: Katya Orlova <e.orlova@ispras.ru>
Date:   Tue Aug 15 16:38:31 2023 +0300

    smb: propagate error code of extract_sharename()

    [ Upstream commit efc0b0bcffcba60d9c6301063d25a22a4744b499 ]

    In addition to the EINVAL, there may be an ENOMEM.

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

```
    Fixes: 70431bfd825d ("cifs: Support fscache indexing rewrite")
    Signed-off-by: Katya Orlova <e.orlova@ispras.ru>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7f56ae02d97421814ce5ab473c3932185da2d603
Author: Phil Sutter <phil@nwl.cc>
Date:   Tue Aug 29 19:51:58 2023 +0200

    netfilter: nf_tables: Audit log rule reset

    [ Upstream commit ea078ae9108e25fc881c84369f7c03931d22e555 ]

    Resetting rules' stateful data happens outside of the transaction logic,
    so 'get' and 'dump' handlers have to emit audit log entries themselves.

    Fixes: 8daa8fde3fc3f ("netfilter: nf_tables: Introduce NFT_MSG_GETRULE_RESET")
    Signed-off-by: Phil Sutter <phil@nwl.cc>
    Reviewed-by: Richard Guy Briggs <rgb@redhat.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 51d68d752995a98408b84acea84b0013917536d3
Author: Phil Sutter <phil@nwl.cc>
Date:   Tue Aug 29 19:51:57 2023 +0200

    netfilter: nf_tables: Audit log setelem reset

    [ Upstream commit 7e9be1124dbe7888907e82cab20164578e3f9ab7 ]

    Since set element reset is not integrated into nf_tables' transaction
    logic, an explicit log call is needed, similar to NFT_MSG_GETOBJ_RESET
    handling.

    For the sake of simplicity, catchall element reset will always generate
    a dedicated log entry. This relieves nf_tables_dump_set() from having to
    adjust the logged element count depending on whether a catchall element
    was found or not.

    Fixes: 079cd633219d7 ("netfilter: nf_tables: Introduce NFT_MSG_GETSETELEM_RESET")
    Signed-off-by: Phil Sutter <phil@nwl.cc>
    Reviewed-by: Richard Guy Briggs <rgb@redhat.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8950b63645033a1745005365db6cc2f964a4d70e
Author: Yu Kuai <yukuai3@huawei.com>
Date:   Wed Aug 16 09:27:08 2023 +0800

    blk-throttle: consider 'carryover_ios/bytes' in throtl_trim_slice()

    [ Upstream commit eead0056648cef49d7b15c07ae612fa217083165 ]

    Currently, 'carryover_ios/bytes' is not handled in throtl_trim_slice(),
    for consequence, 'carryover_ios/bytes' will be used to throttle bio
    multiple times, for example:

    1) set iops limit to 100, and slice start is 0, slice end is 100ms;
    2) current time is 0, and 10 ios are dispatched, those io won't be
       throttled and io_disp is 10;
    3) still at current time 0, update iops limit to 1000, carryover_ios is
       updated to (0 - 10) = -10;
    4) in this slice(0 - 100ms), io_allowed = 100 + (-10) = 90, which means
       only 90 ios can be dispatched without waiting;
```

```
     5) assume that io is throttled in slice(0 - 100ms), and
        throtl_trim_slice() update silce to (100ms - 200ms). In this case,
        'carryover_ios/bytes' is not cleared and still only 90 ios can be
        dispatched between 100ms - 200ms.

     Fix this problem by updating 'carryover_ios/bytes' in
     throtl_trim_slice().

     Fixes: a880ae93e5b5 ("blk-throttle: fix io hung due to configuration updates")
     Reported-by: zhuxiaohui <zhuxiaohui.400@bytedance.com>
     Link: https://lore.kernel.org/all/20230812072116.42321-1-zhuxiaohui.400@bytedance.com/
     Signed-off-by: Yu Kuai <yukuai3@huawei.com>
     Acked-by: Tejun Heo <tj@kernel.org>
     Link: https://lore.kernel.org/r/20230816012708.1193747-5-yukuai1@huaweicloud.com
     Signed-off-by: Jens Axboe <axboe@kernel.dk>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3e76e05e3bdb0eba098dd211eda961aec64c1b47
Author: Yu Kuai <yukuai3@huawei.com>
Date:   Wed Aug 16 09:27:07 2023 +0800

     blk-throttle: use calculate_io/bytes_allowed() for throtl_trim_slice()

     [ Upstream commit e8368b57c006dc0e02dcd8a9dc9f2060ff5476fe ]

     There are no functional changes, just make the code cleaner.

     Signed-off-by: Yu Kuai <yukuai3@huawei.com>
     Acked-by: Tejun Heo <tj@kernel.org>
     Link: https://lore.kernel.org/r/20230816012708.1193747-4-yukuai1@huaweicloud.com
     Signed-off-by: Jens Axboe <axboe@kernel.dk>
     Stable-dep-of: eead0056648c ("blk-throttle: consider 'carryover_ios/bytes' in
throtl_trim_slice()")
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7fb464d52fa41c31a6fd1ad82888e67c65935d94
Author: Andrzej Hajda <andrzej.hajda@intel.com>
Date:   Mon Aug 21 17:30:35 2023 +0200

     drm/i915: mark requests for GuC virtual engines to avoid use-after-free

     [ Upstream commit 5eefc5307c983b59344a4cb89009819f580c84fa ]

     References to i915_requests may be trapped by userspace inside a
     sync_file or dmabuf (dma-resv) and held indefinitely across different
     proceses. To counter-act the memory leaks, we try to not to keep
     references from the request past their completion.
     On the other side on fence release we need to know if rq->engine
     is valid and points to hw engine (true for non-virtual requests).
     To make it possible extra bit has been added to rq->execution_mask,
     for marking virtual engines.

     Fixes: bcb9aa45d5a0 ("Revert "drm/i915: Hold reference to intel_context over life of
i915_request"")
     Signed-off-by: Chris Wilson <chris.p.wilson@linux.intel.com>
     Signed-off-by: Andrzej Hajda <andrzej.hajda@intel.com>
     Reviewed-by: Andi Shyti <andi.shyti@linux.intel.com>
     Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
     Link: https://patchwork.freedesktop.org/patch/msgid/20230821153035.3903006-1-
andrzej.hajda@intel.com
     (cherry picked from commit 280410677af763f3871b93e794a199cfcf6fb580)
     Signed-off-by: Rodrigo Vivi <rodrigo.vivi@intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 33da1bcbfa9fcc10de49f6f9ce4488630dc1d121
```

```
Author: Yonghong Song <yonghong.song@linux.dev>
Date:   Sun Aug 27 08:05:51 2023 -0700

    selftests/bpf: Fix flaky cgroup_iter_sleepable subtest

    [ Upstream commit 5439cfa7fe612e7d02d5a1234feda3fa6e483ba7 ]

    Occasionally, with './test_progs -j' on my vm, I will hit the
    following failure:

      test_cgrp_local_storage:PASS:join_cgroup /cgrp_local_storage 0 nsec
      test_cgroup_iter_sleepable:PASS:skel_open 0 nsec
      test_cgroup_iter_sleepable:PASS:skel_load 0 nsec
      test_cgroup_iter_sleepable:PASS:attach_iter 0 nsec
      test_cgroup_iter_sleepable:PASS:iter_create 0 nsec
      test_cgroup_iter_sleepable:FAIL:cgroup_id unexpected cgroup_id: actual 1 != expected
2812
      #48/5    cgrp_local_storage/cgroup_iter_sleepable:FAIL
      #48      cgrp_local_storage:FAIL

    Finally, I decided to do some investigation since the test is introduced
    by myself. It turns out the reason is due to cgroup_fd with value 0.
    In cgroup_iter, a cgroup_fd of value 0 means the root cgroup.

            /* from cgroup_iter.c */
            if (fd)
                    cgrp = cgroup_v1v2_get_from_fd(fd);
            else if (id)
                    cgrp = cgroup_get_from_id(id);
            else /* walk the entire hierarchy by default. */
                    cgrp = cgroup_get_from_path("/");

    That is why we got cgroup_id 1 instead of expected 2812.

    Why we got a cgroup_fd 0? Nobody should really touch 'stdin' (fd 0) in
    test_progs. I traced 'close' syscall with stack trace and found the root
    cause, which is a bug in bpf_obj_pinning.c. Basically, the code closed
    fd 0 although it should not. Fixing the bug in bpf_obj_pinning.c also
    resolved the above cgroup_iter_sleepable subtest failure.

    Fixes: 3b22f98e5a05 ("selftests/bpf: Add path_fd-based BPF_OBJ_PIN and BPF_OBJ_GET
tests")
    Signed-off-by: Yonghong Song <yonghong.song@linux.dev>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Link: https://lore.kernel.org/bpf/20230827150551.1743497-1-yonghong.song@linux.dev
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2d2b4d003a167d13bb5d7f4f418337108726ed2e
Author: Vincent Whitchurch <vincent.whitchurch@axis.com>
Date:   Tue Aug 29 16:04:12 2023 +0200

    regulator: tps6287x: Fix n_voltages

    [ Upstream commit c69290557c7571dff3d995fa27619b965915e8a1 ]

    There are 256 possible voltage settings for each range, not 256 possible
    voltage settings in total.

    Fixes: 15a1cd245d5b ("regulator: tps6287x: Fix missing .n_voltages setting")
    Signed-off-by: Vincent Whitchurch <vincent.whitchurch@axis.com
    Link: https://lore.kernel.org/r/20230829-tps-voltages-v1-1-7ba4f958a194@axis.com
    Signed-off-by: Mark Brown <broonie@kernel.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3f87e6f331d28af27c9a7a9aff32645c9bcccdc
```

```
Author: Namhyung Kim <namhyung@kernel.org>
Date:   Fri Aug 25 09:41:52 2023 -0700

    perf test stat_bpf_counters_cgrp: Enhance perf stat cgroup BPF counter test

    [ Upstream commit a84260e314029e6dc9904fd6eabf8d9fd7965351 ]

    It has system-wide test and cpu-list test but the cpu-list test fails
    sometimes.  It runs sleep command on CPU1 and measure both user.slice
    and system.slice cgroups by default (on systemd-based systems).

    But if the system was idle enough, sometime the system.slice gets no
    count and it makes the test failing.  Maybe that's because it only looks
    at the CPU1, let's add CPU0 to increase the chance it finds some tasks.

    Fixes: 7901086014bbaa3a ("perf test: Add a new test for perf stat cgroup BPF counter")
    Reported-by: Arnaldo Carvalho de Melo <acme@kernel.org>
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
    Link: https://lore.kernel.org/r/20230825164152.165610-3-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fe8b6e5994108d773df953d62945eb67908c8051
Author: Kajol Jain <kjain@linux.ibm.com>
Date:   Sun Jul 9 23:57:39 2023 +0530

    perf test stat_bpf_counters_cgrp: Fix shellcheck issue about logical operators

    [ Upstream commit 0dd1f815545d7210150642741c364521cc5cf116 ]

    Running shellcheck on lock_contention.sh generates below warning:

    In stat_bpf_counters_cgrp.sh line 28:
            if [ -d /sys/fs/cgroup/system.slice -a -d /sys/fs/cgroup/user.slice ]; then
                                                ^-- SC2166 (warning): Prefer [ p ] && [ q ]
as [ p -a q ] is not well defined.

    In stat_bpf_counters_cgrp.sh line 34:
            local self_cgrp=$(grep perf_event /proc/self/cgroup | cut -d: -f3)
            ^-------------^ SC3043 (warning): In POSIX sh, 'local' is undefined.
                  ^-------^ SC2155 (warning): Declare and assign separately to avoid masking
return values.
                            ^-- SC2046 (warning): Quote this to prevent word splitting.

    In stat_bpf_counters_cgrp.sh line 51:
            local output
            ^----------^ SC3043 (warning): In POSIX sh, 'local' is undefined.

    In stat_bpf_counters_cgrp.sh line 65:
            local output
            ^----------^ SC3043 (warning): In POSIX sh, 'local' is undefined.

    Fixed above warnings by:
    - Changing the expression [p -a q] to [p] && [q].
    - Fixing shellcheck warnings for local usage, by prefixing
      function name to the variable.

    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
```

```
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Disha Goel <disgoel@linux.vnet.ibm.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230709182800.53002-6-atrajeev@linux.vnet.ibm.com
    Signed-off-by: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Stable-dep-of: a84260e31402 ("perf test stat_bpf_counters_cgrp: Enhance perf stat cgroup
BPF counter test")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1fb133ca6d95a07f91674efcfa2ce4ca2ac88a7
Author: Miquel Raynal <miquel.raynal@bootlin.com>
Date:   Thu Aug 17 12:18:53 2023 +0200

    i3c: master: svc: Describe member 'saved_regs'

    [ Upstream commit 5496eac6ad7428fa06811a8c34b3a15beb93b86d ]

    The 'saved_regs' member of the 'svc_i3c_master' structure is not
    described in the kernel doc, which produces the following warning:

        Function parameter or member 'saved_regs' not described in 'svc_i3c_master'

    Add the missing line in the kernel documentation of the parent
    structure.

    Fixes: 1c5ee2a77b1b ("i3c: master: svc: fix i3c suspend/resume issue")
    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/oe-kbuild-all/202308171435.0xQ82lvu-lkp@intel.com/
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/r/20230817101853.16805-1-miquel.raynal@bootlin.com
    Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4e387cd28631721829eab19a4d9cb4b6fe49c63c
Author: Ian Rogers <irogers@google.com>
Date:   Thu Aug 24 19:39:57 2023 -0700

    perf header: Fix missing PMU caps

    [ Upstream commit 9897009eecae821efc684ecdd1d04584f5501509 ]

    PMU caps are written as HEADER_PMU_CAPS or for the special case of the
    PMU "cpu" as HEADER_CPU_PMU_CAPS. As the PMU "cpu" is special, and not
    any "core" PMU, the logic had become broken and core PMUs not called
    "cpu" were not having their caps written.

    This affects ARM and s390 non-hybrid PMUs.

    Simplify the PMU caps writing logic to scan one fewer time and to be
    more explicit in its behavior.

    Fixes: 178ddf3bad981380 ("perf header: Avoid hybrid PMU list in write_pmu_caps")
    Reported-by: Wei Li <liwei391@huawei.com>
    Signed-off-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Huacai Chen <chenhuacai@kernel.org>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: James Clark <james.clark@arm.com>
```

```
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: John Garry <john.g.garry@oracle.com>
    Cc: K Prateek Nayak <kprateek.nayak@amd.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Leo Yan <leo.yan@linaro.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Mike Leach <mike.leach@linaro.org>
    Cc: Ming Wang <wangming01@loongson.cn>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Ravi Bangoria <ravi.bangoria@amd.com>
    Cc: Sean Christopherson <seanjc@google.com>
    Cc: Suzuki Poulouse <suzuki.poulose@arm.com>
    Cc: Will Deacon <will@kernel.org>
    Cc: linux-arm-kernel@lists.infradead.org
    Link: https://lore.kernel.org/r/20230825024002.801955-2-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4f314ab4a5e78b6c5853c2633d15dc58e9323a88
Author: Justin Stitt <justinstitt@google.com>
Date:   Thu Aug 24 21:20:25 2023 +0000

    accel/ivpu: refactor deprecated strncpy

    [ Upstream commit 4b2fd81f2af7147e844ecec0c5c07a16bca6b86e ]

    `strncpy` is deprecated for use on NUL-terminated destination strings [1].

    A suitable replacement is `strscpy` [2] due to the fact that it
    guarantees NUL-termination on its destination buffer argument which is
    _not_ the case for `strncpy`!

    Also remove extraneous if-statement as it can never be entered. The
    return value from `strncpy` is it's first argument. In this case,
    `...dyndbg_cmd` is an array:
    |       char dyndbg_cmd[VPU_DYNDBG_CMD_MAX_LEN];
                 ^^^^^^^^^^
    This can never be NULL which means `strncpy`'s return value cannot be
    NULL here. Just use `strscpy` which is more robust and results in
    simpler and less ambiguous code.

    Moreover, remove needless `... - 1` as `strscpy`'s implementation
    ensures NUL-termination and we do not need to carefully dance around
    ending boundaries with a "- 1" anymore.

    Fixes: 5d7422cfb498 ("accel/ivpu: Add IPC driver and JSM messages")
    Link: www.kernel.org/doc/html/latest/process/deprecated.html#strncpy-on-nul-terminated-
strings [1]
    Link: https://manpages.debian.org/testing/linux-manual-4.8/strscpy.9.en.html [2]
    Link: https://github.com/KSPP/linux/issues/90
    Cc: linux-hardening@vger.kernel.org
    Signed-off-by: Justin Stitt <justinstitt@google.com>
    Reviewed-by: Stanislaw Gruszka <stanislaw.gruszka@linux.intel.com>
    Signed-off-by: Stanislaw Gruszka <stanislaw.gruszka@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230824-strncpy-drivers-accel-ivpu-
ivpu_jsm_msg-c-v1-1-12d9b52d2dff@google.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e3a0ddbaf7f1f9ffc070718b417461ced3268758
Author: Vladimir Zapolskiy <vz@mleia.com>
Date:   Mon Jul 17 17:52:57 2023 +0200

    pwm: lpc32xx: Remove handling of PWM channels
```

```
    [ Upstream commit 4aae44f65827f0213a7361cf9c32cfe06114473f ]

    Because LPC32xx PWM controllers have only a single output which is
    registered as the only PWM device/channel per controller, it is known in
    advance that pwm->hwpwm value is always 0. On basis of this fact
    simplify the code by removing operations with pwm->hwpwm, there is no
    controls which require channel number as input.

    Even though I wasn't aware at the time when I forward ported that patch,
    this fixes a null pointer dereference as lpc32xx->chip.pwms is NULL
    before devm_pwmchip_add() is called.

    Reported-by: Dan Carpenter <dan.carpenter@linaro.org>
    Signed-off-by: Vladimir Zapolskiy <vz@mleia.com>
    Signed-off-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Fixes: 3d2813fb17e5 ("pwm: lpc32xx: Don't modify HW state in .probe() after the PWM chip
was registered")
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f11ccbdb9e1693006e6dcf188b1998823e6bbb3f
Author: Ilkka Koskinen <ilkka@os.amperecomputing.com>
Date:   Thu Aug 3 14:13:28 2023 -0700

    perf vendor events arm64: Remove L1D_CACHE_LMISS from AmpereOne list

    [ Upstream commit b8af10062df3c23fe002c3f187389bb263b3eb20 ]

    amperene/cache.json file tried to include L1D_CACHE_LMISS while it
    doesn't exist in common-and-microarch.json. While this bug doesn't seem to
    cause issue in newer kernels with jevents.py script, it prevents building
    older perf tools with the backported patch.

    Fixes: a9650b7f6fc09d16 ("perf vendor events arm64: Add AmpereOne core PMU events")
    Reported-by: Dave Kleikamp <dave.kleikamp@oracle.com>
    Reviewed-by: Ian Rogers <irogers@google.com>
    Reviewed-by: John Garry <john.g.garry@oracle.com>
    Signed-off-by: Ilkka Koskinen <ilkka@os.amperecomputing.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Ilkka Koskinen <ilkka@os.amperecomputing.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: James Clark <james.clark@arm.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Leo Yan <leo.yan@linaro.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Mike Leach <mike.leach@linaro.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Will Deacon <will@kernel.org>
    Cc: linux-arm-kernel@lists.infradead.org
    Closes: https://lore.kernel.org/all/76bb2e47-ce44-76ae-838e-53279047084d@oracle.com/
    Link: https://lore.kernel.org/r/20230803211331.140553-2-ilkka@os.amperecomputing.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1edc8ab3dd8fe650700fe89d72f15897b6b7e7ab
Author: Raag Jadav <raag.jadav@intel.com>
Date:   Fri Aug 11 17:32:20 2023 +0530

    watchdog: intel-mid_wdt: add MODULE_ALIAS() to allow auto-load

    [ Upstream commit cf38e7691c85f1b09973b22a0b89bf1e1228d2f9 ]
```

```
    When built with CONFIG_INTEL_MID_WATCHDOG=m, currently the driver
    needs to be loaded manually, for the lack of module alias.
    This causes unintended resets in cases where watchdog timer is
    set-up by bootloader and the driver is not explicitly loaded.
    Add MODULE_ALIAS() to load the driver automatically at boot and
    avoid this issue.

    Fixes: 87a1ef8058d9 ("watchdog: add Intel MID watchdog driver support")
    Signed-off-by: Raag Jadav <raag.jadav@intel.com>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20230811120220.31578-1-raag.jadav@intel.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b12265092d33429d80ed07d93b50d02b4a6c2b02
Author: Arnaldo Carvalho de Melo <acme@redhat.com>
Date:   Thu Aug 17 09:11:21 2023 -0300

    perf lock: Don't pass an ERR_PTR() directly to perf_session__delete()

    [ Upstream commit abaf1e0355abb050f9c11d2d13a513caec80f7ad ]

    While debugging a segfault on 'perf lock contention' without an
    available perf.data file I noticed that it was basically calling:

            perf_session__delete(ERR_PTR(-1))

    Resulting in:

      (gdb) run lock contention
      Starting program: /root/bin/perf lock contention
      [Thread debugging using libthread_db enabled]
      Using host libthread_db library "/lib64/libthread_db.so.1".
      failed to open perf.data: No such file or directory  (try 'perf record' first)
      Initializing perf session failed

      Program received signal SIGSEGV, Segmentation fault.
      0x00000000005e7515 in auxtrace__free (session=0xffffffffffffffff) at
util/auxtrace.c:2858
      2858          if (!session->auxtrace)
      (gdb) p session
      $1 = (struct perf_session *) 0xffffffffffffffff
      (gdb) bt
      #0  0x00000000005e7515 in auxtrace__free (session=0xffffffffffffffff) at
util/auxtrace.c:2858
      #1  0x000000000057bb4d in perf_session__delete (session=0xffffffffffffffff) at
util/session.c:300
      #2  0x000000000047c421 in __cmd_contention (argc=0, argv=0x7fffffffe200) at builtin-
lock.c:2161
      #3  0x000000000047dc95 in cmd_lock (argc=0, argv=0x7fffffffe200) at builtin-
lock.c:2604
      #4  0x0000000000501466 in run_builtin (p=0xe597a8 <commands+552>, argc=2,
argv=0x7fffffffe200) at perf.c:322
      #5  0x00000000005016d5 in handle_internal_command (argc=2, argv=0x7fffffffe200) at
perf.c:375
      #6  0x0000000000501824 in run_argv (argcp=0x7fffffffe02c, argv=0x7fffffffe020) at
perf.c:419
      #7  0x0000000000501b11 in main (argc=2, argv=0x7fffffffe200) at perf.c:535
      (gdb)

    So just set it to NULL after using PTR_ERR(session) to decode the error
    as perf_session__delete(NULL) is supported.
```

```
    Fixes: eef4fee5e52071d5 ("perf lock: Dynamically allocate lockhash_table")
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Andi Kleen <ak@linux.intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: K Prateek Nayak <kprateek.nayak@amd.com>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Leo Yan <leo.yan@linaro.org>
    Cc: Mamatha Inamdar <mamatha4@linux.vnet.ibm.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Masami Hiramatsu <mhiramat@kernel.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Paolo Bonzini <pbonzini@redhat.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Ravi Bangoria <ravi.bangoria@amd.com>
    Cc: Ross Zwisler <zwisler@chromium.org>
    Cc: Sean Christopherson <seanjc@google.com>
    Cc: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Cc: Tiezhu Yang <yangtiezhu@loongson.cn>
    Cc: Yang Jihong <yangjihong1@huawei.com>
    Link: https://lore.kernel.org/lkml/ZN4R1AYfsD2J8lRs@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6b66e3a098bb35cd9fba146720806bbf2b361e74
Author: Arnaldo Carvalho de Melo <acme@redhat.com>
Date:   Thu Aug 17 09:11:21 2023 -0300

    perf top: Don't pass an ERR_PTR() directly to perf_session__delete()

    [ Upstream commit ef23cb593304bde0cc046fd4cc83ae7ea2e24f16 ]

    While debugging a segfault on 'perf lock contention' without an
    available perf.data file I noticed that it was basically calling:

            perf_session__delete(ERR_PTR(-1))

    Resulting in:

      (gdb) run lock contention
      Starting program: /root/bin/perf lock contention
      [Thread debugging using libthread_db enabled]
      Using host libthread_db library "/lib64/libthread_db.so.1".
      failed to open perf.data: No such file or directory  (try 'perf record' first)
      Initializing perf session failed

      Program received signal SIGSEGV, Segmentation fault.
      0x00000000005e7515 in auxtrace__free (session=0xffffffffffffffff) at
util/auxtrace.c:2858
      2858          if (!session->auxtrace)
      (gdb) p session
      $1 = (struct perf_session *) 0xffffffffffffffff
      (gdb) bt
      #0  0x00000000005e7515 in auxtrace__free (session=0xffffffffffffffff) at
util/auxtrace.c:2858
      #1  0x000000000057bb4d in perf_session__delete (session=0xffffffffffffffff) at
util/session.c:300
      #2  0x000000000047c421 in __cmd_contention (argc=0, argv=0x7fffffffe200) at builtin-
lock.c:2161
      #3  0x000000000047dc95 in cmd_lock (argc=0, argv=0x7fffffffe200) at builtin-
lock.c:2604
      #4  0x0000000000501466 in run_builtin (p=0xe597a8 <commands+552>, argc=2,
argv=0x7fffffffe200) at perf.c:322
```

```
       #5  0x00000000005016d5 in handle_internal_command (argc=2, argv=0x7fffffffe200) at
 perf.c:375
       #6  0x0000000000501824 in run_argv (argcp=0x7fffffffe02c, argv=0x7fffffffe020) at
 perf.c:419
       #7  0x0000000000501b11 in main (argc=2, argv=0x7fffffffe200) at perf.c:535
       (gdb)

    So just set it to NULL after using PTR_ERR(session) to decode the error
    as perf_session__delete(NULL) is supported.

    The same problem was found in 'perf top' after an audit of all
    perf_session__new() failure handling.

    Fixes: 6ef81c55a2b6584c ("perf session: Return error code for perf_session__new()
 function on failure")
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Alexey Budankov <alexey.budankov@linux.intel.com>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Jeremie Galarneau <jeremie.galarneau@efficios.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kate Stewart <kstewart@linuxfoundation.org>
    Cc: Mamatha Inamdar <mamatha4@linux.vnet.ibm.com>
    Cc: Mukesh Ojha <mojha@codeaurora.org>
    Cc: Nageswara R Sastry <rnsastry@linux.vnet.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Ravi Bangoria <ravi.bangoria@linux.ibm.com>
    Cc: Shawn Landden <shawn@git.icu>
    Cc: Song Liu <songliubraving@fb.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Tzvetomir Stoyanov <tstoyanov@vmware.com>
    Link: https://lore.kernel.org/lkml/ZN4Q2rxxsL08A8rd@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 2405d1eb5e9c046be15f1d3dc1ec8440fe81ff1b
 Author: Kajol Jain <kjain@linux.ibm.com>
 Date:   Mon Aug 14 16:58:02 2023 +0530

    perf vendor events: Update metric event names for power10 platform

    [ Upstream commit edd65d2bc55fb84d7b80c2ffe3b74d9b11ac4e2f ]

    Update metric event name for some of the JSON/metric events for
    power10 platform.

    Fixes: 3ca3af7d1f230d1f ("perf vendor events power10: Add metric events JSON file for
 power10 platform")
    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230814112803.1508296-6-kjain@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit f31933014c280a654c79fd809004a34498a874ba
 Author: Kajol Jain <kjain@linux.ibm.com>
 Date:   Mon Aug 14 16:58:00 2023 +0530
```

```
    perf vendor events: Move JSON/events to appropriate files for power10 platform

    [ Upstream commit 7d473f475b2aff7e7c5d63b6f701c54590f84781 ]

    Move some of the power10 JSON/events to appropriate files.

    Fixes: 32daa5d7899e0343 ("perf vendor events: Initial JSON/events list for power10
platform")
    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230814112803.1508296-4-kjain@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d3b9594c82615d5c8e7e66a126e56ac54e80454f
Author: Kajol Jain <kjain@linux.ibm.com>
Date:   Mon Aug 14 16:57:59 2023 +0530

    perf vendor events: Drop STORES_PER_INST metric event for power10 platform

    [ Upstream commit 4836b9a85ef148c7c9779b66fab3f7279e488d90 ]

    Drop STORES_PER_INST metric event for the power10 platform, as the
    metric expression of STORES_PER_INST metric event using dropped event
    PM_ST_FIN.

    Fixes: 3ca3af7d1f230d1f ("perf vendor events power10: Add metric events JSON file for
power10 platform")
    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230814112803.1508296-3-kjain@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f85bca192b9fc37938636bfffc72c69658afd0e7
Author: Kajol Jain <kjain@linux.ibm.com>
Date:   Mon Aug 14 16:57:58 2023 +0530

    perf vendor events: Drop some of the JSON/events for power10 platform

    [ Upstream commit e104df97b8dcfbab2e42de634b99bf03f0805d85 ]

    Drop some of the JSON/events for power10 platform due to counter
    data mismatch.

    Fixes: 32daa5d7899e0343 ("perf vendor events: Initial JSON/events list for power10
platform")
    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
```

```
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230814112803.1508296-2-kjain@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1d160e27a6461c9afc805feb817e723a28d8e9f2
Author: Kajol Jain <kjain@linux.ibm.com>
Date:   Mon Aug 14 16:57:57 2023 +0530

    perf vendor events: Update the JSON/events descriptions for power10 platform

    [ Upstream commit 3286f88f31da060ac2789cee247153961ba57e49 ]

    Update the description for some of the JSON/events for power10 platform.

    Fixes: 32daa5d7899e0343 ("perf vendor events: Initial JSON/events list for power10
platform")
    Signed-off-by: Kajol Jain <kjain@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Link: https://lore.kernel.org/r/20230814112803.1508296-1-kjain@linux.ibm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dd6df6c9c2f53f0c1812c9c46ab3b94ec159ef28
Author: Adrian Hunter <adrian.hunter@intel.com>
Date:   Mon Jul 31 12:18:57 2023 +0300

    perf dlfilter: Add al_cleanup()

    [ Upstream commit 82b0a10390e5f198a4e23c9cc6a7307d2cf099f3 ]

    Add perf_dlfilter_fns.al_cleanup() to do addr_location__exit() on data
    passed via perf_dlfilter_fns.resolve_address().

    Add dlfilter-test-api-v2 to the "dlfilter C API" test to test it.

    Update documentation, clarifying that data returned by APIs should not
    be dereferenced after filter_event() and filter_event_early() return.

    Fixes: 0dd5041c9a0eaf8c ("perf addr_location: Add init/exit/copy functions")
    Reviewed-by: Ian Rogers <irogers@google.com>
    Signed-off-by: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Link: https://lore.kernel.org/r/20230731091857.10681-3-adrian.hunter@intel.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb95d578d172bd3285f7fa96c0d3ae2fe0cf44a5
Author: Arnaldo Carvalho de Melo <acme@kernel.org>
Date:   Mon Jul 31 12:18:56 2023 +0300

    perf dlfilter: Initialize addr_location before passing it to thread__find_symbol_fb()

    [ Upstream commit 42c6dd9d23019ff339d0aca80a444eb71087050e ]

    As thread__find_symbol_fb() will end up calling thread__find_map() and
    it in turn will call these on uninitialized memory:
```

```
            maps__zput(al->maps);
            map__zput(al->map);
            thread__zput(al->thread);
```

    Fixes: 0dd5041c9a0eaf8c ("perf addr_location: Add init/exit/copy functions")
    Reviewed-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Disha Goel <disgoel@linux.vnet.ibm.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kajol Jain <kjain@linux.ibm.com>
    Cc: Madhavan Srinivasan <maddy@linux.ibm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Link: https://lore.kernel.org/r/20230731091857.10681-2-adrian.hunter@intel.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cc91cf171b8d6ab4acd394b3f287f2d691159e11
Author: Namhyung Kim <namhyung@kernel.org>
Date: Thu Aug 10 19:58:21 2023 -0700

    perf bpf-filter: Fix sample flag check with ||

    [Upstream commit dc7f01f1bceca38839992b3371e0be8a3c9d5acf]

    For logical OR operator, the actual sample_flags are in the 'groups'
    list so it needs to check entries in the list instead. Otherwise it
    would show the following error message.

      $ sudo perf record -a -e cycles:p --filter 'period > 100 || weight > 0' sleep 1
      Error: cycles:p event does not have sample flags 0
      failed to set filter "BPF" on event cycles:p with 2 (No such file or directory)

    Actually it should warn on 'weight' is used without WEIGHT flag.

      Error: cycles:p event does not have PERF_SAMPLE_WEIGHT
       Hint: please add -W option to perf record
      failed to set filter "BPF" on event cycles:p with 2 (No such file or directory)

    Fixes: 4310551b76e0d676 ("perf bpf filter: Show warning for missing sample flags")
    Reviewed-by: Ian Rogers <irogers@google.com>
    Signed-off-by: Namhyung Kim <namhyung@kernel.org>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ingo Molnar <mingo@kernel.org>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link: https://lore.kernel.org/r/20230811025822.3859771-1-namhyung@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6125d4cc36e94bbac7dd292b19d79e8d9042271f
Author: Ivan Babrou <ivan@cloudflare.com>
Date: Mon Jul 17 17:07:37 2023 -0700

    perf script: Print "cgroup" field on the same line as "comm"

    [Upstream commit 8c49c6e1a7b790c4cb9f464c5485117451d91c60]

    Commit 3fd7a168bf51 ("perf script: Add 'cgroup' field for output")
    added support for printing cgroup path in perf script output.

    It was okay if you didn't want any stacks:

```
        $ sudo perf script --comms jpegtran:23f4bf -F comm,tid,cpu,time,cgroup
        jpegtran:23f4bf 3321915 [013] 404718.587488:  /idle.slice/polish.service
        jpegtran:23f4bf 3321915 [031] 404718.592073:  /idle.slice/polish.service

    With stacks it gets messier as cgroup is printed after the stack:

        $ perf script --comms jpegtran:23f4bf -F comm,tid,cpu,time,cgroup,ip,sym
        jpegtran:23f4bf 3321915 [013] 404718.587488:
                        5c554 compress_output
                        570d9 jpeg_finish_compress
                        3476e jpegtran_main
                        330ee jpegtran::main
                        326e2 core::ops::function::FnOnce::call_once (inlined)
                        326e2 std::sys_common::backtrace::__rust_begin_short_backtrace
        /idle.slice/polish.service
        jpegtran:23f4bf 3321915 [031] 404718.592073:
                        8474d jsimd_encode_mcu_AC_first_prepare_sse2.PADDING
                    55af68e62fff [unknown]
        /idle.slice/polish.service

    Let's instead print cgroup on the same line as comm:

        $ perf script --comms jpegtran:23f4bf -F comm,tid,cpu,time,cgroup,ip,sym
        jpegtran:23f4bf 3321915 [013] 404718.587488:  /idle.slice/polish.service
                        5c554 compress_output
                        570d9 jpeg_finish_compress
                        3476e jpegtran_main
                        330ee jpegtran::main
                        326e2 core::ops::function::FnOnce::call_once (inlined)
                        326e2 std::sys_common::backtrace::__rust_begin_short_backtrace

        jpegtran:23f4bf 3321915 [031] 404718.592073:  /idle.slice/polish.service
                        8474d jsimd_encode_mcu_AC_first_prepare_sse2.PADDING
                    55af68e62fff [unknown]

    Fixes: 3fd7a168bf514979 ("perf script: Add 'cgroup' field for output")
    Signed-off-by: Ivan Babrou <ivan@cloudflare.com>
    Acked-by: Ian Rogers <irogers@google.com>
    Acked-by: Namhyung Kim <namhyung@kernel.org>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: kernel-team@cloudflare.com
    Link: https://lore.kernel.org/r/20230718000737.49077-1-ivan@cloudflare.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 09af247f5bd610f60f7245e04cf5f89e2756b634
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 21 13:18:52 2023 -0700

    x86/virt: Drop unnecessary check on extended CPUID level in cpu_has_svm()

    [ Upstream commit 5df8ecfe3632d5879d1f154f7aa8de441b5d1c89 ]

    Drop the explicit check on the extended CPUID level in cpu_has_svm(), the
    kernel's cached CPUID info will leave the entire SVM leaf unset if said
    leaf is not supported by hardware.  Prior to using cached information,
    the check was needed to avoid false positives due to Intel's rather crazy
    CPUID behavior of returning the values of the maximum supported leaf if
    the specified leaf is unsupported.
```

```
    Fixes: 682a8108872f ("x86/kvm/svm: Simplify cpu_has_svm()")
    Link: https://lore.kernel.org/r/20230721201859.2307736-13-seanjc@google.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 95101e1cf94abc69ba158f0fbd4777784e5a0ea0
Author: Arnaldo Carvalho de Melo <acme@redhat.com>
Date:   Wed Aug 2 18:22:14 2023 -0300

    perf annotate bpf: Don't enclose non-debug code with an assert()

    [ Upstream commit 979e9c9fc9c2a761303585e07fe2699bdd88182f ]

    In 616b14b47a86d880 ("perf build: Conditionally define NDEBUG") we
    started using NDEBUG=1 when DEBUG=1 isn't present, so code that is
    enclosed with assert() is not called.

    In dd317df072071903 ("perf build: Make binutil libraries opt in") we
    stopped linking against binutils-devel, for licensing reasons.

    Recently people asked me why annotation of BPF programs wasn't working,
    i.e. this:

      $ perf annotate bpf_prog_5280546344e3f45c_kfree_skb

    was returning:

      case SYMBOL_ANNOTATE_ERRNO__NO_LIBOPCODES_FOR_BPF:
         scnprintf(buf, buflen, "Please link with binutils's libopcode to enable BPF
annotation");

    This was on a fedora rpm, so its new enough that I had to try to test by
    rebuilding using BUILD_NONDISTRO=1, only to get it segfaulting on me.

    This combination made this libopcode function not to be called:

            assert(bfd_check_format(bfdf, bfd_object));

    Changing it to:

            if (!bfd_check_format(bfdf, bfd_object))
                    abort();

    Made it work, looking at this "check" function made me realize it
    changes the 'bfdf' internal state, i.e. we better call it.

    So stop using assert() on it, just call it and abort if it fails.

    Probably it is better to propagate the error, etc, but it seems it is
    unlikely to fail from the usage done so far and we really need to stop
    using libopcodes, so do the quick fix above and move on.

    With it we have BPF annotation back working when built with
    BUILD_NONDISTRO=1:

      ⬢[acme@toolbox perf-tools-next]$ perf annotate --stdio2
bpf_prog_5280546344e3f45c_kfree_skb   | head
      No kallsyms or vmlinux with build-id 939bc71a1a51cdc434e60af93c7e734f7d5c0e7e was
found
      Samples: 12  of event 'cpu-clock:ppp', 4000 Hz, Event count (approx.): 3000000,
[percent: local period]
      bpf_prog_5280546344e3f45c_kfree_skb() bpf_prog_5280546344e3f45c_kfree_skb
      Percent      int kfree_skb(struct trace_event_raw_kfree_skb *args) {
                     nop
       33.33         xchg   %ax,%ax
```

```
                    push   %rbp
                    mov    %rsp,%rbp
                    sub    $0x180,%rsp
                    push   %rbx
                    push   %r13
      ⬢[acme@toolbox perf-tools-next]$
```

```
    Fixes: 6987561c9e86eace ("perf annotate: Enable annotation of BPF programs")
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Mohamed Mahmoud <mmahmoud@redhat.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Dave Tucker <datucker@redhat.com>
    Cc: Derek Barbosa <debarbos@redhat.com>
    Cc: Song Liu <songliubraving@fb.com>
    Link: https://lore.kernel.org/lkml/ZMrMzoQBe0yqMek1@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 52385cfce5be36e6fb222f58a96b73d27d7644ec
Author: Dmitry Torokhov <dmitry.torokhov@gmail.com>
Date:   Sun Jul 23 22:30:20 2023 -0700

    Input: tca6416-keypad - fix interrupt enable disbalance

    [ Upstream commit cc141c35af873c6796e043adcb820833bd8ef8c5 ]

    The driver has been switched to use IRQF_NO_AUTOEN, but in the error
    unwinding and remove paths calls to enable_irq() were left in place, which
    will lead to an incorrect enable counter value.

    Fixes: bcd9730a04a1 ("Input: move to use request_irq by IRQF_NO_AUTOEN flag")
    Link: https://lore.kernel.org/r/20230724053024.352054-3-dmitry.torokhov@gmail.com
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e1c2e3c03ae68777c929079067f96cf63fac68b7
Author: Dmitry Torokhov <dmitry.torokhov@gmail.com>
Date:   Sun Jul 23 22:30:18 2023 -0700

    Input: tca6416-keypad - always expect proper IRQ number in i2c client

    [ Upstream commit 687fe7dfb736b03ab820d172ea5dbfc1ec447135 ]

    Remove option having i2c client contain raw gpio number instead of proper
    IRQ number. There are no users of this facility in mainline and it will
    allow cleaning up the driver code with regard to wakeup handling, etc.

    Link: https://lore.kernel.org/r/20230724053024.352054-1-dmitry.torokhov@gmail.com
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Stable-dep-of: cc141c35af87 ("Input: tca6416-keypad - fix interrupt enable disbalance")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f518e18b2c13ab3edf38325e5ebdbe9ff78e440e
Author: Sean Christopherson <seanjc@google.com>
Date:   Thu Jun 15 16:37:56 2023 +1000

    KVM: SVM: Don't defer NMI unblocking until next exit for SEV-ES guests

    [ Upstream commit 389fbbec261b2842fd0e34b26a2b288b122cc406 ]

    Immediately mark NMIs as unmasked in response to #VMGEXIT(NMI complete)
    instead of setting awaiting_iret_completion and waiting until the *next*
    VM-Exit to unmask NMIs.  The whole point of "NMI complete" is that the
```

```
    guest is responsible for telling the hypervisor when it's safe to inject
    an NMI, i.e. there's no need to wait.  And because there's no IRET to
    single-step, the next VM-Exit could be a long time coming, i.e. KVM could
    incorrectly hold an NMI pending for far longer than what is required and
    expected.

    Opportunistically fix a stale reference to HF_IRET_MASK.

    Fixes: 916b54a7688b ("KVM: x86: Move HF_NMI_MASK and HF_IRET_MASK into "struct
vcpu_svm"")
    Fixes: 4444dfe4050b ("KVM: SVM: Add NMI support for an SEV-ES guest")
    Cc: Tom Lendacky <thomas.lendacky@amd.com>
    Link: https://lore.kernel.org/r/20230615063757.3039121-9-aik@amd.com
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6f9da1f65fd8014cfcef7533703981c6e6d2d9fb
Author: Ian Rogers <irogers@google.com>
Date:   Tue Jun 27 11:10:27 2023 -0700

    perf parse-events: Additional error reporting

    [ Upstream commit b30d4f0b695428f513c561eeaea52e042ef48550 ]

    When no events or PMUs match report an error for event_pmu:

    Before:
    ```
    $ perf stat -e 'asdfasdf' -a sleep 1
    Run 'perf list' for a list of valid events

     Usage: perf stat [<options>] [<command>]

        -e, --event <event>   event selector. use 'perf list' to list available events
    ```

    After:
    ```
    $ perf stat -e 'asdfasdf' -a sleep 1
    event syntax error: 'asdfasdf'
                         ___ Bad event name

    Unabled to find PMU or event on a PMU of 'asdfasdf'
    Run 'perf list' for a list of valid events

     Usage: perf stat [<options>] [<command>]

        -e, --event <event>   event selector. use 'perf list' to list available events
    ```

    Fixes the inadvertent removal when hybrid parsing was modified.

    Fixes: 70c90e4a6b2fbe77 ("perf parse-events: Avoid scanning PMUs before parsing")
    Signed-off-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
```

```
    Link: https://lore.kernel.org/r/20230627181030.95608-11-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit be3835ca6a3e612a67e4c62c77fa030d4a14c7fe
Author: Ian Rogers <irogers@google.com>
Date:   Tue Jun 27 11:10:26 2023 -0700

    perf parse-events: Separate ENOMEM memory handling

    [ Upstream commit b52cb995f1a559bc6e1a7cdc0ed0375503528541 ]

    Add PE_ABORT that will YYNOMEM or YYABORT accordingly.

    Signed-off-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
    Link: https://lore.kernel.org/r/20230627181030.95608-10-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Stable-dep-of: b30d4f0b6954 ("perf parse-events: Additional error reporting")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0e8501c8a936e0b2ff4ffdff2c4866ba74969189
Author: Ian Rogers <irogers@google.com>
Date:   Tue Jun 27 11:10:25 2023 -0700

    perf parse-events: Move instances of YYABORT to YYNOMEM

    [ Upstream commit 77cdd787fc45e3426b8e0b5038b85c276540dfb4 ]

    Migration to improve error reporting as YYABORT cases should carry
    event parsing errors.

    Signed-off-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
    Link: https://lore.kernel.org/r/20230627181030.95608-9-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Stable-dep-of: b30d4f0b6954 ("perf parse-events: Additional error reporting")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7e7bca9dd3e7fe5cc4c7ef78610c9cb0ce93c9fd
Author: Ian Rogers <irogers@google.com>
Date:   Tue Jun 27 11:10:24 2023 -0700

    perf parse-events: Separate YYABORT and YYNOMEM cases

    [ Upstream commit a7a3252dad354a9e5c173156dab959e4019b9467 ]
```

```
    Split cases in event_pmu for greater accuracy.

    Signed-off-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Athira Rajeev <atrajeev@linux.vnet.ibm.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: bpf@vger.kernel.org
    Link: https://lore.kernel.org/r/20230627181030.95608-8-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Stable-dep-of: b30d4f0b6954 ("perf parse-events: Additional error reporting")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2a6a2572a0f77874c008a723b4fbe4965163b5f2
Author: Ying Liu <victor.liu@nxp.com>
Date:   Fri Jul 21 09:29:03 2023 +0000

    backlight: gpio_backlight: Drop output GPIO direction check for initial power state

    [ Upstream commit fe1328b5b2a087221e31da77e617f4c2b70f3b7f ]

    So, let's drop output GPIO direction check and only check GPIO value to set
    the initial power state.

    Fixes: 706dc68102bc ("backlight: gpio: Explicitly set the direction of the GPIO")
    Signed-off-by: Liu Ying <victor.liu@nxp.com>
    Reviewed-by: Andy Shevchenko <andy@kernel.org>
    Acked-by: Linus Walleij <linus.walleij@linaro.org>
    Acked-by: Bartosz Golaszewski <bartosz.golaszewski@linaro.org>
    Link: https://lore.kernel.org/r/20230721093342.1532531-1-victor.liu@nxp.com
    Signed-off-by: Lee Jones <lee@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f60b2ca659d69313595f291354c1a3da8221a7c0
Author: Artur Weber <aweber.kernel@gmail.com>
Date:   Fri Jul 14 14:14:39 2023 +0200

    backlight: lp855x: Initialize PWM state on first brightness change

    [ Upstream commit 4c09e20b3c85f60353ace21092e34f35f5e3ab00 ]

    As pointed out by Uwe Kleine-König[1], the changes introduced in
    commit c1ff7da03e16 ("video: backlight: lp855x: Get PWM for PWM mode
    during probe") caused the PWM state set up by the bootloader to be
    re-set when the driver is probed. This differs from the behavior from
    before that patch, where the PWM state would be initialized on the
    first brightness change.

    Fix this by moving the PWM state initialization into the PWM control
    function. Add a new variable, needs_pwm_init, to the device info struct
    to allow us to check whether we need the initialization, or whether it
    has already been done.

    [1] https://lore.kernel.org/lkml/20230614083953.e4kkweddjz7wztby@pengutronix.de/

    Fixes: c1ff7da03e16 ("video: backlight: lp855x: Get PWM for PWM mode during probe")
    Signed-off-by: Artur Weber <aweber.kernel@gmail.com>
    Reviewed-by: Daniel Thompson <daniel.thompson@linaro.org>
    Link: https://lore.kernel.org/r/20230714121440.7717-2-aweber.kernel@gmail.com
    Signed-off-by: Lee Jones <lee@kernel.org>
```

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

commit 053a8ea0be4d96713716393df63638704c2b760c
Author: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
Date:   Wed Jul 19 21:20:10 2023 +0200

    pwm: atmel-tcb: Fix resource freeing in error path and remove

    [ Upstream commit c11622324c023415fb69196c5fc3782d2b8cced0 ]

    Several resources were not freed in the error path and the remove
    function. Add the forgotten items.

    Fixes: 34cbcd72588f ("pwm: atmel-tcb: Add sama5d2 support")
    Fixes: 061f8572a31c ("pwm: atmel-tcb: Switch to new binding")
    Signed-off-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Reviewed-by: Claudiu Beznea <claudiu.beznea@tuxon.dev>
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 13ffd3dd8fd4f7605eb55ed09d2305592775ae25
Author: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
Date:   Wed Jul 19 21:20:09 2023 +0200

    pwm: atmel-tcb: Harmonize resource allocation order

    [ Upstream commit 0323e8fedd1ef25342cf7abf3a2024f5670362b8 ]

    Allocate driver data as first resource in the probe function. This way it
    can be used during allocation of the other resources (instead of assigning
    these to local variables first and update driver data only when it's
    allocated). Also as driver data is allocated using a devm function this
    should happen first to have the order of freeing resources in the error
    path and the remove function in reverse.

    Signed-off-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Stable-dep-of: c11622324c02 ("pwm: atmel-tcb: Fix resource freeing in error path and
remove")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 27f396f64537b1ae48d0644d7cbf0d250b3c0b33
Author: Arnaldo Carvalho de Melo <acme@redhat.com>
Date:   Wed Jul 19 15:37:14 2023 -0300

    perf trace: Really free the evsel->priv area

    [ Upstream commit 7962ef13651a9163f07b530607392ea123482e8a ]

    In 3cb4d5e00e037c70 ("perf trace: Free syscall tp fields in
    evsel->priv") it only was freeing if strcmp(evsel->tp_format->system,
    "syscalls") returned zero, while the corresponding initialization of
    evsel->priv was being performed if it was _not_ zero, i.e. if the tp
    system wasn't 'syscalls'.

    Just stop looking for that and free it if evsel->priv was set, which
    should be equivalent.

    Also use the pre-existing evsel_trace__delete() function.

    This resolves these leaks, detected with:

      $ make EXTRA_CFLAGS="-fsanitize=address" BUILD_BPF_SKEL=1 CORESIGHT=1
O=/tmp/build/perf-tools-next -C tools/perf install-bin
```

```
      =================================================================
      ==481565==ERROR: LeakSanitizer: detected memory leaks

      Direct leak of 40 byte(s) in 1 object(s) allocated from:
          #0 0x7f7343cba097 in calloc (/lib64/libasan.so.8+0xba097)
          #1 0x987966 in zalloc (/home/acme/bin/perf+0x987966)
          #2 0x52f9b9 in evsel_trace__new /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:307
          #3 0x52f9b9 in evsel__syscall_tp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:333
          #4 0x52f9b9 in evsel__init_raw_syscall_tp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:458
          #5 0x52f9b9 in perf_evsel__raw_syscall_newtp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:480
          #6 0x540e8b in trace__add_syscall_newtp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:3212
          #7 0x540e8b in trace__run /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:3891
          #8 0x540e8b in cmd_trace /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:5156
          #9 0x5ef262 in run_builtin /home/acme/git/perf-tools-next/tools/perf/perf.c:323
          #10 0x4196da in handle_internal_command /home/acme/git/perf-tools-
next/tools/perf/perf.c:377
          #11 0x4196da in run_argv /home/acme/git/perf-tools-next/tools/perf/perf.c:421
          #12 0x4196da in main /home/acme/git/perf-tools-next/tools/perf/perf.c:537
          #13 0x7f7342c4a50f in __libc_start_call_main (/lib64/libc.so.6+0x2750f)

      Direct leak of 40 byte(s) in 1 object(s) allocated from:
          #0 0x7f7343cba097 in calloc (/lib64/libasan.so.8+0xba097)
          #1 0x987966 in zalloc (/home/acme/bin/perf+0x987966)
          #2 0x52f9b9 in evsel_trace__new /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:307
          #3 0x52f9b9 in evsel__syscall_tp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:333
          #4 0x52f9b9 in evsel__init_raw_syscall_tp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:458
          #5 0x52f9b9 in perf_evsel__raw_syscall_newtp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:480
          #6 0x540dd1 in trace__add_syscall_newtp /home/acme/git/perf-tools-
next/tools/perf/builtin-trace.c:3205
          #7 0x540dd1 in trace__run /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:3891
          #8 0x540dd1 in cmd_trace /home/acme/git/perf-tools-next/tools/perf/builtin-
trace.c:5156
          #9 0x5ef262 in run_builtin /home/acme/git/perf-tools-next/tools/perf/perf.c:323
          #10 0x4196da in handle_internal_command /home/acme/git/perf-tools-
next/tools/perf/perf.c:377
          #11 0x4196da in run_argv /home/acme/git/perf-tools-next/tools/perf/perf.c:421
          #12 0x4196da in main /home/acme/git/perf-tools-next/tools/perf/perf.c:537
          #13 0x7f7342c4a50f in __libc_start_call_main (/lib64/libc.so.6+0x2750f)

      SUMMARY: AddressSanitizer: 80 byte(s) leaked in 2 allocation(s).
      [root@quaco ~]#

    With this we plug all leaks with "perf trace sleep 1".

    Fixes: 3cb4d5e00e037c70 ("perf trace: Free syscall tp fields in evsel->priv")
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Riccardo Mancini <rickyman7@gmail.com>
    Link: https://lore.kernel.org/lkml/20230719202951.534582-5-acme@kernel.org
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit f6fa0e0760a7748341c289c76728ef33871c4f8c
Author: Jeff LaBundy <jeff@labundy.com>
Date:   Sun Jul 9 12:06:37 2023 -0500

    Input: iqs7222 - configure power mode before triggering ATI

    [ Upstream commit 2e00b8bf5624767f6be7427b6eb532524793463e ]

    If the device drops into ultra-low-power mode before being placed
    into normal-power mode as part of ATI being triggered, the device
    does not assert any interrupts until the ATI routine is restarted
    two seconds later.

    Solve this problem by adopting the vendor's recommendation, which
    calls for the device to be placed into normal-power mode prior to
    being configured and ATI being triggered.

    The original implementation followed this sequence, but the order
    was inadvertently changed as part of the resolution of a separate
    erratum.

    Fixes: 1e4189d8af27 ("Input: iqs7222 - protect volatile registers")
    Signed-off-by: Jeff LaBundy <jeff@labundy.com>
    Link: https://lore.kernel.org/r/ZKrpHc2Ji9qR25r2@nixie71
    Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f20377d4e0200ef53fa1d0d042fb542170be4f86
Author: Xie XiuQi <xiexiuqi@huawei.com>
Date:   Thu Aug 31 11:42:05 2023 +0800

    tools/mm: fix undefined reference to pthread_once

    [ Upstream commit 7f33105cdd59a99d068d3d147723a865d10e2260 ]

    Commit 97d5f2e9ee12 ("tools api fs: More thread safety for global
    filesystem variables") introduces pthread_once, so the libpthread
    should be added at link time, or we'll meet the following compile
    error when 'make -C tools/mm':

      gcc -Wall -Wextra -I../lib/ -o page-types page-types.c ../lib/api/libapi.a
      ~/linux/tools/lib/api/fs/fs.c:146: undefined reference to `pthread_once'
      ~/linux/tools/lib/api/fs/fs.c:147: undefined reference to `pthread_once'
      ~/linux/tools/lib/api/fs/fs.c:148: undefined reference to `pthread_once'
      ~/linux/tools/lib/api/fs/fs.c:149: undefined reference to `pthread_once'
      ~/linux/tools/lib/api/fs/fs.c:150: undefined reference to `pthread_once'
      /usr/bin/ld: ../lib/api/libapi.a(libapi-in.o):~/linux/tools/lib/api/fs/fs.c:151:
      more undefined references to `pthread_once' follow
      collect2: error: ld returned 1 exit status
      make: *** [Makefile:22: page-types] Error 1

    Link: https://lkml.kernel.org/r/20230831034205.2376653-1-xiexiuqi@huaweicloud.com
    Fixes: 97d5f2e9ee12 ("tools api fs: More thread safety for global filesystem variables")
    Signed-off-by: Xie XiuQi <xiexiuqi@huawei.com>
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit adf27ed04c836018c64b08f3608736694c82aa46
Author: Konstantin Meskhidze <konstantin.meskhidze@huawei.com>
Date:   Tue Sep 5 17:59:14 2023 +0800

    kconfig: fix possible buffer overflow
```

```
    [ Upstream commit a3b7039bb2b22fcd2ad20d59c00ed4e606ce3754 ]

    Buffer 'new_argv' is accessed without bound check after accessing with
    bound check via 'new_argc' index.

    Fixes: e298f3b49def ("kconfig: add built-in function support")
    Co-developed-by: Ivanov Mikhail <ivanov.mikhail1@huawei-partners.com>
    Signed-off-by: Konstantin Meskhidze <konstantin.meskhidze@huawei.com>
    Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 30e4f88dfaa94f6fbeefea61e4a1b4f6bb816042
Author: Jonathan Marek <jonathan@marek.ca>
Date:   Wed Aug 2 09:52:22 2023 -0400

    mailbox: qcom-ipcc: fix incorrect num_chans counting

    [ Upstream commit a493208079e299aefdc15169dc80e3da3ebb718a ]

    Breaking out early when a match is found leads to an incorrect num_chans
    value when more than one ipcc mailbox channel is used by the same device.

    Fixes: e9d50e4b4d04 ("mailbox: qcom-ipcc: Dynamic alloc for channel arrangement")
    Signed-off-by: Jonathan Marek <jonathan@marek.ca>
    Signed-off-by: Jassi Brar <jaswinder.singh@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1a1623087d8d472e6acfed0e3875e5bb52d9ea18
Author: Andreas Gruenbacher <agruenba@redhat.com>
Date:   Thu Aug 10 17:15:46 2023 +0200

    gfs2: low-memory forced flush fixes

    [ Upstream commit b74cd55aa9a9d0aca760028a51343ec79812e410 ]

    First, function gfs2_ail_flush_reqd checks the SDF_FORCE_AIL_FLUSH flag
    to determine if an AIL flush should be forced in low-memory situations.
    However, it also immediately clears the flag, and when called repeatedly
    as in function gfs2_logd, the flag will be lost.  Fix that by pulling
    the SDF_FORCE_AIL_FLUSH flag check out of gfs2_ail_flush_reqd.

    Second, function gfs2_writepages sets the SDF_FORCE_AIL_FLUSH flag
    whether or not enough pages were written.  If enough pages could be
    written, flushing the AIL is unnecessary, though.

    Third, gfs2_writepages doesn't wake up logd after setting the
    SDF_FORCE_AIL_FLUSH flag, so it can take a long time for logd to react.
    It would be preferable to wake up logd, but that hurts the performance
    of some workloads and we don't quite understand why so far, so don't
    wake up logd so far.

    Fixes: b066a4eebd4f ("gfs2: forcibly flush ail to relieve memory pressure")
    Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 42c45325a5d53093c1e74ad2ab40658680b01fbe
Author: Andreas Gruenbacher <agruenba@redhat.com>
Date:   Thu Aug 17 15:46:16 2023 +0200

    gfs2: Switch to wait_event in gfs2_logd

    [ Upstream commit 6df373b09b1dcf2f7d579f515f653f89a896d417 ]

    In gfs2_logd(), switch from an open-coded wait loop to
```

```
      wait_event_interruptible_timeout().

      Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
      Stable-dep-of: b74cd55aa9a9 ("gfs2: low-memory forced flush fixes")
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit bc8e346651ba12daad06ba2bf03abf21a45cfe63
  Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
  Date:   Sat Feb 25 11:58:48 2023 +0100

      tpm_crb: Fix an error handling path in crb_acpi_add()

      [ Upstream commit 9c377852ddfdc557b1370f196b0cfdf28d233460 ]

      Some error paths don't call acpi_put_table() before returning.
      Branch to the correct place instead of doing some direct return.

      Fixes: 4d2732882703 ("tpm_crb: Add support for CRB devices based on Pluton")
      Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
      Acked-by: Matthew Garrett <mgarrett@aurora.tech>
      Reviewed-by: Jarkko Sakkinen <jarkko@kernel.org>
      Signed-off-by: Jarkko Sakkinen <jarkko@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit ad2c639f4f717d9ba23d6e982375fb900ac316c0
  Author: Jiri Slaby <jirislaby@kernel.org>
  Date:   Tue Aug 29 12:51:06 2023 +0200

      kbuild: dummy-tools: make MPROFILE_KERNEL checks work on BE

      [ Upstream commit bfb41e46d0b040ae83c1c4a50292298208b10f73 ]

      Commit 2eab791f940b ("kbuild: dummy-tools: support MPROFILE_KERNEL
      checks for ppc") added support for ppc64le's checks for
      -mprofile-kernel.

      Now, commit aec0ba7472a7 ("powerpc/64: Use -mprofile-kernel for big
      endian ELFv2 kernels") added support for -mprofile-kernel even on
      big-endian ppc.

      So lift the check in gcc-check-mprofile-kernel.sh to support big-endian too.

      Fixes: aec0ba7472a7 ("powerpc/64: Use -mprofile-kernel for big endian ELFv2 kernels")
      Signed-off-by: Jiri Slaby <jslaby@suse.cz>
      Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 8da9dd59dd0b42b586ccfc7b640f215d09698cbc
  Author: Masahiro Yamada <masahiroy@kernel.org>
  Date:   Wed Aug 23 20:50:41 2023 +0900

      kbuild: do not run depmod for 'make modules_sign'

      [ Upstream commit 2429742e506a2b5939a62c629c4a46d91df0ada8 ]

      Commit 961ab4a3cd66 ("kbuild: merge scripts/Makefile.modsign to
      scripts/Makefile.modinst") started to run depmod at the end of
      'make modules_sign'.

      Move the depmod rule to scripts/Makefile.modinst and run it only when
      $(modules_sign_only) is empty.

      Fixes: 961ab4a3cd66 ("kbuild: merge scripts/Makefile.modsign to
  scripts/Makefile.modinst")
      Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
```

```
    Reviewed-by: Nicolas Schier <nicolas@fjasle.eu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit edd005afaefec3f166acf69d897cf9a0d8a1201f
Author: Masahiro Yamada <masahiroy@kernel.org>
Date:   Sat Jul 22 13:47:48 2023 +0900

    kbuild: rpm-pkg: define _arch conditionally

    [ Upstream commit 233046a2afd12a4f699305b92ee634eebf1e4f31 ]

    Commit 3089b2be0cce ("kbuild: rpm-pkg: fix build error when _arch is
    undefined") does not work as intended; _arch is always defined as
    $UTS_MACHINE.

    The intention was to define _arch to $UTS_MACHINE only when it is not
    defined.

    Fixes: 3089b2be0cce ("kbuild: rpm-pkg: fix build error when _arch is undefined")
    Signed-off-by: Masahiro Yamada <masahiroy@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5c6cc19d7378c6bebeb87519a77a2c8be431cac8
Author: Qiang Yu <quic_qianyu@quicinc.com>
Date:   Thu May 18 14:22:39 2023 +0800

    bus: mhi: host: Skip MHI reset if device is in RDDM

    commit cabce92dd805945a090dc6fc73b001bb35ed083a upstream.

    In RDDM EE, device can not process MHI reset issued by host. In case of MHI
    power off, host is issuing MHI reset and polls for it to get cleared until
    it times out. Since this timeout can not be avoided in case of RDDM, skip
    the MHI reset in this scenarios.

    Cc: <stable@vger.kernel.org>
    Fixes: a6e2e3522f29 ("bus: mhi: core: Add support for PM state transitions")
    Signed-off-by: Qiang Yu <quic_qianyu@quicinc.com>
    Reviewed-by: Jeffrey Hugo <quic_jhugo@quicinc.com>
    Reviewed-by: Manivannan Sadhasivam <mani@kernel.org>
    Link: https://lore.kernel.org/r/1684390959-17836-1-git-send-email-
quic_qianyu@quicinc.com
    Signed-off-by: Manivannan Sadhasivam <manivannan.sadhasivam@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b350c662d63add0157931bca0789344053c69c05
Author: Fedor Pchelkin <pchelkin@ispras.ru>
Date:   Thu Jul 20 18:37:51 2023 +0300

    NFSv4/pnfs: minor fix for cleanup path in nfs4_get_device_info

    commit 96562c45af5c31b89a197af28f79bfa838fb8391 upstream.

    It is an almost improbable error case but when page allocating loop in
    nfs4_get_device_info() fails then we should only free the already
    allocated pages, as __free_page() can't deal with NULL arguments.

    Found by Linux Verification Center (linuxtesting.org).

    Cc: stable@vger.kernel.org
    Signed-off-by: Fedor Pchelkin <pchelkin@ispras.ru>
    Reviewed-by: Benjamin Coddington <bcodding@redhat.com>
    Signed-off-by: Anna Schumaker <Anna.Schumaker@Netapp.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 0ec26716e45d615edfff46012e7dedcc0ac5f7ab
Author: Trond Myklebust <trond.myklebust@hammerspace.com>
Date:   Sat Aug 19 17:22:14 2023 -0400

    NFS: Fix a potential data corruption

    commit 88975a55969e11f26fe3846bf4fbf8e7dc8cbbd4 upstream.

    We must ensure that the subrequests are joined back into the head before
    we can retransmit a request. If the head was not on the commit lists,
    because the server wrote it synchronously, we still need to add it back
    to the retransmission list.
    Add a call that mirrors the effect of nfs_cancel_remove_inode() for
    O_DIRECT.

    Fixes: ed5d588fe47f ("NFS: Try to join page groups before an O_DIRECT retransmission")
    Cc: stable@vger.kernel.org
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Anna Schumaker <Anna.Schumaker@Netapp.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit de830dc9625a1dfea6bf0f5de5e725f4b789e800
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:29:01 2023 +0200

    clk: qcom: mss-sc7180: fix missing resume during probe

    commit e2349da0fa7ca822cda72f427345b95795358fe7 upstream.

    Drivers that enable runtime PM must make sure that the controller is
    runtime resumed before accessing its registers to prevent the power
    domain from being disabled.

    Fixes: 8def929c4097 ("clk: qcom: Add modem clock controller driver for SC7180")
    Cc: stable@vger.kernel.org      # 5.7
    Cc: Taniya Das <quic_tdas@quicinc.com>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-8-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0f316479360a89899fe379d519eea81ecb60f062
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:29:00 2023 +0200

    clk: qcom: q6sstop-qcs404: fix missing resume during probe

    commit 97112c83f4671a4a722f99a53be4e91fac4091bc upstream.

    Drivers that enable runtime PM must make sure that the controller is
    runtime resumed before accessing its registers to prevent the power
    domain from being disabled.

    Fixes: 6cdef2738db0 ("clk: qcom: Add Q6SSTOP clock controller for QCS404")
    Cc: stable@vger.kernel.org      # 5.5
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-7-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5a078ec3fa28304a80ad2b5e8b61f29f8f9c1612
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:28:59 2023 +0200

    clk: qcom: lpasscc-sc7280: fix missing resume during probe
```

```
    commit 66af5339d4f8e20c6d89a490570bd94d40f1a7f6 upstream.

    Drivers that enable runtime PM must make sure that the controller is
    runtime resumed before accessing its registers to prevent the power
    domain from being disabled.

    Fixes: 4ab43d171181 ("clk: qcom: Add lpass clock controller driver for SC7280")
    Cc: stable@vger.kernel.org      # 5.16
    Cc: Taniya Das <quic_tdas@quicinc.com>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-6-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 98faf753f00909f67e9eaa092c30f34009f97b21
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:28:57 2023 +0200

    clk: qcom: dispcc-sm8550: fix runtime PM imbalance on probe errors

    commit acaf1b3296a504d4a61b685f78baae771421608d upstream.

    Make sure to decrement the runtime PM usage count before returning in
    case regmap initialisation fails.

    Fixes: 90114ca11476 ("clk: qcom: add SM8550 DISPCC driver")
    Cc: stable@vger.kernel.org      # 6.3
    Cc: Neil Armstrong <neil.armstrong@linaro.org>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-4-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1c1962a912ef4a5996ca5d02fce60b352bdee51d
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:28:56 2023 +0200

    clk: qcom: dispcc-sm8450: fix runtime PM imbalance on probe errors

    commit b0f3d01bda6c3f6f811e70f76d2040ae81f64565 upstream.

    Make sure to decrement the runtime PM usage count before returning in
    case regmap initialisation fails.

    Fixes: 16fb89f92ec4 ("clk: qcom: Add support for Display Clock Controller on SM8450")
    Cc: stable@vger.kernel.org      # 6.1
    Cc: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-3-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 22ee7c9c7f381be178b4457bc54530002e08e938
Author: Chris Lew <quic_clew@quicinc.com>
Date:   Tue Aug 1 12:17:12 2023 +0530

    soc: qcom: qmi_encdec: Restrict string length in decode

    commit 8d207400fd6b79c92aeb2f33bb79f62dff904ea2 upstream.

    The QMI TLV value for strings in a lot of qmi element info structures
    account for null terminated strings with MAX_LEN + 1. If a string is
    actually MAX_LEN + 1 length, this will cause an out of bounds access
    when the NULL character is appended in decoding.
```

```
    Fixes: 9b8a11e82615 ("soc: qcom: Introduce QMI encoder/decoder")
    Cc: stable@vger.kernel.org
    Signed-off-by: Chris Lew <quic_clew@quicinc.com>
    Signed-off-by: Praveenkumar I <quic_ipkumar@quicinc.com>
    Link: https://lore.kernel.org/r/20230801064712.3590128-1-quic_ipkumar@quicinc.com
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c8b48c5ee3c61b041a6a26489eb3127a0b60f5c0
Author: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
Date:   Sat May 13 00:17:23 2023 +0300

    clk: qcom: gcc-mdm9615: use proper parent for pll0_vote clock

    commit 1583694bb4eaf186f17131dbc1b83d6057d2749b upstream.

    The pll0_vote clock definitely should have pll0 as a parent (instead of
    pll8).

    Fixes: 7792a8d6713c ("clk: mdm9615: Add support for MDM9615 Clock Controllers")
    Cc: stable@kernel.org
    Reviewed-by: Neil Armstrong <neil.armstrong@linaro.org>
    Signed-off-by: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Reviewed-by: Konrad Dybcio <konrad.dybcio@linaro.org>
    Link: https://lore.kernel.org/r/20230512211727.3445575-7-dmitry.baryshkov@linaro.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 120367a5ab68bd8d444467872d9e1aa4808f7353
Author: Marco Felsch <m.felsch@pengutronix.de>
Date:   Mon Aug 7 10:47:43 2023 +0200

    clk: imx: pll14xx: align pdiv with reference manual

    commit 37cfd5e457cbdcd030f378127ff2d62776f641e7 upstream.

    The PLL14xx hardware can be found on i.MX8M{M,N,P} SoCs and always come
    with a 6-bit pre-divider. Neither the reference manuals nor the
    datasheets of these SoCs do mention any restrictions. Furthermore the
    current code doesn't respect the restrictions from the comment too.

    Therefore drop the restriction and align the max pre-divider (pdiv)
    value to 63 to get more accurate frequencies.

    Fixes: b09c68dc57c9 ("clk: imx: pll14xx: Support dynamic rates")
    Cc: stable@vger.kernel.org
    Signed-off-by: Marco Felsch <m.felsch@pengutronix.de>
    Reviewed-by: Abel Vesa <abel.vesa@linaro.org>
    Reviewed-by: Adam Ford <aford173@gmail.com>
    Signed-off-by: Philipp Zabel <p.zabel@pengutronix.de>
    Acked-by: Sascha Hauer <s.hauer@pengutronix.de>
    Tested-by: Ahmad Fatoum <a.fatoum@pengutronix.de>
    Link: https://lore.kernel.org/r/20230807084744.1184791-1-m.felsch@pengutronix.de
    Signed-off-by: Abel Vesa <abel.vesa@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 20b9ffdb96b4eea0d771bf238fddc7152cde5ef2
Author: Ahmad Fatoum <a.fatoum@pengutronix.de>
Date:   Mon Aug 7 10:47:44 2023 +0200

    clk: imx: pll14xx: dynamically configure PLL for 393216000/361267200Hz

    commit 72d00e560d10665e6139c9431956a87ded6e9880 upstream.
```

```
    Since commit b09c68dc57c9 ("clk: imx: pll14xx: Support dynamic rates"),
    the driver has the ability to dynamically compute PLL parameters to
    approximate the requested rates. This is not always used, because the
    logic is as follows:

      - Check if the target rate is hardcoded in the frequency table
      - Check if varying only kdiv is possible, so switch over is glitch free
      - Compute rate dynamically by iterating over pdiv range

    If we skip the frequency table for the 1443x PLL, we find that the
    computed values differ to the hardcoded ones. This can be valid if the
    hardcoded values guarantee for example an earlier lock-in or if the
    divisors are chosen, so that other important rates are more likely to
    be reached glitch-free.

    For rates (393216000 and 361267200, this doesn't seem to be the case:
    They are only approximated by existing parameters (393215995 and
    361267196 Hz, respectively) and they aren't reachable glitch-free from
    other hardcoded frequencies. Dropping them from the table allows us
    to lock-in to these frequencies exactly.

    This is immediately noticeable because they are the assigned-clock-rates
    for IMX8MN_AUDIO_PLL1 and IMX8MN_AUDIO_PLL2, respectively and a look
    into clk_summary so far showed that they were a few Hz short of the target:

    imx8mn-board:~# grep audio_pll[12]_out /sys/kernel/debug/clk/clk_summary
    audio_pll2_out           0        0        0   361267196 0     0  50000   N
    audio_pll1_out           1        1        0   393215995 0     0  50000   Y

    and afterwards:

    imx8mn-board:~# grep audio_pll[12]_out /sys/kernel/debug/clk/clk_summary
    audio_pll2_out           0        0        0   361267200 0     0  50000   N
    audio_pll1_out           1        1        0   393216000 0     0  50000   Y

    This change is equivalent to adding following hardcoded values:

      /*               rate     mdiv  pdiv  sdiv   kdiv */
      PLL_1443X_RATE(393216000, 655,    5,    3,  23593),
      PLL_1443X_RATE(361267200, 497,   33,    0, -16882),

    Fixes: 053a4ffe2988 ("clk: imx: imx8mm: fix audio pll setting")
    Cc: stable@vger.kernel.org # v5.18+
    Signed-off-by: Ahmad Fatoum <a.fatoum@pengutronix.de>
    Signed-off-by: Marco Felsch <m.felsch@pengutronix.de>
    Link: https://lore.kernel.org/r/20230807084744.1184791-2-m.felsch@pengutronix.de
    Signed-off-by: Abel Vesa <abel.vesa@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 209a26803ca26ad7c467a2b385edf9b942d5bef3
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Fri Jul 28 18:59:23 2023 +0200

    dt-bindings: clock: xlnx,versal-clk: drop select:false

    commit 172044e30b00977784269e8ab72132a48293c654 upstream.

    select:false makes the schema basically ignored and not effective, which
    is clearly not what we want for a device binding.

    Fixes: 352546805a44 ("dt-bindings: clock: Add bindings for versal clock driver")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Link: https://lore.kernel.org/r/20230728165923.108589-1-krzysztof.kozlowski@linaro.org
    Reviewed-by: Conor Dooley <conor.dooley@microchip.com>
```

```
    Reviewed-by: Shubhrajyoti Datta <shubhrajyoti.datta@amd.com>
    Signed-off-by: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aaf5e6320a7e69ae5c19366c949e0c216a7c1d90
Author: Raag Jadav <raag.jadav@intel.com>
Date:   Tue Aug 22 12:53:40 2023 +0530

    pinctrl: cherryview: fix address_space_handler() argument

    commit d5301c90716a8e20bc961a348182daca00c8e8f0 upstream.

    First argument of acpi_*_address_space_handler() APIs is acpi_handle of
    the device, which is incorrectly passed in driver ->remove() path here.
    Fix it by passing the appropriate argument and while at it, make both
    API calls consistent using ACPI_HANDLE().

    Fixes: a0b028597d59 ("pinctrl: cherryview: Add support for GMMR GPIO opregion")
    Cc: stable@vger.kernel.org
    Signed-off-by: Raag Jadav <raag.jadav@intel.com>
    Acked-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b0fd3e2b095af02030b2411f767d057c036598c3
Author: Bharath SM <bharathsm@microsoft.com>
Date:   Wed Aug 16 19:38:45 2023 +0000

    cifs: update desired access while requesting for directory lease

    commit b6d44d42313baa45a81ce9b299aeee2ccf3d0ee1 upstream.

    We read and cache directory contents when we get directory
    lease, so we should ask for read permission to read contents
    of directory.

    Signed-off-by: Bharath SM <bharathsm@microsoft.com>
    Reviewed-by: Shyam Prasad N <sprasad@microsoft.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0fe7b6d676176886334875ae954d22e8bf69d8d9
Author: Helge Deller <deller@gmx.de>
Date:   Fri Aug 25 17:46:39 2023 +0200

    parisc: led: Reduce CPU overhead for disk & lan LED computation

    commit 358ad816e52d4253b38c2f312e6b1cbd89e0dbf7 upstream.

    Older PA-RISC machines have LEDs which show the disk- and LAN-activity.
    The computation is done in software and takes quite some time, e.g. on a
    J6500 this may take up to 60% time of one CPU if the machine is loaded
    via network traffic.

    Since most people don't care about the LEDs, start with LEDs disabled and
    just show a CPU heartbeat LED. The disk and LAN LEDs can be turned on
    manually via /proc/pdc/led.

    Signed-off-by: Helge Deller <deller@gmx.de>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 81de4949593927fb98fc93a963d1ea595a202a7f
Author: Helge Deller <deller@gmx.de>
```

```
Date:   Sun Aug 27 13:46:11 2023 +0200

    parisc: led: Fix LAN receive and transmit LEDs

    commit 4db89524b084f712a887256391fc19d9f66c8e55 upstream.

    Fix the LAN receive and LAN transmit LEDs, which where swapped
    up to now.

    Signed-off-by: Helge Deller <deller@gmx.de>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6599f72bed2364750e4f6f3a6f5f17ad0b3e1a0d
Author: Kalesh Singh <kaleshsingh@google.com>
Date:   Tue Aug 1 19:56:03 2023 -0700

    Multi-gen LRU: avoid race in inc_min_seq()

    commit bb5e7f234eacf34b65be67ebb3613e3b8cf11b87 upstream.

    inc_max_seq() will try to inc_min_seq() if nr_gens == MAX_NR_GENS. This
    is because the generations are reused (the last oldest now empty
    generation will become the next youngest generation).

    inc_min_seq() is retried until successful, dropping the lru_lock
    and yielding the CPU on each failure, and retaking the lock before
    trying again:

            while (!inc_min_seq(lruvec, type, can_swap)) {
                    spin_unlock_irq(&lruvec->lru_lock);
                    cond_resched();
                    spin_lock_irq(&lruvec->lru_lock);
            }

    However, the initial condition that required incrementing the min_seq
    (nr_gens == MAX_NR_GENS) is not retested. This can change by another
    call to inc_max_seq() from run_aging() with force_scan=true from the
    debugfs interface.

    Since the eviction stalls when the nr_gens == MIN_NR_GENS, avoid
    unnecessarily incrementing the min_seq by rechecking the number of
    generations before each attempt.

    This issue was uncovered in previous discussion on the list by Yu Zhao
    and Aneesh Kumar [1].

    [1] https://lore.kernel.org/linux-
mm/CAOUHufbO7CaVm=xjEb1avDhHVvnC8pJmGyKcFf2iY_dpf+zR3w@mail.gmail.com/

    Link: https://lkml.kernel.org/r/20230802025606.346758-2-kaleshsingh@google.com
    Fixes: d6c3af7d8a2b ("mm: multi-gen LRU: debugfs interface")
    Signed-off-by: Kalesh Singh <kaleshsingh@google.com>
    Tested-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
[mediatek]
    Tested-by: Charan Teja Kalla <quic_charante@quicinc.com>
    Cc: Yu Zhao <yuzhao@google.com>
    Cc: Aneesh Kumar K V <aneesh.kumar@linux.ibm.com>
    Cc: Barry Song <baohua@kernel.org>
    Cc: Brian Geffon <bgeffon@google.com>
    Cc: Jan Alexander Steffens (heftig) <heftig@archlinux.org>
    Cc: Lecopzer Chen <lecopzer.chen@mediatek.com>
    Cc: Matthias Brugger <matthias.bgg@gmail.com>
    Cc: Oleksandr Natalenko <oleksandr@natalenko.name>
    Cc: Qi Zheng <zhengqi.arch@bytedance.com>
```

```
    Cc: Steven Barrett <steven@liquorix.net>
    Cc: Suleiman Souhlal <suleiman@google.com>
    Cc: Suren Baghdasaryan <surenb@google.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 16e5d74e123d71fb48b5e7d5aec4b2ecb6af3c91
Author: Andrew Donnellan <ajd@linux.ibm.com>
Date:   Fri Jul 14 11:52:38 2023 +1000

    lib/test_meminit: allocate pages up to order MAX_ORDER

    commit efb78fa86e95832b78ca0ba60f3706788a818938 upstream.

    test_pages() tests the page allocator by calling alloc_pages() with
    different orders up to order 10.

    However, different architectures and platforms support different maximum
    contiguous allocation sizes.  The default maximum allocation order
    (MAX_ORDER) is 10, but architectures can use CONFIG_ARCH_FORCE_MAX_ORDER
    to override this.  On platforms where this is less than 10, test_meminit()
    will blow up with a WARN().  This is expected, so let's not do that.

    Replace the hardcoded "10" with the MAX_ORDER macro so that we test
    allocations up to the expected platform limit.

    Link: https://lkml.kernel.org/r/20230714015238.47931-1-ajd@linux.ibm.com
    Fixes: 5015a300a522 ("lib: introduce test_meminit module")
    Signed-off-by: Andrew Donnellan <ajd@linux.ibm.com>
    Reviewed-by: Alexander Potapenko <glider@google.com>
    Cc: Xiaoke Wang <xkernel.wang@foxmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6a99d82d849987450e7acf2a995d64c72c68f040
Author: Muchun Song <muchun.song@linux.dev>
Date:   Fri Jul 7 11:38:59 2023 +0800

    mm: hugetlb_vmemmap: fix a race between vmemmap pmd split

    commit 3ce2c24cb68f228590a053d6058a5901cd31af61 upstream.

    The local variable @page in __split_vmemmap_huge_pmd() to obtain a pmd
    page without holding page_table_lock may possiblely get the page table
    page instead of a huge pmd page.

    The effect may be in set_pte_at() since we may pass an invalid page
    struct, if set_pte_at() wants to access the page struct (e.g.
    CONFIG_PAGE_TABLE_CHECK is enabled), it may crash the kernel.

    So fix it.  And inline __split_vmemmap_huge_pmd() since it only has one
    user.

    Link: https://lkml.kernel.org/r/20230707033859.16148-1-songmuchun@bytedance.com
    Fixes: d8d55f5616cf ("mm: sparsemem: use page table lock to protect kernel pmd
operations")
    Signed-off-by: Muchun Song <songmuchun@bytedance.com>
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 75bb29b6c44f486e9bd048759669c205dccb2048
```

```
Author: Michal Hocko <mhocko@suse.com>
Date:   Tue Jul 4 13:52:40 2023 +0200

    memcg: drop kmem.limit_in_bytes

    commit 86327e8eb94c52eca4f93cfece2e29d1bf52acbf upstream.

    kmem.limit_in_bytes (v1 way to limit kernel memory usage) has been
    deprecated since 58056f77502f ("memcg, kmem: further deprecate
    kmem.limit_in_bytes") merged in 5.16.  We haven't heard about any serious
    users since then but it seems that the mere presence of the file is
    causing more harm thatn good.  We (SUSE) have had several bug reports from
    customers where Docker based containers started to fail because a write to
    kmem.limit_in_bytes has failed.

    This was unexpected because runc code only expects ENOENT (kmem disabled)
    or EBUSY (tasks already running within cgroup).  So a new error code was
    unexpected and the whole container startup failed.  This has been later
    addressed by
    https://github.com/opencontainers/runc/commit/52390d68040637dfc77f9fda6bbe70952423d380
    so current Docker runtimes do not suffer from the problem anymore.  There
    are still older version of Docker in use and likely hard to get rid of
    completely.

    Address this by wiping out the file completely and effectively get back to
    pre 4.5 era and CONFIG_MEMCG_KMEM=n configuration.

    I would recommend backporting to stable trees which have picked up
    58056f77502f ("memcg, kmem: further deprecate kmem.limit_in_bytes").

    [mhocko@suse.com: restore _KMEM switch case]
      Link: https://lkml.kernel.org/r/ZKe5wxdbvPi5Cwd7@dhcp22.suse.cz
    Link: https://lkml.kernel.org/r/20230704115240.14672-1-mhocko@kernel.org
    Signed-off-by: Michal Hocko <mhocko@suse.com>
    Acked-by: Shakeel Butt <shakeelb@google.com>
    Acked-by: Johannes Weiner <hannes@cmpxchg.org>
    Acked-by: Roman Gushchin <roman.gushchin@linux.dev>
    Cc: Muchun Song <muchun.song@linux.dev>
    Cc: Tejun Heo <tj@kernel.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cbf06d3e0271f3b9c7590d865acb576ce9911291
Author: Steve French <stfrench@microsoft.com>
Date:   Thu Aug 24 23:29:18 2023 -0500

    send channel sequence number in SMB3 requests after reconnects

    commit 09ee7a3bf866c0fa5ee1914d2c65958559eb5b4c upstream.

    The ChannelSequence field in the SMB3 header is supposed to be
    increased after reconnect to allow the server to distinguish
    requests from before and after the reconnect.  We had always
    been setting it to zero.  There are cases where incrementing
    ChannelSequence on requests after network reconnects can reduce
    the chance of data corruptions.

    See MS-SMB2 3.2.4.1 and 3.2.7.1

    Signed-off-by: Steve French <stfrench@microsoft.com>
    Cc: stable@vger.kernel.org # 5.16+
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0286d8947a3c134b2580988141db7083348aebba
```

```
Author: Aleksey Nasibulin <alealexpro100@ya.ru>
Date:   Wed Jul 12 03:40:17 2023 +0200

    ARM: dts: BCM5301X: Extend RAM to full 256MB for Linksys EA6500 V2

    commit 91994e59079dcb455783d3f9ea338eea6f671af3 upstream.

    Linksys ea6500-v2 have 256MB of ram. Currently we only use 128MB.
    Expand the definition to use all the available RAM.

    Fixes: 03e96644d7a8 ("ARM: dts: BCM5301X: Add basic DT for Linksys EA6500 V2")
    Signed-off-by: Aleksey Nasibulin <alealexpro100@ya.ru>
    Signed-off-by: Christian Marangi <ansuelsmth@gmail.com>
    Cc: stable@vger.kernel.org
    Acked-by: Rafał Miłecki <rafal@milecki.pl>
    Link: https://lore.kernel.org/r/20230712014017.28123-1-ansuelsmth@gmail.com
    Signed-off-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 908f16a356230f8f9ee8d38d53472ee15edcf186
Author: Chris Paterson <chris.paterson2@renesas.com>
Date:   Fri Jun 9 23:11:36 2023 +0100

    arm64: dts: renesas: rzg2l: Fix txdv-skew-psec typos

    commit db67345716a52abb750ec8f76d6a5675218715f9 upstream.

    It looks like txdv-skew-psec is a typo from a copy+paste. txdv-skew-psec
    is not present in the PHY bindings nor is it in the driver.

    Correct to txen-skew-psec which is clearly what it was meant to be.

    Given that the default for txen-skew-psec is 0, and the device tree is
    only trying to set it to 0 anyway, there should not be any functional
    change from this fix.

    Fixes: 361b0dcbd7f9 ("arm64: dts: renesas: rzg2l-smarc-som: Enable Ethernet")
    Fixes: 6494e4f90503 ("arm64: dts: renesas: rzg2ul-smarc-som: Enable Ethernet on SMARC
platform")
    Fixes: ce0c63b6a5ef ("arm64: dts: renesas: Add initial device tree for RZ/G2LC SMARC
EVK")
    Cc: stable@vger.kernel.org # 6.1.y
    Reported-by: Tomohiro Komagata <tomohiro.komagata.aj@renesas.com>
    Signed-off-by: Chris Paterson <chris.paterson2@renesas.com>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Link: https://lore.kernel.org/r/20230609221136.7431-1-chris.paterson2@renesas.com
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6190c676fedf7a6ba2bdab662cb8600b32c5b1eb
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Thu Jul 20 13:53:35 2023 +0200

    ARM: dts: qcom: msm8974pro-castor: correct touchscreen syna,nosleep-mode

    commit 7c74379afdfee7b13f1cd8ff1ad6e0f986aec96c upstream.

    There is no syna,nosleep property in Synaptics RMI4 touchscreen:

      qcom-msm8974pro-sony-xperia-shinano-castor.dtb: synaptics@2c: rmi4-f01@1:
'syna,nosleep' does not match any of the regexes: 'pinctrl-[0-9]+'

    Fixes: ab80661883de ("ARM: dts: qcom: msm8974: Add Sony Xperia Z2 Tablet")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
```

```
    Link: https://lore.kernel.org/r/20230720115335.137354-6-krzysztof.kozlowski@linaro.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1d4e79be26748f2881fca5a4cd5a0bf3ec2c3a0d
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Thu Jul 20 13:53:34 2023 +0200

    ARM: dts: qcom: msm8974pro-castor: correct touchscreen function names

    commit 31fba16c19c45b2b3a7c23b0bfef80aed1b29050 upstream.

    The node names for functions of Synaptics RMI4 touchscreen must be as
    "rmi4-fXX", as required by bindings and Linux driver.

      qcom-msm8974pro-sony-xperia-shinano-castor.dtb: synaptics@2c: Unevaluated properties
are not allowed ('rmi-f01@1', 'rmi-f11@11' were unexpected)

    Fixes: ab80661883de ("ARM: dts: qcom: msm8974: Add Sony Xperia Z2 Tablet")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Link: https://lore.kernel.org/r/20230720115335.137354-5-krzysztof.kozlowski@linaro.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0167317a624a99dd1e861b69820efd9f9f4e6c5e
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Thu Jul 20 13:53:30 2023 +0200

    arm64: dts: qcom: msm8953-vince: drop duplicated touschreen parent interrupt

    commit b019cf7e5fbaa7d25f716cb936a9237b47156f2d upstream.

    Interrupts extended already define a parent interrupt controller:

      msm8953-xiaomi-vince.dtb: touchscreen@20: Unevaluated properties are not allowed
('interrupts-parent' was unexpected)

    Fixes: aa17e707e04a ("arm64: dts: qcom: msm8953: Add device tree for Xiaomi Redmi 5
Plus")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Reviewed-by: Konrad Dybcio <konrad.dybcio@linaro.org>
    Link: https://lore.kernel.org/r/20230720115335.137354-1-krzysztof.kozlowski@linaro.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c6ffc6d40f906fd4e731772178d77f9c2653f228
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Thu Jul 20 13:53:33 2023 +0200

    ARM: dts: qcom: msm8974pro-castor: correct inverted X of touchscreen

    commit 43db69268149049540b1d2bbe8a69e59d5cb43b6 upstream.

    There is no syna,f11-flip-x property, so assume intention was to use
    touchscreen-inverted-x.

    Fixes: ab80661883de ("ARM: dts: qcom: msm8974: Add Sony Xperia Z2 Tablet")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Link: https://lore.kernel.org/r/20230720115335.137354-4-krzysztof.kozlowski@linaro.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 0cf21057fb77b6b59273907e550caadf8c013471
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:29:02 2023 +0200

    clk: qcom: turingcc-qcs404: fix missing resume during probe

    commit a9f71a033587c9074059132d34c74eabbe95ef26 upstream.

    Drivers that enable runtime PM must make sure that the controller is
    runtime resumed before accessing its registers to prevent the power
    domain from being disabled.

    Fixes: 892df0191b29 ("clk: qcom: Add QCS404 TuringCC")
    Cc: stable@vger.kernel.org      # 5.2
    Cc: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-9-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ac47243e58daac0b9bf2861457911b1845c8dd2a
Author: Sameer Pujar <spujar@nvidia.com>
Date:   Thu Jun 29 10:42:17 2023 +0530

    arm64: tegra: Update AHUB clock parent and rate

    commit dc6d5d85ed3a3fe566314f388bce4c71a26b1677 upstream.

    I2S data sanity test failures are seen at lower AHUB clock rates
    on Tegra234. The Tegra194 uses the same clock relationship for AHUB
    and it is likely that similar issues would be seen. Thus update the
    AHUB clock parent and rates here as well for Tegra194, Tegra186
    and Tegra210.

    Fixes: 177208f7b06d ("arm64: tegra: Add DT binding for AHUB components")
    Cc: stable@vger.kernel.org
    Signed-off-by: Sameer Pujar <spujar@nvidia.com>
    Signed-off-by: Thierry Reding <treding@nvidia.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e983f0bfa902d1b65f99d9a3f17d6dafb2e864fd
Author: Sheetal <sheetal@nvidia.com>
Date:   Thu Jun 29 10:42:16 2023 +0530

    arm64: tegra: Update AHUB clock parent and rate on Tegra234

    commit e483fe34adab3197558b7284044c1b26f5ede20e upstream.

    I2S data sanity tests fail beyond a bit clock frequency of 6.144MHz.
    This happens because the AHUB clock rate is too low and it shows
    9.83MHz on boot.

    The maximum rate of PLLA_OUT0 is 49.152MHz and is used to serve I/O
    clocks. It is recommended that AHUB clock operates higher than this.
    Thus fix this by using PLLP_OUT0 as parent clock for AHUB instead of
    PLLA_OUT0 and fix the rate to 81.6MHz.

    Fixes: dc94a94daa39 ("arm64: tegra: Add audio devices on Tegra234")
    Cc: stable@vger.kernel.org
    Signed-off-by: Sheetal <sheetal@nvidia.com>
    Signed-off-by: Sameer Pujar <spujar@nvidia.com>
    Reviewed-by: Mohan Kumar D <mkumard@nvidia.com>
    Signed-off-by: Thierry Reding <treding@nvidia.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit bd2a4d3c2a17e1b4f42ba01a5c99084426684392
Author: Paul Cercueil <paul@crapouillou.net>
Date:   Fri Jul 14 17:37:20 2023 +0200

    ARM: dts: samsung: exynos4210-i9100: Fix LCD screen's physical size

    commit b3f3fc32e5ff1e848555af8616318cc667457f90 upstream.

    The previous values were completely bogus, and resulted in the computed
    DPI ratio being much lower than reality, causing applications and UIs to
    misbehave.

    The new values were measured by myself with a ruler.

    Signed-off-by: Paul Cercueil <paul@crapouillou.net>
    Acked-by: Sam Ravnborg <sam@ravnborg.org>
    Fixes: 8620cc2f99b7 ("ARM: dts: exynos: Add devicetree file for the Galaxy S2")
    Cc: <stable@vger.kernel.org> # v5.8+
    Link: https://lore.kernel.org/r/20230714153720.336990-1-paul@crapouillou.net
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7c3d7b8a99801e4cfaae601cf1535231f77114d1
Author: Sheetal <sheetal@nvidia.com>
Date:   Thu Jun 22 17:04:09 2023 +0530

    ASoC: tegra: Fix SFC conversion for few rates

    commit d900d9a435ca95a386f49424f3689cd17ec201da upstream.

    Sample rate conversions for rates greater than 48kHz are found to be
    failing. It means x->y conversions fail when either x or y is greater
    than 48kHz.

    This happens because, tegra210_sfc_rate_to_idx() returns incorrect
    index for rates greater than 48kHz. This actually depends on the
    tegra210_sfc_rates[] array and it is not in sync with frequency
    values of SFC TX/RX register. To be precise, 64kHz entry is missing
    in above array defined in the driver. Due to this wrong index is
    returned and this results in incorrect programming of coefficients.

    To fix this, align the tegra210_sfc_rates[] array with SFC register
    specification and thus add 64kHz entry to it. Also, the coefficient
    table is updated to reflect that none of the conversions are supported
    for 64kHz.

    Fixes: b2f74ec53a6c ("ASoC: tegra: Add Tegra210 based SFC driver")
    Cc: stable@vger.kernel.org
    Signed-off-by: Sheetal <sheetal@nvidia.com>
    Reviewed-by: Mohan Kumar D <mkumard@nvidia.com>
    Reviewed-by: Sameer Pujar <spujar@nvidia.com>
    Link: https://lore.kernel.org/r/Message-Id: <1687433656-7892-2-git-send-email-
spujar@nvidia.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b0cce2c7960bce06fae80a63cc15d31ea43f00a4
Author: Thomas Zimmermann <tzimmermann@suse.de>
Date:   Wed Jun 21 14:53:35 2023 +0200

    drm/ast: Fix DRAM init on AST2200

    commit 4cfe75f0f14f044dae66ad0e6eea812d038465d9 upstream.

    Fix the test for the AST2200 in the DRAM initialization. The value
```

```
    in ast->chip has to be compared against an enum constant instead of
    a numerical value.

    This bug got introduced when the driver was first imported into the
    kernel.

    Signed-off-by: Thomas Zimmermann <tzimmermann@suse.de>
    Fixes: 312fec1405dd ("drm: Initial KMS driver for AST (ASpeed Technologies) 2000 series
(v2)")
    Cc: Dave Airlie <airlied@redhat.com>
    Cc: dri-devel@lists.freedesktop.org
    Cc: <stable@vger.kernel.org> # v3.5+
    Reviewed-by: Sui Jingfeng <suijingfeng@loongson.cn>
    Reviewed-by: Jocelyn Falempe <jfalempe@redhat.com>
    Tested-by: Jocelyn Falempe <jfalempe@redhat.com> # AST2600
    Link: https://patchwork.freedesktop.org/patch/msgid/20230621130032.3568-2-
tzimmermann@suse.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 44bc5f5c252d33623711a3bb1dac5774ceadffc1
Author: Johan Hovold <johan+linaro@kernel.org>
Date:   Tue Jul 18 15:28:55 2023 +0200

    clk: qcom: camcc-sc7180: fix async resume during probe

    commit c948ff727e25297f3a703eb5349dd66aabf004e4 upstream.

    To make sure that the controller is runtime resumed and its power domain
    is enabled before accessing its registers during probe, the synchronous
    runtime PM interface must be used.

    Fixes: 8d4025943e13 ("clk: qcom: camcc-sc7180: Use runtime PM ops instead of clk ones")
    Cc: stable@vger.kernel.org      # 5.11
    Cc: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Johan Hovold <johan+linaro@kernel.org>
    Link: https://lore.kernel.org/r/20230718132902.21430-2-johan+linaro@kernel.org
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0517fc5a71333b315164736bbd32608894fbb872
Author: Thomas Zimmermann <tzimmermann@suse.de>
Date:   Tue Jun 13 13:06:49 2023 +0200

    fbdev/ep93xx-fb: Do not assign to struct fb_info.dev

    commit f90a0e5265b60cdd3c77990e8105f79aa2fac994 upstream.

    Do not assing the Linux device to struct fb_info.dev. The call to
    register_framebuffer() initializes the field to the fbdev device.
    Drivers should not override its value.

    Fixes a bug where the driver incorrectly decreases the hardware
    device's reference counter and leaks the fbdev device.

    v2:
            * add Fixes tag (Dan)

    Signed-off-by: Thomas Zimmermann <tzimmermann@suse.de>
    Fixes: 88017bda96a5 ("ep93xx video driver")
    Cc: <stable@vger.kernel.org> # v2.6.32+
    Reviewed-by: Javier Martinez Canillas <javierm@redhat.com>
    Reviewed-by: Sam Ravnborg <sam@ravnborg.org>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230613110953.24176-15-
tzimmermann@suse.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 9566b301c1a398158f1f03b69373bf46f2af020b
Author: Ian Kent <raven@themaw.net>
Date:   Sun Aug 6 09:26:49 2023 +0800

    kernfs: fix missing kernfs_iattr_rwsem locking

    commit 0559f63057f927d298d68294d6ff77ce09b99255 upstream.

    When the kernfs_iattr_rwsem was introduced a case was missed.

    The update of the kernfs directory node child count was also protected
    by the kernfs_rwsem and needs to be included in the change so that the
    child count (and so the inode n_link attribute) does not change while
    holding the rwsem for read.

    Fixes: 9caf69614225 ("kernfs: Introduce separate rwsem to protect inode attributes.")
    Cc: stable <stable@kernel.org>
    Signed-off-by: Ian Kent <raven@themaw.net>
    Reviewed-By: Imran Khan <imran.f.khan@oracle.com>
    Acked-by: Miklos Szeredi <mszeredi@redhat.com>
    Cc: Anders Roxell <anders.roxell@linaro.org>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Cc: Minchan Kim <minchan@kernel.org>
    Cc: Eric Sandeen <sandeen@sandeen.net>
    Link:
https://lore.kernel.org/r/169128520941.68052.15749253469930138901.stgit@donald.themaw.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a7cb2e709f2927cc3c76781df3e45de2381b3b9d
Author: Chengming Zhou <zhouchengming@bytedance.com>
Date:   Fri Sep 1 20:03:06 2023 +0800

    null_blk: fix poll request timeout handling

    commit 5a26e45edb4690d58406178b5a9ea4c6dcf2c105 upstream.

    When doing io_uring benchmark on /dev/nullb0, it's easy to crash the
    kernel if poll requests timeout triggered, as reported by David. [1]

    BUG: kernel NULL pointer dereference, address: 0000000000000008
    Workqueue: kblockd blk_mq_timeout_work
    RIP: 0010:null_timeout_rq+0x4e/0x91
    Call Trace:
     ? null_timeout_rq+0x4e/0x91
     blk_mq_handle_expired+0x31/0x4b
     bt_iter+0x68/0x84
     ? bt_tags_iter+0x81/0x81
     __sbitmap_for_each_set.constprop.0+0xb0/0xf2
     ? __blk_mq_complete_request_remote+0xf/0xf
     bt_for_each+0x46/0x64
     ? __blk_mq_complete_request_remote+0xf/0xf
     ? percpu_ref_get_many+0xc/0x2a
     blk_mq_queue_tag_busy_iter+0x14d/0x18e
     blk_mq_timeout_work+0x95/0x127
     process_one_work+0x185/0x263
     worker_thread+0x1b5/0x227

    This is indeed a race problem between null_timeout_rq() and null_poll().

    null_poll()                               null_timeout_rq()
      spin_lock(&nq->poll_lock)
      list_splice_init(&nq->poll_list, &list)
      spin_unlock(&nq->poll_lock)
```

```
       while (!list_empty(&list))
         req = list_first_entry()
         list_del_init()
         ...
         blk_mq_add_to_batch()
         // req->rq_next = NULL
                                            spin_lock(&nq->poll_lock)

                                            // rq->queuelist->next == NULL
                                            list_del_init(&rq->queuelist)

                                            spin_unlock(&nq->poll_lock)
```

```
    Fix these problems by setting requests state to MQ_RQ_COMPLETE under
    nq->poll_lock protection, in which null_timeout_rq() can safely detect
    this race and early return.

    Note this patch just fix the kernel panic when request timeout happen.

    [1] https://lore.kernel.org/all/3893581.1691785261@warthog.procyon.org.uk/

    Fixes: 0a593fbbc245 ("null_blk: poll queue support")
    Reported-by: David Howells <dhowells@redhat.com>
    Tested-by: David Howells <dhowells@redhat.com>
    Reviewed-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: Chengming Zhou <zhouchengming@bytedance.com>
    Link: https://lore.kernel.org/r/20230901120306.170520-2-chengming.zhou@linux.dev
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7c059b6a6ec24fa307cd36bdb948756b82972dd2
Author: Quinn Tran <qutran@marvell.com>
Date:   Mon Aug 21 18:30:39 2023 +0530

    scsi: qla2xxx: Fix firmware resource tracking

    commit e370b64c7db96384a0886a09a9d80406e4c663d7 upstream.

    The storage was not draining I/Os and the work load was not spread out
    across different CPUs evenly. This led to firmware resource counters
    getting overrun on the busy CPU. This overrun prevented error recovery from
    happening in a timely manner.

    By switching the counter to atomic, it allows the count to be little more
    accurate to prevent the overrun.

    Cc: stable@vger.kernel.org
    Fixes: da7c21b72aa8 ("scsi: qla2xxx: Fix command flush during TMF")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230821130045.34850-4-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6c3c9b0d13e93198e18516cd981581602ff2d455
Author: Quinn Tran <qutran@marvell.com>
Date:   Mon Aug 21 18:30:41 2023 +0530

    scsi: qla2xxx: Error code did not return to upper layer

    commit 0ba0b018f94525a6b32f5930f980ce9b62b72e6f upstream.

    TMF was returned with an error code. The error code was not preserved to be
    returned to upper layer. Instead, the error code from the Marker was
```

```
    returned.

    Preserve error code from TMF and return it to upper layer.

    Cc: stable@vger.kernel.org
    Fixes: da7c21b72aa8 ("scsi: qla2xxx: Fix command flush during TMF")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230821130045.34850-6-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a923d44190762979025ee5bec8ed733c79cc1954
Author: Nilesh Javali <njavali@marvell.com>
Date:   Mon Aug 21 18:30:43 2023 +0530

    scsi: qla2xxx: Fix smatch warn for qla_init_iocb_limit()

    commit b496953dd0444001b12f425ea07d78c1f47e3193 upstream.

    Fix indentation for warning reported by smatch:

    drivers/scsi/qla2xxx/qla_init.c:4199 qla_init_iocb_limit() warn: inconsistent indenting

    Fixes: efa74a62aaa2 ("scsi: qla2xxx: Adjust IOCB resource on qpair create")
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230821130045.34850-8-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ac67720f5475d3d88baa83f5d037994c4b853822
Author: Quinn Tran <qutran@marvell.com>
Date:   Mon Aug 21 18:30:38 2023 +0530

    scsi: qla2xxx: Flush mailbox commands on chip reset

    commit 6d0b65569c0a10b27c49bacd8d25bcd406003533 upstream.

    Fix race condition between Interrupt thread and Chip reset thread in trying
    to flush the same mailbox. With the race condition, the "ha->mbx_intr_comp"
    will get an extra complete() call. The extra complete call create erroneous
    mailbox timeout condition when the next mailbox is sent where the mailbox
    call does not wait for interrupt to arrive. Instead, it advances without
    waiting.

    Add lock protection around the check for mailbox completion.

    Cc: stable@vger.kernel.org
    Fixes: b2000805a975 ("scsi: qla2xxx: Flush mailbox commands on chip reset")
    Signed-off-by: Quinn Tran <quinn.tran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230821130045.34850-3-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5c4246a3a054945ecaf5dfedf02e2e73b8043fea
Author: Manish Rangankar <mrangankar@marvell.com>
Date:   Mon Aug 21 18:30:42 2023 +0530

    scsi: qla2xxx: Remove unsupported ql2xenabledif option

    commit e9105c4b7a9208a21a9bda133707624f12ddabc2 upstream.
```

```
    User accidently passed module parameter ql2xenabledif=1 which is
    unsupported. However, driver still initialized which lead to guard tag
    errors during device discovery.

    Remove unsupported ql2xenabledif=1 option and validate the user input.

    Cc: stable@vger.kernel.org
    Signed-off-by: Manish Rangankar <mrangankar@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230821130045.34850-7-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 34ccae749d18141e6c34247202f30d1060e55c53
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:31:02 2023 +0530

    scsi: qla2xxx: Fix TMF leak through

    commit 5d3148d8e8b05f084e607ac3bd55a4c317a9f934 upstream.

    Task management can retry up to 5 times when FW resource becomes bottle
    neck. Between the retries, there is a short sleep.  Current code assumes
    the chip has not reset or session has not changed.

    Check for chip reset or session change before sending Task management.

    Cc: stable@vger.kernel.org
    Fixes: 9803fb5d2759 ("scsi: qla2xxx: Fix task management cmd failure")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-9-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bf464c4d0496a40ca89d1ca9e34600cc95e3f50e
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:31:00 2023 +0530

    scsi: qla2xxx: Fix session hang in gnl

    commit 39d22740712c7563a2e18c08f033deeacdaf66e7 upstream.

    Connection does not resume after a host reset / chip reset. The cause of
    the blockage is due to the FCF_ASYNC_ACTIVE left on. The gnl command was
    interrupted by the chip reset. On exiting the command, this flag should be
    turn off to allow relogin to reoccur. Clear this flag to prevent blockage.

    Cc: stable@vger.kernel.org
    Fixes: 17e64648aa47 ("scsi: qla2xxx: Correct fcport flags handling")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-7-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 997f10317ed4f84fb3b326acb2e1363fd33878d9
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:31:01 2023 +0530

    scsi: qla2xxx: Turn off noisy message log
```

```
    commit 8ebaa45163a3fedc885c1dc7d43ea987a2f00a06 upstream.

    Some consider noisy log as test failure.  Turn off noisy message log.

    Cc: stable@vger.kernel.org
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-8-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5add37906f18e9c720795da6f2dbd2b9396b6e2f
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:30:59 2023 +0530

    scsi: qla2xxx: Fix erroneous link up failure

    commit 5b51f35d127e7bef55fa869d2465e2bca4636454 upstream.

    Link up failure occurred where driver failed to see certain events from FW
    indicating link up (AEN 8011) and fabric login completion (AEN 8014).
    Without these 2 events, driver would not proceed forward to scan the
    fabric. The cause of this is due to delay in the receive of interrupt for
    Mailbox 60 that causes qla to set the fw_started flag late.  The late
    setting of this flag causes other interrupts to be dropped.  These dropped
    interrupts happen to be the link up (AEN 8011) and fabric login completion
    (AEN 8014).

    Set fw_started flag early to prevent interrupts being dropped.

    Cc: stable@vger.kernel.org
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-6-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b617809d87377aba8fb2e7830d1ac240332d74c9
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:30:58 2023 +0530

    scsi: qla2xxx: Fix command flush during TMF

    commit da7c21b72aa86e990af5f73bce6590b8d8d148d0 upstream.

    For each TMF request, driver iterates through each qpair and flushes
    commands associated to the TMF. At the end of the qpair flush, a Marker is
    used to complete the flush transaction. This process was repeated for each
    qpair. The multiple flush and marker for this TMF request seems to cause
    confusion for FW.

    Instead, 1 flush is sent to FW. Driver would wait for FW to go through all
    the I/Os on each qpair to be read then return. Driver then closes out the
    transaction with a Marker.

    Cc: stable@vger.kernel.org
    Fixes: d90171dd0da5 ("scsi: qla2xxx: Multi-que support for TMF")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-5-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
```

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

commit 8ef539cac3576a513c705986a4a299d0c2ea48a9
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:31:03 2023 +0530

    scsi: qla2xxx: fix inconsistent TMF timeout

    commit 009e7fe4a1ed52276b332842a6b6e23b07200f2d upstream.

    Different behavior were experienced of session being torn down vs not when
    TMF is timed out. When FW detects the time out, the session is torn down.
    When driver detects the time out, the session is not torn down.

    Allow TMF error to return to upper layer without session tear down.

    Cc: stable@vger.kernel.org
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-10-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b05017cb4ff75eea783583f3d400059507510ab1
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:30:55 2023 +0530

    scsi: qla2xxx: Fix deletion race condition

    commit 6dfe4344c168c6ca20fe7640649aacfcefcccb26 upstream.

    System crash when using debug kernel due to link list corruption. The cause
    of the link list corruption is due to session deletion was allowed to queue
    up twice.  Here's the internal trace that show the same port was allowed to
    double queue for deletion on different cpu.

    20808683956 015 qla2xxx [0000:13:00.1]-e801:4: Scheduling sess ffff93ebf9306800 for
deletion 50:06:0e:80:12:48:ff:50 fc4_type 1
    20808683957 027 qla2xxx [0000:13:00.1]-e801:4: Scheduling sess ffff93ebf9306800 for
deletion 50:06:0e:80:12:48:ff:50 fc4_type 1

    Move the clearing/setting of deleted flag lock.

    Cc: stable@vger.kernel.org
    Fixes: 726b85487067 ("qla2xxx: Add framework for async fabric discovery")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-2-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 69228675683e345d2dbf3be8b0ea4f24c4d60d96
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:30:57 2023 +0530

    scsi: qla2xxx: Limit TMF to 8 per function

    commit a8ec192427e0516436e61f9ca9eb49c54eadfe0a upstream.

    Per FW recommendation, 8 TMF's can be outstanding for each
    function. Previously, it allowed 8 per target.

    Limit TMF to 8 per function.
```

```
    Cc: stable@vger.kernel.org
    Fixes: 6a87679626b5 ("scsi: qla2xxx: Fix task management cmd fail due to unavailable
resource")
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-4-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0e2f41d082285c9a5788c97372088ed1f30ab02c
Author: Quinn Tran <qutran@marvell.com>
Date:   Fri Jul 14 12:30:56 2023 +0530

    scsi: qla2xxx: Adjust IOCB resource on qpair create

    commit efa74a62aaa2429c04fe6cb277b3bf6739747d86 upstream.

    During NVMe queue creation, a new qpair is created. FW resource limit needs
    to be re-adjusted to take into account the new qpair. Otherwise, NVMe
    command can not go through.  This issue was discovered while
    testing/forcing FW execution to fail at load time.

    Add call to readjust IOCB and exchange limit.

    In addition, get FW state command and require FW to be running. Otherwise,
    error is generated.

    Cc: stable@vger.kernel.org
    Signed-off-by: Quinn Tran <qutran@marvell.com>
    Signed-off-by: Nilesh Javali <njavali@marvell.com>
    Link: https://lore.kernel.org/r/20230714070104.40052-3-njavali@marvell.com
    Reviewed-by: Himanshu Madhani <himanshu.madhani@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ea834e6609b82293abffe810428ee4e1881ee457
Author: Bean Huo <beanhuo@micron.com>
Date:   Wed Aug 9 20:18:46 2023 +0200

    scsi: ufs: core: Add advanced RPMB support where UFSHCI 4.0 does not support EHS length
in UTRD

    commit c91e585cfb3dd7d076e9ba0967908fc504d32def upstream.

    According to UFSHCI 4.0 specification:

    5.2 Host Controller Capabilities Registers
    5.2.1 Offset 00h: CAP – Controller Capabilities:

     "EHS Length in UTRD Supported (EHSLUTRDS): Indicates whether the host
      controller supports EHS Length field in UTRD.

      0 – Host controller takes EHS length from CMD UPIU, and SW driver use EHS
      Length field in CMD UPIU.

      1 – HW controller takes EHS length from UTRD, and SW driver use EHS
      Length field in UTRD.

      NOTE Recommend Host controllers move to taking EHS length from UTRD, and
      in UFS-5, it will be mandatory."

    So, when UFSHCI 4.0 doesn't support EHS Length field in UTRD, we could use
    EHS Length field in CMD UPIU. Remove the limitation that advanced RPMB only
```

```
    works when EHS length is supported in UTRD.

    Fixes: 6ff265fc5ef6 ("scsi: ufs: core: bsg: Add advanced RPMB support in ufs_bsg")
    Co-developed-by: "jonghwi.rha" <jonghwi.rha@samsung.com>
    Signed-off-by: "jonghwi.rha" <jonghwi.rha@samsung.com>
    Signed-off-by: Bean Huo <beanhuo@micron.com>
    Link: https://lore.kernel.org/r/20230809181847.102123-2-beanhuo@iokpp.de
    Reviewed-by: Bart Van Assche <bvanassche@acm.org>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4037dcd70c3335600a2a95ec6229d70859bd37a1
Author: Gurchetan Singh <gurchetansingh@chromium.org>
Date:   Fri Jul 7 14:31:24 2023 -0700

    drm/virtio: Conditionally allocate virtio_gpu_fence

    commit 70d1ace56db6c79d39dbe9c0d5244452b67e2fde upstream.

    We don't want to create a fence for every command submission.  It's
    only necessary when userspace provides a waitable token for submission.
    This could be:

    1) bo_handles, to be used with VIRTGPU_WAIT
    2) out_fence_fd, to be used with dma_fence apis
    3) a ring_idx provided with VIRTGPU_CONTEXT_PARAM_POLL_RINGS_MASK
       + DRM event API
    4) syncobjs in the future

    The use case for just submitting a command to the host, and expecting
    no response.  For example, gfxstream has GFXSTREAM_CONTEXT_PING that
    just wakes up the host side worker threads.  There's also
    CROSS_DOMAIN_CMD_SEND which just sends data to the Wayland server.

    This prevents the need to signal the automatically created
    virtio_gpu_fence.

    In addition, VIRTGPU_EXECBUF_RING_IDX is checked when creating a
    DRM event object.  VIRTGPU_CONTEXT_PARAM_POLL_RINGS_MASK is
    already defined in terms of per-context rings.  It was theoretically
    possible to create a DRM event on the global timeline (ring_idx == 0),
    if the context enabled DRM event polling.  However, that wouldn't
    work and userspace (Sommelier).  Explicitly disallow it for
    clarity.

    Signed-off-by: Gurchetan Singh <gurchetansingh@chromium.org>
    Reviewed-by: Dmitry Osipenko <dmitry.osipenko@collabora.com>
    Tested-by: Dmitry Osipenko <dmitry.osipenko@collabora.com>
    Signed-off-by: Dmitry Osipenko <dmitry.osipenko@collabora.com> # edited coding style
    Link: https://patchwork.freedesktop.org/patch/msgid/20230707213124.494-1-
gurchetansingh@chromium.org
    Signed-off-by: Alyssa Ross <hi@alyssa.is>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 23e0882db356a403ca92c467b2a7c6e0cc149fe6
Author: Quan Tian <qtian@vmware.com>
Date:   Tue Sep 5 10:36:10 2023 +0000

    net/ipv6: SKB symmetric hash should incorporate transport ports

    commit a5e2151ff9d5852d0ababbbcaeebd9646af9c8d9 upstream.

    __skb_get_hash_symmetric() was added to compute a symmetric hash over
    the protocol, addresses and transport ports, by commit eb70db875671
    ("packet: Use symmetric hash for PACKET_FANOUT_HASH."). It uses
```

```
    flow_keys_dissector_symmetric_keys as the flow_dissector to incorporate
    IPv4 addresses, IPv6 addresses and ports. However, it should not specify
    the flag as FLOW_DISSECTOR_F_STOP_AT_FLOW_LABEL, which stops further
    dissection when an IPv6 flow label is encountered, making transport
    ports not being incorporated in such case.

    As a consequence, the symmetric hash is based on 5-tuple for IPv4 but
    3-tuple for IPv6 when flow label is present. It caused a few problems,
    e.g. when nft symhash and openvswitch l4_sym rely on the symmetric hash
    to perform load balancing as different L4 flows between two given IPv6
    addresses would always get the same symmetric hash, leading to uneven
    traffic distribution.

    Removing the use of FLOW_DISSECTOR_F_STOP_AT_FLOW_LABEL makes sure the
    symmetric hash is based on 5-tuple for both IPv4 and IPv6 consistently.

    Fixes: eb70db875671 ("packet: Use symmetric hash for PACKET_FANOUT_HASH.")
    Reported-by: Lars Ekman <uablrek@gmail.com>
    Closes: https://github.com/antrea-io/antrea/issues/5457
    Signed-off-by: Quan Tian <qtian@vmware.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```