```
commit 5c2fc53857eb993952e932da8222b11b063c2581
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed May 18 10:28:23 2022 +0200

    Linux 5.17.9

    Link: https://lore.kernel.org/r/20220516193625.489108457@linuxfoundation.org
    Tested-by: Zan Aziz <zanaziz313@gmail.com>
    Tested-by: Fox Chen <foxhlchen@gmail.com>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Fenil Jain<fkjainco@gmail.com>
    Tested-by: Florian Fainelli <f.fainelli@gmail.com>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 25d8511ed7c2ead1723e24f93d94882cb9c05b50
Author: Hyeonggon Yoo <42.hyeyoo@gmail.com>
Date:   Mon May 9 17:34:29 2022 -0700

    mm/kfence: reset PG_slab and memcg_data before freeing __kfence_pool

    commit 2839b0999c20c9f6bf353849c69370e121e2fa1a upstream.

    When kfence fails to initialize kfence pool, it frees the pool.  But it
    does not reset memcg_data and PG_slab flag.

    Below is a BUG because of this. Let's fix it by resetting memcg_data
    and PG_slab flag before free.

    [    0.089149] BUG: Bad page state in process swapper/0  pfn:3d8e06
    [    0.089149] page:ffffea46cf638180 refcount:0 mapcount:0 mapping:0000000000000000
index:0x0 pfn:0x3d8e06
    [    0.089150] memcg:ffffffff94a475d1
    [    0.089150] flags: 0x17ffffc0000200(slab|node=0|zone=2|lastcpupid=0x1fffff)
    [    0.089151] raw: 0017ffffc0000200 ffffea46cf638188 ffffea46cf638188 0000000000000000
    [    0.089152] raw: 0000000000000000 0000000000000000 00000000ffffffff ffffffff94a475d1
    [    0.089152] page dumped because: page still charged to cgroup
    [    0.089153] Modules linked in:
    [    0.089153] CPU: 0 PID: 0 Comm: swapper/0 Tainted: G    B   W         5.18.0-rc1+
#965
    [    0.089154] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-2
04/01/2014
    [    0.089154] Call Trace:
    [    0.089155]  <TASK>
    [    0.089155]  dump_stack_lvl+0x49/0x5f
    [    0.089157]  dump_stack+0x10/0x12
    [    0.089158]  bad_page.cold+0x63/0x94
    [    0.089159]  check_free_page_bad+0x66/0x70
    [    0.089160]  __free_pages_ok+0x423/0x530
    [    0.089161]  __free_pages_core+0x8e/0xa0
    [    0.089162]  memblock_free_pages+0x10/0x12
    [    0.089164]  memblock_free_late+0x8f/0xb9
    [    0.089165]  kfence_init+0x68/0x92
    [    0.089166]  start_kernel+0x789/0x992
    [    0.089167]  x86_64_start_reservations+0x24/0x26
    [    0.089168]  x86_64_start_kernel+0xa9/0xaf
    [    0.089170]  secondary_startup_64_no_verify+0xd5/0xdb
    [    0.089171]  </TASK>

    Link: https://lkml.kernel.org/r/YnPG3pQrqfcgOlVa@hyeyoo
```

```
    Fixes: 0ce20dd84089 ("mm: add Kernel Electric-Fence infrastructure")
    Fixes: 8f0b36497303 ("mm: kfence: fix objcgs vector allocation")
    Signed-off-by: Hyeonggon Yoo <42.hyeyoo@gmail.com>
    Reviewed-by: Marco Elver <elver@google.com>
    Reviewed-by: Muchun Song <songmuchun@bytedance.com>
    Cc: Alexander Potapenko <glider@google.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 837c8c6f623b8991962f148ceb316c5b840e50a7
Author: Wan Jiabing <wanjiabing@vivo.com>
Date:   Mon May 9 22:45:19 2022 +0800

    net: phy: micrel: Fix incorrect variable type in micrel

    commit 12a4d677b1c34717443470c1492fe520638ef39a upstream.

    In lanphy_read_page_reg, calling __phy_read() might return a negative
    error code. Use 'int' to check the error code.

    Fixes: 7c2dcfa295b1 ("net: phy: micrel: Add support for LAN8804 PHY")
    Signed-off-by: Wan Jiabing <wanjiabing@vivo.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20220509144519.2343399-1-wanjiabing@vivo.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4f6d1d30ce0f56b24e71f359d7a98f93dad573a3
Author: Naoya Horiguchi <naoya.horiguchi@nec.com>
Date:   Thu Apr 28 23:14:44 2022 -0700

    mm/hwpoison: use pr_err() instead of dump_page() in get_any_page()

    commit 1825b93b626e99eb9a0f9f50342c7b2fa201b387 upstream.

    The following VM_BUG_ON_FOLIO() is triggered when memory error event
    happens on the (thp/folio) pages which are about to be freed:

      [ 1160.232771] page:00000000b36a8a0f refcount:1 mapcount:0 mapping:0000000000000000
index:0x1 pfn:0x16a000
      [ 1160.236916] page:00000000b36a8a0f refcount:0 mapcount:0 mapping:0000000000000000
index:0x1 pfn:0x16a000
      [ 1160.240684] flags: 0x57ffffc0800000(hwpoison|node=1|zone=2|lastcpupid=0x1fffff)
      [ 1160.243458] raw: 0057ffffc0800000 dead000000000100 dead000000000122
0000000000000000
      [ 1160.246268] raw: 0000000000000001 0000000000000000 00000000ffffffff
0000000000000000
      [ 1160.249197] page dumped because: VM_BUG_ON_FOLIO(!folio_test_large(folio))
      [ 1160.251815] ------------[ cut here ]------------
      [ 1160.253438] kernel BUG at include/linux/mm.h:788!
      [ 1160.256162] invalid opcode: 0000 [#1] PREEMPT SMP PTI
      [ 1160.258172] CPU: 2 PID: 115368 Comm: mceinj.sh Tainted: G            E     5.18.0-
rc1-v5.18-rc1-220404-2353-005-g83111+ #3
      [ 1160.262049] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-
1.fc35 04/01/2014
      [ 1160.265103] RIP: 0010:dump_page.cold+0x27e/0x2bd
      [ 1160.266757] Code: fe ff ff 48 c7 c6 81 f1 5a 98 e9 4c fe ff ff 48 c7 c6 a1 95 59 98
e9 40 fe ff ff 48 c7 c6 50 bf 5a 98 48 89 ef e8 9d 04 6d ff <0f> 0b 41 f7 c4 ff 0f 00 00 0f
85 9f fd ff ff 49 8b 04 24 a9 00 00
      [ 1160.273180] RSP: 0018:ffffaa2c4d59fd18 EFLAGS: 00010292
      [ 1160.274969] RAX: 000000000000003e RBX: 0000000000000001 RCX: 0000000000000000
      [ 1160.277263] RDX: 0000000000000001 RSI: ffffffff985995a1 RDI: 00000000ffffffff
      [ 1160.279571] RBP: ffffdc9c45a80000 R08: 0000000000000000 R09: 00000000ffffdfff
      [ 1160.281794] R10: ffffaa2c4d59fb08 R11: ffffffff98940d08 R12: ffffdc9c45a80000
```

```
       [ 1160.283920] R13: ffffffff985b6f94 R14: 0000000000000000 R15: ffffdc9c45a80000
       [ 1160.286641] FS:  00007eff54ce1740(0000) GS:ffff99c67bd00000(0000)
  knlGS:0000000000000000
       [ 1160.289498] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
       [ 1160.291106] CR2: 00005628381a5f68 CR3: 0000000104712003 CR4: 0000000000170ee0
       [ 1160.293031] Call Trace:
       [ 1160.293724]  <TASK>
       [ 1160.294334]  get_hwpoison_page+0x47d/0x570
       [ 1160.295474]  memory_failure+0x106/0xaa0
       [ 1160.296474]  ? security_capable+0x36/0x50
       [ 1160.297524]  hard_offline_page_store+0x43/0x80
       [ 1160.298684]  kernfs_fop_write_iter+0x11c/0x1b0
       [ 1160.299829]  new_sync_write+0xf9/0x160
       [ 1160.300810]  vfs_write+0x209/0x290
       [ 1160.301835]  ksys_write+0x4f/0xc0
       [ 1160.302718]  do_syscall_64+0x3b/0x90
       [ 1160.303664]  entry_SYSCALL_64_after_hwframe+0x44/0xae
       [ 1160.304981] RIP: 0033:0x7eff54b018b7

    As shown in the RIP address, this VM_BUG_ON in folio_entire_mapcount() is
    called from dump_page("hwpoison: unhandlable page") in get_any_page().
    The below explains the mechanism of the race:

      CPU 0                                          CPU 1

        memory_failure
          get_hwpoison_page
            get_any_page
              dump_page
                compound = PageCompound
                                                       free_pages_prepare
                                                         page->flags &=
  ~PAGE_FLAGS_CHECK_AT_PREP
                folio_entire_mapcount
                  VM_BUG_ON_FOLIO(!folio_test_large(folio))

    So replace dump_page() with safer one, pr_err().

    Link: https://lkml.kernel.org/r/20220427053220.719866-1-naoya.horiguchi@linux.dev
    Fixes: 74e8ee4708a8 ("mm: Turn head_compound_mapcount() into folio_entire_mapcount()")
    Signed-off-by: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Reviewed-by: John Hubbard <jhubbard@nvidia.com>
    Reviewed-by: Miaohe Lin <linmiaohe@huawei.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Christoph Hellwig <hch@infradead.org>
    Cc: Jason Gunthorpe <jgg@nvidia.com>
    Cc: William Kucharski <william.kucharski@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 3a322e9954cee9d4f2d15ca315d1cd2b54d1641d
  Author: Charan Teja Reddy <quic_charante@quicinc.com>
  Date:   Tue May 10 01:19:57 2022 +0530

    dma-buf: call dma_buf_stats_setup after dmabuf is in valid list

    commit ef3a6b70507a2add2cd2e01f5eb9b54d561bacb9 upstream.

    When dma_buf_stats_setup() fails, it closes the dmabuf file which
    results into the calling of dma_buf_file_release() where it does
    list_del(&dmabuf->list_node) with out first adding it to the proper
    list. This is resulting into panic in the below path:
    __list_del_entry_valid+0x38/0xac
    dma_buf_file_release+0x74/0x158
```

```
      __fput+0xf4/0x428
      ____fput+0x14/0x24
      task_work_run+0x178/0x24c
      do_notify_resume+0x194/0x264
      work_pending+0xc/0x5f0

      Fix it by moving the dma_buf_stats_setup() after dmabuf is added to the
      list.

      Fixes: bdb8d06dfefd ("dmabuf: Add the capability to expose DMA-BUF stats in sysfs")
      Signed-off-by: Charan Teja Reddy <quic_charante@quicinc.com>
      Tested-by: T.J. Mercier <tjmercier@google.com>
      Acked-by: T.J. Mercier <tjmercier@google.com>
      Cc: <stable@vger.kernel.org> # 5.15.x+
      Reviewed-by: Christian König <christian.koenig@amd.com>
      Signed-off-by: Christian König <christian.koenig@amd.com>
      Link: https://patchwork.freedesktop.org/patch/msgid/1652125797-2043-1-git-send-email-
  quic_charante@quicinc.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit cfcd74c197b2ff882a488233af2d8eeb37b77d6a
  Author: Alex Deucher <alexander.deucher@amd.com>
  Date:   Tue May 10 09:37:06 2022 -0400

      Revert "drm/amd/pm: keep the BACO feature enabled for suspend"

      commit a56f445f807b0276fc0660c330bf93a9ea78e8ea upstream.

      This reverts commit eaa090538e8d21801c6d5f94590c3799e6a528b5.

      Commit ebc002e3ee78 ("drm/amdgpu: don't use BACO for reset in S3")
      stops using BACO for reset during suspend, so it's no longer
      necessary to leave BACO enabled during suspend.  This fixes
      resume from suspend on the navy flounder dGPU in the ASUS ROG
      Strix G513QY.

      Bug: https://gitlab.freedesktop.org/drm/amd/-/issues/2008
      Bug: https://gitlab.freedesktop.org/drm/amd/-/issues/1982
      Reviewed-by: Lijo Lazar <lijo.lazar@amd.com>
      Reviewed-by: Mario Limonciello <mario.limonciello@amd.com>
      Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
      Cc: stable@vger.kernel.org
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit e199707bec9875e8fe30b171332c4bacd42823d4
  Author: Zack Rusin <zackr@vmware.com>
  Date:   Wed Mar 2 10:24:24 2022 -0500

      drm/vmwgfx: Initialize drm_mode_fb_cmd2

      commit 3059d9b9f6aa433a55b9d0d21b566396d5497c33 upstream.

      Transition to drm_mode_fb_cmd2 from drm_mode_fb_cmd left the structure
      unitialized. drm_mode_fb_cmd2 adds a few additional members, e.g. flags
      and modifiers which were never initialized. Garbage in those members
      can cause random failures during the bringup of the fbcon.

      Initializing the structure fixes random blank screens after bootup due
      to flags/modifiers mismatches during the fbcon bring up.

      Fixes: dabdcdc9822a ("drm/vmwgfx: Switch to mode_cmd2")
      Signed-off-by: Zack Rusin <zackr@vmware.com>
      Cc: Daniel Vetter <daniel.vetter@intel.com>
      Cc: <stable@vger.kernel.org> # v4.10+
      Reviewed-by: Martin Krastev <krastevm@vmware.com>
```

```
    Reviewed-by: Maaz Mombasawala <mombasawalam@vmware.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220302152426.885214-7-zack@kde.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit eef49e281a1fc43acd85327e5a630647c45c13bf
Author: Trond Myklebust <trond.myklebust@hammerspace.com>
Date:   Sat May 7 13:53:59 2022 -0400

    SUNRPC: Ensure that the gssproxy client can start in a connected state

    commit fd13359f54ee854f00134abc6be32da94ec53dbf upstream.

    Ensure that the gssproxy client connects to the server from the gssproxy
    daemon process context so that the AF_LOCAL socket connection is done
    using the correct path and namespaces.

    Fixes: 1d658336b05f ("SUNRPC: Add RPC based upcall mechanism for RPCGSS auth")
    Cc: stable@vger.kernel.org
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8822e08d81d55635f902f3c6660f927f9c8d8c27
Author: Fabio Estevam <festevam@denx.de>
Date:   Wed May 4 11:31:04 2022 -0300

    net: phy: micrel: Pass .probe for KS8737

    commit 15f03ffe4bb951e982457f44b6cf6b06ef4cbb93 upstream.

    Since commit f1131b9c23fb ("net: phy: micrel: use
    kszphy_suspend()/kszphy_resume for irq aware devices") the kszphy_suspend/
    resume hooks are used.

    These functions require the probe function to be called so that
    priv can be allocated.

    Otherwise, a NULL pointer dereference happens inside
    kszphy_config_reset().

    Cc: stable@vger.kernel.org
    Fixes: f1131b9c23fb ("net: phy: micrel: use kszphy_suspend()/kszphy_resume for irq aware
devices")
    Reported-by: Andrew Lunn <andrew@lunn.ch>
    Signed-off-by: Fabio Estevam <festevam@denx.de>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20220504143104.1286960-2-festevam@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 01b651d572a6cdc3a2cc089482c6b8463cfb1b3a
Author: Fabio Estevam <festevam@denx.de>
Date:   Wed May 4 11:31:03 2022 -0300

    net: phy: micrel: Do not use kszphy_suspend/resume for KSZ8061

    commit e333eed63a091a09bd0db191b7710c594c6e995b upstream.

    Since commit f1131b9c23fb ("net: phy: micrel: use
    kszphy_suspend()/kszphy_resume for irq aware devices") the following
    NULL pointer dereference is observed on a board with KSZ8061:

     # udhcpc -i eth0
    udhcpc: started, v1.35.0
    8<--- cut here ---
    Unable to handle kernel NULL pointer dereference at virtual address 00000008
```

```
    pgd = f73cef4e
    [00000008] *pgd=00000000
    Internal error: Oops: 5 [#1] SMP ARM
    Modules linked in:
    CPU: 0 PID: 196 Comm: ifconfig Not tainted 5.15.37-dirty #94
    Hardware name: Freescale i.MX6 SoloX (Device Tree)
    PC is at kszphy_config_reset+0x10/0x114
    LR is at kszphy_resume+0x24/0x64
    ...

    The KSZ8061 phy_driver structure does not have the .probe/..driver_data
    fields, which means that priv is not allocated.

    This causes the NULL pointer dereference inside kszphy_config_reset().

    Fix the problem by using the generic suspend/resume functions as before.

    Another alternative would be to provide the .probe and .driver_data
    information into the structure, but to be on the safe side, let's
    just restore Ethernet functionality by using the generic suspend/resume.

    Cc: stable@vger.kernel.org
    Fixes: f1131b9c23fb ("net: phy: micrel: use kszphy_suspend()/kszphy_resume for irq aware
devices")
    Signed-off-by: Fabio Estevam <festevam@denx.de>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20220504143104.1286960-1-festevam@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0e9a7116138bb7107accafd6b057729234b3b57d
Author: Mike Rapoport <rppt@kernel.org>
Date:   Mon May 9 17:34:28 2022 -0700

    arm[64]/memremap: don't abuse pfn_valid() to ensure presence of linear map

    commit 260364d112bc822005224667c0c9b1b17a53eafd upstream.

    The semantics of pfn_valid() is to check presence of the memory map for a
    PFN and not whether a PFN is covered by the linear map.  The memory map
    may be present for NOMAP memory regions, but they won't be mapped in the
    linear mapping.  Accessing such regions via __va() when they are
    memremap()'ed will cause a crash.

    On v5.4.y the crash happens on qemu-arm with UEFI [1]:

    <1>[    0.084476] 8<--- cut here ---
    <1>[    0.084595] Unable to handle kernel paging request at virtual address dfb76000
    <1>[    0.084938] pgd = (ptrval)
    <1>[    0.085038] [dfb76000] *pgd=5f7fe801, *pte=00000000, *ppte=00000000

    ...

    <4>[    0.093923] [<c0ed6ce8>] (memcpy) from [<c16a06f8>] (dmi_setup+0x60/0x418)
    <4>[    0.094204] [<c16a06f8>] (dmi_setup) from [<c16a38d4>] (arm_dmi_init+0x8/0x10)
    <4>[    0.094408] [<c16a38d4>] (arm_dmi_init) from [<c0302e9c>]
(do_one_initcall+0x50/0x228)
    <4>[    0.094619] [<c0302e9c>] (do_one_initcall) from [<c16011e4>]
(kernel_init_freeable+0x15c/0x1f8)
    <4>[    0.094841] [<c16011e4>] (kernel_init_freeable) from [<c0f028cc>]
(kernel_init+0x8/0x10c)
    <4>[    0.095057] [<c0f028cc>] (kernel_init) from [<c03010e8>] (ret_from_fork+0x14/0x2c)

    On kernels v5.10.y and newer the same crash won't reproduce on ARM because
    commit b10d6bca8720 ("arch, drivers: replace for_each_membock() with
```

```
    for_each_mem_range()") changed the way memory regions are registered in
    the resource tree, but that merely covers up the problem.

    On ARM64 memory resources registered in yet another way and there the
    issue of wrong usage of pfn_valid() to ensure availability of the linear
    map is also covered.

    Implement arch_memremap_can_ram_remap() on ARM and ARM64 to prevent access
    to NOMAP regions via the linear mapping in memremap().

    Link: https://lore.kernel.org/all/Yl65zxGgFzF1Okac@sirena.org.uk
    Link: https://lkml.kernel.org/r/20220426060107.7618-1-rppt@kernel.org
    Signed-off-by: Mike Rapoport <rppt@linux.ibm.com>
    Reported-by: "kernelci.org bot" <bot@kernelci.org>
    Tested-by: Mark Brown <broonie@kernel.org>
    Reviewed-by: Ard Biesheuvel <ardb@kernel.org>
    Acked-by: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Mark Brown <broonie@kernel.org>
    Cc: Mark-PK Tsai <mark-pk.tsai@mediatek.com>
    Cc: Russell King <linux@armlinux.org.uk>
    Cc: Tony Lindgren <tony@atomide.com>
    Cc: Will Deacon <will@kernel.org>
    Cc: <stable@vger.kernel.org>    [5.4+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d19c3a70c4102b2a86560038a9076bd00cb794c5
Author: Wen Gong <quic_wgong@quicinc.com>
Date:   Wed Apr 27 14:16:19 2022 +0300

    ath11k: reduce the wait time of 11d scan and hw scan while add interface

    commit bb300130e47fcefbe938f06dbacaef0312e28416 upstream.

    (cherry picked from commit 1f682dc9fb3790aa7ec27d3d122ff32b1eda1365 in wireless-next)

    Currently ath11k will wait 11d scan complete while add interface in
    ath11k_mac_op_add_interface(), when system resume without enable
    wowlan, ath11k_mac_op_add_interface() is called for each resume, thus
    it increase the resume time of system. And ath11k_mac_op_hw_scan()
    after ath11k_mac_op_add_interface() also needs some time cost because
    the previous 11d scan need more than 5 seconds when 6 GHz is enabled,
    then the scan started event will indicated to ath11k after the 11d
    scan completed.

    While 11d scan/hw scan is running in firmware, if ath11k update channel
    list to firmware by WMI_SCAN_CHAN_LIST_CMDID, then firmware will cancel
    the current scan which is running, it lead the scan failed. The patch
    commit 9dcf6808b253 ("ath11k: add 11d scan offload support") used
    finish_11d_scan/finish_11d_ch_list/pending_11d to synchronize the 11d
    scan/hw scan/channel list between ath11k/firmware/mac80211 and to avoid
    the scan fail.

    Add wait operation before ath11k update channel list, function
    ath11k_reg_update_chan_list() will wait until the current 11d scan/hw
    scan completed. And remove the wait operation of start 11d scan and
    waiting channel list complete in hw scan. After these changes, resume
    time cost reduce about 5 seconds and also hw scan time cost reduced
    obviously, and scan failed not seen.

    The 11d scan is sent to firmware only one time for each interface added
    in mac.c, and it is moved after the 1st hw scan because 11d scan will
    cost some time and thus leads the AP scan result update to UI delay.
    Currently priority of ath11k's hw scan is WMI_SCAN_PRIORITY_LOW, and
```

```
    priority of 11d scan in firmware is WMI_SCAN_PRIORITY_MEDIUM, then the
    11d scan which sent after hw scan will cancel the hw scan in firmware,
    so change the priority to WMI_SCAN_PRIORITY_MEDIUM for the hw scan which
    is in front of the 11d scan, thus it will not happen scan cancel in
    firmware.

    Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3

    Fixes: 9dcf6808b253 ("ath11k: add 11d scan offload support")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=215777
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Wen Gong <quic_wgong@quicinc.com>
    Signed-off-by: Kalle Valo <quic_kvalo@quicinc.com>
    Link: https://lore.kernel.org/r/20220328035832.14122-1-quic_wgong@quicinc.com
    Signed-off-by: Kalle Valo <kvalo@kernel.org>
    Link: https://lore.kernel.org/r/20220427111619.9758-1-kvalo@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5a71708ceb018e9e030e7469f8536e65d7bc6096
Author: Nicolas Dichtel <nicolas.dichtel@6wind.com>
Date:   Wed May 4 11:07:38 2022 +0200

    ping: fix address binding wrt vrf

    commit e1a7ac6f3ba6e157adcd0ca94d92a401f1943f56 upstream.

    When ping_group_range is updated, 'ping' uses the DGRAM ICMP socket,
    instead of an IP raw socket. In this case, 'ping' is unable to bind its
    socket to a local address owned by a vrflite.

    Before the patch:
    $ sysctl -w net.ipv4.ping_group_range='0  2147483647'
    $ ip link add blue type vrf table 10
    $ ip link add foo type dummy
    $ ip link set foo master blue
    $ ip link set foo up
    $ ip addr add 192.168.1.1/24 dev foo
    $ ip addr add 2001::1/64 dev foo
    $ ip vrf exec blue ping -c1 -I 192.168.1.1 192.168.1.2
    ping: bind: Cannot assign requested address
    $ ip vrf exec blue ping6 -c1 -I 2001::1 2001::2
    ping6: bind icmp socket: Cannot assign requested address

    CC: stable@vger.kernel.org
    Fixes: 1b69c6d0ae90 ("net: Introduce L3 Master device abstraction")
    Signed-off-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8d83493a462a0a51f194c90b559072e9c7c738e0
Author: Waiman Long <longman@redhat.com>
Date:   Wed Apr 27 10:54:28 2022 -0400

    cgroup/cpuset: Remove cpus_allowed/mems_allowed setup in cpuset_init_smp()

    commit 2685027fca387b602ae565bff17895188b803988 upstream.

    There are 3 places where the cpu and node masks of the top cpuset can
    be initialized in the order they are executed:
     1) start_kernel -> cpuset_init()
     2) start_kernel -> cgroup_init() -> cpuset_bind()
     3) kernel_init_freeable() -> do_basic_setup() -> cpuset_init_smp()

    The first cpuset_init() call just sets all the bits in the masks.
```

```
    The second cpuset_bind() call sets cpus_allowed and mems_allowed to the
    default v2 values. The third cpuset_init_smp() call sets them back to
    v1 values.

    For systems with cgroup v2 setup, cpuset_bind() is called once.  As a
    result, cpu and memory node hot add may fail to update the cpu and node
    masks of the top cpuset to include the newly added cpu or node in a
    cgroup v2 environment.

    For systems with cgroup v1 setup, cpuset_bind() is called again by
    rebind_subsystem() when the v1 cpuset filesystem is mounted as shown
    in the dmesg log below with an instrumented kernel.

      [    2.609781] cpuset_bind() called - v2 = 1
      [    3.079473] cpuset_init_smp() called
      [    7.103710] cpuset_bind() called - v2 = 0

    smp_init() is called after the first two init functions.  So we don't
    have a complete list of active cpus and memory nodes until later in
    cpuset_init_smp() which is the right time to set up effective_cpus
    and effective_mems.

    To fix this cgroup v2 mask setup problem, the potentially incorrect
    cpus_allowed & mems_allowed setting in cpuset_init_smp() are removed.
    For cgroup v2 systems, the initial cpuset_bind() call will set the masks
    correctly.  For cgroup v1 systems, the second call to cpuset_bind()
    will do the right setup.

    cc: stable@vger.kernel.org
    Signed-off-by: Waiman Long <longman@redhat.com>
    Tested-by: Feng Tang <feng.tang@intel.com>
    Reviewed-by: Michal Koutný <mkoutny@suse.com>
    Signed-off-by: Tejun Heo <tj@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 42d3af388150e280d1d1091435f91e496f2b0a0d
Author: Jing Xia <jing.xia@unisoc.com>
Date:   Tue May 10 10:35:14 2022 +0800

    writeback: Avoid skipping inode writeback

    commit 846a3351ddfe4a86eede4bb26a205c3f38ef84d3 upstream.

    We have run into an issue that a task gets stuck in
    balance_dirty_pages_ratelimited() when perform I/O stress testing.
    The reason we observed is that an I_DIRTY_PAGES inode with lots
    of dirty pages is in b_dirty_time list and standard background
    writeback cannot writeback the inode.
    After studing the relevant code, the following scenario may lead
    to the issue:

    task1                                   task2
    -----                                   -----
    fuse_flush
     write_inode_now //in b_dirty_time
      writeback_single_inode
       __writeback_single_inode
                                    fuse_write_end
                                     filemap_dirty_folio
                                      __xa_set_mark:PAGECACHE_TAG_DIRTY
         lock inode->i_lock
         if mapping tagged PAGECACHE_TAG_DIRTY
         inode->i_state |= I_DIRTY_PAGES
         unlock inode->i_lock
                                       __mark_inode_dirty:I_DIRTY_PAGES
```

```
                                        lock inode->i_lock
                                        -was dirty,inode stays in
                                        -b_dirty_time
                                        unlock inode->i_lock

       if(!(inode->i_state & I_DIRTY_All))
          -not true,so nothing done

    This patch moves the dirty inode to b_dirty list when the inode
    currently is not queued in b_io or b_more_io list at the end of
    writeback_single_inode.

    Reviewed-by: Jan Kara <jack@suse.cz>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    CC: stable@vger.kernel.org
    Fixes: 0ae45f63d4ef ("vfs: add support for a lazytime mount option")
    Signed-off-by: Jing Xia <jing.xia@unisoc.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20220510023514.27399-1-jing.xia@unisoc.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0c29394c66330386d34c7483cd9f8412ad0347c8
Author: Francesco Dolcini <francesco.dolcini@toradex.com>
Date:   Fri May 6 08:08:15 2022 +0200

    net: phy: Fix race condition on link status change

    commit 91a7cda1f4b8bdf770000a3b60640576dafe0cec upstream.

    This fixes the following error caused by a race condition between
    phydev->adjust_link() and a MDIO transaction in the phy interrupt
    handler. The issue was reproduced with the ethernet FEC driver and a
    micrel KSZ9031 phy.

    [  146.195696] fec 2188000.ethernet eth0: MDIO read timeout
    [  146.201779] ------------[ cut here ]------------
    [  146.206671] WARNING: CPU: 0 PID: 571 at drivers/net/phy/phy.c:942 phy_error+0x24/0x6c
    [  146.214744] Modules linked in: bnep imx_vdoa imx_sdma evbug
    [  146.220640] CPU: 0 PID: 571 Comm: irq/128-2188000 Not tainted 5.18.0-rc3-00080-
gd569e86915b7 #9
    [  146.229563] Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
    [  146.236257]  unwind_backtrace from show_stack+0x10/0x14
    [  146.241640]  show_stack from dump_stack_lvl+0x58/0x70
    [  146.246841]  dump_stack_lvl from __warn+0xb4/0x24c
    [  146.251772]  __warn from warn_slowpath_fmt+0x5c/0xd4
    [  146.256873]  warn_slowpath_fmt from phy_error+0x24/0x6c
    [  146.262249]  phy_error from kszphy_handle_interrupt+0x40/0x48
    [  146.268159]  kszphy_handle_interrupt from irq_thread_fn+0x1c/0x78
    [  146.274417]  irq_thread_fn from irq_thread+0xf0/0x1dc
    [  146.279605]  irq_thread from kthread+0xe4/0x104
    [  146.284267]  kthread from ret_from_fork+0x14/0x28
    [  146.289164] Exception stack(0xe6fa1fb0 to 0xe6fa1ff8)
    [  146.294448] 1fa0:                                     00000000 00000000 00000000
00000000
    [  146.302842] 1fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000
    [  146.311281] 1fe0: 00000000 00000000 00000000 00000000 00000013 00000000
    [  146.318262] irq event stamp: 12325
    [  146.321780] hardirqs last  enabled at (12333): [<c01984c4>]
__up_console_sem+0x50/0x60
    [  146.330013] hardirqs last disabled at (12342): [<c01984b0>]
__up_console_sem+0x3c/0x60
    [  146.338259] softirqs last  enabled at (12324): [<c01017f0>] __do_softirq+0x2c0/0x624
    [  146.346311] softirqs last disabled at (12319): [<c01300ac>]
__irq_exit_rcu+0x138/0x178
```

```
    [  146.354447] ---[ end trace 0000000000000000 ]---

    With the FEC driver phydev->adjust_link() calls fec_enet_adjust_link()
    calls fec_stop()/fec_restart() and both these function reset and
    temporary disable the FEC disrupting any MII transaction that
    could be happening at the same time.

    fec_enet_adjust_link() and phy_read() can be running at the same time
    when we have one additional interrupt before the phy_state_machine() is
    able to terminate.

    Thread 1 (phylib WQ)       | Thread 2 (phy interrupt)
                               |
                               | phy_interrupt()            <-- PHY IRQ
                               |  handle_interrupt()
                               |   phy_read()
                               |   phy_trigger_machine()
                               |    --> schedule phylib WQ
                               |
                               |
    phy_state_machine()        |
     phy_check_link_status()   |
      phy_link_change()        |
       phydev->adjust_link()   |
        fec_enet_adjust_link() |
         --> FEC reset         | phy_interrupt()            <-- PHY IRQ
                               |  phy_read()
                               |

    Fix this by acquiring the phydev lock in phy_interrupt().

    Link: https://lore.kernel.org/all/20220422152612.GA510015@francesco-nb.int.toradex.com/
    Fixes: c974bdbc3e77 ("net: phy: Use threaded IRQ, to allow IRQ from sleeping devices")
    cc: <stable@vger.kernel.org>
    Signed-off-by: Francesco Dolcini <francesco.dolcini@toradex.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20220506060815.327382-1-francesco.dolcini@toradex.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fb45f24264b94e8c37657789dbb1966a07933d12
Author: Manuel Ullmann <labre@posteo.de>
Date:   Wed May 4 21:30:44 2022 +0200

    net: atlantic: always deep reset on pm op, fixing up my null deref regression

    commit 1809c30b6e5a83a1de1435fe01aaa4de4d626a7c upstream.

    The impact of this regression is the same for resume that I saw on
    thaw: the kernel hangs and nothing except SysRq rebooting can be done.

    Fixes regression in commit cbe6c3a8f8f4 ("net: atlantic: invert deep
    par in pm functions, preventing null derefs"), where I disabled deep
    pm resets in suspend and resume, trying to make sense of the
    atl_resume_common() deep parameter in the first place.

    It turns out, that atlantic always has to deep reset on pm
    operations. Even though I expected that and tested resume, I screwed
    up by kexec-rebooting into an unpatched kernel, thus missing the
    breakage.

    This fixup obsoletes the deep parameter of atl_resume_common, but I
    leave the cleanup for the maintainers to post to mainline.

    Suspend and hibernation were successfully tested by the reporters.
```

```
    Fixes: cbe6c3a8f8f4 ("net: atlantic: invert deep par in pm functions, preventing null
derefs")
    Link: https://lore.kernel.org/regressions/9-
Ehc_xXSwdXcvZqKD5aSqsqeNj5Izco4MYEwnx5cySXVEc9-x_WC4C3kAoCqNTi-
H38frroUK17iobNVnkLtW36V6VWGSQEOHXhmVMm5iQ=@protonmail.com/
    Reported-by: Jordan Leppert <jordanleppert@protonmail.com>
    Reported-by: Holger Hoffstaette <holger@applied-asynchrony.com>
    Tested-by: Jordan Leppert <jordanleppert@protonmail.com>
    Tested-by: Holger Hoffstaette <holger@applied-asynchrony.com>
    CC: <stable@vger.kernel.org> # 5.10+
    Signed-off-by: Manuel Ullmann <labre@posteo.de>
    Link: https://lore.kernel.org/r/87bkw8dfmp.fsf@posteo.de
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d2a5e5b06158b4b87e2592368f809343f65c3ec
Author: Xiaomeng Tong <xiam0nd.tong@gmail.com>
Date:   Tue May 10 13:48:46 2022 -0700

    i40e: i40e_main: fix a missing check on list iterator

    commit 3f95a7472d14abef284d8968734fe2ae7ff4845f upstream.

    The bug is here:
            ret = i40e_add_macvlan_filter(hw, ch->seid, vdev->dev_addr, &aq_err);

    The list iterator 'ch' will point to a bogus position containing
    HEAD if the list is empty or no element is found. This case must
    be checked before any use of the iterator, otherwise it will
    lead to a invalid memory access.

    To fix this bug, use a new variable 'iter' as the list iterator,
    while use the origin variable 'ch' as a dedicated pointer to
    point to the found element.

    Cc: stable@vger.kernel.org
    Fixes: 1d8d80b4e4ff6 ("i40e: Add macvlan support on i40e")
    Signed-off-by: Xiaomeng Tong <xiam0nd.tong@gmail.com>
    Tested-by: Gurucharan <gurucharanx.g@intel.com> (A Contingent worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Link: https://lore.kernel.org/r/20220510204846.2166999-1-anthony.l.nguyen@intel.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 751654995bc89b7b175e8fbed2b1ab027231e551
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Tue Apr 5 15:21:34 2022 +0100

    drm/nouveau/tegra: Stop using iommu_present()

    commit 87fd2b091fb33871a7f812658a0971e8e26f903f upstream.

    Even if some IOMMU has registered itself on the platform "bus", that
    doesn't necessarily mean it provides translation for the device we
    care about. Replace iommu_present() with a more appropriate check.

    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    [added cc for stable]
    Signed-off-by: Lyude Paul <lyude@redhat.com>
    Cc: stable@vger.kernel.org # v5.0+
    Link:
https://patchwork.freedesktop.org/patch/msgid/70d40ea441da3663c2824d54102b471e9a621f8a.16491
68494.git.robin.murphy@arm.com
```

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

commit e1e84d3adb689e978d2f546b76d62d616bf3aba7
Author: Zack Rusin <zackr@vmware.com>
Date:   Fri Mar 18 13:43:31 2022 -0400

    drm/vmwgfx: Disable command buffers on svga3 without gbobjects

    commit 21d1d192890ced87f2f04f8f4dea92406e0b162a upstream.

    With very limited vram on svga3 it's difficult to handle all the surface
    migrations. Without gbobjects, i.e. the ability to store surfaces in
    guest mobs, there's no reason to support intermediate svga2 features,
    especially because we can fall back to fb traces and svga3 will never
    support those in-between features.

    On svga3 we wither want to use fb traces or screen targets
    (i.e. gbobjects), nothing in between. This fixes presentation on a lot
    of fusion/esxi tech previews where the exposed svga3 caps haven't been
    finalized yet.

    Signed-off-by: Zack Rusin <zackr@vmware.com>
    Fixes: 2cd80dbd3551 ("drm/vmwgfx: Add basic support for SVGA3")
    Cc: <stable@vger.kernel.org> # v5.14+
    Reviewed-by: Martin Krastev <krastevm@vmware.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220318174332.440068-5-zack@kde.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f6fd9552ef33ff76535d163aa8d75b0e46222450
Author: Niels Dossche <dossche.niels@gmail.com>
Date:   Mon May 9 17:34:28 2022 -0700

    mm: mremap: fix sign for EFAULT error return value

    commit 7d1e6496616275f3830e2f2f91fa69a66953e95b upstream.

    The mremap syscall is supposed to return a pointer to the new virtual
    memory area on success, and a negative value of the error code in case of
    failure.  Currently, EFAULT is returned when the VMA is not found, instead
    of -EFAULT.  The users of this syscall will therefore believe the syscall
    succeeded in case the VMA didn't exist, as it returns a pointer to address
    0xe (0xe being the value of EFAULT).  Fix the sign of the error value.

    Link: https://lkml.kernel.org/r/20220427224439.23828-2-dossche.niels@gmail.com
    Fixes: 550a7d60bd5e ("mm, hugepages: add mremap() support for hugepage backed vma")
    Signed-off-by: Niels Dossche <dossche.niels@gmail.com>
    Cc: Mina Almasry <almasrymina@google.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 13d9b8cd12f37d133b07ea5b323583e8a0c6b738
Author: Xu Yu <xuyu@linux.alibaba.com>
Date:   Thu Apr 28 23:14:43 2022 -0700

    mm/huge_memory: do not overkill when splitting huge_zero_page

    commit 478d134e9506c7e9bfe2830ed03dd85e97966313 upstream.

    Kernel panic when injecting memory_failure for the global huge_zero_page,
    when CONFIG_DEBUG_VM is enabled, as follows.

      Injecting memory failure for pfn 0x109ff9 at process virtual address 0x20ff9000
      page:00000000fb053fc3 refcount:2 mapcount:0 mapping:0000000000000000 index:0x0
```

```
pfn:0x109e00
      head:00000000fb053fc3 order:9 compound_mapcount:0 compound_pincount:0
      flags: 0x17fffc000010001(locked|head|node=0|zone=2|lastcpupid=0x1ffff)
      raw: 017fffc000010001 0000000000000000 dead000000000122 0000000000000000
      raw: 0000000000000000 0000000000000000 00000002ffffffff 0000000000000000
      page dumped because: VM_BUG_ON_PAGE(is_huge_zero_page(head))
      ------------[ cut here ]------------
      kernel BUG at mm/huge_memory.c:2499!
      invalid opcode: 0000 [#1] PREEMPT SMP PTI
      CPU: 6 PID: 553 Comm: split_bug Not tainted 5.18.0-rc1+ #11
      Hardware name: Alibaba Cloud Alibaba Cloud ECS, BIOS 3288b3c 04/01/2014
      RIP: 0010:split_huge_page_to_list+0x66a/0x880
      Code: 84 9b fb ff ff 48 8b 7c 24 08 31 f6 e8 9f 5d 2a 00 b8 b8 02 00 00 e9 e8 fb ff ff
48 c7 c6 e8 47 3c 82 4c b
      RSP: 0018:ffffc90000dcbdf8 EFLAGS: 00010246
      RAX: 000000000000003c RBX: 0000000000000001 RCX: 0000000000000000
      RDX: 0000000000000000 RSI: ffffffff823e4c4f RDI: 00000000ffffffff
      RBP: ffff88843fffdb40 R08: 0000000000000000 R09: 00000000fffeffff
      R10: ffffc90000dcbc48 R11: ffffffff82d68448 R12: ffffea0004278000
      R13: ffffffff823c6203 R14: 0000000000109ff9 R15: ffffea000427fe40
      FS:  00007fc375a26740(0000) GS:ffff88842fd80000(0000) knlGS:0000000000000000
      CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      CR2: 00007fc3757c9290 CR3: 0000000102174006 CR4: 00000000003706e0
      DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
      DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
      Call Trace:
      try_to_split_thp_page+0x3a/0x130
      memory_failure+0x128/0x800
      madvise_inject_error.cold+0x8b/0xa1
      __x64_sys_madvise+0x54/0x60
      do_syscall_64+0x35/0x80
      entry_SYSCALL_64_after_hwframe+0x44/0xae
      RIP: 0033:0x7fc3754f8bf9
      Code: 01 00 48 81 c4 80 00 00 00 e9 f1 fe ff ff 0f 1f 00 48 89 f8 48 89 f7 48 89 d6 48
89 ca 4d 89 c2 4d 89 c8 8
      RSP: 002b:00007ffeda93a1d8 EFLAGS: 00000217 ORIG_RAX: 000000000000001c
      RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007fc3754f8bf9
      RDX: 0000000000000064 RSI: 0000000000003000 RDI: 0000000020ff9000
      RBP: 00007ffeda93a200 R08: 0000000000000000 R09: 0000000000000000
      R10: 00000000ffffffff R11: 0000000000000217 R12: 0000000000400490
      R13: 00007ffeda93a2e0 R14: 0000000000000000 R15: 0000000000000000

    We think that raising BUG is overkilling for splitting huge_zero_page, the
    huge_zero_page can't be met from normal paths other than memory failure,
    but memory failure is a valid caller.  So we tend to replace the BUG to
    WARN + returning -EBUSY, and thus the panic above won't happen again.

    Link:
https://lkml.kernel.org/r/f35f8b97377d5d3ede1bc5ac3114da888c57cbce.1651052574.git.xuyu@linux
.alibaba.com
    Fixes: d173d5417fb6 ("mm/memory-failure.c: skip huge_zero_page in memory_failure()")
    Fixes: 6a46079cf57a ("HWPOISON: The high level memory error handler in the VM v7")
    Signed-off-by: Xu Yu <xuyu@linux.alibaba.com>
    Suggested-by: Yang Shi <shy828301@gmail.com>
    Reported-by: kernel test robot <lkp@intel.com>
    Reviewed-by: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Reviewed-by: Yang Shi <shy828301@gmail.com>
    Reviewed-by: Miaohe Lin <linmiaohe@huawei.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 86079983c0f255e78649b41885762d2907801274
Author: Xu Yu <xuyu@linux.alibaba.com>
Date:   Thu Apr 28 23:14:43 2022 -0700
```

```
    Revert "mm/memory-failure.c: skip huge_zero_page in memory_failure()"

    commit b4e61fc031b11dd807dffc46cebbf0e25966d3d1 upstream.

    Patch series "mm/memory-failure: rework fix on huge_zero_page splitting".


    This patch (of 2):

    This reverts commit d173d5417fb67411e623d394aab986d847e47dad.

    The commit d173d5417fb6 ("mm/memory-failure.c: skip huge_zero_page in
    memory_failure()") explicitly skips huge_zero_page in memory_failure(), in
    order to avoid triggering VM_BUG_ON_PAGE on huge_zero_page in
    split_huge_page_to_list().

    This works, but Yang Shi thinks that,

        Raising BUG is overkilling for splitting huge_zero_page. The
        huge_zero_page can't be met from normal paths other than memory
        failure, but memory failure is a valid caller. So I tend to replace
        the BUG to WARN + returning -EBUSY. If we don't care about the
        reason code in memory failure, we don't have to touch memory
        failure.

    And for the issue that huge_zero_page will be set PG_has_hwpoisoned,
    Yang Shi comments that,

        The anonymous page fault doesn't check if the page is poisoned or
        not since it typically gets a fresh allocated page and assumes the
        poisoned page (isolated successfully) can't be reallocated again.
        But huge zero page and base zero page are reused every time. So no
        matter what fix we pick, the issue is always there.

    Finally, Yang, David, Anshuman and Naoya all agree to fix the bug, i.e.,
    to split huge_zero_page, in split_huge_page_to_list().

    This reverts the commit d173d5417fb6 ("mm/memory-failure.c: skip
    huge_zero_page in memory_failure()"), and the original bug will be fixed
    by the next patch.

    Link:
https://lkml.kernel.org/r/872cefb182ba1dd686b0e7db1e6b2ebe5a4fff87.1651039624.git.xuyu@linux
.alibaba.com
    Fixes: d173d5417fb6 ("mm/memory-failure.c: skip huge_zero_page in memory_failure()")
    Fixes: 6a46079cf57a ("HWPOISON: The high level memory error handler in the VM v7")
    Signed-off-by: Xu Yu <xuyu@linux.alibaba.com>
    Suggested-by: Yang Shi <shy828301@gmail.com>
    Reviewed-by: Yang Shi <shy828301@gmail.com>
    Reviewed-by: Miaohe Lin <linmiaohe@huawei.com>
    Cc: Naoya Horiguchi <naoya.horiguchi@nec.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 25633e355cbea61e5a18b938a56f391b7185cf60
Author: Jeff Layton <jlayton@kernel.org>
Date:   Mon Apr 25 15:54:27 2022 -0400

    ceph: fix setting of xattrs on async created inodes

    commit 620239d9a32e9fe27c9204ec11e40058671aeeb6 upstream.

    Currently when we create a file, we spin up an xattr buffer to send
```

```
    along with the create request. If we end up doing an async create
    however, then we currently pass down a zero-length xattr buffer.

    Fix the code to send down the xattr buffer in req->r_pagelist. If the
    xattrs span more than a page, however give up and don't try to do an
    async create.

    Cc: stable@vger.kernel.org
    URL: https://bugzilla.redhat.com/show_bug.cgi?id=2063929
    Fixes: 9a8d03ca2e2c ("ceph: attempt to do async create when possible")
    Reported-by: John Fortin <fortinj66@gmail.com>
    Reported-by: Sri Ramanujam <sri@ramanujam.io>
    Signed-off-by: Jeff Layton <jlayton@kernel.org>
    Reviewed-by: Xiubo Li <xiubli@redhat.com>
    Signed-off-by: Ilya Dryomov <idryomov@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a75811da73f3a661eef450941cbcbe9c5d90c439
Author: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
Date:   Wed Apr 27 15:23:28 2022 +0200

    serial: 8250_mtk: Fix register address for XON/XOFF character

    commit e1bfdbc7daca171c74a577b3dd0b36d76bb0ffcc upstream.

    The XON1/XOFF1 character registers are at offset 0xa0 and 0xa8
    respectively, so we cannot use the definition in serial_port.h.

    Fixes: bdbd0a7f8f03 ("serial: 8250-mtk: modify baudrate setting")
    Signed-off-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20220427132328.228297-4-
angelogioacchino.delregno@collabora.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 36723b9a2bcd9b60e1ebd34783bbc5e10554e91e
Author: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
Date:   Wed Apr 27 15:23:26 2022 +0200

    serial: 8250_mtk: Fix UART_EFR register address

    commit bb0b197aadd928f52ce6f01f0ee977f0a08cf1be upstream.

    On MediaTek SoCs, the UART IP is 16550A compatible, but there are some
    specific quirks: we are declaring a register shift of 2, but this is
    only valid for the majority of the registers, as there are some that
    are out of the standard layout.

    Specifically, this driver is using definitions from serial_reg.h, where
    we have a UART_EFR register defined as 2: this results in a 0x8 offset,
    but there we have the FCR register instead.

    The right offset for the EFR register on MediaTek UART is at 0x98,
    so, following the decimal definition convention in serial_reg.h and
    accounting for the register left shift of two, add and use the correct
    register address for this IP, defined as decimal 38, so that the final
    calculation results in (0x26 << 2) = 0x98.

    Fixes: bdbd0a7f8f03 ("serial: 8250-mtk: modify baudrate setting")
    Signed-off-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20220427132328.228297-2-
angelogioacchino.delregno@collabora.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit a8ef54640b3b44cc52204c40db79d010c895ea5c
Author: Lukas Wunner <lukas@wunner.de>
Date:   Tue May 10 09:56:05 2022 +0200

    genirq: Remove WARN_ON_ONCE() in generic_handle_domain_irq()

    commit 792ea6a074ae7ea5ab6f1b8b31f76bb0297de66c upstream.

    Since commit 0953fb263714 ("irq: remove handle_domain_{irq,nmi}()"),
    generic_handle_domain_irq() warns if called outside hardirq context, even
    though the function calls down to handle_irq_desc(), which warns about the
    same, but conditionally on handle_enforce_irqctx().

    The newly added warning is a false positive if the interrupt originates
    from any other irqchip than x86 APIC or ARM GIC/GICv3.  Those are the only
    ones for which handle_enforce_irqctx() returns true.  Per commit
    c16816acd086 ("genirq: Add protection against unsafe usage of
    generic_handle_irq()"):

     "In general calling generic_handle_irq() with interrupts disabled from non
      interrupt context is harmless. For some interrupt controllers like the
      x86 trainwrecks this is outright dangerous as it might corrupt state if
      an interrupt affinity change is pending."

    Examples for interrupt chips where the warning is a false positive are
    USB-attached GPIO controllers such as drivers/gpio/gpio-dln2.c:

      USB gadgets are incapable of directly signaling an interrupt because they
      cannot initiate a bus transaction by themselves.  All communication on
      the bus is initiated by the host controller, which polls a gadget's
      Interrupt Endpoint in regular intervals.  If an interrupt is pending,
      that information is passed up the stack in softirq context, from which a
      hardirq is synthesized via generic_handle_domain_irq().

    Remove the warning to eliminate such false positives.

    Fixes: 0953fb263714 ("irq: remove handle_domain_{irq,nmi}()")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: Marc Zyngier <maz@kernel.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Jakub Kicinski <kuba@kernel.org>
    CC: Linus Walleij <linus.walleij@linaro.org>
    Cc: Bartosz Golaszewski <brgl@bgdev.pl>
    Cc: Octavian Purdila <octavian.purdila@nxp.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20220505113207.487861b2@kernel.org
    Link: https://lore.kernel.org/r/20220506203242.GA1855@wunner.de
    Link:
https://lore.kernel.org/r/c3caf60bfa78e5fdbdf483096b7174da65d1813a.1652168866.git.lukas@wunn
er.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f8a0a51f90fb09518d9cb662fdddcf58066bd30c
Author: Indan Zupancic <Indan.Zupancic@mep-info.com>
Date:   Thu May 5 13:47:50 2022 +0200

    fsl_lpuart: Don't enable interrupts too early

    commit 401fb66a355eb0f22096cf26864324f8e63c7d78 upstream.

    If an irq is pending when devm_request_irq() is called, the irq
    handler will cause a NULL pointer access because initialisation
    is not done yet.
```

```
    Fixes: 9d7ee0e28da59 ("tty: serial: lpuart: avoid report NULL interrupt")
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Indan Zupancic <Indan.Zupancic@mep-info.com>
    Link: https://lore.kernel.org/r/20220505114750.45423-1-Indan.Zupancic@mep-info.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 80406a1bad81e271e9ebf416a917b590a2272876
Author: Miaoqian Lin <linmq006@gmail.com>
Date:   Fri Apr 29 17:49:17 2022 +0100

    slimbus: qcom: Fix IRQ check in qcom_slim_probe

    commit fe503887eed6ea528e144ec8dacfa1d47aa701ac upstream.

    platform_get_irq() returns non-zero IRQ number on success,
    negative error number on failure.
    And the doc of platform_get_irq() provides a usage example:

        int irq = platform_get_irq(pdev, 0);
        if (irq < 0)
            return irq;

    Fix the check of return value to catch errors correctly.

    Fixes: ad7fcbc308b0 ("slimbus: qcom: Add Qualcomm Slimbus controller driver")
    Cc: stable@vger.kernel.org
    Signed-off-by: Miaoqian Lin <linmq006@gmail.com>
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Link: https://lore.kernel.org/r/20220429164917.5202-2-srinivas.kandagatla@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ae2ba69d6a8c896eacaf119c81844c307fff642e
Author: Sven Schwermer <sven.schwermer@disruptive-technologies.com>
Date:   Mon Apr 25 16:34:50 2022 +0200

    USB: serial: option: add Fibocom MA510 modem

    commit 07989eb981d862f7f2be68d233d753f2e7ccc119 upstream.

    The MA510 modem has 3 USB configurations that are configurable via the AT
    command AT+GTUSBMODE={30,31,32} which make the modem enumerate with the
    following interfaces, respectively:

    30: Diag + QDSS + Modem + RMNET
    31: Diag + Modem + AT + ECM
    32: Modem + AT + ECM

    The first configuration (30) reuses u-blox R410M's VID/PID with
    identical interface configuration.

    A detailed description of the USB configuration for each mode follows:

    +GTUSBMODE: 30
    --------------
    T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#= 19 Spd=480  MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=05c6 ProdID=90b2 Rev= 0.00
    S:  Manufacturer=Fibocom MA510 Modem
    S:  Product=Fibocom MA510 Modem
    S:  SerialNumber=55e2695b
    C:* #Ifs= 4 Cfg#= 1 Atr=e0 MxPwr=500mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
    E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 1 Alt= 0 #EPs= 1 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
```

```
E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=83(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=qmi_wwan
E:  Ad=85(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

+GTUSBMODE: 31
--------------
T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#= 99 Spd=480  MxCh= 0
D:  Ver= 2.00 Cls=ef(misc ) Sub=02 Prot=01 MxPS=64 #Cfgs=  1
P:  Vendor=2cb7 ProdID=0106 Rev= 0.00
S:  Manufacturer=Fibocom MA510 Modem
S:  Product=Fibocom MA510 Modem
S:  SerialNumber=55e2695b
C:* #Ifs= 5 Cfg#= 1 Atr=e0 MxPwr=500mA
A:  FirstIf#= 3 IfCount= 2 Cls=02(comm.) Sub=00 Prot=00
I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=82(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fe Prot=ff Driver=option
E:  Ad=84(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 3 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=06 Prot=00 Driver=cdc_ether
E:  Ad=86(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
I:  If#= 4 Alt= 0 #EPs= 0 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
I:* If#= 4 Alt= 1 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
E:  Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

+GTUSBMODE: 32
--------------
T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#=100 Spd=480  MxCh= 0
D:  Ver= 2.00 Cls=ef(misc ) Sub=02 Prot=01 MxPS=64 #Cfgs=  1
P:  Vendor=2cb7 ProdID=010a Rev= 0.00
S:  Manufacturer=Fibocom MA510 Modem
S:  Product=Fibocom MA510 Modem
S:  SerialNumber=55e2695b
C:* #Ifs= 4 Cfg#= 1 Atr=e0 MxPwr=500mA
A:  FirstIf#= 2 IfCount= 2 Cls=02(comm.) Sub=00 Prot=00
I:* If#= 0 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
E:  Ad=81(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=fe Prot=ff Driver=option
E:  Ad=83(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
I:* If#= 2 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=06 Prot=00 Driver=cdc_ether
E:  Ad=85(I) Atr=03(Int.) MxPS=  64 Ivl=2ms
I:  If#= 3 Alt= 0 #EPs= 0 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
I:* If#= 3 Alt= 1 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

Signed-off-by: Sven Schwermer <sven.schwermer@disruptive-technologies.com>
Cc: stable@vger.kernel.org
```

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

commit 0ae6a67bd169ec2835f47112993b04027050ec74
Author: Sven Schwermer <sven.schwermer@disruptive-technologies.com>
Date:   Mon Apr 25 16:34:49 2022 +0200

    USB: serial: option: add Fibocom L610 modem

    commit 714adff9a6271b5f1664b04c944b598141ebfe73 upstream.

    The L610 modem has 3 USB configurations that are configurable via the AT
    command AT+GTUSBMODE={31,32,33} which make the modem enumerate with the
    following interfaces, respectively:

    31: Modem + NV + MOS + Diag + LOG + AT + AT
    32: ECM + Modem + NV + MOS + Diag + LOG + AT + AT
    33: RNDIS + Modem + NV + MOS + Diag + LOG + AT + AT

    A detailed description of the USB configuration for each mode follows:

    +GTUSBMODE: 31
    --------------
    T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#=124 Spd=480  MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=1782 ProdID=4d10 Rev= 0.00
    S:  Manufacturer=FIBOCOM
    S:  Product=L610
    C:* #Ifs= 7 Cfg#= 1 Atr=e0 MxPwr=400mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 2 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 3 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 4 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 5 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=06(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 6 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=07(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

    +GTUSBMODE: 32
    --------------
    T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#=122 Spd=480  MxCh= 0
    D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=1782 ProdID=4d11 Rev= 0.00
    S:  Manufacturer=FIBOCOM
    S:  Product=L610
    C:* #Ifs= 9 Cfg#= 1 Atr=e0 MxPwr=400mA
    A:  FirstIf#= 0 IfCount= 2 Cls=02(comm.) Sub=06 Prot=00
    I:* If#= 0 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=06 Prot=00 Driver=cdc_ether
    E:  Ad=81(I) Atr=03(Int.) MxPS=  16 Ivl=32ms
    I:  If#= 1 Alt= 0 #EPs= 0 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
    I:* If#= 1 Alt= 1 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=cdc_ether
    E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
```

```
      E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 2 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 3 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 4 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 5 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 6 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=06(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 7 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=88(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=07(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 8 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=89(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=08(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

      +GTUSBMODE: 33
      --------------
      T:  Bus=03 Lev=01 Prnt=01 Port=06 Cnt=04 Dev#=126 Spd=480  MxCh= 0
      D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
      P:  Vendor=1782 ProdID=4d11 Rev= 0.00
      S:  Manufacturer=FIBOCOM
      S:  Product=L610
      C:* #Ifs= 9 Cfg#= 1 Atr=e0 MxPwr=400mA
      A:  FirstIf#= 0 IfCount= 2 Cls=e0(wlcon) Sub=01 Prot=03
      I:* If#= 0 Alt= 0 #EPs= 1 Cls=e0(wlcon) Sub=01 Prot=03 Driver=rndis_host
      E:  Ad=82(I) Atr=03(Int.) MxPS=   8 Ivl=4096ms
      I:* If#= 1 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=rndis_host
      E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 2 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 3 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 4 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 5 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 6 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=06(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 7 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=88(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=07(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 8 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=89(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=08(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

      Signed-off-by: Sven Schwermer <sven.schwermer@disruptive-technologies.com>
      Cc: stable@vger.kernel.org
      Signed-off-by: Johan Hovold <johan@kernel.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 7c31fc80c0d860a5405e90799e23b06b76eeed37
```

```
Author: Ethan Yang <etyang@sierrawireless.com>
Date:   Mon Apr 25 13:58:40 2022 +0800

    USB: serial: qcserial: add support for Sierra Wireless EM7590

    commit 870b1eee2d844727b06e238c121d260bc5645580 upstream.

    Add support for Sierra Wireless EM7590 0xc080/0xc081 compositions.

    Signed-off-by: Ethan Yang <etyang@sierrawireless.com>
    Link: https://lore.kernel.org/r/20220425055840.5693-1-etyang@sierrawireless.com
    Cc: stable@vger.kernel.org
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 441b3c5dc84dd421c9c09f56a745a00c44c4257a
Author: Scott Chen <scott@labau.com.tw>
Date:   Mon Apr 25 17:00:20 2022 +0800

    USB: serial: pl2303: add device id for HP LM930 Display

    commit 26a08f8bad3e1f98d3153f939fb8cd330da4cb26 upstream.

    Add the device id for the HPLM930Display which is a PL2303GC based
    device.

    Signed-off-by: Scott Chen <scott@labau.com.tw>
    Cc: stable@vger.kernel.org
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f1d4911bf0971f55a61e59fc7acf78a78e585336
Author: ChiYuan Huang <cy_huang@richtek.com>
Date:   Tue May 10 13:13:00 2022 +0800

    usb: typec: tcpci_mt6360: Update for BMC PHY setting

    commit 4031cd95cba70c72e4cadc2d46624bcd31e5a6c0 upstream.

    Update MT6360 BMC PHY Tx/Rx setting for the compatibility.

    Macpaul reported this CtoDP cable attention message cannot be received from
    MT6360 TCPC. But actually, attention message really sent from UFP_D
    device.

    After RD's comment, there may be BMC PHY Tx/Rx setting causes this issue.

    Below's the detailed TCPM log and DP attention message didn't received from 6360
    TCPCI.
    [ 1206.367775] Identity: 0000:0000.0000
    [ 1206.416570] Alternate mode 0: SVID 0xff01, VDO 1: 0x00000405
    [ 1206.447378] AMS DFP_TO_UFP_ENTER_MODE start
    [ 1206.447383] PD TX, header: 0x1d6f
    [ 1206.449393] PD TX complete, status: 0
    [ 1206.454110] PD RX, header: 0x184f [1]
    [ 1206.456867] Rx VDM cmd 0xff018144 type 1 cmd 4 len 1
    [ 1206.456872] AMS DFP_TO_UFP_ENTER_MODE finished
    [ 1206.456873] cc:=4
    [ 1206.473100] AMS STRUCTURED_VDMS start
    [ 1206.473103] PD TX, header: 0x2f6f
    [ 1206.475397] PD TX complete, status: 0
    [ 1206.480442] PD RX, header: 0x2a4f [1]
    [ 1206.483145] Rx VDM cmd 0xff018150 type 1 cmd 16 len 2
    [ 1206.483150] AMS STRUCTURED_VDMS finished
    [ 1206.483151] cc:=4
```

```
    [ 1206.505643] AMS STRUCTURED_VDMS start
    [ 1206.505646] PD TX, header: 0x216f
    [ 1206.507933] PD TX complete, status: 0
    [ 1206.512664] PD RX, header: 0x1c4f [1]
    [ 1206.515456] Rx VDM cmd 0xff018151 type 1 cmd 17 len 1
    [ 1206.515460] AMS STRUCTURED_VDMS finished
    [ 1206.515461] cc:=4

    Fixes: e1aefcdd394fd ("usb typec: mt6360: Add support for mt6360 Type-C driver")
    Cc: stable <stable@vger.kernel.org>
    Reported-by: Macpaul Lin <macpaul.lin@mediatek.com>
    Tested-by: Macpaul Lin <macpaul.lin@mediatek.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Acked-by: Heikki Krogerus <heikki.krogerus@linux.intel.com>
    Signed-off-by: ChiYuan Huang <cy_huang@richtek.com>
    Signed-off-by: Fabien Parent <fparent@baylibre.com>
    Link: https://lore.kernel.org/r/1652159580-30959-1-git-send-email-u0084500@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0b448ed552cff7f8310e371fd15cd10f3c3b489a
Author: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
Date:   Mon May 2 10:04:56 2022 +0200

    usb: typec: tcpci: Don't skip cleanup in .remove() on error

    commit bbc126ae381cf0a27822c1f822d0aeed74cc40d9 upstream.

    Returning an error value in an i2c remove callback results in an error
    message being emitted by the i2c core, but otherwise it doesn't make a
    difference. The device goes away anyhow and the devm cleanups are
    called.

    In this case the remove callback even returns early without stopping the
    tcpm worker thread and various timers. A work scheduled on the work
    queue, or a firing timer after tcpci_remove() returned probably results
    in a use-after-free situation because the regmap and driver data were
    freed. So better make sure that tcpci_unregister_port() is called even
    if disabling the irq failed.

    Also emit a more specific error message instead of the i2c core's
    "remove failed (EIO), will be ignored" and return 0 to suppress the
    core's warning.

    This patch is (also) a preparation for making i2c remove callbacks
    return void.

    Fixes: 3ba76256fc4e ("usb: typec: tcpci: mask event interrupts when remove driver")
    Signed-off-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Cc: stable <stable@vger.kernel.org>
    Acked-by: Heikki Krogerus <heikki.krogerus@linux.intel.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20220502080456.21568-1-u.kleine-koenig@pengutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 387c1f181d4702ca096146f719e0d649649bad70
Author: Sergey Ryazanov <ryazanov.s.a@gmail.com>
Date:   Sun May 1 20:58:28 2022 +0300

    usb: cdc-wdm: fix reading stuck on device close

    commit 01e01f5c89773c600a9f0b32c888de0146066c3a upstream.

    cdc-wdm tracks whether a response reading request is in-progress and
    blocks the next request from being sent until the previous request is
    completed. As soon as last user closes the cdc-wdm device file, the
```

```
    driver cancels any ongoing requests, resets the pending response
    counter, but leaves the response reading in-progress flag
    (WDM_RESPONDING) untouched.

    So if the user closes the device file during the response receive
    request is being performed, no more data will be obtained from the
    modem. The request will be cancelled, effectively preventing the
    WDM_RESPONDING flag from being reseted. Keeping the flag set will
    prevent a new response receive request from being sent, permanently
    blocking the read path. The read path will staying blocked until the
    module will be reloaded or till the modem will be re-attached.

    This stuck has been observed with a Huawei E3372 modem attached to an
    OpenWrt router and using the comgt utility to set up a network
    connection.

    Fix this issue by clearing the WDM_RESPONDING flag on the device file
    close.

    Without this fix, the device reading stuck can be easily reproduced in a
    few connection establishing attempts. With this fix, a load test for
    modem connection re-establishing worked for several hours without any
    issues.

    Fixes: 922a5eadd5a3 ("usb: cdc-wdm: Fix race between autosuspend and reading from the
device")
    Signed-off-by: Sergey Ryazanov <ryazanov.s.a@gmail.com>
    Cc: stable <stable@vger.kernel.org>
    Acked-by: Oliver Neukum <oneukum@suse.com>
    Link: https://lore.kernel.org/r/20220501175828.8185-1-ryazanov.s.a@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e10735ce87502b07e171727e574afdd6c0890a08
Author: Dan Vacura <w36195@motorola.com>
Date:   Tue May 3 15:10:38 2022 -0500

    usb: gadget: uvc: allow for application to cleanly shutdown

    commit b81ac4395bbeaf36e078dea1a48c02dd97b76235 upstream.

    Several types of kernel panics can occur due to timing during the uvc
    gadget removal. This appears to be a problem with gadget resources being
    managed by both the client application's v4l2 open/close and the UDC
    gadget bind/unbind. Since the concept of USB_GADGET_DELAYED_STATUS
    doesn't exist for unbind, add a wait to allow for the application to
    close out.

    Some examples of the panics that can occur are:

    <1>[ 1147.652313] Unable to handle kernel NULL pointer dereference at
    virtual address 0000000000000028
    <4>[ 1147.652510] Call trace:
    <4>[ 1147.652514]  usb_gadget_disconnect+0x74/0x1f0
    <4>[ 1147.652516]  usb_gadget_deactivate+0x38/0x168
    <4>[ 1147.652520]  usb_function_deactivate+0x54/0x90
    <4>[ 1147.652524]  uvc_function_disconnect+0x14/0x38
    <4>[ 1147.652527]  uvc_v4l2_release+0x34/0xa0
    <4>[ 1147.652537]  __fput+0xdc/0x2c0
    <4>[ 1147.652540]  ____fput+0x10/0x1c
    <4>[ 1147.652545]  task_work_run+0xe4/0x12c
    <4>[ 1147.652549]  do_notify_resume+0x108/0x168

    <1>[  282.950561][ T1472] Unable to handle kernel NULL pointer
    dereference at virtual address 00000000000005b8
    <6>[  282.953111][ T1472] Call trace:
```

```
      <6>[  282.953121][ T1472]  usb_function_deactivate+0x54/0xd4
      <6>[  282.953134][ T1472]  uvc_v4l2_release+0xac/0x1e4
      <6>[  282.953145][ T1472]  v4l2_release+0x134/0x1f0
      <6>[  282.953167][ T1472]  __fput+0xf4/0x428
      <6>[  282.953178][ T1472]  ____fput+0x14/0x24
      <6>[  282.953193][ T1472]  task_work_run+0xac/0x130

      <3>[  213.410077][   T29] configfs-gadget gadget: uvc: Failed to queue
      request (-108).
      <1>[  213.410116][   T29] Unable to handle kernel NULL pointer
      dereference at virtual address 0000000000000003
      <6>[  213.413460][   T29] Call trace:
      <6>[  213.413474][   T29]  uvcg_video_pump+0x1f0/0x384
      <6>[  213.413489][   T29]  process_one_work+0x2a4/0x544
      <6>[  213.413502][   T29]  worker_thread+0x350/0x784
      <6>[  213.413515][   T29]  kthread+0x2ac/0x320
      <6>[  213.413528][   T29]  ret_from_fork+0x10/0x30

      Signed-off-by: Dan Vacura <w36195@motorola.com>
      Cc: stable <stable@vger.kernel.org>
      Link: https://lore.kernel.org/r/20220503201039.71720-1-w36195@motorola.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit fbbe7fbd4d52897dcad41b1a48aa04e552487134
  Author: Daniel Starke <daniel.starke@siemens.com>
  Date:   Wed May 4 10:17:33 2022 +0200

      tty: n_gsm: fix invalid gsmtty_write_room() result

      commit 9361ebfbb79fd1bc8594a487c01ad52cdaa391ea upstream.

      gsmtty_write() does not prevent the user to use the full fifo size of 4096
      bytes as allocated in gsm_dlci_alloc(). However, gsmtty_write_room() tries
      to limit the return value by 'TX_SIZE' and returns a negative value if the
      fifo has more than 'TX_SIZE' bytes stored. This is obviously wrong as
      'TX_SIZE' is defined as 512.
      Define 'TX_SIZE' to the fifo size and use it accordingly for allocation to
      keep the current behavior. Return the correct remaining size of the fifo in
      gsmtty_write_room() via kfifo_avail().

      Fixes: e1eaea46bb40 ("tty: n_gsm line discipline")
      Cc: stable@vger.kernel.org
      Signed-off-by: Daniel Starke <daniel.starke@siemens.com>
      Link: https://lore.kernel.org/r/20220504081733.3494-3-daniel.starke@siemens.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 6e31ead009b2c7d4fb856b3416f43748fb9b5d12
  Author: Daniel Starke <daniel.starke@siemens.com>
  Date:   Wed May 4 10:17:32 2022 +0200

      tty: n_gsm: fix mux activation issues in gsm_config()

      commit edd5f60c340086891fab094ad61270d6c80f9ca4 upstream.

      The current implementation activates the mux if it was restarted and opens
      the control channel if the mux was previously closed and we are now acting
      as initiator instead of responder, which is the default setting.
      This has two issues.
      1) No mux is activated if we keep all default values and only switch to
      initiator. The control channel is not allocated but will be opened next
      which results in a NULL pointer dereference.
      2) Switching the configuration after it was once configured while keeping
      the initiator value the same will not reopen the control channel if it was
      closed due to parameter incompatibilities. The mux remains dead.
```

```
    Fix 1) by always activating the mux if it is dead after configuration.
    Fix 2) by always opening the control channel after mux activation.

    Fixes: e1eaea46bb40 ("tty: n_gsm line discipline")
    Cc: stable@vger.kernel.org
    Signed-off-by: Daniel Starke <daniel.starke@siemens.com>
    Link: https://lore.kernel.org/r/20220504081733.3494-2-daniel.starke@siemens.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8b1ce42be3a6cde06e07555304ca56b5afcf65b0
Author: Daniel Starke <daniel.starke@siemens.com>
Date:   Wed May 4 10:17:31 2022 +0200

    tty: n_gsm: fix buffer over-read in gsm_dlci_data()

    commit fd442e5ba30aaa75ea47b32149e7a3110dc20a46 upstream.

    'len' is decreased after each octet that has its EA bit set to 0, which
    means that the value is encoded with additional octets. However, the final
    octet does not decreases 'len' which results in 'len' being one byte too
    long. A buffer over-read may occur in tty_insert_flip_string() as it tries
    to read one byte more than the passed content size of 'data'.
    Decrease 'len' also for the final octet which has the EA bit set to 1 to
    write the correct number of bytes from the internal receive buffer to the
    virtual tty.

    Fixes: 2e124b4a390c ("TTY: switch tty_flip_buffer_push")
    Cc: stable@vger.kernel.org
    Signed-off-by: Daniel Starke <daniel.starke@siemens.com>
    Link: https://lore.kernel.org/r/20220504081733.3494-1-daniel.starke@siemens.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4bb92ab559d75a0d04256c9b9e1032582a40a183
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Thu May 5 20:46:21 2022 +0800

    tty/serial: digicolor: fix possible null-ptr-deref in digicolor_uart_probe()

    commit 447ee1516f19f534a228dda237eddb202f23e163 upstream.

    It will cause null-ptr-deref when using 'res', if platform_get_resource()
    returns NULL, so move using 'res' after devm_ioremap_resource() that
    will check it to avoid null-ptr-deref.
    And use devm_platform_get_and_ioremap_resource() to simplify code.

    Fixes: 5930cb3511df ("serial: driver for Conexant Digicolor USART")
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Reviewed-by: Baruch Siach <baruch@tkos.co.il>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20220505124621.1592697-1-yangyingliang@huawei.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 53a48b2728539037fdb2f0fbc1301f7b080f6eb0
Author: Adrian-Ken Rueegsegger <ken@codelabs.ch>
Date:   Mon May 9 11:06:37 2022 +0200

    x86/mm: Fix marking of unused sub-pmd ranges

    commit 280abe14b6e0a38de9cc86fe6a019523aadd8f70 upstream.

    The unused part precedes the new range spanned by the start, end parameters
    of vmemmap_use_new_sub_pmd(). This means it actually goes from
    ALIGN_DOWN(start, PMD_SIZE) up to start.

    Use the correct address when applying the mark using memset.
```

```
    Fixes: 8d400913c231 ("x86/vmemmap: handle unpopulated sub-pmd ranges")
    Signed-off-by: Adrian-Ken Rueegsegger <ken@codelabs.ch>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Reviewed-by: Oscar Salvador <osalvador@suse.de>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20220509090637.24152-2-ken@codelabs.ch
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b130f1e5c9e47310a5b96c3a3fbaddd30735355c
Author: Chunfeng Yun <chunfeng.yun@mediatek.com>
Date:   Thu May 12 14:49:30 2022 +0800

    usb: xhci-mtk: fix fs isoc's transfer error

    commit c237566b78ad8c72bc0431c5d6171db8d12e6f94 upstream.

    Due to the scheduler allocates the optimal bandwidth for FS ISOC endpoints,
    this may be not enough actually and causes data transfer error, so come up
    with an estimate that is no less than the worst case bandwidth used for
    any one mframe, but may be an over-estimate.

    Fixes: 451d3912586a ("usb: xhci-mtk: update fs bus bandwidth by bw_budget_table")
    Cc: stable@vger.kernel.org
    Signed-off-by: Chunfeng Yun <chunfeng.yun@mediatek.com>
    Link: https://lore.kernel.org/r/20220512064931.31670-1-chunfeng.yun@mediatek.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 87bed367311e6d28b072d191dcb78c58e213ec3d
Author: Alexander Graf <graf@amazon.com>
Date:   Tue May 10 14:37:17 2022 +0200

    KVM: PPC: Book3S PR: Enable MSR_DR for switch_mmu_context()

    commit ee8348496c77e3737d0a6cda307a521f2cff954f upstream.

    Commit 863771a28e27 ("powerpc/32s: Convert switch_mmu_context() to C")
    moved the switch_mmu_context() to C. While in principle a good idea, it
    meant that the function now uses the stack. The stack is not accessible
    from real mode though.

    So to keep calling the function, let's turn on MSR_DR while we call it.
    That way, all pointer references to the stack are handled virtually.

    In addition, make sure to save/restore r12 on the stack, as it may get
    clobbered by the C function.

    Fixes: 863771a28e27 ("powerpc/32s: Convert switch_mmu_context() to C")
    Cc: stable@vger.kernel.org # v5.14+
    Reported-by: Matt Evans <matt@ozlabs.org>
    Signed-off-by: Alexander Graf <graf@amazon.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/20220510123717.24508-1-graf@amazon.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 61093e5e790abf01196a844ab00d8f163c0b74d6
Author: Thiébaud Weksteen <tweek@google.com>
Date:   Mon May 2 10:49:52 2022 +1000

    firmware_loader: use kernel credentials when reading firmware

    commit 581dd69830341d299b0c097fc366097ab497d679 upstream.

    Device drivers may decide to not load firmware when probed to avoid
```

```
    slowing down the boot process should the firmware filesystem not be
    available yet. In this case, the firmware loading request may be done
    when a device file associated with the driver is first accessed. The
    credentials of the userspace process accessing the device file may be
    used to validate access to the firmware files requested by the driver.
    Ensure that the kernel assumes the responsibility of reading the
    firmware.

    This was observed on Android for a graphic driver loading their firmware
    when the device file (e.g. /dev/mali0) was first opened by userspace
    (i.e. surfaceflinger). The security context of surfaceflinger was used
    to validate the access to the firmware file (e.g.
    /vendor/firmware/mali.bin).

    Previously, Android configurations were not setting up the
    firmware_class.path command line argument and were relying on the
    userspace fallback mechanism. In this case, the security context of the
    userspace daemon (i.e. ueventd) was consistently used to read firmware
    files. More Android devices are now found to set firmware_class.path
    which gives the kernel the opportunity to read the firmware directly
    (via kernel_read_file_from_path_initns). In this scenario, the current
    process credentials were used, even if unrelated to the loading of the
    firmware file.

    Signed-off-by: Thiébaud Weksteen <tweek@google.com>
    Cc: <stable@vger.kernel.org> # 5.10
    Reviewed-by: Paul Moore <paul@paul-moore.com>
    Acked-by: Luis Chamberlain <mcgrof@kernel.org>
    Link: https://lore.kernel.org/r/20220502004952.3970800-1-tweek@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 79f7efd0a0fb21831a9778ff07276c88b7ca6cea
Author: Jeremy Linton <jeremy.linton@arm.com>
Date:   Thu Apr 28 10:19:47 2022 -0500

    perf tests: Fix coresight `perf test` failure.

    [ Upstream commit 45fa7c38696bae632310c2876ba81fdfa25cc9c2 ]

    Currently the `perf test` always fails the coresight test like:

      89: Check Arm CoreSight trace data recording and synthesized samples: FAILED!

    That is because the test_arm_coresight.sh is attempting to SIGINT the
    parent but is using $$ rather than $PPID and it sigint's itself when
    run under the perf test framework.

    Since this is done in a trap clause it ends up returning a non zero
    return.

    Since $PPID is a bash ism and not all distros are linking /bin/sh to
    bash, the alternative parent pid lookups are uglier than just dropping
    the kill, and its not strictly needed, lets pick the simple solution and
    drop the sigint.

    Fixes: 133fe2e617e48ca0 ("perf tests: Improve temp file cleanup in
test_arm_coresight.sh")
    Reviewed-by: James Clark <james.clark@arm.com>
    Signed-off-by: Jeremy Linton <jeremy.linton@arm.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    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: Leo Yan <leo.yan@linaro.org>
```

```
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Jeremy Linton <jeremy.linton@arm.com>
    Link: https://lore.kernel.org/r/20220428151947.290146-1-jeremy.linton@arm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94c4a7f9eaf0f16934737816a73948a238871af6
Author: Stephen Boyd <swboyd@chromium.org>
Date:   Tue Apr 26 18:32:26 2022 -0700

    interconnect: Restore sync state by ignoring ipa-virt in provider count

    [ Upstream commit 20ce30fb4750f2ffc130cdcb26232b1dd87cd0a5 ]

    Ignore compatible strings for the IPA virt drivers that were removed in
    commits 2fb251c26560 ("interconnect: qcom: sdx55: Drop IP0
    interconnects") and 2f3724930eb4 ("interconnect: qcom: sc7180: Drop IP0
    interconnects") so that the sync state logic can kick in again.
    Otherwise all the interconnects in the system will stay pegged at max
    speeds because 'providers_count' is always going to be one larger than
    the number of drivers that will ever probe on sc7180 or sdx55. This
    fixes suspend on sc7180 and sdx55 devices when you don't have a
    devicetree patch to remove the ipa-virt compatible node.

    Cc: Bjorn Andersson <bjorn.andersson@linaro.org>
    Cc: Doug Anderson <dianders@chromium.org>
    Cc: Alex Elder <elder@linaro.org>
    Cc: Taniya Das <quic_tdas@quicinc.com>
    Cc: Mike Tipton <quic_mdtipton@quicinc.com>
    Fixes: 2fb251c26560 ("interconnect: qcom: sdx55: Drop IP0 interconnects")
    Fixes: 2f3724930eb4 ("interconnect: qcom: sc7180: Drop IP0 interconnects")
    Signed-off-by: Stephen Boyd <swboyd@chromium.org>
    Reviewed-by: Alex Elder <elder@linaro.org>
    Reviewed-by: Douglas Anderson <dianders@chromium.org>
    Link: https://lore.kernel.org/r/20220427013226.341209-1-swboyd@chromium.org
    Signed-off-by: Georgi Djakov <djakov@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 819864279cd030f00cc7ccc5b2d2fd413d29c3a4
Author: Matthew Wilcox (Oracle) <willy@infradead.org>
Date:   Tue May 3 00:09:31 2022 -0400

    block: Do not call folio_next() on an unreferenced folio

    [ Upstream commit 170f37d6aa6ad4582eefd7459015de79e244536e ]

    It is unsafe to call folio_next() on a folio unless you hold a reference
    on it that prevents it from being split or freed.  After returning
    from the iterator, iomap calls folio_end_writeback() which may drop
    the last reference to the page, or allow the page to be split.  If that
    happens, the iterator will not advance far enough through the bio_vec,
    leading to assertion failures like the BUG() in folio_end_writeback()
    that checks we're not trying to end writeback on a page not currently
    under writeback.  Other assertion failures were also seen, but they're
    all explained by this one bug.

    Fix the bug by remembering where the next folio starts before returning
    from the iterator.  There are other ways of fixing this bug, but this
    seems the simplest.

    Reported-by: Darrick J. Wong <djwong@kernel.org>
    Tested-by: Darrick J. Wong <djwong@kernel.org>
    Reported-by: Brian Foster <bfoster@redhat.com>
```

```
    Tested-by: Brian Foster <bfoster@redhat.com>
    Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5034cbb361e1c447911a15b1d3982d5df7aa17b9
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:14 2022 +0200

    tcp: drop the hash_32() part from the index calculation

    [ Upstream commit e8161345ddbb66e449abde10d2fdce93f867eba9 ]

    In commit 190cc82489f4 ("tcp: change source port randomizarion at
    connect() time"), the table_perturb[] array was introduced and an
    index was taken from the port_offset via hash_32(). But it turns
    out that hash_32() performs a multiplication while the input here
    comes from the output of SipHash in secure_seq, that is well
    distributed enough to avoid the need for yet another hash.

    Suggested-by: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e3ee7bb47d6509c3e8a3e96e5d8e3bf21549b6e8
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:13 2022 +0200

    tcp: increase source port perturb table to 2^16

    [ Upstream commit 4c2c8f03a5ab7cb04ec64724d7d176d00bcc91e5 ]

    Moshe Kol, Amit Klein, and Yossi Gilad reported being able to accurately
    identify a client by forcing it to emit only 40 times more connections
    than there are entries in the table_perturb[] table. The previous two
    improvements consisting in resalting the secret every 10s and adding
    randomness to each port selection only slightly improved the situation,
    and the current value of 2^8 was too small as it's not very difficult
    to make a client emit 10k connections in less than 10 seconds.

    Thus we're increasing the perturb table from 2^8 to 2^16 so that the
    same precision now requires 2.6M connections, which is more difficult in
    this time frame and harder to hide as a background activity. The impact
    is that the table now uses 256 kB instead of 1 kB, which could mostly
    affect devices making frequent outgoing connections. However such
    components usually target a small set of destinations (load balancers,
    database clients, perf assessment tools), and in practice only a few
    entries will be visited, like before.

    A live test at 1 million connections per second showed no performance
    difference from the previous value.

    Reported-by: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Reported-by: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Reported-by: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 01e16c23823a057667feb5cf26ba0c963fef6afd
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:12 2022 +0200
```

```
    tcp: dynamically allocate the perturb table used by source ports

    [ Upstream commit e9261476184be1abd486c9434164b2acbe0ed6c2 ]

    We'll need to further increase the size of this table and it's likely
    that at some point its size will not be suitable anymore for a static
    table. Let's allocate it on boot from inet_hashinfo2_init(), which is
    called from tcp_init().

    Cc: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Cc: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Cc: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2cef1db8f8aa81330fee4538a1158e1f6fd5bd1
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:11 2022 +0200

    tcp: add small random increments to the source port

    [ Upstream commit ca7af0402550f9a0b3316d5f1c30904e42ed257d ]

    Here we're randomly adding between 0 and 7 random increments to the
    selected source port in order to add some noise in the source port
    selection that will make the next port less predictable.

    With the default port range of 32768-60999 this means a worst case
    reuse scenario of 14116/8=1764 connections between two consecutive
    uses of the same port, with an average of 14116/4.5=3137. This code
    was stressed at more than 800000 connections per second to a fixed
    target with all connections closed by the client using RSTs (worst
    condition) and only 2 connections failed among 13 billion, despite
    the hash being reseeded every 10 seconds, indicating a perfectly
    safe situation.

    Cc: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Cc: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Cc: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a8081f81323e1550c241157244318db166b660e
Author: Eric Dumazet <edumazet@google.com>
Date:   Mon May 2 10:46:10 2022 +0200

    tcp: resalt the secret every 10 seconds

    [ Upstream commit 4dfa9b438ee34caca4e6a4e5e961641807367f6f ]

    In order to limit the ability for an observer to recognize the source
    ports sequence used to contact a set of destinations, we should
    periodically shuffle the secret. 10 seconds looks effective enough
    without causing particular issues.

    Cc: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Cc: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Cc: Amit Klein <aksecurity@gmail.com>
    Cc: Jason A. Donenfeld <Jason@zx2c4.com>
    Tested-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Eric Dumazet <edumazet@google.com>
```

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

commit 27003fa8b581098aa9768bc03f82d5654368cb02
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:09 2022 +0200

    tcp: use different parts of the port_offset for index and offset

    [ Upstream commit 9e9b70ae923baf2b5e8a0ea4fd0c8451801ac526 ]

    Amit Klein suggests that we use different parts of port_offset for the
    table's index and the port offset so that there is no direct relation
    between them.

    Cc: Jason A. Donenfeld <Jason@zx2c4.com>
    Cc: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Cc: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Cc: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6976724355f5fdada89de528730f9a7b4928f2e3
Author: Willy Tarreau <w@1wt.eu>
Date:   Mon May 2 10:46:08 2022 +0200

    secure_seq: use the 64 bits of the siphash for port offset calculation

    [ Upstream commit b2d057560b8107c633b39aabe517ff9d93f285e3 ]

    SipHash replaced MD5 in secure_ipv{4,6}_port_ephemeral() via commit
    7cd23e5300c1 ("secure_seq: use SipHash in place of MD5"), but the output
    remained truncated to 32-bit only. In order to exploit more bits from the
    hash, let's make the functions return the full 64-bit of siphash_3u32().
    We also make sure the port offset calculation in __inet_hash_connect()
    remains done on 32-bit to avoid the need for div_u64_rem() and an extra
    cost on 32-bit systems.

    Cc: Jason A. Donenfeld <Jason@zx2c4.com>
    Cc: Moshe Kol <moshe.kol@mail.huji.ac.il>
    Cc: Yossi Gilad <yossi.gilad@mail.huji.ac.il>
    Cc: Amit Klein <aksecurity@gmail.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Willy Tarreau <w@1wt.eu>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 181010c92c062f50ccdd0521e04fee8983e85468
Author: Matthew Hagan <mnhagan88@gmail.com>
Date:   Mon May 2 23:33:15 2022 +0100

    net: sfp: Add tx-fault workaround for Huawei MA5671A SFP ONT

    [ Upstream commit 2069624dac19d62c558bb6468fe03678553ab01d ]

    As noted elsewhere, various GPON SFP modules exhibit non-standard
    TX-fault behaviour. In the tested case, the Huawei MA5671A, when used
    in combination with a Marvell mv88e6085 switch, was found to
    persistently assert TX-fault, resulting in the module being disabled.

    This patch adds a quirk to ignore the SFP_F_TX_FAULT state, allowing the
    module to function.
```

```
    Change from v1: removal of erroneous return statment (Andrew Lunn)

    Signed-off-by: Matthew Hagan <mnhagan88@gmail.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20220502223315.1973376-1-mnhagan88@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 859f7fc86a6b2e8343d4e2bf53a27593ce5cd40b
Author: Shravya Kumbham <shravya.kumbham@xilinx.com>
Date:   Mon May 2 12:57:49 2022 +0530

    net: emaclite: Don't advertise 1000BASE-T and do auto negotiation

    [ Upstream commit b800528b97d0adc3a5ba42d78a8b0d3f07a31f44 ]

    In xemaclite_open() function we are setting the max speed of
    emaclite to 100Mb using phy_set_max_speed() function so,
    there is no need to write the advertising registers to stop
    giga-bit speed and the phy_start() function starts the
    auto-negotiation so, there is no need to handle it separately
    using advertising registers. Remove the phy_read and phy_write
    of advertising registers in xemaclite_open() function.

    Signed-off-by: Shravya Kumbham <shravya.kumbham@xilinx.com>
    Signed-off-by: Radhey Shyam Pandey <radhey.shyam.pandey@xilinx.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cef27a7d36026bd3c86f4fdfb4611ebbe2814af4
Author: Jens Axboe <axboe@kernel.dk>
Date:   Sun May 1 21:19:50 2022 -0600

    io_uring: assign non-fixed early for async work

    [ Upstream commit a196c78b5443fc61af2c0490213b9d125482cbd1 ]

    We defer file assignment to ensure that fixed files work with links
    between a direct accept/open and the links that follow it. But this has
    the side effect that normal file assignment is then not complete by the
    time that request submission has been done.

    For deferred execution, if the file is a regular file, assign it when
    we do the async prep anyway.

    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a1c27ea040e47cbe9bc03b703196a2b506c75905
Author: Ajit Kumar Pandey <AjitKumar.Pandey@amd.com>
Date:   Tue Apr 26 13:33:57 2022 -0500

    ASoC: SOF: Fix NULL pointer exception in sof_pci_probe callback

    [ Upstream commit c61711c1c95791850be48dd65a1d72eb34ba719f ]

    We are accessing "desc->ops" in sof_pci_probe without checking "desc"
    pointer. This results in NULL pointer exception if pci_id->driver_data
    i.e desc pointer isn't defined in sof device probe:

    BUG: kernel NULL pointer dereference, address: 0000000000000060
    PGD 0 P4D 0
    Oops: 0000 [#1] PREEMPT SMP NOPTI
    RIP: 0010:sof_pci_probe+0x1e/0x17f [snd_sof_pci]
```

```
    Code: Unable to access opcode bytes at RIP 0xffffffffc043dff4.
    RSP: 0018:ffffac4b03b9b8d8 EFLAGS: 00010246

    Add NULL pointer check for sof_dev_desc pointer to avoid such exception.

    Reviewed-by: Ranjani Sridharan <ranjani.sridharan@linux.intel.com>
    Signed-off-by: Ajit Kumar Pandey <AjitKumar.Pandey@amd.com>
    Signed-off-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Link: https://lore.kernel.org/r/20220426183357.102155-1-pierre-
louis.bossart@linux.intel.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9013a73042fca3ff22f822632f784cee44ebe63f
Author: Sven Schnelle <svens@linux.ibm.com>
Date:   Mon Apr 25 14:17:42 2022 +0200

    s390: disable -Warray-bounds

    [ Upstream commit 8b202ee218395319aec1ef44f72043e1fbaccdd6 ]

    gcc-12 shows a lot of array bound warnings on s390. This is caused
    by the S390_lowcore macro which uses a hardcoded address of 0.

    Wrapping that with absolute_pointer() works, but gcc no longer knows
    that a 12 bit displacement is sufficient to access lowcore. So it
    emits instructions like 'lghi %r1,0; l %rx,xxx(%r1)' instead of a
    single load/store instruction. As s390 stores variables often
    read/written in lowcore, this is considered problematic. Therefore
    disable -Warray-bounds on s390 for gcc-12 for the time being, until
    there is a better solution.

    Signed-off-by: Sven Schnelle <svens@linux.ibm.com>
    Link: https://lore.kernel.org/r/yt9dzgkelelc.fsf@linux.ibm.com
    Link: https://lore.kernel.org/r/20220422134308.1613610-1-svens@linux.ibm.com
    Link: https://lore.kernel.org/r/20220425121742.3222133-1-svens@linux.ibm.com
    Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e8e477ce0974c1463d7af29f358b27bee2ef8f5f
Author: Mark Brown <broonie@kernel.org>
Date:   Sat Apr 23 14:12:39 2022 +0100

    ASoC: ops: Validate input values in snd_soc_put_volsw_range()

    [ Upstream commit aa22125c57f9e577f0a667e4fa07fc3fa8ca1e60 ]

    Check that values written via snd_soc_put_volsw_range() are
    within the range advertised by the control, ensuring that we
    don't write out of spec values to the hardware.

    Signed-off-by: Mark Brown <broonie@kernel.org>
    Link: https://lore.kernel.org/r/20220423131239.3375261-1-broonie@kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0a0897169f7db53f67132986a569acb82b91eab7
Author: Mark Brown <broonie@kernel.org>
Date:   Wed Apr 20 20:34:54 2022 +0100

    ASoC: max98090: Generate notifications on changes for custom control

    [ Upstream commit 13fcf676d9e102594effc686d98521ff5c90b925 ]

    The max98090 driver has some custom controls which share a put() function
```

```
    which returns 0 unconditionally, meaning that events are not generated
    when the value changes. Fix that.

    Signed-off-by: Mark Brown <broonie@kernel.org>
    Link: https://lore.kernel.org/r/20220420193454.2647908-2-broonie@kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d87c00a9b00fc29566ce6852d976217101bc8d7f
Author: Mark Brown <broonie@kernel.org>
Date:   Wed Apr 20 20:34:53 2022 +0100

    ASoC: max98090: Reject invalid values in custom control put()

    [ Upstream commit 2fbe467bcbfc760a08f08475eea6bbd4c2874319 ]

    The max98090 driver has a custom put function for some controls which can
    only be updated in certain circumstances which makes no effort to validate
    that input is suitable for the control, allowing out of spec values to be
    written to the hardware and presented to userspace. Fix this by returning
    an error when invalid values are written.

    Signed-off-by: Mark Brown <broonie@kernel.org>
    Link: https://lore.kernel.org/r/20220420193454.2647908-1-broonie@kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit de650e604a99cf24f8f85e0817de0ed406361124
Author: Ashish Mhetre <amhetre@nvidia.com>
Date:   Thu Apr 21 13:45:04 2022 +0530

    iommu: arm-smmu: disable large page mappings for Nvidia arm-smmu

    [ Upstream commit 4a25f2ea0e030b2fc852c4059a50181bfc5b2f57 ]

    Tegra194 and Tegra234 SoCs have the erratum that causes walk cache
    entries to not be invalidated correctly. The problem is that the walk
    cache index generated for IOVA is not same across translation and
    invalidation requests. This is leading to page faults when PMD entry is
    released during unmap and populated with new PTE table during subsequent
    map request. Disabling large page mappings avoids the release of PMD
    entry and avoid translations seeing stale PMD entry in walk cache.
    Fix this by limiting the page mappings to PAGE_SIZE for Tegra194 and
    Tegra234 devices. This is recommended fix from Tegra hardware design
    team.

    Acked-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Krishna Reddy <vdumpa@nvidia.com>
    Co-developed-by: Pritesh Raithatha <praithatha@nvidia.com>
    Signed-off-by: Pritesh Raithatha <praithatha@nvidia.com>
    Signed-off-by: Ashish Mhetre <amhetre@nvidia.com>
    Link: https://lore.kernel.org/r/20220421081504.24678-1-amhetre@nvidia.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8b8df503be22cdd2cee12fc9e52bc7b8b88cbe0c
Author: Duoming Zhou <duoming@zju.edu.cn>
Date:   Mon Apr 18 23:33:22 2022 +0800

    RDMA/irdma: Fix deadlock in irdma_cleanup_cm_core()

    [ Upstream commit 679ab61bf5f5f519377d812afb4fb93634782c74 ]

    There is a deadlock in irdma_cleanup_cm_core(), which is shown below:
```

```
        (Thread 1)              |      (Thread 2)
                                | irdma_schedule_cm_timer()
    irdma_cleanup_cm_core()     |  add_timer()
     spin_lock_irqsave() //(1) |  (wait a time)
     ...                        | irdma_cm_timer_tick()
     del_timer_sync()           |  spin_lock_irqsave() //(2)
     (wait timer to stop)       |  ...
```

    We hold cm_core->ht_lock in position (1) of thread 1 and use
    del_timer_sync() to wait timer to stop, but timer handler also need
    cm_core->ht_lock in position (2) of thread 2. As a result,
    irdma_cleanup_cm_core() will block forever.

    This patch removes the check of timer_pending() in
    irdma_cleanup_cm_core(), because the del_timer_sync() function will just
    return directly if there isn't a pending timer. As a result, the lock is
    redundant, because there is no resource it could protect.

    Link: https://lore.kernel.org/r/20220418153322.42524-1-duoming@zju.edu.cn
    Signed-off-by: Duoming Zhou <duoming@zju.edu.cn>
    Reviewed-by: Shiraz Saleem <shiraz.saleem@intel.com>
    Signed-off-by: Jason Gunthorpe <jgg@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a98c5ef03de51a7cad20c401cf12719806b0084
Author: Ji-Ze Hong (Peter Hong) <hpeter@gmail.com>
Date: Mon Apr 18 17:07:06 2022 +0800

    hwmon: (f71882fg) Fix negative temperature

    [Upstream commit 4aaaaf0f279836f06d3b9d0ffeec7a1e1a04ceef]

    All temperature of Fintek superio hwmonitor that using 1-byte reg will use
    2's complement.

    In show_temp()
            temp = data->temp[nr] * 1000;

    When data->temp[nr] read as 255, it indicate -1C, but this code will report
    255C to userspace. It'll be ok when change to:
            temp = ((s8)data->temp[nr]) * 1000;

    Signed-off-by: Ji-Ze Hong (Peter Hong) <hpeter+linux_kernel@gmail.com>
    Link: https://lore.kernel.org/r/20220418090706.6339-1-hpeter+linux_kernel@gmail.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4c1e6cdd676de6c397ad24fef330e91993890ed4
Author: Denis Pauk <pauk.denis@gmail.com>
Date: Sun Apr 3 22:34:54 2022 +0300

    hwmon: (asus_wmi_sensors) Fix CROSSHAIR VI HERO name

    [Upstream commit 4fd45cc8568e6086272d3036f2c29d61e9b776a1]

    CROSSHAIR VI HERO motherboard is incorrectly named as
    ROG CROSSHAIR VI HERO.

    Signed-off-by: Denis Pauk <pauk.denis@gmail.com>
    Link: https://lore.kernel.org/r/20220403193455.1363-1-pauk.denis@gmail.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8eeeb89cc49ff5c09ab0eed55a94385349ea9b08
Author: Andreas Gruenbacher <agruenba@redhat.com>

```
Date:   Thu Apr 14 17:52:39 2022 +0200

    gfs2: Fix filesystem block deallocation for short writes

    [ Upstream commit d031a8866e709c9d1ee5537a321b6192b4d2dc5b ]

    When a write cannot be carried out in full, gfs2_iomap_end() releases
    blocks that have been allocated for this write but haven't been used.

    To compute the end of the allocation, gfs2_iomap_end() incorrectly
    rounded the end of the attempted write down to the next block boundary
    to arrive at the end of the allocation.  It would have to round up, but
    the end of the allocation is also available as iomap->offset +
    iomap->length, so just use that instead.

    In addition, use round_up() for computing the start of the unused range.

    Fixes: 64bc06bb32ee ("gfs2: iomap buffered write support")
    Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4d962c47f2c495da2c60141acacb74388c0487af
Author: Zack Rusin <zackr@vmware.com>
Date:   Wed Mar 2 10:24:22 2022 -0500

    drm/vmwgfx: Fix fencing on SVGAv3

    [ Upstream commit 1d6595b4cd47acfd824550f48f10b54a6f0e93ee ]

    Port of the vmwgfx to SVGAv3 lacked support for fencing. SVGAv3 removed
    FIFO's and replaced them with command buffers and extra registers.
    The initial version of SVGAv3 lacked support for most advanced features
    (e.g. 3D) which made fences unnecessary. That is no longer the case,
    especially as 3D support is being turned on.

    Switch from FIFO commands and capabilities to command buffers and extra
    registers to enable fences on SVGAv3.

    Fixes: 2cd80dbd3551 ("drm/vmwgfx: Add basic support for SVGA3")
    Signed-off-by: Zack Rusin <zackr@vmware.com>
    Reviewed-by: Martin Krastev <krastevm@vmware.com>
    Reviewed-by: Maaz Mombasawala <mombasawalam@vmware.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220302152426.885214-5-zack@kde.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d553181718b1627221f65b48bc3e8577151c4229
Author: Maxim Mikityanskiy <maximmi@nvidia.com>
Date:   Thu May 12 12:18:30 2022 +0300

    tls: Fix context leak on tls_device_down

    [ Upstream commit 3740651bf7e200109dd42d5b2fb22226b26f960a ]

    The commit cited below claims to fix a use-after-free condition after
    tls_device_down. Apparently, the description wasn't fully accurate. The
    context stayed alive, but ctx->netdev became NULL, and the offload was
    torn down without a proper fallback, so a bug was present, but a
    different kind of bug.

    Due to misunderstanding of the issue, the original patch dropped the
    refcount_dec_and_test line for the context to avoid the alleged
    premature deallocation. That line has to be restored, because it matches
    the refcount_inc_not_zero from the same function, otherwise the contexts
    that survived tls_device_down are leaked.
```

```
    This patch fixes the described issue by restoring refcount_dec_and_test.
    After this change, there is no leak anymore, and the fallback to
    software kTLS still works.

    Fixes: c55dcdd435aa ("net/tls: Fix use-after-free after the TLS device goes down and
up")
    Signed-off-by: Maxim Mikityanskiy <maximmi@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Link: https://lore.kernel.org/r/20220512091830.678684-1-maximmi@nvidia.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b02a6c368da69aead20b2f2bc8645a17bbe4983f
Author: Taehee Yoo <ap420073@gmail.com>
Date:   Thu May 12 05:47:09 2022 +0000

    net: sfc: ef10: fix memory leak in efx_ef10_mtd_probe()

    [ Upstream commit 1fa89ffbc04545b7582518e57f4b63e2a062870f ]

    In the NIC ->probe() callback, ->mtd_probe() callback is called.
    If NIC has 2 ports, ->probe() is called twice and ->mtd_probe() too.
    In the ->mtd_probe(), which is efx_ef10_mtd_probe() it allocates and
    initializes mtd partiion.
    But mtd partition for sfc is shared data.
    So that allocated mtd partition data from last called
    efx_ef10_mtd_probe() will not be used.
    Therefore it must be freed.
    But it doesn't free a not used mtd partition data in efx_ef10_mtd_probe().

    kmemleak reports:
    unreferenced object 0xffff88811ddb0000 (size 63168):
      comm "systemd-udevd", pid 265, jiffies 4294681048 (age 348.586s)
      hex dump (first 32 bytes):
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
      backtrace:
        [<ffffffffa3767749>] kmalloc_order_trace+0x19/0x120
        [<ffffffffa3873f0e>] __kmalloc+0x20e/0x250
        [<ffffffffc041389f>] efx_ef10_mtd_probe+0x11f/0x270 [sfc]
        [<ffffffffc0484c8a>] efx_pci_probe.cold.17+0x3df/0x53d [sfc]
        [<ffffffffa414192c>] local_pci_probe+0xdc/0x170
        [<ffffffffa4145df5>] pci_device_probe+0x235/0x680
        [<ffffffffa443dd52>] really_probe+0x1c2/0x8f0
        [<ffffffffa443e72b>] __driver_probe_device+0x2ab/0x460
        [<ffffffffa443e92a>] driver_probe_device+0x4a/0x120
        [<ffffffffa443f2ae>] __driver_attach+0x16e/0x320
        [<ffffffffa4437a90>] bus_for_each_dev+0x110/0x190
        [<ffffffffa443b75e>] bus_add_driver+0x39e/0x560
        [<ffffffffa4440b1e>] driver_register+0x18e/0x310
        [<ffffffffc02e2055>] 0xffffffffc02e2055
        [<ffffffffa3001af3>] do_one_initcall+0xc3/0x450
        [<ffffffffa33ca574>] do_init_module+0x1b4/0x700

    Acked-by: Martin Habets <habetsm.xilinx@gmail.com>
    Fixes: 8127d661e77f ("sfc: Add support for Solarflare SFC9100 family")
    Signed-off-by: Taehee Yoo <ap420073@gmail.com>
    Link: https://lore.kernel.org/r/20220512054709.12513-1-ap420073@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7b42621ce5514adb7570ce70bb0acef5c8396608
Author: Guangguan Wang <guangguan.wang@linux.alibaba.com>
Date:   Thu May 12 11:08:20 2022 +0800
```

```
    net/smc: non blocking recvmsg() return -EAGAIN when no data and signal_pending

    [ Upstream commit f3c46e41b32b6266cf60b0985c61748f53bf1c61 ]

    Non blocking sendmsg will return -EAGAIN when any signal pending
    and no send space left, while non blocking recvmsg return -EINTR
    when signal pending and no data received. This may makes confused.
    As TCP returns -EAGAIN in the conditions described above. Align the
    behavior of smc with TCP.

    Fixes: 846e344eb722 ("net/smc: add receive timeout check")
    Signed-off-by: Guangguan Wang <guangguan.wang@linux.alibaba.com>
    Reviewed-by: Tony Lu <tonylu@linux.alibaba.com>
    Acked-by: Karsten Graul <kgraul@linux.ibm.com>
    Link: https://lore.kernel.org/r/20220512030820.73848-1-guangguan.wang@linux.alibaba.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5fe928809f838a2b6a91f1c743be6559aca64612
Author: Florian Fainelli <f.fainelli@gmail.com>
Date:   Wed May 11 19:17:31 2022 -0700

    net: dsa: bcm_sf2: Fix Wake-on-LAN with mac_link_down()

    [ Upstream commit b7be130c5d52e5224ac7d89568737b37b4c4b785 ]

    After commit 2d1f90f9ba83 ("net: dsa/bcm_sf2: fix incorrect usage of
    state->link") the interface suspend path would call our mac_link_down()
    call back which would forcibly set the link down, thus preventing
    Wake-on-LAN packets from reaching our management port.

    Fix this by looking at whether the port is enabled for Wake-on-LAN and
    not clearing the link status in that case to let packets go through.

    Fixes: 2d1f90f9ba83 ("net: dsa/bcm_sf2: fix incorrect usage of state->link")
    Signed-off-by: Florian Fainelli <f.fainelli@gmail.com>
    Link: https://lore.kernel.org/r/20220512021731.2494261-1-f.fainelli@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2d5140ddf2e00d4b8ce30231e182ac3371f44c32
Author: Amit Cohen <amcohen@nvidia.com>
Date:   Wed May 11 14:57:47 2022 +0300

    mlxsw: Avoid warning during ip6gre device removal

    [ Upstream commit 810c2f0a3f86158c1e02e74947b66d811473434a ]

    IPv6 addresses which are used for tunnels are stored in a hash table
    with reference counting. When a new GRE tunnel is configured, the driver
    is notified and configures it in hardware.

    Currently, any change in the tunnel is not applied in the driver. It
    means that if the remote address is changed, the driver is not aware of
    this change and the first address will be used.

    This behavior results in a warning [1] in scenarios such as the
    following:

     # ip link add name gre1 type ip6gre local 2000::3 remote 2000::fffe tos inherit ttl
inherit
     # ip link set name gre1 type ip6gre local 2000::3 remote 2000::ffff ttl inherit
     # ip link delete gre1

    The change of the address is not applied in the driver. Currently, the
```

```
    driver uses the remote address which is stored in the 'parms' of the
    overlay device. When the tunnel is removed, the new IPv6 address is
    used, the driver tries to release it, but as it is not aware of the
    change, this address is not configured and it warns about releasing non
    existing IPv6 address.

    Fix it by using the IPv6 address which is cached in the IPIP entry, this
    address is the last one that the driver used, so even in cases such the
    above, the first address will be released, without any warning.

    [1]:

    WARNING: CPU: 1 PID: 2197 at drivers/net/ethernet/mellanox/mlxsw/spectrum.c:2920
mlxsw_sp_ipv6_addr_put+0x146/0x220 [mlxsw_spectrum]
    ...
    CPU: 1 PID: 2197 Comm: ip Not tainted 5.17.0-rc8-custom-95062-gc1e5ded51a9a #84
    Hardware name: Mellanox Technologies Ltd. MSN4700/VMOD0010, BIOS 5.11 07/12/2021
    RIP: 0010:mlxsw_sp_ipv6_addr_put+0x146/0x220 [mlxsw_spectrum]
    ...
    Call Trace:
     <TASK>
     mlxsw_sp2_ipip_rem_addr_unset_gre6+0xf1/0x120 [mlxsw_spectrum]
     mlxsw_sp_netdevice_ipip_ol_event+0xdb/0x640 [mlxsw_spectrum]
     mlxsw_sp_netdevice_event+0xc4/0x850 [mlxsw_spectrum]
     raw_notifier_call_chain+0x3c/0x50
     call_netdevice_notifiers_info+0x2f/0x80
     unregister_netdevice_many+0x311/0x6d0
     rtnl_dellink+0x136/0x360
     rtnetlink_rcv_msg+0x12f/0x380
     netlink_rcv_skb+0x49/0xf0
     netlink_unicast+0x233/0x340
     netlink_sendmsg+0x202/0x440
     ____sys_sendmsg+0x1f3/0x220
     ___sys_sendmsg+0x70/0xb0
     __sys_sendmsg+0x54/0xa0
     do_syscall_64+0x35/0x80
     entry_SYSCALL_64_after_hwframe+0x44/0xae

    Fixes: e846efe2737b ("mlxsw: spectrum: Add hash table for IPv6 address mapping")
    Reported-by: Maksym Yaremchuk <maksymy@nvidia.com>
    Signed-off-by: Amit Cohen <amcohen@nvidia.com>
    Signed-off-by: Ido Schimmel <idosch@nvidia.com>
    Link: https://lore.kernel.org/r/20220511115747.238602-1-idosch@nvidia.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 87b810e1790c3ac5cc4934f301ba9a4baa31009e
Author: Hui Tang <tanghui20@huawei.com>
Date:   Tue May 10 21:51:48 2022 +0800

    drm/vc4: hdmi: Fix build error for implicit function declaration

    [ Upstream commit 6fed53de560768bde6d701a7c79c253b45b259e3 ]

    drivers/gpu/drm/vc4/vc4_hdmi.c: In function ‘vc4_hdmi_connector_detect’:
    drivers/gpu/drm/vc4/vc4_hdmi.c:228:7: error: implicit declaration of function
‘gpiod_get_value_cansleep’; did you mean ‘gpio_get_value_cansleep’? [-Werror=implicit-
function-declaration]
       if (gpiod_get_value_cansleep(vc4_hdmi->hpd_gpio))
           ^~~~~~~~~~~~~~~~~~~~~~~~
           gpio_get_value_cansleep
      CC [M]  drivers/gpu/drm/vc4/vc4_validate.o
      CC [M]  drivers/gpu/drm/vc4/vc4_v3d.o
      CC [M]  drivers/gpu/drm/vc4/vc4_validate_shaders.o
      CC [M]  drivers/gpu/drm/vc4/vc4_debugfs.o
```

```
    drivers/gpu/drm/vc4/vc4_hdmi.c: In function ‘vc4_hdmi_bind’:
    drivers/gpu/drm/vc4/vc4_hdmi.c:2883:23: error: implicit declaration of function
 ‘devm_gpiod_get_optional’; did you mean ‘devm_clk_get_optional’? [-Werror=implicit-function-
 declaration]
      vc4_hdmi->hpd_gpio = devm_gpiod_get_optional(dev, "hpd", GPIOD_IN);
                           ^~~~~~~~~~~~~~~~~~~~~~
                           devm_clk_get_optional
    drivers/gpu/drm/vc4/vc4_hdmi.c:2883:59: error: ‘GPIOD_IN’ undeclared (first use in this
 function); did you mean ‘GPIOF_IN’?
      vc4_hdmi->hpd_gpio = devm_gpiod_get_optional(dev, "hpd", GPIOD_IN);
                                                               ^~~~~~~~
                                                               GPIOF_IN
    drivers/gpu/drm/vc4/vc4_hdmi.c:2883:59: note: each undeclared identifier is reported
 only once for each function it appears in
    cc1: all warnings being treated as errors

    Fixes: 6800234ceee0 ("drm/vc4: hdmi: Convert to gpiod")
    Signed-off-by: Hui Tang <tanghui20@huawei.com>
    Signed-off-by: Maxime Ripard <maxime@cerno.tech>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220510135148.247719-1-
 tanghui20@huawei.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 8a92ca503e8cbd83730631f23b0b01faf3e8ca53
 Author: Florian Fainelli <f.fainelli@gmail.com>
 Date:   Tue May 10 20:17:51 2022 -0700

    net: bcmgenet: Check for Wake-on-LAN interrupt probe deferral

    [ Upstream commit 6b77c06655b8a749c1a3d9ebc51e9717003f7e5a ]

    The interrupt controller supplying the Wake-on-LAN interrupt line maybe
    modular on some platforms (irq-bcm7038-l1.c) and might be probed at a
    later time than the GENET driver. We need to specifically check for
    -EPROBE_DEFER and propagate that error to ensure that we eventually
    fetch the interrupt descriptor.

    Fixes: 9deb48b53e7f ("bcmgenet: add WOL IRQ check")
    Fixes: 5b1f0e62941b ("net: bcmgenet: Avoid touching non-existent interrupt")
    Signed-off-by: Florian Fainelli <f.fainelli@gmail.com>
    Reviewed-by: Stefan Wahren <stefan.wahren@i2se.com>
    Link: https://lore.kernel.org/r/20220511031752.2245566-1-f.fainelli@gmail.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit f5da26f89fc984063feaeb4bacd30fc074859c77
 Author: Yang Yingliang <yangyingliang@huawei.com>
 Date:   Wed May 11 11:08:29 2022 +0800

    net: ethernet: mediatek: ppe: fix wrong size passed to memset()

    [ Upstream commit 00832b1d1a393dfb1b9491d085e5b27e8c25d103 ]

    'foe_table' is a pointer, the real size of struct mtk_foe_entry
    should be pass to memset().

    Fixes: ba37b7caf1ed ("net: ethernet: mtk_eth_soc: add support for initializing the PPE")
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Acked-by: Felix Fietkau <nbd@nbd.name>
    Link: https://lore.kernel.org/r/20220511030829.3308094-1-yangyingliang@huawei.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit c19cc520b3d69904e9518d401ad0df7f4702aca0
 Author: Paolo Abeni <pabeni@redhat.com>
```

```
Date:   Tue May 10 16:57:34 2022 +0200

    net/sched: act_pedit: really ensure the skb is writable

    [ Upstream commit 8b796475fd7882663a870456466a4fb315cc1bd6 ]

    Currently pedit tries to ensure that the accessed skb offset
    is writable via skb_unclone(). The action potentially allows
    touching any skb bytes, so it may end-up modifying shared data.

    The above causes some sporadic MPTCP self-test failures, due to
    this code:

            tc -n $ns2 filter add dev ns2eth$i egress \
                    protocol ip prio 1000 \
                    handle 42 fw \
                    action pedit munge offset 148 u8 invert \
                    pipe csum tcp \
                    index 100

    The above modifies a data byte outside the skb head and the skb is
    a cloned one, carrying a TCP output packet.

    This change addresses the issue by keeping track of a rough
    over-estimate highest skb offset accessed by the action and ensuring
    such offset is really writable.

    Note that this may cause performance regressions in some scenarios,
    but hopefully pedit is not in the critical path.

    Fixes: db2c24175d14 ("act_pedit: access skb->data safely")
    Acked-by: Mat Martineau <mathew.j.martineau@linux.intel.com>
    Tested-by: Geliang Tang <geliang.tang@suse.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Acked-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Link:
https://lore.kernel.org/r/1fcf78e6679d0a287dd61bb0f04730ce33b3255d.1652194627.git.pabeni@red
hat.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 796e9b31fdb144b58bf293e743a04fbbc2406fdd
Author: Alexandra Winter <wintera@linux.ibm.com>
Date:   Tue May 10 09:05:08 2022 +0200

    s390/lcs: fix variable dereferenced before check

    [ Upstream commit 671bb35c8e746439f0ed70815968f9a4f20a8deb ]

    smatch complains about
    drivers/s390/net/lcs.c:1741 lcs_get_control() warn: variable dereferenced before check
'card->dev' (see line 1739)

    Fixes: 27eb5ac8f015 ("[PATCH] s390: lcs driver bug fixes and improvements [1/2]")
    Signed-off-by: Alexandra Winter <wintera@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea0053af5dab4d63a9c44563973fb2f3bfd9eb2b
Author: Alexandra Winter <wintera@linux.ibm.com>
Date:   Tue May 10 09:05:07 2022 +0200

    s390/ctcm: fix potential memory leak

    [ Upstream commit 0c0b20587b9f25a2ad14db7f80ebe49bdf29920a ]
```

```
    smatch complains about
    drivers/s390/net/ctcm_mpc.c:1210 ctcmpc_unpack_skb() warn: possible memory leak of
'mpcginfo'

    mpc_action_discontact() did not free mpcginfo. Consolidate the freeing in
    ctcmpc_unpack_skb().

    Fixes: 293d984f0e36 ("ctcm: infrastructure for replaced ctc driver")
    Signed-off-by: Alexandra Winter <wintera@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0104f830ddbb8bc93ebb45ea69568bfcc5336efe
Author: Alexandra Winter <wintera@linux.ibm.com>
Date:   Tue May 10 09:05:06 2022 +0200

    s390/ctcm: fix variable dereferenced before check

    [ Upstream commit 2c50c6867c85afee6f2b3bcbc50fc9d0083d1343 ]

    Found by cppcheck and smatch.
    smatch complains about
    drivers/s390/net/ctcm_sysfs.c:43 ctcm_buffer_write() warn: variable dereferenced before
check 'priv' (see line 42)

    Fixes: 3c09e2647b5e ("ctcm: rename READ/WRITE defines to avoid redefinitions")
    Reported-by: Colin Ian King <colin.i.king@gmail.com>
    Signed-off-by: Alexandra Winter <wintera@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3c91d1007feee23b768b42e02d7fa7fff9e8d99b
Author: Shunsuke Mie <mie@igel.co.jp>
Date:   Tue May 10 19:27:23 2022 +0900

    virtio: fix virtio transitional ids

    [ Upstream commit 7ff960a6fe399fdcbca6159063684671ae57eee9 ]

    This commit fixes the transitional PCI device ID.

    Fixes: d61914ea6ada ("virtio: update virtio id table, add transitional ids")
    Signed-off-by: Shunsuke Mie <mie@igel.co.jp>
    Link: https://lore.kernel.org/r/20220510102723.87666-1-mie@igel.co.jp
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit beb869607209404af4297f0b19723e725fb24587
Author: Joey Gouly <joey.gouly@arm.com>
Date:   Tue May 10 11:27:21 2022 +0100

    arm64: vdso: fix makefile dependency on vdso.so

    [ Upstream commit 205f3991a273cac6008ef4db3d1c0dc54d14fb56 ]

    There is currently no dependency for vdso*-wrap.S on vdso*.so, which means that
    you can get a build that uses a stale vdso*-wrap.o.

    In commit a5b8ca97fbf8, the file that includes the vdso.so was moved and renamed
    from arch/arm64/kernel/vdso/vdso.S to arch/arm64/kernel/vdso-wrap.S, when this
    happened the Makefile was not updated to force the dependcy on vdso.so.

    Fixes: a5b8ca97fbf8 ("arm64: do not descend to vdso directories twice")
    Signed-off-by: Joey Gouly <joey.gouly@arm.com>
```

```
    Cc: Masahiro Yamada <masahiroy@kernel.org>
    Cc: Vincenzo Frascino <vincenzo.frascino@arm.com>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Will Deacon <will@kernel.org>
    Link: https://lore.kernel.org/r/20220510102721.50811-1-joey.gouly@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d44f5ffa36ef190930a1f8a62be76508db1d1f2c
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Sat May 7 16:45:50 2022 +0300

    net: dsa: flush switchdev workqueue on bridge join error path

    [ Upstream commit 630fd4822af2374cd75c682b7665dcb367613765 ]

    There is a race between switchdev_bridge_port_offload() and the
    dsa_port_switchdev_sync_attrs() call right below it.

    When switchdev_bridge_port_offload() finishes, FDB entries have been
    replayed by the bridge, but are scheduled for deferred execution later.

    However dsa_port_switchdev_sync_attrs -> dsa_port_can_apply_vlan_filtering()
    may impose restrictions on the vlan_filtering attribute and refuse
    offloading.

    When this happens, the delayed FDB entries will dereference dp->bridge,
    which is a NULL pointer because we have stopped the process of
    offloading this bridge.

    Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000
    Workqueue: dsa_ordered dsa_slave_switchdev_event_work
    pc : dsa_port_bridge_host_fdb_del+0x64/0x100
    lr : dsa_slave_switchdev_event_work+0x130/0x1bc
    Call trace:
     dsa_port_bridge_host_fdb_del+0x64/0x100
     dsa_slave_switchdev_event_work+0x130/0x1bc
     process_one_work+0x294/0x670
     worker_thread+0x80/0x460
    ---[ end trace 0000000000000000 ]---
    Error: dsa_core: Must first remove VLAN uppers having VIDs also present in bridge.

    Fix the bug by doing what we do on the normal bridge leave path as well,
    which is to wait until the deferred FDB entries complete executing, then
    exit.

    The placement of dsa_flush_workqueue() after switchdev_bridge_port_unoffload()
    guarantees that both the FDB additions and deletions on rollback are waited for.

    Fixes: d7d0d423dbaa ("net: dsa: flush switchdev workqueue when leaving the bridge")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Link: https://lore.kernel.org/r/20220507134550.1849834-1-vladimir.oltean@nxp.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 854b878885e3c4bd7411d85c3d5cfe7f727bf8fb
Author: Joel Savitz <jsavitz@redhat.com>
Date:   Mon May 9 17:34:29 2022 -0700

    selftests: vm: Makefile: rename TARGETS to VMTARGETS

    [ Upstream commit 41c240099fe09377b6b9f8272e45d2267c843d3e ]

    The tools/testing/selftests/vm/Makefile uses the variable TARGETS
    internally to generate a list of platform-specific binary build targets
```

```
    suffixed with _{32,64}.  When building the selftests using its own
    Makefile directly, such as via the following command run in a kernel tree:

    One receives an error such as the following:

    make: Entering directory '/root/linux/tools/testing/selftests'
    make --no-builtin-rules ARCH=x86 -C ../../.. headers_install
    make[1]: Entering directory '/root/linux'
      INSTALL ./usr/include
    make[1]: Leaving directory '/root/linux'
    make[1]: Entering directory '/root/linux/tools/testing/selftests/vm'
    make[1]: *** No rule to make target 'vm.c', needed by
'/root/linux/tools/testing/selftests/vm/vm_64'.  Stop.
    make[1]: Leaving directory '/root/linux/tools/testing/selftests/vm'
    make: *** [Makefile:175: all] Error 2
    make: Leaving directory '/root/linux/tools/testing/selftests'

    The TARGETS variable passed to tools/testing/selftests/Makefile collides
    with the TARGETS used in tools/testing/selftests/vm/Makefile, so rename
    the latter to VMTARGETS, eliminating the collision with no functional
    change.

    Link: https://lkml.kernel.org/r/20220504213454.1282532-1-jsavitz@redhat.com
    Fixes: f21fda8f6453 ("selftests: vm: pkeys: fix multilib builds for x86")
    Signed-off-by: Joel Savitz <jsavitz@redhat.com>
    Acked-by: Nico Pache <npache@redhat.com>
    Cc: Joel Savitz <jsavitz@redhat.com>
    Cc: Shuah Khan <shuah@kernel.org>
    Cc: Sandipan Das <sandipan@linux.ibm.com>
    Cc: Dave Hansen <dave.hansen@intel.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bce0fac239b06297000d0a65dba9a6f1238a881a
Author: Kalesh Singh <kaleshsingh@google.com>
Date:   Mon May 9 17:34:28 2022 -0700

    procfs: prevent unprivileged processes accessing fdinfo dir

    [ Upstream commit 1927e498aee1757b3df755a194cbfc5cc0f2b663 ]

    The file permissions on the fdinfo dir from were changed from
    S_IRUSR|S_IXUSR to S_IRUGO|S_IXUGO, and a PTRACE_MODE_READ check was added
    for opening the fdinfo files [1].  However, the ptrace permission check
    was not added to the directory, allowing anyone to get the open FD numbers
    by reading the fdinfo directory.

    Add the missing ptrace permission check for opening the fdinfo directory.

    [1] https://lkml.kernel.org/r/20210308170651.919148-1-kaleshsingh@google.com

    Link: https://lkml.kernel.org/r/20210713162008.1056986-1-kaleshsingh@google.com
    Fixes: 7bc3fa0172a4 ("procfs: allow reading fdinfo with PTRACE_MODE_READ")
    Signed-off-by: Kalesh Singh <kaleshsingh@google.com>
    Cc: Kees Cook <keescook@chromium.org>
    Cc: Eric W. Biederman <ebiederm@xmission.com>
    Cc: Christian Brauner <christian.brauner@ubuntu.com>
    Cc: Suren Baghdasaryan <surenb@google.com>
    Cc: Hridya Valsaraju <hridya@google.com>
    Cc: Jann Horn <jannh@google.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ca869e57be2ad91345da3a217f0e608fce0d3e2f
Author: Randy Dunlap <rdunlap@infradead.org>
```

```
Date:   Mon May 9 16:47:40 2022 -0700

    hwmon: (ltq-cputemp) restrict it to SOC_XWAY

    [ Upstream commit 151d6dcbed836270c6c240932da66f147950cbdb ]

    Building with SENSORS_LTQ_CPUTEMP=y with SOC_FALCON=y causes build
    errors since FALCON does not support the same features as XWAY.

    Change this symbol to depend on SOC_XWAY since that provides the
    necessary interfaces.

    Repairs these build errors:

    ../drivers/hwmon/ltq-cputemp.c: In function 'ltq_cputemp_enable':
    ../drivers/hwmon/ltq-cputemp.c:23:9: error: implicit declaration of function
'ltq_cgu_w32'; did you mean 'ltq_ebu_w32'? [-Werror=implicit-function-declaration]
       23 |         ltq_cgu_w32(ltq_cgu_r32(CGU_GPHY1_CR) | CGU_TEMP_PD, CGU_GPHY1_CR);
    ../drivers/hwmon/ltq-cputemp.c:23:21: error: implicit declaration of function
'ltq_cgu_r32'; did you mean 'ltq_ebu_r32'? [-Werror=implicit-function-declaration]
       23 |         ltq_cgu_w32(ltq_cgu_r32(CGU_GPHY1_CR) | CGU_TEMP_PD, CGU_GPHY1_CR);
    ../drivers/hwmon/ltq-cputemp.c: In function 'ltq_cputemp_probe':
    ../drivers/hwmon/ltq-cputemp.c:92:31: error: 'SOC_TYPE_VR9_2' undeclared (first use in
this function)
       92 |         if (ltq_soc_type() != SOC_TYPE_VR9_2)

    Fixes: 7074d0a92758 ("hwmon: (ltq-cputemp) add cpu temp sensor driver")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Reported-by: kernel test robot <lkp@intel.com>
    Cc: Florian Eckert <fe@dev.tdt.de>
    Cc: Guenter Roeck <linux@roeck-us.net>
    Cc: Jean Delvare <jdelvare@suse.com>
    Cc: linux-hwmon@vger.kernel.org
    Link: https://lore.kernel.org/r/20220509234740.26841-1-rdunlap@infradead.org
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a872f3bed07930fd7b10550c441c7b7f83749bb5
Author: Jesse Brandeburg <jesse.brandeburg@intel.com>
Date:   Fri May 6 18:10:38 2022 -0700

    dim: initialize all struct fields

    [ Upstream commit ee1444b5e1df4155b591d0d9b1e72853a99ea861 ]

    The W=2 build pointed out that the code wasn't initializing all the
    variables in the dim_cq_moder declarations with the struct initializers.
    The net change here is zero since these structs were already static
    const globals and were initialized with zeros by the compiler, but
    removing compiler warnings has value in and of itself.

    lib/dim/net_dim.c: At top level:
    lib/dim/net_dim.c:54:9: warning: missing initializer for field ‘comps’ of ‘const struct
dim_cq_moder’ [-Wmissing-field-initializers]
       54 |         NET_DIM_RX_EQE_PROFILES,
          |         ^~~~~~~~~~~~~~~~~~~~~~~
    In file included from lib/dim/net_dim.c:6:
    ./include/linux/dim.h:45:13: note: ‘comps’ declared here
       45 |         u16 comps;
          |             ^~~~~

    and repeats for the tx struct, and once you fix the comps entry then
    the cq_period_mode field needs the same treatment.

    Use the commonly accepted style to indicate to the compiler that we
```

```
    know what we're doing, and add a comma at the end of each struct
    initializer to clean up the issue, and use explicit initializers
    for the fields we are initializing which makes the compiler happy.

    While here and fixing these lines, clean up the code slightly with
    a fix for the super long lines by removing the word "_MODERATION" from a
    couple defines only used in this file.

    Fixes: f8be17b81d44 ("lib/dim: Fix -Wunused-const-variable warnings")
    Signed-off-by: Jesse Brandeburg <jesse.brandeburg@intel.com>
    Link: https://lore.kernel.org/r/20220507011038.14568-1-jesse.brandeburg@intel.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a49d70d8e8a012e48384677747536c89f7d8e194
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Fri May 6 11:40:40 2022 +0800

    ionic: fix missing pci_release_regions() on error in ionic_probe()

    [ Upstream commit e4b1045bf9cfec6f70ac6d3783be06c3a88dcb25 ]

    If ionic_map_bars() fails, pci_release_regions() need be called.

    Fixes: fbfb8031533c ("ionic: Add hardware init and device commands")
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Link: https://lore.kernel.org/r/20220506034040.2614129-1-yangyingliang@huawei.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3f1a4b3fc3c5da16e6de47346c4d160d1190892f
Author: Dan Aloni <dan.aloni@vastdata.com>
Date:   Sun May 8 15:54:50 2022 +0300

    nfs: fix broken handling of the softreval mount option

    [ Upstream commit 085d16d5f949b64713d5e960d6c9bbf51bc1d511 ]

    Turns out that ever since this mount option was added, passing
    `softreval` in NFS mount options cancelled all other flags while not
    affecting the underlying flag `NFS_MOUNT_SOFTREVAL`.

    Fixes: c74dfe97c104 ("NFS: Add mount option 'softreval'")
    Signed-off-by: Dan Aloni <dan.aloni@vastdata.com>
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f0ce2d072c8ed0ea5934e31e782c6679541264ed
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Thu May 5 23:04:22 2022 +0200

    mac80211_hwsim: call ieee80211_tx_prepare_skb under RCU protection

    [ Upstream commit 9e2db50f1ef2238fc2f71c5de1c0418b7a5b0ea2 ]

    This is needed since it might use (and pass out) pointers to
    e.g. keys protected by RCU. Can't really happen here as the
    frames aren't encrypted, but we need to still adhere to the
    rules.

    Fixes: cacfddf82baf ("mac80211_hwsim: initialize ieee80211_tx_info at hw_scan_work")
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Link:
https://lore.kernel.org/r/20220505230421.5f139f9de173.I77ae111a28f7c0e9fd1ebcee7f39dbec5c606
770@changeid
```

```
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4a21979cfe2d71b80ca2111d8b3bc63d739f9391
Author: Amir Goldstein <amir73il@gmail.com>
Date:   Sat May 7 11:00:28 2022 +0300

    fanotify: do not allow setting dirent events in mask of non-dir

    [ Upstream commit ceaf69f8eadcafb323392be88e7a5248c415d423 ]

    Dirent events (create/delete/move) are only reported on watched
    directory inodes, but in fanotify as well as in legacy inotify, it was
    always allowed to set them on non-dir inode, which does not result in
    any meaningful outcome.

    Until kernel v5.17, dirent events in fanotify also differed from events
    "on child" (e.g. FAN_OPEN) in the information provided in the event.
    For example, FAN_OPEN could be set in the mask of a non-dir or the mask
    of its parent and event would report the fid of the child regardless of
    the marked object.
    By contrast, FAN_DELETE is not reported if the child is marked and the
    child fid was not reported in the events.

    Since kernel v5.17, with fanotify group flag FAN_REPORT_TARGET_FID, the
    fid of the child is reported with dirent events, like events "on child",
    which may create confusion for users expecting the same behavior as
    events "on child" when setting events in the mask on a child.

    The desired semantics of setting dirent events in the mask of a child
    are not clear, so for now, deny this action for a group initialized
    with flag FAN_REPORT_TARGET_FID and for the new event FAN_RENAME.
    We may relax this restriction in the future if we decide on the
    semantics and implement them.

    Fixes: d61fd650e9d2 ("fanotify: introduce group flag FAN_REPORT_TARGET_FID")
    Fixes: 8cc3b1ccd930 ("fanotify: wire up FAN_RENAME event")
    Link: https://lore.kernel.org/linux-fsdevel/20220505133057.zm5t6vumc4xdcnsg@quack3.lan/
    Signed-off-by: Amir Goldstein <amir73il@gmail.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20220507080028.219826-1-amir73il@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b73548c9ef237ea0f77fe133784388469757193d
Author: Taehee Yoo <ap420073@gmail.com>
Date:   Wed May 4 12:32:27 2022 +0000

    net: sfc: fix memory leak due to ptp channel

    [ Upstream commit 49e6123c65dac6393b04f39ceabf79c44f66b8be ]

    It fixes memory leak in ring buffer change logic.

    When ring buffer size is changed(ethtool -G eth0 rx 4096), sfc driver
    works like below.
    1. stop all channels and remove ring buffers.
    2. allocates new buffer array.
    3. allocates rx buffers.
    4. start channels.

    While the above steps are working, it skips some steps if the channel
    doesn't have a ->copy callback function.
    Due to ptp channel doesn't have ->copy callback, these above steps are
    skipped for ptp channel.
    It eventually makes some problems.
```

```
     a. ptp channel's ring buffer size is not changed, it works only
        1024(default).
     b. memory leak.

     The reason for memory leak is to use the wrong ring buffer values.
     There are some values, which is related to ring buffer size.
     a. efx->rxq_entries
      - This is global value of rx queue size.
     b. rx_queue->ptr_mask
      - used for access ring buffer as circular ring.
      - roundup_pow_of_two(efx->rxq_entries) - 1
     c. rx_queue->max_fill
      - efx->rxq_entries - EFX_RXD_HEAD_ROOM

     These all values should be based on ring buffer size consistently.
     But ptp channel's values are not.
     a. efx->rxq_entries
      - This is global(for sfc) value, always new ring buffer size.
     b. rx_queue->ptr_mask
      - This is always 1023(default).
     c. rx_queue->max_fill
      - This is new ring buffer size - EFX_RXD_HEAD_ROOM.

     Let's assume we set 4096 for rx ring buffer,

                            normal channel     ptp channel
     efx->rxq_entries       4096               4096
     rx_queue->ptr_mask     4095               1023
     rx_queue->max_fill     4086               4086

     sfc driver allocates rx ring buffers based on these values.
     When it allocates ptp channel's ring buffer, 4086 ring buffers are
     allocated then, these buffers are attached to the allocated array.
     But ptp channel's ring buffer array size is still 1024(default)
     and ptr_mask is still 1023 too.
     So, 3062 ring buffers will be overwritten to the array.
     This is the reason for memory leak.

     Test commands:
        ethtool -G <interface name> rx 4096
        while :
        do
            ip link set <interface name> up
            ip link set <interface name> down
        done

     In order to avoid this problem, it adds ->copy callback to ptp channel
     type.
     So that rx_queue->ptr_mask value will be updated correctly.

     Fixes: 7c236c43b838 ("sfc: Add support for IEEE-1588 PTP")
     Signed-off-by: Taehee Yoo <ap420073@gmail.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 4a66d823194f258ea145c3b2c01d565e1ce5d0ae
  Author: Javier Martinez Canillas <javierm@redhat.com>
  Date:   Fri May 6 15:22:25 2022 +0200

     fbdev: efifb: Fix a use-after-free due early fb_info cleanup

     [ Upstream commit 1b5853dfab7fdde450f00f145327342238135c8a ]

     Commit d258d00fb9c7 ("fbdev: efifb: Cleanup fb_info in .fb_destroy rather
     than .remove") attempted to fix a use-after-free error due driver freeing
```

```
    the fb_info in the .remove handler instead of doing it in .fb_destroy.

    But ironically that change introduced yet another use-after-free since the
    fb_info was still used after the free.

    This should fix for good by freeing the fb_info at the end of the handler.

    Fixes: d258d00fb9c7 ("fbdev: efifb: Cleanup fb_info in .fb_destroy rather than .remove")
    Reported-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Reported-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Signed-off-by: Javier Martinez Canillas <javierm@redhat.com>
    Reviewed-by: Andi Shyti <andi.shyti@linux.intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Reviewed-by: Thomas Zimmermann <tzimemrmann@suse.de>
    Signed-off-by: Lucas De Marchi <lucas.demarchi@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220506132225.588379-1-
javierm@redhat.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b524200f5fe1fd07952a522e75b0f52f5a2da405
Author: Kees Cook <keescook@chromium.org>
Date:   Thu May 5 16:31:01 2022 -0700

    net: chelsio: cxgb4: Avoid potential negative array offset

    [ Upstream commit 1c7ab9cd98b78bef1657a5db7204d8d437e24c94 ]

    Using min_t(int, ...) as a potential array index implies to the compiler
    that negative offsets should be allowed. This is not the case, though.
    Replace "int" with "unsigned int". Fixes the following warning exposed
    under future CONFIG_FORTIFY_SOURCE improvements:

    In file included from include/linux/string.h:253,
                     from include/linux/bitmap.h:11,
                     from include/linux/cpumask.h:12,
                     from include/linux/smp.h:13,
                     from include/linux/lockdep.h:14,
                     from include/linux/rcupdate.h:29,
                     from include/linux/rculist.h:11,
                     from include/linux/pid.h:5,
                     from include/linux/sched.h:14,
                     from include/linux/delay.h:23,
                     from drivers/net/ethernet/chelsio/cxgb4/t4_hw.c:35:
    drivers/net/ethernet/chelsio/cxgb4/t4_hw.c: In function 't4_get_raw_vpd_params':
    include/linux/fortify-string.h:46:33: warning: '__builtin_memcpy' pointer overflow
between offset 29 and size [2147483648, 4294967295] [-Warray-bounds]
       46 | #define __underlying_memcpy     __builtin_memcpy
          |                                 ^
    include/linux/fortify-string.h:388:9: note: in expansion of macro '__underlying_memcpy'
      388 |         __underlying_##op(p, q, __fortify_size);                        \
          |         ^~~~~~~~~~~~~
    include/linux/fortify-string.h:433:26: note: in expansion of macro
'__fortify_memcpy_chk'
      433 | #define memcpy(p, q, s)  __fortify_memcpy_chk(p, q, s,                 \
          |                          ^~~~~~~~~~~~~~~~~~~~
    drivers/net/ethernet/chelsio/cxgb4/t4_hw.c:2796:9: note: in expansion of macro 'memcpy'
     2796 |         memcpy(p->id, vpd + id, min_t(int, id_len, ID_LEN));
          |         ^~~~~~
    include/linux/fortify-string.h:46:33: warning: '__builtin_memcpy' pointer overflow
between offset 0 and size [2147483648, 4294967295] [-Warray-bounds]
       46 | #define __underlying_memcpy     __builtin_memcpy
          |                                 ^
    include/linux/fortify-string.h:388:9: note: in expansion of macro '__underlying_memcpy'
      388 |         __underlying_##op(p, q, __fortify_size);                        \
          |         ^~~~~~~~~~~~~
          |
```

```
    include/linux/fortify-string.h:433:26: note: in expansion of macro
'__fortify_memcpy_chk'
     433 | #define memcpy(p, q, s)  __fortify_memcpy_chk(p, q, s,                 \
         |                          ^~~~~~~~~~~~~~~~~~~~
    drivers/net/ethernet/chelsio/cxgb4/t4_hw.c:2798:9: note: in expansion of macro 'memcpy'
     2798 |         memcpy(p->sn, vpd + sn, min_t(int, sn_len, SERNUM_LEN));
          |         ^~~~~~
```

    Additionally remove needless cast from u8[] to char * in last strim()
    call.

    Reported-by: kernel test robot <lkp@intel.com>
    Link: https://lore.kernel.org/lkml/202205031926.FVP7epJM-lkp@intel.com
    Fixes: fc9279298e3a ("cxgb4: Search VPD with pci_vpd_find_ro_info_keyword()")
    Fixes: 24c521f81c30 ("cxgb4: Use pci_vpd_find_id_string() to find VPD ID string")
    Cc: Raju Rangoju <rajur@chelsio.com>
    Cc: Eric Dumazet <edumazet@google.com>
    Cc: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/20220505233101.1224230-1-keescook@chromium.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 133db730567095ec662b0252572ed5038c11a4f2
Author: Eric Dumazet <edumazet@google.com>
Date: Thu May 5 09:19:46 2022 -0700

    netlink: do not reset transport header in netlink_recvmsg()

    [Upstream commit d5076fe4049cadef1f040eda4aaa001bb5424225]

    netlink_recvmsg() does not need to change transport header.

    If transport header was needed, it should have been reset
    by the producer (netlink_dump()), not the consumer(s).

    The following trace probably happened when multiple threads
    were using MSG_PEEK.

    BUG: KCSAN: data-race in netlink_recvmsg / netlink_recvmsg

    write to 0xffff88811e9f15b2 of 2 bytes by task 32012 on cpu 1:
     skb_reset_transport_header include/linux/skbuff.h:2760 [inline]
     netlink_recvmsg+0x1de/0x790 net/netlink/af_netlink.c:1978
     sock_recvmsg_nosec net/socket.c:948 [inline]
     sock_recvmsg net/socket.c:966 [inline]
     __sys_recvfrom+0x204/0x2c0 net/socket.c:2097
     __do_sys_recvfrom net/socket.c:2115 [inline]
     __se_sys_recvfrom net/socket.c:2111 [inline]
     __x64_sys_recvfrom+0x74/0x90 net/socket.c:2111
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x2b/0x70 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x44/0xae

    write to 0xffff88811e9f15b2 of 2 bytes by task 32005 on cpu 0:
     skb_reset_transport_header include/linux/skbuff.h:2760 [inline]
     netlink_recvmsg+0x1de/0x790 net/netlink/af_netlink.c:1978
     ____sys_recvmsg+0x162/0x2f0
     ___sys_recvmsg net/socket.c:2674 [inline]
     __sys_recvmsg+0x209/0x3f0 net/socket.c:2704
     __do_sys_recvmsg net/socket.c:2714 [inline]
     __se_sys_recvmsg net/socket.c:2711 [inline]
     __x64_sys_recvmsg+0x42/0x50 net/socket.c:2711
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x2b/0x70 arch/x86/entry/common.c:80

```
      entry_SYSCALL_64_after_hwframe+0x44/0xae

    value changed: 0xffff -> 0x0000

    Reported by Kernel Concurrency Sanitizer on:
    CPU: 0 PID: 32005 Comm: syz-executor.4 Not tainted 5.18.0-rc1-syzkaller-00328-
ge1f700ebd6be-dirty #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
01/01/2011

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Link: https://lore.kernel.org/r/20220505161946.2867638-1-eric.dumazet@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8e75186e398e62e7fdd4c82e809ab9c9497d28a
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Wed Feb 9 07:03:11 2022 +0100

    drm/nouveau: Fix a potential theorical leak in nouveau_get_backlight_name()

    [ Upstream commit ab244be47a8f111bc82496a8a20c907236e37f95 ]

    If successful ida_simple_get() calls are not undone when needed, some
    additional memory may be allocated and wasted.

    Here, an ID between 0 and MAX_INT is required. If this ID is >=100, it is
    not taken into account and is wasted. It should be released.

    Instead of calling ida_simple_remove(), take advantage of the 'max'
    parameter to require the ID not to be too big. Should it be too big, it
    is not allocated and don't need to be freed.

    While at it, use ida_alloc_xxx()/ida_free() instead to
    ida_simple_get()/ida_simple_remove().
    The latter is deprecated and more verbose.

    Fixes: db1a0ae21461 ("drm/nouveau/bl: Assign different names to interfaces")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    [Fixed formatting warning from checkpatch]
    Signed-off-by: Lyude Paul <lyude@redhat.com>
    Link:
https://patchwork.freedesktop.org/patch/msgid/9ba85bca59df6813dc029e743a836451d5173221.16443
86541.git.christophe.jaillet@wanadoo.fr
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fa1db3661b5c4958120feafed6a5ad54a9287430
Author: Lokesh Dhoundiyal <lokesh.dhoundiyal@alliedtelesis.co.nz>
Date:   Thu May 5 14:00:17 2022 +1200

    ipv4: drop dst in multicast routing path

    [ Upstream commit 9e6c6d17d1d6a3f1515ce399f9a011629ec79aa0 ]

    kmemleak reports the following when routing multicast traffic over an
    ipsec tunnel.

    Kmemleak output:
    unreferenced object 0x8000000044bebb00 (size 256):
      comm "softirq", pid 0, jiffies 4294985356 (age 126.810s)
      hex dump (first 32 bytes):
        00 00 00 00 00 00 00 00 80 00 00 00 05 13 74 80  ..............t.
```

```
          80 00 00 00 04 9b bf f9 00 00 00 00 00 00 00 00  ................
        backtrace:
          [<00000000f83947e0>] __kmalloc+0x1e8/0x300
          [<00000000b7ed8dca>] metadata_dst_alloc+0x24/0x58
          [<0000000081d32c20>] __ipgre_rcv+0x100/0x2b8
          [<00000000824f6cf1>] gre_rcv+0x178/0x540
          [<00000000ccd4e162>] gre_rcv+0x7c/0xd8
          [<00000000c024b148>] ip_protocol_deliver_rcu+0x124/0x350
          [<000000006a483377>] ip_local_deliver_finish+0x54/0x68
          [<00000000d9271b3a>] ip_local_deliver+0x128/0x168
          [<00000000bd4968ae>] xfrm_trans_reinject+0xb8/0xf8
          [<0000000071672a19>] tasklet_action_common.isra.16+0xc4/0x1b0
          [<0000000062e9c336>] __do_softirq+0x1fc/0x3e0
          [<00000000013d7914>] irq_exit+0xc4/0xe0
          [<00000000a4d73e90>] plat_irq_dispatch+0x7c/0x108
          [<000000000751eb8e>] handle_int+0x16c/0x178
          [<000000001668023b>] _raw_spin_unlock_irqrestore+0x1c/0x28

      The metadata dst is leaked when ip_route_input_mc() updates the dst for
      the skb. Commit f38a9eb1f77b ("dst: Metadata destinations") correctly
      handled dropping the dst in ip_route_input_slow() but missed the
      multicast case which is handled by ip_route_input_mc(). Drop the dst in
      ip_route_input_mc() avoiding the leak.

      Fixes: f38a9eb1f77b ("dst: Metadata destinations")
      Signed-off-by: Lokesh Dhoundiyal <lokesh.dhoundiyal@alliedtelesis.co.nz>
      Signed-off-by: Chris Packham <chris.packham@alliedtelesis.co.nz>
      Reviewed-by: David Ahern <dsahern@kernel.org>
      Link: https://lore.kernel.org/r/20220505020017.3111846-1-
  chris.packham@alliedtelesis.co.nz
      Signed-off-by: Jakub Kicinski <kuba@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 2eb88e7a8c519ae36d6101fc01e77e4eeaa01173
  Author: Michal Michalik <michal.michalik@intel.com>
  Date:   Wed Apr 20 14:23:02 2022 +0200

      ice: fix PTP stale Tx timestamps cleanup

      [ Upstream commit a11b6c1a383ff092f432e040c20e032503785d47 ]

      Read stale PTP Tx timestamps from PHY on cleanup.

      After running out of Tx timestamps request handlers, hardware (HW) stops
      reporting finished requests. Function ice_ptp_tx_tstamp_cleanup() used
      to only clean up stale handlers in driver and was leaving the hardware
      registers not read. Not reading stale PTP Tx timestamps prevents next
      interrupts from arriving and makes timestamping unusable.

      Fixes: ea9b847cda64 ("ice: enable transmit timestamps for E810 devices")
      Signed-off-by: Michal Michalik <michal.michalik@intel.com>
      Reviewed-by: Jacob Keller <jacob.e.keller@intel.com>
      Reviewed-by: Paul Menzel <pmenzel@molgen.mpg.de>
      Tested-by: Gurucharan <gurucharanx.g@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 351afae8b061efbe91be16640fa13f23aca72076
  Author: Anatolii Gerasymenko <anatolii.gerasymenko@intel.com>
  Date:   Thu Apr 28 12:01:00 2022 +0000

      ice: clear stale Tx queue settings before configuring

      [ Upstream commit 6096dae926a22e2892ef9169f582589c16d39639 ]
```

```
    The iAVF driver uses 3 virtchnl op codes to communicate with the PF
    regarding the VF Tx queues:

    * VIRTCHNL_OP_CONFIG_VSI_QUEUES configures the hardware and firmware
    logic for the Tx queues

    * VIRTCHNL_OP_ENABLE_QUEUES configures the queue interrupts

    * VIRTCHNL_OP_DISABLE_QUEUES disables the queue interrupts and Tx rings.

    There is a bug in the iAVF driver due to the race condition between VF
    reset request and shutdown being executed in parallel. This leads to a
    break in logic and VIRTCHNL_OP_DISABLE_QUEUES is not being sent.

    If this occurs, the PF driver never cleans up the Tx queues. This results
    in leaving behind stale Tx queue settings in the hardware and firmware.

    The most obvious outcome is that upon the next
    VIRTCHNL_OP_CONFIG_VSI_QUEUES, the PF will fail to program the Tx
    scheduler node due to a lack of space.

    We need to protect ICE driver against such situation.

    To fix this, make sure we clear existing stale settings out when
    handling VIRTCHNL_OP_CONFIG_VSI_QUEUES. This ensures we remove the
    previous settings.

    Calling ice_vf_vsi_dis_single_txq should be safe as it will do nothing if
    the queue is not configured. The function already handles the case when the
    Tx queue is not currently configured and exits with a 0 return in that
    case.

    Fixes: 7ad15440acf8 ("ice: Refactor VIRTCHNL_OP_CONFIG_VSI_QUEUES handling")
    Signed-off-by: Jacob Keller <jacob.e.keller@intel.com>
    Signed-off-by: Anatolii Gerasymenko <anatolii.gerasymenko@intel.com>
    Tested-by: Konrad Jankowski <konrad0.jankowski@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit addd0d3421d138230b5b761a4060b8f2d86601fb
Author: Ivan Vecera <ivecera@redhat.com>
Date:   Sat Apr 23 12:20:21 2022 +0200

    ice: Fix race during aux device (un)plugging

    [ Upstream commit 486b9eee57ddca5c9a2d59fc41153f36002e0a00 ]

    Function ice_plug_aux_dev() assigns pf->adev field too early prior
    aux device initialization and on other side ice_unplug_aux_dev()
    starts aux device deinit and at the end assigns NULL to pf->adev.
    This is wrong because pf->adev should always be non-NULL only when
    aux device is fully initialized and ready. This wrong order causes
    a crash when ice_send_event_to_aux() call occurs because that function
    depends on non-NULL value of pf->adev and does not assume that
    aux device is half-initialized or half-destroyed.
    After order correction the race window is tiny but it is still there,
    as Leon mentioned and manipulation with pf->adev needs to be protected
    by mutex.

    Fix (un-)plugging functions so pf->adev field is set after aux device
    init and prior aux device destroy and protect pf->adev assignment by
    new mutex. This mutex is also held during ice_send_event_to_aux()
    call to ensure that aux device is valid during that call.
    Note that device lock used ice_send_event_to_aux() needs to be kept
    to avoid race with aux drv unload.
```

```
    Reproducer:
    cycle=1
    while :;do
            echo "#### Cycle: $cycle"

            ip link set ens7f0 mtu 9000
            ip link add bond0 type bond mode 1 miimon 100
            ip link set bond0 up
            ifenslave bond0 ens7f0
            ip link set bond0 mtu 9000
            ethtool -L ens7f0 combined 1
            ip link del bond0
            ip link set ens7f0 mtu 1500
            sleep 1

            let cycle++
    done

    In short when the device is added/removed to/from bond the aux device
    is unplugged/plugged. When MTU of the device is changed an event is
    sent to aux device asynchronously. This can race with (un)plugging
    operation and because pf->adev is set too early (plug) or too late
    (unplug) the function ice_send_event_to_aux() can touch uninitialized
    or destroyed fields. In the case of crash below pf->adev->dev.mutex.

    Crash:
    [   53.372066] bond0: (slave ens7f0): making interface the new active one
    [   53.378622] bond0: (slave ens7f0): Enslaving as an active interface with an u
    p link
    [   53.386294] IPv6: ADDRCONF(NETDEV_CHANGE): bond0: link becomes ready
    [   53.549104] bond0: (slave ens7f1): Enslaving as a backup interface with an up
     link
    [   54.118906] ice 0000:ca:00.0 ens7f0: Number of in use tx queues changed inval
    idating tc mappings. Priority traffic classification disabled!
    [   54.233374] ice 0000:ca:00.1 ens7f1: Number of in use tx queues changed inval
    idating tc mappings. Priority traffic classification disabled!
    [   54.248204] bond0: (slave ens7f0): Releasing backup interface
    [   54.253955] bond0: (slave ens7f1): making interface the new active one
    [   54.274875] bond0: (slave ens7f1): Releasing backup interface
    [   54.289153] bond0 (unregistering): Released all slaves
    [   55.383179] MII link monitoring set to 100 ms
    [   55.398696] bond0: (slave ens7f0): making interface the new active one
    [   55.405241] BUG: kernel NULL pointer dereference, address: 0000000000000080
    [   55.405289] bond0: (slave ens7f0): Enslaving as an active interface with an u
    p link
    [   55.412198] #PF: supervisor write access in kernel mode
    [   55.412200] #PF: error_code(0x0002) - not-present page
    [   55.412201] PGD 25d2ad067 P4D 0
    [   55.412204] Oops: 0002 [#1] PREEMPT SMP NOPTI
    [   55.412207] CPU: 0 PID: 403 Comm: kworker/0:2 Kdump: loaded Tainted: G S
                5.17.0-13579-g57f2d6540f03 #1
    [   55.429094] bond0: (slave ens7f1): Enslaving as a backup interface with an up
     link
    [   55.430224] Hardware name: Dell Inc. PowerEdge R750/06V45N, BIOS 1.4.4 10/07/
    2021
    [   55.430226] Workqueue: ice ice_service_task [ice]
    [   55.468169] RIP: 0010:mutex_unlock+0x10/0x20
    [   55.472439] Code: 0f b1 13 74 96 eb e0 4c 89 ee eb d8 e8 79 54 ff ff 66 0f 1f 84 00
 00 00 00 00 0f 1f 44 00 00 65 48 8b 04 25 40 ef 01 00 31 d2 <f0> 48 0f b1 17 75 01 c3 e9 e3
 fe ff ff 0f 1f 00 0f 1f 44 00 00 48
    [   55.491186] RSP: 0018:ff4454230d7d7e28 EFLAGS: 00010246
    [   55.496413] RAX: ff1a79b208b08000 RBX: ff1a79b2182e8880 RCX: 0000000000000001
    [   55.503545] RDX: 0000000000000000 RSI: ff4454230d7d7db0 RDI: 0000000000000080
    [   55.510678] RBP: ff1a79d1c7e48b68 R08: ff4454230d7d7db0 R09: 0000000000000041
```

```
    [   55.517812] R10: 00000000000000a5 R11: 00000000000006e6 R12: ff1a79d1c7e48bc0
    [   55.524945] R13: 0000000000000000 R14: ff1a79d0ffc305c0 R15: 0000000000000000
    [   55.532076] FS:  0000000000000000(0000) GS:ff1a79d0ffc00000(0000)
knlGS:0000000000000000
    [   55.540163] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [   55.545908] CR2: 0000000000000080 CR3: 00000003487ae003 CR4: 0000000000771ef0
    [   55.553041] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [   55.560173] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [   55.567305] PKRU: 55555554
    [   55.570018] Call Trace:
    [   55.572474]  <TASK>
    [   55.574579]  ice_service_task+0xaab/0xef0 [ice]
    [   55.579130]  process_one_work+0x1c5/0x390
    [   55.583141]  ? process_one_work+0x390/0x390
    [   55.587326]  worker_thread+0x30/0x360
    [   55.590994]  ? process_one_work+0x390/0x390
    [   55.595180]  kthread+0xe6/0x110
    [   55.598325]  ? kthread_complete_and_exit+0x20/0x20
    [   55.603116]  ret_from_fork+0x1f/0x30
    [   55.606698]  </TASK>

    Fixes: f9f5301e7e2d ("ice: Register auxiliary device to provide RDMA")
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Signed-off-by: Ivan Vecera <ivecera@redhat.com>
    Reviewed-by: Dave Ertman <david.m.ertman@intel.com>
    Tested-by: Gurucharan <gurucharanx.g@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 5db0f897ea7cf807f9817a062ee074de5e9f15f1
Author: Maximilian Luz <luzmaximilian@gmail.com>
Date:   Fri Apr 29 21:57:38 2022 +0200

    platform/surface: aggregator: Fix initialization order when compiling as builtin module

    [ Upstream commit 44acfc22c7d055d9c4f8f0974ee28422405b971a ]

    When building the Surface Aggregator Module (SAM) core, registry, and
    other SAM client drivers as builtin modules (=y), proper initialization
    order is not guaranteed. Due to this, client driver registration
    (triggered by device registration in the registry) races against bus
    initialization in the core.

    If any attempt is made at registering the device driver before the bus
    has been initialized (i.e. if bus initialization fails this race) driver
    registration will fail with a message similar to:

        Driver surface_battery was unable to register with bus_type surface_aggregator
because the bus was not initialized

    Switch from module_init() to subsys_initcall() to resolve this issue.
    Note that the serdev subsystem uses postcore_initcall() so we are still
    able to safely register the serdev device driver for the core.

    Fixes: c167b9c7e3d6 ("platform/surface: Add Surface Aggregator subsystem")
    Reported-by: Blaž Hrastnik <blaz@mxxn.io>
    Signed-off-by: Maximilian Luz <luzmaximilian@gmail.com>
    Link: https://lore.kernel.org/r/20220429195738.535751-1-luzmaximilian@gmail.com
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a1aac13288de2935dc1a9330a93b1ac92f1e2b72
Author: Javier Martinez Canillas <javierm@redhat.com>
Date:   Fri May 6 00:06:31 2022 +0200
```

```
    fbdev: vesafb: Cleanup fb_info in .fb_destroy rather than .remove

    [ Upstream commit b3c9a924aab61adbc29df110006aa03afe1a78ba ]

    The driver is calling framebuffer_release() in its .remove callback, but
    this will cause the struct fb_info to be freed too early. Since it could
    be that a reference is still hold to it if user-space opened the fbdev.

    This would lead to a use-after-free error if the framebuffer device was
    unregistered but later a user-space process tries to close the fbdev fd.

    To prevent this, move the framebuffer_release() call to fb_ops.fb_destroy
    instead of doing it in the driver's .remove callback.

    Strictly speaking, the code flow in the driver is still wrong because all
    the hardware cleanupd (i.e: iounmap) should be done in .remove while the
    software cleanup (i.e: releasing the framebuffer) should be done in the
    .fb_destroy handler. But this at least makes to match the behavior before
    commit 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal").

    Fixes: 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal")
    Suggested-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Signed-off-by: Javier Martinez Canillas <javierm@redhat.com>
    Reviewed-by: Thomas Zimmermann <tzimmermann@suse.de>
    Reviewed-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220505220631.366371-1-
javierm@redhat.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 02109faee127f73bb27106394691c452c42a451e
Author: Javier Martinez Canillas <javierm@redhat.com>
Date:   Fri May 6 00:05:40 2022 +0200

    fbdev: efifb: Cleanup fb_info in .fb_destroy rather than .remove

    [ Upstream commit d258d00fb9c7c0cdf9d10c1ded84f10339d2d349 ]

    The driver is calling framebuffer_release() in its .remove callback, but
    this will cause the struct fb_info to be freed too early. Since it could
    be that a reference is still hold to it if user-space opened the fbdev.

    This would lead to a use-after-free error if the framebuffer device was
    unregistered but later a user-space process tries to close the fbdev fd.

    To prevent this, move the framebuffer_release() call to fb_ops.fb_destroy
    instead of doing it in the driver's .remove callback.

    Strictly speaking, the code flow in the driver is still wrong because all
    the hardware cleanupd (i.e: iounmap) should be done in .remove while the
    software cleanup (i.e: releasing the framebuffer) should be done in the
    .fb_destroy handler. But this at least makes to match the behavior before
    commit 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal").

    Fixes: 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal")
    Suggested-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Signed-off-by: Javier Martinez Canillas <javierm@redhat.com>
    Reviewed-by: Thomas Zimmermann <tzimmermann@suse.de>
    Reviewed-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220505220540.366218-1-
javierm@redhat.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8872a31f204b1197003ae501d4d6670eb56352c8
Author: Javier Martinez Canillas <javierm@redhat.com>
```

```
Date:   Fri May 6 00:04:56 2022 +0200

    fbdev: simplefb: Cleanup fb_info in .fb_destroy rather than .remove

    [ Upstream commit 666b90b3ce9e4aac1e1deba266c3a230fb3913b0 ]

    The driver is calling framebuffer_release() in its .remove callback, but
    this will cause the struct fb_info to be freed too early. Since it could
    be that a reference is still hold to it if user-space opened the fbdev.

    This would lead to a use-after-free error if the framebuffer device was
    unregistered but later a user-space process tries to close the fbdev fd.

    To prevent this, move the framebuffer_release() call to fb_ops.fb_destroy
    instead of doing it in the driver's .remove callback.

    Strictly speaking, the code flow in the driver is still wrong because all
    the hardware cleanupd (i.e: iounmap) should be done in .remove while the
    software cleanup (i.e: releasing the framebuffer) should be done in the
    .fb_destroy handler. But this at least makes to match the behavior before
    commit 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal").

    Fixes: 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal")
    Suggested-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Signed-off-by: Javier Martinez Canillas <javierm@redhat.com>
    Reviewed-by: Thomas Zimmermann <tzimmermann@suse.de>
    Reviewed-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220505220456.366090-1-
javierm@redhat.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cd30d7b1b4173a423685a58e9ad19a73b0cf3fbe
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Thu May 5 02:55:03 2022 +0300

    net: mscc: ocelot: avoid corrupting hardware counters when moving VCAP filters

    [ Upstream commit 93a8417088ea570b5721d2b526337a2d3aed9fa3 ]

    Given the following order of operations:

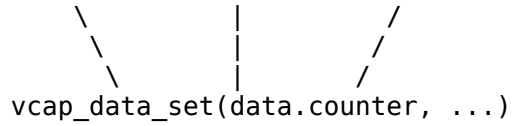
    (1) we add filter A using tc-flower
    (2) we send a packet that matches it
    (3) we read the filter's statistics to find a hit count of 1
    (4) we add a second filter B with a higher preference than A, and A
        moves one position to the right to make room in the TCAM for it
    (5) we send another packet, and this matches the second filter B
    (6) we read the filter statistics again.

    When this happens, the hit count of filter A is 2 and of filter B is 1,
    despite a single packet having matched each filter.

    Furthermore, in an alternate history, reading the filter stats a second
    time between steps (3) and (4) makes the hit count of filter A remain at
    1 after step (6), as expected.

    The reason why this happens has to do with the filter->stats.pkts field,
    which is written to hardware through the call path below:

                   vcap_entry_set
                   /      |      \
                  /       |       \
                 /        |        \
                /         |         \
    es0_entry_set   is1_entry_set   is2_entry_set
```



    The primary role of filter->stats.pkts is to transport the filter hit
    counters from the last readout all the way from vcap_entry_get() ->
    ocelot_vcap_filter_stats_update() -> ocelot_cls_flower_stats().
    The reason why vcap_entry_set() writes it to hardware is so that the
    counters (saturating and having a limited bit width) are cleared
    after each user space readout.

    The writing of filter->stats.pkts to hardware during the TCAM entry
    movement procedure is an unintentional consequence of the code design,
    because the hit count isn't up to date at this point.

    So at step (4), when filter A is moved by ocelot_vcap_filter_add() to
    make room for filter B, the hardware hit count is 0 (no packet matched
    on it in the meantime), but filter->stats.pkts is 1, because the last
    readout saw the earlier packet. The movement procedure programs the old
    hit count back to hardware, so this creates the impression to user space
    that more packets have been matched than they really were.

    The bug can be seen when running the gact_drop_and_ok_test() from the
    tc_actions.sh selftest.

    Fix the issue by reading back the hit count to tmp->stats.pkts before
    migrating the VCAP filter. Sure, this is a best-effort technique, since
    the packets that hit the rule between vcap_entry_get() and
    vcap_entry_set() won't be counted, but at least it allows the counters
    to be reliably used for selftests where the traffic is under control.

    The vcap_entry_get() name is a bit unintuitive, but it only reads back
    the counter portion of the TCAM entry, not the entire entry.

    The index from which we retrieve the counter is also a bit unintuitive
    (i - 1 during add, i + 1 during del), but this is the way in which TCAM
    entry movement works. The "entry index" isn't a stored integer for a
    TCAM filter, instead it is dynamically computed by
    ocelot_vcap_block_get_filter_index() based on the entry's position in
    the &block->rules list. That position (as well as block->count) is
    automatically updated by ocelot_vcap_filter_add_to_block() on add, and
    by ocelot_vcap_block_remove_filter() on del. So "i" is the new filter
    index, and "i - 1" or "i + 1" respectively are the old addresses of that
    TCAM entry (we only support installing/deleting one filter at a time).

    Fixes: b596229448dd ("net: mscc: ocelot: Add support for tcam")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 200004e98b666ba110c4b5672eaad77778486aa2
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date: Thu May 5 02:55:02 2022 +0300

    net: mscc: ocelot: restrict tc-trap actions to VCAP IS2 lookup 0

    [Upstream commit 477d2b91623e682e9a8126ea92acb8f684969cc7]

    Once the CPU port was added to the destination port mask of a packet, it
    can never be cleared, so even packets marked as dropped by the MASK_MODE
    of a VCAP IS2 filter will still reach it. This is why we need the
    OCELOT_POLICER_DISCARD to "kill dropped packets dead" and make software
    stop seeing them.

```
    We disallow policer rules from being put on any other chain than the one
    for the first lookup, but we don't do this for "drop" rules, although we
    should. This change is merely ascertaining that the rules dont't
    (completely) work and letting the user know.

    The blamed commit is the one that introduced the multi-chain architecture
    in ocelot. Prior to that, we should have always offloaded the filters to
    VCAP IS2 lookup 0, where they did work.

    Fixes: 1397a2eb52e2 ("net: mscc: ocelot: create TCAM skeleton from tc filter chains")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c6da1d2aa9e18d89efffd776e48b02161bcf4064
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Thu May 5 02:55:01 2022 +0300

    net: mscc: ocelot: fix VCAP IS2 filters matching on both lookups

    [ Upstream commit 6741e11880003e35802d78cc58035057934f4dab ]

    The VCAP IS2 TCAM is looked up twice per packet, and each filter can be
    configured to only match during the first, second lookup, or both, or
    none.

    The blamed commit wrote the code for making VCAP IS2 filters match only
    on the given lookup. But right below that code, there was another line
    that explicitly made the lookup a "don't care", and this is overwriting
    the lookup we've selected. So the code had no effect.

    Some of the more noticeable effects of having filters match on both
    lookups:

    - in "tc -s filter show dev swp0 ingress", we see each packet matching a
      VCAP IS2 filter counted twice. This throws off scripts such as
      tools/testing/selftests/net/forwarding/tc_actions.sh and makes them
      fail.

    - a "tc-drop" action offloaded to VCAP IS2 needs a policer as well,
      because once the CPU port becomes a member of the destination port
      mask of a packet, nothing removes it, not even a PERMIT/DENY mask mode
      with a port mask of 0. But VCAP IS2 rules with the POLICE_ENA bit in
      the action vector can only appear in the first lookup. What happens
      when a filter matches both lookups is that the action vector is
      combined, and this makes the POLICE_ENA bit ineffective, since the
      last lookup in which it has appeared is the second one. In other
      words, "tc-drop" actions do not drop packets for the CPU port, dropped
      packets are still seen by software unless there was an FDB entry that
      directed those packets to some other place different from the CPU.

    The last bit used to work, because in the initial commit b596229448dd
    ("net: mscc: ocelot: Add support for tcam"), we were writing the FIRST
    field of the VCAP IS2 half key with a 1, not with a "don't care".
    The change to "don't care" was made inadvertently by me in commit
    c1c3993edb7c ("net: mscc: ocelot: generalize existing code for VCAP"),
    which I just realized, and which needs a separate fix from this one,
    for "stable" kernels that lack the commit blamed below.

    Fixes: 226e9cd82a96 ("net: mscc: ocelot: only install TCAM entries into a specific
lookup and PAG")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit ac0878d4d67b2158ccaecf420e9a31fa0270ccc0
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Thu May 5 02:55:00 2022 +0300

    net: mscc: ocelot: fix last VCAP IS1/IS2 filter persisting in hardware when deleted

    [ Upstream commit 16bbebd35629c93a8c68c6d8d28557e100bcee73 ]

    ocelot_vcap_filter_del() works by moving the next filters over the
    current one, and then deleting the last filter by calling vcap_entry_set()
    with a del_filter which was specially created by memsetting its memory
    to zeroes. vcap_entry_set() then programs this to the TCAM and action
    RAM via the cache registers.

    The problem is that vcap_entry_set() is a dispatch function which looks
    at del_filter->block_id. But since del_filter is zeroized memory, the
    block_id is 0, or otherwise said, VCAP_ES0. So practically, what we do
    is delete the entry at the same TCAM index from VCAP ES0 instead of IS1
    or IS2.

    The code was not always like this. vcap_entry_set() used to simply be
    is2_entry_set(), and then, the logic used to work.

    Restore the functionality by populating the block_id of the del_filter
    based on the VCAP block of the filter that we're deleting. This makes
    vcap_entry_set() know what to do.

    Fixes: 1397a2eb52e2 ("net: mscc: ocelot: create TCAM skeleton from tc filter chains")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 231ef3aa5727c836062bee0be3b28c7e6dcf9750
Author: Tariq Toukan <tariqt@nvidia.com>
Date:   Wed May 4 11:09:14 2022 +0300

    net: Fix features skip in for_each_netdev_feature()

    [ Upstream commit 85db6352fc8a158a893151baa1716463d34a20d0 ]

    The find_next_netdev_feature() macro gets the "remaining length",
    not bit index.
    Passing "bit - 1" for the following iteration is wrong as it skips
    the adjacent bit. Pass "bit" instead.

    Fixes: 3b89ea9c5902 ("net: Fix for_each_netdev_feature on Big endian")
    Signed-off-by: Tariq Toukan <tariqt@nvidia.com>
    Reviewed-by: Gal Pressman <gal@nvidia.com>
    Link: https://lore.kernel.org/r/20220504080914.1918-1-tariqt@nvidia.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 27067b27433b221b4d18760bae526e52c1f4dac1
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Thu May 5 10:53:53 2022 +0900

    net: rds: use maybe_get_net() when acquiring refcount on TCP sockets

    [ Upstream commit 6997fbd7a3dafa754f81d541498ace35b43246d8 ]

    Eric Dumazet is reporting addition on 0 problem at rds_tcp_tune(), for
    delayed works queued in rds_wq might be invoked after a net namespace's
    refcount already reached 0.

    Since rds_tcp_exit_net() from cleanup_net() calls flush_workqueue(rds_wq),
```

```
    it is guaranteed that we can instead use maybe_get_net() from delayed work
    functions until rds_tcp_exit_net() returns.

    Note that I'm not convinced that all works which might access a net
    namespace are already queued in rds_wq by the moment rds_tcp_exit_net()
    calls flush_workqueue(rds_wq). If some race is there, rds_tcp_exit_net()
    will fail to wait for work functions, and kmem_cache_free() could be
    called from net_free() before maybe_get_net() is called from
    rds_tcp_tune().

    Reported-by: Eric Dumazet <edumazet@google.com>
    Fixes: 3a58f13a881ed351 ("net: rds: acquire refcount on TCP sockets")
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Link: https://lore.kernel.org/r/41d09faf-bc78-1a87-dfd1-c6d1b5984b61@I-love.SAKURA.ne.jp
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 15d7df5241accfad05e73d9da38f07bcd6efd462
Author: Manikanta Pubbisetty <quic_mpubbise@quicinc.com>
Date:   Thu Apr 28 10:57:44 2022 +0530

    mac80211: Reset MBSSID parameters upon connection

    [ Upstream commit 86af062f40a73bf63321694e6bf637144f0383fe ]

    Currently MBSSID parameters in struct ieee80211_bss_conf
    are not reset upon connection. This could be problematic
    with some drivers in a scenario where the device first
    connects to a non-transmit BSS and then connects to a
    transmit BSS of a Multi BSS AP. The MBSSID parameters
    which are set after connecting to a non-transmit BSS will
    not be reset and the same parameters will be passed on to
    the driver during the subsequent connection to a transmit
    BSS of a Multi BSS AP.

    For example, firmware running on the ath11k device uses the
    Multi BSS data for tracking the beacon of a non-transmit BSS
    and reports the driver when there is a beacon miss. If we do
    not reset the MBSSID parameters during the subsequent
    connection to a transmit BSS, then the driver would have
    wrong MBSSID data and FW would be looking for an incorrect
    BSSID in the MBSSID beacon of a Multi BSS AP and reports
    beacon loss leading to an unstable connection.

    Reset the MBSSID parameters upon every connection to solve this
    problem.

    Fixes: 78ac51f81532 ("mac80211: support multi-bssid")
    Signed-off-by: Manikanta Pubbisetty <quic_mpubbise@quicinc.com>
    Link: https://lore.kernel.org/r/20220428052744.27040-1-quic_mpubbise@quicinc.com
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7d4396e682c7dd6b1f959114dad59df093830ed9
Author: Camel Guo <camel.guo@axis.com>
Date:   Tue May 3 13:43:33 2022 +0200

    hwmon: (tmp401) Add OF device ID table

    [ Upstream commit 3481551f035725fdc46885425eac3ef9b58ae7b7 ]

    This driver doesn't have of_match_table. This makes the kernel module
    tmp401.ko lack alias patterns (e.g: of:N*T*Cti,tmp411) to match DT node
    of the supported devices hence this kernel module will not be
```

```
    automatically loaded.

    After adding of_match_table to this driver, the folllowing alias will be
    added into tmp401.ko.
    $ modinfo drivers/hwmon/tmp401.ko
    filename: drivers/hwmon/tmp401.ko
    ......
    author:         Hans de Goede <hdegoede@redhat.com>
    alias:          of:N*T*Cti,tmp435C*
    alias:          of:N*T*Cti,tmp435
    alias:          of:N*T*Cti,tmp432C*
    alias:          of:N*T*Cti,tmp432
    alias:          of:N*T*Cti,tmp431C*
    alias:          of:N*T*Cti,tmp431
    alias:          of:N*T*Cti,tmp411C*
    alias:          of:N*T*Cti,tmp411
    alias:          of:N*T*Cti,tmp401C*
    alias:          of:N*T*Cti,tmp401
    ......

    Fixes: af503716ac14 ("i2c: core: report OF style module alias for devices registered via
OF")
    Signed-off-by: Camel Guo <camel.guo@axis.com>
    Link: https://lore.kernel.org/r/20220503114333.456476-1-camel.guo@axis.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 18f95c31b085843f93c441b3ba4ab5a74859041e
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Mon Apr 11 08:42:10 2022 -0700

    iwlwifi: iwl-dbg: Use del_timer_sync() before freeing

    [ Upstream commit 7635a1ad8d92dcc8247b53f949e37795154b5b6f ]

    In Chrome OS, a large number of crashes is observed due to corrupted timer
    lists. Steven Rostedt pointed out that this usually happens when a timer
    is freed while still active, and that the problem is often triggered
    by code calling del_timer() instead of del_timer_sync() just before
    freeing.

    Steven also identified the iwlwifi driver as one of the possible culprits
    since it does exactly that.

    Reported-by: Steven Rostedt <rostedt@goodmis.org>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: Johannes Berg <johannes.berg@intel.com>
    Cc: Gregory Greenman <gregory.greenman@intel.com>
    Fixes: 60e8abd9d3e91 ("iwlwifi: dbg_ini: add periodic trigger new API support")
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Acked-by: Gregory Greenman <gregory.greenman@intel.com>
    Tested-by: Sedat Dilek <sedat.dilek@gmail.com> # Linux v5.17.3-rc1 and Debian LLVM-14
    Signed-off-by: Kalle Valo <kvalo@kernel.org>
    Link: https://lore.kernel.org/r/20220411154210.1870008-1-linux@roeck-us.net
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bea410635595f8efbec90c948da04c3613ef87ea
Author: Sven Eckelmann <sven@narfation.org>
Date:   Sat Apr 16 13:51:10 2022 +0200

    batman-adv: Don't skb_split skbuffs with frag_list

    [ Upstream commit a063f2fba3fa633a599253b62561051ac185fa99 ]

    The receiving interface might have used GRO to receive more fragments than
```

```
    MAX_SKB_FRAGS fragments. In this case, these will not be stored in
    skb_shinfo(skb)->frags but merged into the frag list.

    batman-adv relies on the function skb_split to split packets up into
    multiple smaller packets which are not larger than the MTU on the outgoing
    interface. But this function cannot handle frag_list entries and is only
    operating on skb_shinfo(skb)->frags. If it is still trying to split such an
    skb and xmit'ing it on an interface without support for NETIF_F_FRAGLIST,
    then validate_xmit_skb() will try to linearize it. But this fails due to
    inconsistent information. And __pskb_pull_tail will trigger a BUG_ON after
    skb_copy_bits() returns an error.

    In case of entries in frag_list, just linearize the skb before operating on
    it with skb_split().

    Reported-by: Felix Kaechele <felix@kaechele.ca>
    Fixes: c6c8fea29769 ("net: Add batman-adv meshing protocol")
    Signed-off-by: Sven Eckelmann <sven@narfation.org>
    Tested-by: Felix Kaechele <felix@kaechele.ca>
    Signed-off-by: Simon Wunderlich <sw@simonwunderlich.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```