```
commit f4282872cef2e98b3358ea7aa93a03125051fec1
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Sat Mar 11 13:50:50 2023 +0100

    Linux 6.2.5

    Link: https://lore.kernel.org/r/20230310133718.689332661@linuxfoundation.org
    Tested-by: Markus Reichelt <lkt+2023@mareichelt.com>
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Conor Dooley <conor.dooley@microchip.com>
    Tested-by: Sudip Mukherjee <sudip.mukherjee@codethink.co.uk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c3ccb01e0fed3e8afffc34a73d99a52bae0f2329
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Mon Feb 13 15:09:26 2023 +0800

    usb: gadget: uvc: fix missing mutex_unlock() if kstrtou8() fails

    commit 7ebb605d2283fb2647b4fa82030307ce00bee436 upstream.

    If kstrtou8() fails, the mutex_unlock() is missed, move kstrtou8()
    before mutex_lock() to fix it up.

    Fixes: 0525210c9840 ("usb: gadget: uvc: Allow definition of XUs in configfs")
    Fixes: b3c839bd8a07 ("usb: gadget: uvc: Make bSourceID read/write")
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Link: https://lore.kernel.org/r/20230213070926.776447-1-yangyingliang@huawei.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4e8f7d998b582a99aadedd07ae6086e99b89c97a
Author: Pierre Gondois <pierre.gondois@arm.com>
Date:   Wed Feb 15 17:10:47 2023 +0100

    arm64: efi: Make efi_rt_lock a raw_spinlock

    commit 0e68b5517d3767562889f1d83fdb828c26adb24f upstream.

    Running a rt-kernel base on 6.2.0-rc3-rt1 on an Ampere Altra outputs
    the following:
      BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:46
      in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 9, name: kworker/u320:0
      preempt_count: 2, expected: 0
      RCU nest depth: 0, expected: 0
      3 locks held by kworker/u320:0/9:
      #0: ffff3fff8c27d128 ((wq_completion)efi_rts_wq){+.+.}-{0:0}, at: process_one_work
(./include/linux/atomic/atomic-long.h:41)
      #1: ffff80000861bdd0 ((work_completion)(&efi_rts_work.work)){+.+.}-{0:0}, at:
process_one_work (./include/linux/atomic/atomic-long.h:41)
      #2: ffffdf7e1ed3e460 (efi_rt_lock){+.+.}-{3:3}, at: efi_call_rts
(drivers/firmware/efi/runtime-wrappers.c:101)
      Preemption disabled at:
      efi_virtmap_load (./arch/arm64/include/asm/mmu_context.h:248)
      CPU: 0 PID: 9 Comm: kworker/u320:0 Tainted: G        W          6.2.0-rc3-rt1
      Hardware name: WIWYNN Mt.Jade Server System B81.03001.0005/Mt.Jade Motherboard, BIOS
1.08.20220218 (SCP: 1.08.20220218) 2022/02/18
      Workqueue: efi_rts_wq efi_call_rts
      Call trace:
      dump_backtrace (arch/arm64/kernel/stacktrace.c:158)
      show_stack (arch/arm64/kernel/stacktrace.c:165)
```

```
        dump_stack_lvl (lib/dump_stack.c:107 (discriminator 4))
        dump_stack (lib/dump_stack.c:114)
        __might_resched (kernel/sched/core.c:10134)
        rt_spin_lock (kernel/locking/rtmutex.c:1769 (discriminator 4))
        efi_call_rts (drivers/firmware/efi/runtime-wrappers.c:101)
        [...]

      This seems to come from commit ff7a167961d1 ("arm64: efi: Execute
      runtime services from a dedicated stack") which adds a spinlock. This
      spinlock is taken through:
      efi_call_rts()
      \-efi_call_virt()
        \-efi_call_virt_pointer()
          \-arch_efi_call_virt_setup()

      Make 'efi_rt_lock' a raw_spinlock to avoid being preempted.

      [ardb: The EFI runtime services are called with a different set of
             translation tables, and are permitted to use the SIMD registers.
             The context switch code preserves/restores neither, and so EFI
             calls must be made with preemption disabled, rather than only
             disabling migration.]

      Fixes: ff7a167961d1 ("arm64: efi: Execute runtime services from a dedicated stack")
      Signed-off-by: Pierre Gondois <pierre.gondois@arm.com>
      Cc: <stable@vger.kernel.org> # v6.1+
      Signed-off-by: Ard Biesheuvel <ardb@kernel.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit ab1d856fbbffda71bf799378b470478b7144e780
  Author: Ricardo Ribalda <ribalda@chromium.org>
  Date:   Thu Jan 5 15:31:29 2023 +0100

      media: uvcvideo: Fix race condition with usb_kill_urb

      commit 619d9b710cf06f7a00a17120ca92333684ac45a8 upstream.

      usb_kill_urb warranties that all the handlers are finished when it
      returns, but does not protect against threads that might be handling
      asynchronously the urb.

      For UVC, the function uvc_ctrl_status_event_async() takes care of
      control changes asynchronously.

      If the code is executed in the following order:

      CPU 0                                   CPU 1
      =====                                   =====
      uvc_status_complete()
                                              uvc_status_stop()
      uvc_ctrl_status_event_work()
                                              uvc_status_start() -> FAIL

      Then uvc_status_start will keep failing and this error will be shown:

      <4>[    5.540139] URB 0000000000000000 submitted while active
      drivers/usb/core/urb.c:378 usb_submit_urb+0x4c3/0x528

      Let's improve the current situation, by not re-submiting the urb if
      we are stopping the status event. Also process the queued work
      (if any) during stop.

      CPU 0                                   CPU 1
      =====                                   =====
      uvc_status_complete()
```

```
                                            uvc_status_stop()
                                            uvc_status_start()
    uvc_ctrl_status_event_work() -> FAIL

    Hopefully, with the usb layer protection this should be enough to cover
    all the cases.

    Cc: stable@vger.kernel.org
    Fixes: e5225c820c05 ("media: uvcvideo: Send a control event when a Control Change
interrupt arrives")
    Reviewed-by: Yunke Cao <yunkec@chromium.org>
    Signed-off-by: Ricardo Ribalda <ribalda@chromium.org>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8ed572d5a0f1509e691a75a0e3d3588050371f1e
Author: Imre Deak <imre.deak@intel.com>
Date:   Wed Feb 8 13:42:57 2023 +0200

    drm/i915: Fix system suspend without fbdev being initialized

    commit 8038510b1fe443ffbc0e356db5f47cbb8678a594 upstream.

    If fbdev is not initialized for some reason - in practice on platforms
    without display - suspending fbdev should be skipped during system
    suspend, fix this up. While at it add an assert that suspending fbdev
    only happens with the display present.

    This fixes the following:

    [   91.227923] PM: suspend entry (s2idle)
    [   91.254598] Filesystems sync: 0.025 seconds
    [   91.270518] Freezing user space processes
    [   91.272266] Freezing user space processes completed (elapsed 0.001 seconds)
    [   91.272686] OOM killer disabled.
    [   91.272872] Freezing remaining freezable tasks
    [   91.274295] Freezing remaining freezable tasks completed (elapsed 0.001 seconds)
    [   91.659622] BUG: kernel NULL pointer dereference, address: 00000000000001c8
    [   91.659981] #PF: supervisor write access in kernel mode
    [   91.660252] #PF: error_code(0x0002) - not-present page
    [   91.660511] PGD 0 P4D 0
    [   91.660647] Oops: 0002 [#1] PREEMPT SMP NOPTI
    [   91.660875] CPU: 4 PID: 917 Comm: bash Not tainted 6.2.0-rc7+ #54
    [   91.661185] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS edk2-
20221117gitfff6d81270b5-9.fc37 unknown
    [   91.661680] RIP: 0010:mutex_lock+0x19/0x30
    [   91.661914] Code: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 0f 1f 44 00 00
53 48 89 fb e8 62 d3 ff ff 31 c0 65 48 8b 14 25 00 15 03 00 <f0> 48 0f b1 13 75 06 5b c3 cc
cc cc cc 48 89 df 5b eb b4 0f 1f 40
    [   91.662840] RSP: 0018:ffffa1e8011ffc08 EFLAGS: 00010246
    [   91.663087] RAX: 0000000000000000 RBX: 00000000000001c8 RCX: 0000000000000000
    [   91.663440] RDX: ffff8be455eb0000 RSI: 0000000000000001 RDI: 00000000000001c8
    [   91.663802] RBP: ffff8be459440000 R08: ffff8be459441f08 R09: ffffffff8e1432c0
    [   91.664167] R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000001
    [   91.664532] R13: 00000000000001c8 R14: 0000000000000000 R15: ffff8be442f4fb20
    [   91.664905] FS:  00007f28ffc16740(0000) GS:ffff8be4bb900000(0000)
knlGS:0000000000000000
    [   91.665334] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [   91.665626] CR2: 00000000000001c8 CR3: 0000000114926006 CR4: 0000000000770ee0
    [   91.665988] PKRU: 55555554
    [   91.666131] Call Trace:
    [   91.666265]  <TASK>
    [   91.666381]  intel_fbdev_set_suspend+0x97/0x1b0 [i915]
    [   91.666738]  i915_drm_suspend+0xb9/0x100 [i915]
```

```
      [   91.667029]  pci_pm_suspend+0x78/0x170
      [   91.667234]  ? __pfx_pci_pm_suspend+0x10/0x10
      [   91.667461]  dpm_run_callback+0x47/0x150
      [   91.667673]  __device_suspend+0x10a/0x4e0
      [   91.667880]  dpm_suspend+0x134/0x270
      [   91.668069]  dpm_suspend_start+0x79/0x80
      [   91.668272]  suspend_devices_and_enter+0x11b/0x890
      [   91.668526]  pm_suspend.cold+0x270/0x2fc
      [   91.668737]  state_store+0x46/0x90
      [   91.668916]  kernfs_fop_write_iter+0x11b/0x200
      [   91.669153]  vfs_write+0x1e1/0x3a0
      [   91.669336]  ksys_write+0x53/0xd0
      [   91.669510]  do_syscall_64+0x58/0xc0
      [   91.669699]  ? syscall_exit_to_user_mode_prepare+0x18e/0x1c0
      [   91.669980]  ? syscall_exit_to_user_mode_prepare+0x18e/0x1c0
      [   91.670278]  ? syscall_exit_to_user_mode+0x17/0x40
      [   91.670524]  ? do_syscall_64+0x67/0xc0
      [   91.670717]  ? __irq_exit_rcu+0x3d/0x140
      [   91.670931]  entry_SYSCALL_64_after_hwframe+0x72/0xdc
      [   91.671202] RIP: 0033:0x7f28ffd14284

      v2: CC stable. (Jani)

      Fixes: f8cc091e0530 ("drm/i915/fbdev: suspend HPD before fbdev unregistration")
      References: https://gitlab.freedesktop.org/drm/intel/-/issues/8015
      Reported-and-tested-by: iczero <iczero@hellomouse.net>
      Cc: Andrzej Hajda <andrzej.hajda@intel.com>
      Cc: iczero <iczero@hellomouse.net>
      Cc: <stable@vger.kernel.org> # v6.1+
      Reviewed-by: Jani Nikula <jani.nikula@intel.com>
      Signed-off-by: Imre Deak <imre.deak@intel.com>
      Link: https://patchwork.freedesktop.org/patch/msgid/20230208114300.3123934-2-
  imre.deak@intel.com
      (cherry picked from commit 9542d708409a41449e99c9a464deb5e062c4bee2)
      Signed-off-by: Jani Nikula <jani.nikula@intel.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 982aa70e619ab0da1fe9665decf095878ac041c4
  Author: Imre Deak <imre.deak@intel.com>
  Date:   Mon Feb 6 13:48:56 2023 +0200

      drm/i915/dp_mst: Fix payload removal during output disabling

      commit eb50912ec931913e70640cecf75cb993fd26995f upstream.

      Use the correct old/new topology and payload states in
      intel_mst_disable_dp(). So far drm_atomic_get_mst_topology_state() it
      used returned either the old state, in case the state was added already
      earlier during the atomic check phase or otherwise the new state (but
      the latter could fail, which can't be handled in the enable/disable
      hooks). After the first patch in the patchset, the state should always
      get added already during the check phase, so here we can get the
      old/new states without a failure.

      drm_dp_remove_payload() should use time_slots from the old payload state
      and vc_start_slot in the new one. It should update the new payload
      states to reflect the sink's current payload table after the payload is
      removed. Pass the new topology state and the old and new payload states
      accordingly.

      This also fixes a problem where the payload allocations for multiple MST
      streams on the same link got inconsistent after a few commits, as
      during payload removal the old instead of the new payload state got
      updated, so the subsequent enabling sequence and commits used a stale
      payload state.
```

```
    v2: Constify the old payload state pointer. (Ville)

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Cc: stable@vger.kernel.org # 6.1
    Reviewed-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Daniel Vetter <daniel@ffwll.ch>
    Acked-by: Wayne Lin <wayne.lin@amd.com>
    Acked-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230206114856.2665066-4-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2909a3dff6b21448c806c92dcf5852cb6c6c516
Author: Imre Deak <imre.deak@intel.com>
Date:   Mon Feb 6 13:48:54 2023 +0200

    drm/display/dp_mst: Handle old/new payload states in drm_dp_remove_payload()

    commit e761cc20946a0094df71cb31a565a6a0d03bd8be upstream.

    Atm, drm_dp_remove_payload() uses the same payload state to both get the
    vc_start_slot required for the payload removal DPCD message and to
    deduct time_slots from vc_start_slot of all payloads after the one being
    removed.

    The above isn't always correct, as vc_start_slot must be the up-to-date
    version contained in the new payload state, but time_slots must be the
    one used when the payload was previously added, contained in the old
    payload state. The new payload's time_slots can change vs. the old one
    if the current atomic commit changes the corresponding mode.

    This patch let's drivers pass the old and new payload states to
    drm_dp_remove_payload(), but keeps these the same for now in all drivers
    not to change the behavior. A follow-up i915 patch will pass in that
    driver the correct old and new states to the function.

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Cc: Ben Skeggs <bskeggs@redhat.com>
    Cc: Karol Herbst <kherbst@redhat.com>
    Cc: Harry Wentland <harry.wentland@amd.com>
    Cc: Alex Deucher <alexander.deucher@amd.com>
    Cc: Wayne Lin <Wayne.Lin@amd.com>
    Cc: stable@vger.kernel.org # 6.1
    Cc: dri-devel@lists.freedesktop.org
    Reviewed-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Daniel Vetter <daniel@ffwll.ch>
    Acked-by: Wayne Lin <wayne.lin@amd.com>
    Acked-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230206114856.2665066-2-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6a74ead8813a911f051aaf06f4abc2df92be0fee
Author: Imre Deak <imre.deak@intel.com>
Date:   Mon Feb 6 13:48:53 2023 +0200
```

```
    drm/i915/dp_mst: Add the MST topology state for modesetted CRTCs

    commit 326b1e792ff08b4d8ecb9605aec98e4e5feef56e upstream.

    Add the MST topology for a CRTC to the atomic state if the driver
    needs to force a modeset on the CRTC after the encoder compute config
    functions are called.

    Later the MST encoder's disable hook also adds the state, but that isn't
    guaranteed to work (since in that hook getting the state may fail, which
    can't be handled there). This should fix that, while a later patch fixes
    the use of the MST state in the disable hook.

    v2: Add missing forward struct declartions, caught by hdrtest.
    v3: Factor out intel_dp_mst_add_topology_state_for_connector() used
        later in the patchset.

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Cc: stable@vger.kernel.org # 6.1
    Reviewed-by: Ville Syrjälä <ville.syrjala@linux.intel.com> # v2
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Lyude Paul <lyude@redhat.com>
    Acked-by: Daniel Vetter <daniel@ffwll.ch>
    Acked-by: Wayne Lin <wayne.lin@amd.com>
    Acked-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230206114856.2665066-1-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5b8f79af9f270021b9d3a90f2d92869b64430125
Author: Imre Deak <imre.deak@intel.com>
Date:   Wed Dec 14 20:42:58 2022 +0200

    drm/display/dp_mst: Fix payload addition on a disconnected sink

    commit 33f960e23c29d113fe3193e0bdc19ac4f3776f20 upstream.

    If an MST stream is enabled on a disconnected sink, the payload for the
    stream is not created and the MST manager's payload count/next start VC
    slot is not updated. Since the payload's start VC slot may still contain
    a valid value (!= -1) the subsequent disabling of such a stream could
    cause an incorrect decrease of the payload count/next start VC slot in
    drm_dp_remove_payload() and hence later payload additions will fail.

    Fix the above by marking the payload as invalid in the above case, so
    that it's skipped during payload removal. While at it add a debug print
    for this case.

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: <stable@vger.kernel.org> # v6.1+
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20221214184258.2869417-3-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 50f7704d1bbe2a9dd2993443ed941d464c75b253
Author: Imre Deak <imre.deak@intel.com>
Date:   Wed Dec 14 20:42:57 2022 +0200

    drm/display/dp_mst: Fix down message handling after a packet reception error

    commit 1241aedb6b5c7a5a8ad73e5eb3a41cfe18a3e00e upstream.
```

```
    After an error during receiving a packet for a multi-packet DP MST
    sideband message, the state tracking which packets have been received
    already is not reset. This prevents the reception of subsequent down
    messages (due to the pending message not yet completed with an
    end-of-message-transfer packet).

    Fix the above by resetting the reception state after a packet error.

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: <stable@vger.kernel.org> # v3.17+
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20221214184258.2869417-2-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a579ed4613b5a64074963988ad481e43cf3b917b
Author: Imre Deak <imre.deak@intel.com>
Date:   Wed Dec 14 20:42:56 2022 +0200

    drm/display/dp_mst: Fix down/up message handling after sink disconnect

    commit 1d082618bbf3b6755b8cc68c0a8122af2842d593 upstream.

    If the sink gets disconnected during receiving a multi-packet DP MST AUX
    down-reply/up-request sideband message, the state keeping track of which
    packets have been received already is not reset. This results in a failed
    sanity check for the subsequent message packet received after a sink is
    reconnected (due to the pending message not yet completed with an
    end-of-message-transfer packet), indicated by the

    "sideband msg set header failed"

    error.

    Fix the above by resetting the up/down message reception state after a
    disconnect event.

    Cc: Lyude Paul <lyude@redhat.com>
    Cc: <stable@vger.kernel.org> # v3.17+
    Signed-off-by: Imre Deak <imre.deak@intel.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20221214184258.2869417-1-
imre.deak@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bf9215c5dd5491be60139fb8eff474e456250298
Author: Imre Deak <imre.deak@intel.com>
Date:   Mon Feb 6 13:48:55 2023 +0200

    drm/display/dp_mst: Add drm_atomic_get_old_mst_topology_state()

    commit 9ffdb67af0ee625ae127711845532f670cc6a4e7 upstream.

    Add a function to get the old MST topology state, required by a
    follow-up i915 patch.

    While at it clarify the code comment of
    drm_atomic_get_new_mst_topology_state() and add _new prefix
    to the new state pointer to remind about its difference from the old
    state.

    v2: Use old_/new_ prefixes for the state pointers. (Ville)
```

```
      Cc: Lyude Paul <lyude@redhat.com>
      Cc: Ville Syrjälä <ville.syrjala@linux.intel.com>
      Cc: stable@vger.kernel.org # 6.1
      Cc: dri-devel@lists.freedesktop.org
      Reviewed-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
      Reviewed-by: Lyude Paul <lyude@redhat.com>
      Acked-by: Lyude Paul <lyude@redhat.com>
      Acked-by: Daniel Vetter <daniel@ffwll.ch>
      Acked-by: Wayne Lin <wayne.lin@amd.com>
      Acked-by: Jani Nikula <jani.nikula@intel.com>
      Signed-off-by: Imre Deak <imre.deak@intel.com>
      Link: https://patchwork.freedesktop.org/patch/msgid/20230206114856.2665066-3-
  imre.deak@intel.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 16c1da390704734ca725d20b6b3320570f7122b6
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:22 2022 +0800

      vDPA/ifcvf: allocate the adapter in dev_add()

      commit 93139037b582134deb1ed894bbc4bc1d34ff35e7 upstream.

      The adapter is the container of the vdpa_device,
      this commits allocate the adapter in dev_add()
      rather than in probe(). So that the vdpa_device()
      could be re-created when the userspace creates
      the vdpa device, and free-ed in dev_del()

      Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
      Cc: stable@vger.kernel.org
      Message-Id: <20221125145724.1129962-11-lingshan.zhu@intel.com>
      Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 3c2e391a141aa1902f9fed54384188caedf29d74
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:21 2022 +0800

      vDPA/ifcvf: manage ifcvf_hw in the mgmt_dev

      commit 6a3b2f179b49f2c6452ecc37b4778a43848b454c upstream.

      This commit allocates the hw structure in the
      management device structure. So the hardware
      can be initialized once the management device
      is allocated in probe.

      Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
      Cc: stable@vger.kernel.org
      Message-Id: <20221125145724.1129962-10-lingshan.zhu@intel.com>
      Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 25e6cbbe25b231d019940888c1d6ac1a6fe5eae6
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:20 2022 +0800

      vDPA/ifcvf: ifcvf_request_irq works on ifcvf_hw

      commit 7cfd36b7e8be6bdaeb5af0f9729871b732a7a3c8 upstream.

      All ifcvf_request_irq's callees are refactored
      to work on ifcvf_hw, so it should be decoupled
      from the adapter as well
```

```
    Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
    Cc: stable@vger.kernel.org
    Message-Id: <20221125145724.1129962-9-lingshan.zhu@intel.com>
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 133b62d15171dba7eacec2038ac3e2fed3775195
Author: Zhu Lingshan <lingshan.zhu@intel.com>
Date:   Fri Nov 25 22:57:19 2022 +0800

    vDPA/ifcvf: decouple config/dev IRQ requester and vectors allocator from the adapter

    commit a70d833e696e538a0feff5e539086c74a90ddf90 upstream.

    This commit decouples the config irq requester, the device
    shared irq requester and the MSI vectors allocator from
    the adapter. So they can be safely invoked since probe
    before the adapter is allocated.

    Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
    Cc: stable@vger.kernel.org
    Message-Id: <20221125145724.1129962-8-lingshan.zhu@intel.com>
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e73cc1854c8fb5272dfd0929b99ec82cdcb273cb
Author: Zhu Lingshan <lingshan.zhu@intel.com>
Date:   Fri Nov 25 22:57:18 2022 +0800

    vDPA/ifcvf: decouple vq irq requester from the adapter

    commit f9a9ffb2e4dbde81090416fc51662441c2a7b73b upstream.

    This commit decouples the vq irq requester from the adapter,
    so that these functions can be invoked since probe.

    Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
    Cc: stable@vger.kernel.org
    Message-Id: <20221125145724.1129962-7-lingshan.zhu@intel.com>
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c357b317e575ccaa36241da3b3c42169982eca21
Author: Zhu Lingshan <lingshan.zhu@intel.com>
Date:   Fri Nov 25 22:57:17 2022 +0800

    vDPA/ifcvf: decouple config IRQ releaser from the adapter

    commit 23dac55cec3afdbc1b4eaed1c79f2cee00477f8b upstream.

    This commit decouples config IRQ releaser from the adapter,
    so that it could be invoked once probe or in err handlers.
    ifcvf_free_irq() works on ifcvf_hw in this commit

    Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
    Cc: stable@vger.kernel.org
    Message-Id: <20221125145724.1129962-6-lingshan.zhu@intel.com>
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 027561804cb5a3e81b6ce14eb8de9d28f2bbf85e
Author: Zhu Lingshan <lingshan.zhu@intel.com>
Date:   Fri Nov 25 22:57:16 2022 +0800
```

```
      vDPA/ifcvf: decouple vq IRQ releasers from the adapter

      commit 004cbcabab46d9346e2524c4eedd71ea57fe4f3c upstream.

      This commit decouples the IRQ releasers from the
      adapter, so that these functions could be
      safely invoked once probe

      Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
      Cc: stable@vger.kernel.org
      Message-Id: <20221125145724.1129962-5-lingshan.zhu@intel.com>
      Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit f35933f80d7d8a5a894aea92f47f5d6e73e6cfbb
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:15 2022 +0800

      vDPA/ifcvf: alloc the mgmt_dev before the adapter

      commit 66e3970b16d1e960afbece65739a3628273633f1 upstream.

      This commit reverses the order of allocating the
      management device and the adapter. So that it would
      be possible to move the allocation of the adapter
      to dev_add().

      Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
      Cc: stable@vger.kernel.org
      Message-Id: <20221125145724.1129962-4-lingshan.zhu@intel.com>
      Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit ba95d05e7b0af8ac878eabd527362296e52dd3b1
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:14 2022 +0800

      vDPA/ifcvf: decouple config space ops from the adapter

      commit af8eb69a62b73a2ce5f91575453534ac07f06eb4 upstream.

      This commit decopules the config space ops from the
      adapter layer, so these functions can be invoked
      once the device is probed.

      Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
      Cc: stable@vger.kernel.org
      Message-Id: <20221125145724.1129962-3-lingshan.zhu@intel.com>
      Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 8dc360fe87ba29b1d2fe6530619c284690c37cf7
  Author: Zhu Lingshan <lingshan.zhu@intel.com>
  Date:   Fri Nov 25 22:57:13 2022 +0800

      vDPA/ifcvf: decouple hw features manipulators from the adapter

      commit d59f633dd05940739b5f46f5d4403cafb91d2742 upstream.

      This commit gets rid of ifcvf_adapter in hw features related
      functions in ifcvf_base. Then these functions are more rubust
      and de-coupling from the ifcvf_adapter layer. So these
      functions could be invoded once the device is probed, even
      before the adapter is allocaed.
```

```
    Signed-off-by: Zhu Lingshan <lingshan.zhu@intel.com>
    Cc: stable@vger.kernel.org
    Message-Id: <20221125145724.1129962-2-lingshan.zhu@intel.com>
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a367a31de29febe9979e78a56b9319234b8425cf
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Tue Mar 7 13:06:29 2023 -0800

    x86/resctl: fix scheduler confusion with 'current'

    commit 7fef099702527c3b2c5234a2ea6a24411485a13a upstream.

    The implementation of 'current' on x86 is very intentionally special: it
    is a very common thing to look up, and it uses 'this_cpu_read_stable()'
    to get the current thread pointer efficiently from per-cpu storage.

    And the keyword in there is 'stable': the current thread pointer never
    changes as far as a single thread is concerned.  Even if when a thread
    is preempted, or moved to another CPU, or even across an explicit call
    'schedule()' that thread will still have the same value for 'current'.

    It is, after all, the kernel base pointer to thread-local storage.
    That's why it's stable to begin with, but it's also why it's important
    enough that we have that special 'this_cpu_read_stable()' access for it.

    So this is all done very intentionally to allow the compiler to treat
    'current' as a value that never visibly changes, so that the compiler
    can do CSE and combine multiple different 'current' accesses into one.

    However, there is obviously one very special situation when the
    currently running thread does actually change: inside the scheduler
    itself.

    So the scheduler code paths are special, and do not have a 'current'
    thread at all.  Instead there are _two_ threads: the previous and the
    next thread - typically called 'prev' and 'next' (or prev_p/next_p)
    internally.

    So this is all actually quite straightforward and simple, and not all
    that complicated.

    Except for when you then have special code that is run in scheduler
    context, that code then has to be aware that 'current' isn't really a
    valid thing.  Did you mean 'prev'? Did you mean 'next'?

    In fact, even if then look at the code, and you use 'current' after the
    new value has been assigned to the percpu variable, we have explicitly
    told the compiler that 'current' is magical and always stable.  So the
    compiler is quite free to use an older (or newer) value of 'current',
    and the actual assignment to the percpu storage is not relevant even if
    it might look that way.

    Which is exactly what happened in the resctl code, that blithely used
    'current' in '__resctrl_sched_in()' when it really wanted the new
    process state (as implied by the name: we're scheduling 'into' that new
    resctl state).  And clang would end up just using the old thread pointer
    value at least in some configurations.

    This could have happened with gcc too, and purely depends on random
    compiler details.  Clang just seems to have been more aggressive about
    moving the read of the per-cpu current_task pointer around.

    The fix is trivial: just make the resctl code adhere to the scheduler
```

```
    rules of using the prev/next thread pointer explicitly, instead of using
    'current' in a situation where it just wasn't valid.

    That same code is then also used outside of the scheduler context (when
    a thread resctl state is explicitly changed), and then we will just pass
    in 'current' as that pointer, of course.  There is no ambiguity in that
    case.

    The fix may be trivial, but noticing and figuring out what went wrong
    was not.  The credit for that goes to Stephane Eranian.

    Reported-by: Stephane Eranian <eranian@google.com>
    Link: https://lore.kernel.org/lkml/20230303231133.1486085-1-eranian@google.com/
    Link:
https://lore.kernel.org/lkml/alpine.LFD.2.01.0908011214330.3304@localhost.localdomain/
    Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
    Tested-by: Tony Luck <tony.luck@intel.com>
    Tested-by: Stephane Eranian <eranian@google.com>
    Tested-by: Babu Moger <babu.moger@amd.com>
    Cc: stable@kernel.org
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ccf1ccdc5926907befbe880b562b2a4b5f44c087
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Tue Feb 28 16:28:57 2023 -0800

    net: tls: avoid hanging tasks on the tx_lock

    commit f3221361dc85d4de22586ce8441ec2c67b454f5d upstream.

    syzbot sent a hung task report and Eric explains that adversarial
    receiver may keep RWIN at 0 for a long time, so we are not guaranteed
    to make forward progress. Thread which took tx_lock and went to sleep
    may not release tx_lock for hours. Use interruptible sleep where
    possible and reschedule the work if it can't take the lock.

    Testing: existing selftest passes

    Reported-by: syzbot+9c0268252b8ef967c62e@syzkaller.appspotmail.com
    Fixes: 79ffe6087e91 ("net/tls: add a TX lock")
    Link: https://lore.kernel.org/all/000000000000e412e905f5b46201@google.com/
    Cc: stable@vger.kernel.org # wait 4 weeks
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Link: https://lore.kernel.org/r/20230301002857.2101894-1-kuba@kernel.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c2e4ac2ff9038b8d94629fe3957ddd0e8f59fb23
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Tue Mar 7 09:19:30 2023 -0800

    eth: fealnx: bring back this old driver

    commit 8f14820801042c221bb9fe51643a2585cac5dec2 upstream.

    This reverts commit d5e2d038dbece821f1af57acbeded3aa9a1832c1.

    We have a report of this chip being used on a

      SURECOM EP-320X-S 100/10M Ethernet PCI Adapter

    which could still have been purchased in some parts
    of the world 3 years ago.
```

```
    Cc: stable@vger.kernel.org
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=217151
    Fixes: d5e2d038dbec ("eth: fealnx: delete the driver for Myson MTD-800")
    Link: https://lore.kernel.org/r/20230307171930.4008454-1-kuba@kernel.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db5839b04f5afa9fe27f0427e7bf1e748ba11909
Author: Richard Fitzgerald <rf@opensource.cirrus.com>
Date:   Fri Dec 2 16:18:12 2022 +0000

    soundwire: cadence: Drain the RX FIFO after an IO timeout

    [ Upstream commit 0603a47bd3a8f439d7844b841eee1819353063e0 ]

    If wait_for_completion_timeout() times-out in _cdns_xfer_msg() it
    is possible that something could have been written to the RX FIFO.
    In this case, we should drain the RX FIFO so that anything in it
    doesn't carry over and mess up the next transfer.

    Obviously, if we got to this state something went wrong, and we
    don't really know the state of everything. The cleanup in this
    situation cannot be bullet-proof but we should attempt to avoid
    breaking future transaction, if only to reduce the amount of
    error noise when debugging the failure from a kernel log.

    Note that this patch only implements the draining for blocking
    (non-deferred) transfers. The deferred API doesn't have any proper
    handling of error conditions and would need some re-design before
    implementing cleanup. That is a task for a separate patch...

    Signed-off-by: Richard Fitzgerald <rf@opensource.cirrus.com>
    Reviewed-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Link: https://lore.kernel.org/r/20221202161812.4186897-4-rf@opensource.cirrus.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ebc5e61b60e67cbc14cc54159d567456f7ec5fd8
Author: Richard Fitzgerald <rf@opensource.cirrus.com>
Date:   Fri Dec 2 16:18:11 2022 +0000

    soundwire: cadence: Remove wasted space in response_buf

    [ Upstream commit 827c32d0df4bbe0d1c47d79f6a5eabfe9ac75216 ]

    The response_buf was declared much larger (128 entries) than the number
    of responses that could ever be written into it. The Cadence IP is
    configurable up to a maximum of 32 entries, and the datasheet says
    that RX_FIFO_AVAIL can be 2 larger than this. So allow up to 34
    responses.

    Also add checking in cdns_read_response() to prevent overflowing
    reponse_buf if RX_FIFO_AVAIL contains an unexpectedly large number.

    Signed-off-by: Richard Fitzgerald <rf@opensource.cirrus.com>
    Reviewed-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Link: https://lore.kernel.org/r/20221202161812.4186897-3-rf@opensource.cirrus.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b2e3f72186ff7e6a4c63c28626d84094247f2f0f
Author: Kees Cook <keescook@chromium.org>
Date:   Wed Feb 8 15:25:53 2023 -0800

    RDMA/cma: Distinguish between sockaddr_in and sockaddr_in6 by size
```

```
    [ Upstream commit 876e480da2f74715fc70e37723e77ca16a631e35 ]

    Clang can do some aggressive inlining, which provides it with greater
    visibility into the sizes of various objects that are passed into
    helpers. Specifically, compare_netdev_and_ip() can see through the type
    given to the "sa" argument, which means it can generate code for "struct
    sockaddr_in" that would have been passed to ipv6_addr_cmp() (that expects
    to operate on the larger "struct sockaddr_in6"), which would result in a
    compile-time buffer overflow condition detected by memcmp(). Logically,
    this state isn't reachable due to the sa_family assignment two callers
    above and the check in compare_netdev_and_ip(). Instead, provide a
    compile-time check on sizes so the size-mismatched code will be elided
    when inlining. Avoids the following warning from Clang:

    ../include/linux/fortify-string.h:652:4: error: call to '__read_overflow' declared with
'error' attribute: detected read beyond size of object (1st parameter)
                            __read_overflow();
                            ^
    note: In function 'cma_netevent_callback'
    note:   which inlined function 'node_from_ndev_ip'
    1 error generated.

    When the underlying object size is not known (e.g. with GCC and older
    Clang), the result of __builtin_object_size() is SIZE_MAX, which will also
    compile away, leaving the code as it was originally.

    Link: https://lore.kernel.org/r/20230208232549.never.139-kees@kernel.org
    Link: https://github.com/ClangBuiltLinux/linux/issues/1687
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Tested-by: Nathan Chancellor <nathan@kernel.org> # build
    Signed-off-by: Jason Gunthorpe <jgg@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1256516724375cb08383d1d6f0191b3b31f67785
Author: Jiapeng Chong <jiapeng.chong@linux.alibaba.com>
Date:   Mon Feb 13 11:57:09 2023 +0800

    phy: rockchip-typec: Fix unsigned comparison with less than zero

    [ Upstream commit f765c59c5a72546a2d74a92ae5d0eb0329d8e247 ]

    The dp and ufp are defined as bool type, the return value type of
    function extcon_get_state should be int, so the type of dp and ufp
    are modified to int.

    ./drivers/phy/rockchip/phy-rockchip-typec.c:827:12-14: WARNING: Unsigned expression
compared with zero: dp > 0.

    Reported-by: Abaci Robot <abaci@linux.alibaba.com>
    Link: https://bugzilla.openanolis.cn/show_bug.cgi?id=3962
    Signed-off-by: Jiapeng Chong <jiapeng.chong@linux.alibaba.com>
    Link: https://lore.kernel.org/r/20230213035709.99027-1-jiapeng.chong@linux.alibaba.com
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa2295bf75714085d58c6c907cdf9d19e2892611
Author: Manivannan Sadhasivam <mani@kernel.org>
Date:   Mon Feb 13 20:19:22 2023 +0530

    PCI: pciehp: Add Qualcomm quirk for Command Completed erratum

    [ Upstream commit 82b34b0800af8c9fc9988c290cdc813e0ca0df31 ]

    The Qualcomm PCI bridge device (Device ID 0x010e) found in chipsets such as
```

```
    SC8280XP used in Lenovo Thinkpad X13s, does not set the Command Completed
    bit unless writes to the Slot Command register change "Control" bits.

    This results in timeouts like below during boot and resume from suspend:

      pcieport 0002:00:00.0: pciehp: Timeout on hotplug command 0x03c0 (issued 2020 msec
ago)
      ...
      pcieport 0002:00:00.0: pciehp: Timeout on hotplug command 0x13f1 (issued 107724 msec
ago)

    Add the device to the Command Completed quirk to mark commands "completed"
    immediately unless they change the "Control" bits.

    Link: https://lore.kernel.org/r/20230213144922.89982-1-manivannan.sadhasivam@linaro.org
    Signed-off-by: Manivannan Sadhasivam <manivannan.sadhasivam@linaro.org>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1200162c2e873ed4b863c38261f9ee32206ff4e
Author: Mengyuan Lou <mengyuanlou@net-swift.com>
Date:   Tue Feb 7 18:24:19 2023 +0800

    PCI: Add ACS quirk for Wangxun NICs

    [ Upstream commit a2b9b123ccac913e9f9b80337d687a2fe786a634 ]

    Wangxun has verified there is no peer-to-peer between functions for the
    below selection of SFxxx, RP1000 and RP2000 NICS.  They may be
    multi-function devices, but the hardware does not advertise ACS capability.

    Add an ACS quirk for these devices so the functions can be in independent
    IOMMU groups.

    Link: https://lore.kernel.org/r/20230207102419.44326-1-mengyuanlou@net-swift.com
    Signed-off-by: Mengyuan Lou <mengyuanlou@net-swift.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3dd596f248e22bafe9d9eb4615be5e11be10de40
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Sat Feb 11 10:33:21 2023 +0800

    PCI: loongson: Add more devices that need MRRS quirk

    [ Upstream commit c768f8c5f40fcdc6f058cc2f02592163d6c6716c ]

    Loongson-2K SOC and LS7A2000 chipset add new PCI IDs that need MRRS
    quirk.  Add them.

    Link: https://lore.kernel.org/r/20230211023321.3530080-1-chenhuacai@loongson.cn
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94f68f3e059c478e240f65fcb64746fe371295df
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:16:33 2023 +0100

    kernel/fail_function: fix memory leak with using debugfs_lookup()

    [ Upstream commit 2bb3669f576559db273efe49e0e69f82450efbca ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
```

```
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Andrew Morton <akpm@linux-foundation.org>
    Reviewed-by: Yang Yingliang <yangyingliang@huawei.com>
    Link: https://lore.kernel.org/r/20230202151633.2310897-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8e47e2bf78812adbd73c45c941d3c51add30b58d
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 15:16:21 2023 +0100

    drivers: base: dd: fix memory leak with using debugfs_lookup()

    [ Upstream commit 36c893d3a759ae7c91ee7d4871ebfc7504f08c40 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: "Rafael J. Wysocki" <rafael@kernel.org>
    Link: https://lore.kernel.org/r/20230202141621.2296458-2-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bf0fd01c7cc1061fb2cfda3e2044371642108e6c
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 15:16:20 2023 +0100

    drivers: base: component: fix memory leak with using debugfs_lookup()

    [ Upstream commit 8deb87b1e810dd558371e88ffd44339fbef27870 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: "Rafael J. Wysocki" <rafael@kernel.org>
    Link: https://lore.kernel.org/r/20230202141621.2296458-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f7651fa88b17c2d7af949981a2423179db5e9453
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 15:11:00 2023 +0100

    misc: vmw_balloon: fix memory leak with using debugfs_lookup()

    [ Upstream commit 209cdbd07cfaa4b7385bad4eeb47e5ec1887d33d ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic at
    once.

    Cc: Nadav Amit <namit@vmware.com>
    Cc: VMware PV-Drivers Reviewers <pv-drivers@vmware.com>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Link: https://lore.kernel.org/r/20230202141100.2291188-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 139769c4bd8273b5e3f85ea474aa37018fe7e436
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 15:12:21 2023 +0100

    tty: pcn_uart: fix memory leak with using debugfs_lookup()

    [ Upstream commit 04a189c720aa2b6091442113ce9b9bc93552dff8 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Jiri Slaby <jirislaby@kernel.org>
    Link: https://lore.kernel.org/r/20230202141221.2293012-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa6030a4d0ce460eed66656703ef5130ab80c834
Author: Mika Westerberg <mika.westerberg@linux.intel.com>
Date:   Tue Jan 31 11:24:05 2023 +0200

    PCI: Distribute available resources for root buses, too

    [ Upstream commit 7180c1d08639f28e63110ad35815f7a1785b8a19 ]

    Previously we distributed spare resources only upon hot-add, so if the
    initial root bus scan found devices that had not been fully configured by
    the BIOS, we allocated only enough resources to cover what was then
    present. If some of those devices were hotplug bridges, we did not leave
    any additional resource space for future expansion.

    Distribute the available resources for root buses, too, to make this work
    the same way as the normal hotplug case.

    A previous commit to do this was reverted due to a regression reported by
    Jonathan Cameron:

      e96e27fc6f79 ("PCI: Distribute available resources for root buses, too")
      5632e2beaf9d ("Revert "PCI: Distribute available resources for root buses, too"")

    This commit changes pci_bridge_resources_not_assigned() to work with
    bridges that do not have all the resource windows programmed by the boot
    firmware (previously we expected all I/O, memory and prefetchable memory
    were programmed).

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216000
    Link: https://lore.kernel.org/r/20220905080232.36087-5-mika.westerberg@linux.intel.com
    Link: https://lore.kernel.org/r/20230131092405.29121-4-mika.westerberg@linux.intel.com
    Reported-by: Chris Chiu <chris.chiu@canonical.com>
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6208bdb65475da25eabc5e0223dd880c6903a96b
Author: Mika Westerberg <mika.westerberg@linux.intel.com>
Date:   Tue Jan 31 11:24:04 2023 +0200

    PCI: Take other bus devices into account when distributing resources

    [ Upstream commit 9db0b9b6a14249ef65a5f1e5e3b37762af96f425 ]

    A PCI bridge may reside on a bus with other devices as well. The resource
    distribution code does not take this into account and therefore it expands
    the bridge resource windows too much, not leaving space for the other
```

```
    devices (or functions of a multifunction device).  This leads to an issue
    that Jonathan reported when running QEMU with the following topology (QEMU
    parameters):

      -device pcie-root-port,port=0,id=root_port13,chassis=0,slot=2  \
      -device x3130-upstream,id=sw1,bus=root_port13,multifunction=on \
      -device e1000,bus=root_port13,addr=0.1                         \
      -device xio3130-downstream,id=fun1,bus=sw1,chassis=0,slot=3    \
      -device e1000,bus=fun1

    The first e1000 NIC here is another function in the switch upstream port.
    This leads to following errors:

      pci 0000:00:04.0: bridge window [mem 0x10200000-0x103fffff] to [bus 02-04]
      pci 0000:02:00.0: bridge window [mem 0x10200000-0x103fffff] to [bus 03-04]
      pci 0000:02:00.1: BAR 0: failed to assign [mem size 0x00020000]
      e1000 0000:02:00.1: can't ioremap BAR 0: [??? 0x00000000 flags 0x0]

    Fix this by taking into account bridge windows, device BARs and SR-IOV PF
    BARs on the bus (PF BARs include space for VF BARS so only account PF
    BARs), including the ones belonging to bridges themselves if it has any.

    Link: https://lore.kernel.org/linux-pci/20221014124553.0000696f@huawei.com/
    Link: https://lore.kernel.org/linux-pci/6053736d-1923-41e7-def9-
7585ce1772d9@ixsystems.com/
    Link: https://lore.kernel.org/r/20230131092405.29121-3-mika.westerberg@linux.intel.com
    Reported-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Reported-by: Alexander Motin <mav@ixsystems.com>
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 730b81ea892c5b2dfacd638027d09343984c322e
Author: Mika Westerberg <mika.westerberg@linux.intel.com>
Date:   Tue Jan 31 11:24:03 2023 +0200

    PCI: Align extra resources for hotplug bridges properly

    [ Upstream commit 08f0a15ee8adb4846b08ca5d5c175fbf0f652bc9 ]

    After division the extra resource space per hotplug bridge may not be
    aligned according to the window alignment, so align it before passing it
    down for further distribution.

    Link: https://lore.kernel.org/r/20230131092405.29121-2-mika.westerberg@linux.intel.com
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ec8a0f412cbb8dfd228161f259a09c13a472e564
Author: Daniel Scally <dan.scally@ideasonboard.com>
Date:   Mon Feb 6 16:17:52 2023 +0000

    usb: gadget: uvc: Make bSourceID read/write

    [ Upstream commit b3c839bd8a07d303bc59a900d55dd35c7826562c ]

    At the moment, the UVC function graph is hardcoded IT -> PU -> OT.
    To add XU support we need the ability to insert the XU descriptors
    into the chain. To facilitate that, make the output terminal's
    bSourceID attribute writeable so that we can configure its source.

    Signed-off-by: Daniel Scally <dan.scally@ideasonboard.com>
    Link: https://lore.kernel.org/r/20230206161802.892954-2-dan.scally@ideasonboard.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

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

commit 00b42e3b203d5e3b17dfe511fd8eb604f0610d3d
Author: Daniel Scally <dan.scally@ideasonboard.com>
Date:   Thu Feb 2 11:41:37 2023 +0000

    usb: uvc: Enumerate valid values for color matching

    [ Upstream commit e16cab9c1596e251761d2bfb5e1467950d616963 ]

    The color matching descriptors defined in the UVC Specification
    contain 3 fields with discrete numeric values representing particular
    settings. Enumerate those values so that later code setting them can
    be more readable.

    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Daniel Scally <dan.scally@ideasonboard.com>
    Link: https://lore.kernel.org/r/20230202114142.300858-2-dan.scally@ideasonboard.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5ee39d2216dc98a894ffc38d15e1ef2d41e2266b
Author: Kees Cook <keescook@chromium.org>
Date:   Sat Feb 4 10:35:46 2023 -0800

    USB: ene_usb6250: Allocate enough memory for full object

    [ Upstream commit ce33e64c1788912976b61314b56935abd4bc97ef ]

    The allocation of PageBuffer is 512 bytes in size, but the dereferencing
    of struct ms_bootblock_idi (also size 512) happens at a calculated offset
    within the allocation, which means the object could potentially extend
    beyond the end of the allocation. Avoid this case by just allocating
    enough space to catch any accesses beyond the end. Seen with GCC 13:

    ../drivers/usb/storage/ene_ub6250.c: In function 'ms_lib_process_bootblock':
    ../drivers/usb/storage/ene_ub6250.c:1050:44: warning: array subscript 'struct
ms_bootblock_idi[0]' is partly outside array bounds of 'unsigned char[512]' [-Warray-
bounds=]
     1050 |                         if (le16_to_cpu(idi->wIDIgeneralConfiguration) !=
MS_IDI_GENERAL_CONF)
          |                                                ^~
    ../include/uapi/linux/byteorder/little_endian.h:37:51: note: in definition of macro
'__le16_to_cpu'
       37 | #define __le16_to_cpu(x) ((__force __u16)(__le16)(x))
          |                                                   ^
    ../drivers/usb/storage/ene_ub6250.c:1050:29: note: in expansion of macro 'le16_to_cpu'
     1050 |                         if (le16_to_cpu(idi->wIDIgeneralConfiguration) !=
MS_IDI_GENERAL_CONF)
          |                             ^~~~~~~~~~
    In file included from ../drivers/usb/storage/ene_ub6250.c:5:
    In function 'kmalloc',
        inlined from 'ms_lib_process_bootblock' at
../drivers/usb/storage/ene_ub6250.c:942:15:
    ../include/linux/slab.h:580:24: note: at offset [256, 512] into object of size 512
allocated by 'kmalloc_trace'
      580 |                 return kmalloc_trace(
          |                        ^~~~~~~~~~~~~~
      581 |                                 kmalloc_caches[kmalloc_type(flags)][index],
          |                                 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      582 |                                 flags, size);
          |                                 ~~~~~~~~~~~~

    Cc: Alan Stern <stern@rowland.harvard.edu>
    Signed-off-by: Kees Cook <keescook@chromium.org>
```

```
    Link: https://lore.kernel.org/r/20230204183546.never.849-kees@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cc5472b517ef2321a106ab9d93484dadf995136a
Author: Kees Cook <keescook@chromium.org>
Date:   Sat Feb 4 10:36:52 2023 -0800

    usb: host: xhci: mvebu: Iterate over array indexes instead of using pointer math

    [ Upstream commit 0fbd2cda92cdb00f72080665554a586f88bca821 ]

    Walking the dram->cs array was seen as accesses beyond the first array
    item by the compiler. Instead, use the array index directly. This allows
    for run-time bounds checking under CONFIG_UBSAN_BOUNDS as well. Seen
    with GCC 13 with -fstrict-flex-arrays:

    In function 'xhci_mvebu_mbus_config',
        inlined from 'xhci_mvebu_mbus_init_quirk' at ../drivers/usb/host/xhci-mvebu.c:66:2:
    ../drivers/usb/host/xhci-mvebu.c:37:28: warning: array subscript 0 is outside array
bounds of 'const struct mbus_dram_window[0]' [-Warray-bounds=]
       37 |                 writel(((cs->size - 1) & 0xffff0000) | (cs->mbus_attr << 8) |
          |                            ~~^~~~~~

    Cc: Mathias Nyman <mathias.nyman@intel.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/20230204183651.never.663-kees@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 67c931a3f2f061bf457995fd21fff114325e0c30
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:34 2023 +0100

    USB: gadget: pxa27x_udc: fix memory leak with using debugfs_lookup()

    [ Upstream commit 7a6952fa0366d4408eb8695af1a0578c39ec718a ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Daniel Mack <daniel@zonque.org>
    Cc: Haojian Zhuang <haojian.zhuang@gmail.com>
    Cc: Robert Jarzmik <robert.jarzmik@free.fr>
    Link: https://lore.kernel.org/r/20230202153235.2412790-12-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8d48a7887dbca22e064c20caf20ae7949019fe9b
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:33 2023 +0100

    USB: gadget: pxa25x_udc: fix memory leak with using debugfs_lookup()

    [ Upstream commit 7a038a681b7df78362d9fc7013e5395a694a9d3a ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Daniel Mack <daniel@zonque.org>
    Cc: Haojian Zhuang <haojian.zhuang@gmail.com>
```

```
    Cc: Robert Jarzmik <robert.jarzmik@free.fr>
    Link: https://lore.kernel.org/r/20230202153235.2412790-11-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 72c25eb9ae4993ccac4821354ff34eb1f32e4781
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:32 2023 +0100

    USB: gadget: lpc32xx_udc: fix memory leak with using debugfs_lookup()

    [ Upstream commit e3965acaf3739fde9d74ad82979b46d37c6c208f ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Jakob Koschel <jakobkoschel@gmail.com>
    Cc: Miaoqian Lin <linmq006@gmail.com>
    Acked-by: Vladimir Zapolskiy <vz@mleia.com>
    Link: https://lore.kernel.org/r/20230202153235.2412790-10-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f30c7046dfa2748520a8045bb43ed2fbca0373b5
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:31 2023 +0100

    USB: gadget: bcm63xx_udc: fix memory leak with using debugfs_lookup()

    [ Upstream commit a91c99b1fe5c6f7e52fb932ad9e57ec7cfe913ec ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Kevin Cernekee <cernekee@gmail.com>
    Link: https://lore.kernel.org/r/20230202153235.2412790-9-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0933eca15f5223b5c2412080c8c3de8758465c78
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:30 2023 +0100

    USB: gadget: gr_udc: fix memory leak with using debugfs_lookup()

    [ Upstream commit 73f4451368663ad28daa67980c6dd11d83b303eb ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Jakob Koschel <jakobkoschel@gmail.com>
    Link: https://lore.kernel.org/r/20230202153235.2412790-8-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9d537c35e48feba9d450acca0ff14a55ce1ec450
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:29 2023 +0100
```

```
    USB: isp1362: fix memory leak with using debugfs_lookup()

    [ Upstream commit c26e682afc14caa87d44beed271eec8991e93c65 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Vincent Mailhol <mailhol.vincent@wanadoo.fr>
    Link: https://lore.kernel.org/r/20230202153235.2412790-7-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a60b4902a626dda08a31d9cf89ccce11bef8dd33
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:28 2023 +0100

    USB: isp116x: fix memory leak with using debugfs_lookup()

    [ Upstream commit a95f62d5813facbec20ec087472eb313ee5fa8af ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Olav Kongas <ok@artecdesign.ee>
    Link: https://lore.kernel.org/r/20230202153235.2412790-6-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 55c2ffc534928f4732199617e3b746d79a57898f
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:27 2023 +0100

    USB: fotg210: fix memory leak with using debugfs_lookup()

    [ Upstream commit 6b4040f452037a7e95472577891d57c6b18c89c5 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Reviewed-by: Linus Walleij <linus.walleij@linaro.org>
    Link: https://lore.kernel.org/r/20230202153235.2412790-5-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 04fdfec7b0286972cb5457ef958c92585447a39f
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:26 2023 +0100

    USB: sl811: fix memory leak with using debugfs_lookup()

    [ Upstream commit e1523c4dbc54e164638ff8729d511cf91e27be04 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Vincent Mailhol <mailhol.vincent@wanadoo.fr>
    Link: https://lore.kernel.org/r/20230202153235.2412790-4-gregkh@linuxfoundation.org
```

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

commit 9cb88847b8b86f132309030022a23dca895b6f61
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:25 2023 +0100

    USB: uhci: fix memory leak with using debugfs_lookup()

    [ Upstream commit 0a3f82c79c86278e7f144564b1cb6cc5c3657144 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Alan Stern <stern@rowland.harvard.edu>
    Link: https://lore.kernel.org/r/20230202153235.2412790-3-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2b8aa879e28df11e45855b04788050c61fb6b02a
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:24 2023 +0100

    USB: ULPI: fix memory leak with using debugfs_lookup()

    [ Upstream commit 8f4d25eba599c4bd4b5ea8ae8752cda480a9d563 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Acked-by: Heikki Krogerus <heikki.krogerus@linux.intel.com>
    Link: https://lore.kernel.org/r/20230202153235.2412790-2-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 972e0682f6e3ee6ecf002657df4aaa511d51dd6c
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:32:23 2023 +0100

    USB: chipidea: fix memory leak with using debugfs_lookup()

    [ Upstream commit ff35f3ea3baba5b81416ac02d005cfbf6dd182fa ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Peter Chen <peter.chen@kernel.org>
    Link: https://lore.kernel.org/r/20230202153235.2412790-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bab872b638130a18fd54d9adfad7db77ed6457be
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:28:20 2023 +0100

    USB: dwc3: fix memory leak with using debugfs_lookup()

    [ Upstream commit be308d68785b205e483b3a0c61ba3a82da468f2c ]
```

```
    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Note, the root dentry for the debugfs directory for the device needs to
    be saved so we don't have to keep looking it up, which required a bit
    more refactoring to properly create and remove it when needed.

    Reported-by: Bruce Chen <bruce.chen@unisoc.com>
    Reported-by: Cixi Geng <cixi.geng1@unisoc.com>
    Tested-by: Cixi Geng <gengcixi@gmail.com>
    Acked-by: Thinh Nguyen <Thinh.Nguyen@synopsys.com>
    Link: https://lore.kernel.org/r/20230202152820.2409908-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb16f3102607b69e1a0233f4b73c6e337f86ef8d
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 15:11:38 2023 +0100

    staging: pi433: fix memory leak with using debugfs_lookup()

    [ Upstream commit 2f36e789e540df6a9fbf471b3a2ba62a8b361586 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.  This requires saving off the root directory dentry to make
    creation of individual device subdirectories easier.

    Cc: Paulo Miguel Almeida <paulo.miguel.almeida.rodenas@gmail.com>
    Cc: Dan Carpenter <error27@gmail.com>
    Cc: Sidong Yang <realwakka@gmail.com>
    Cc: Liu Shixin <liushixin2@huawei.com>
    Cc: "Uwe Kleine-König" <u.kleine-koenig@pengutronix.de>
    Link: https://lore.kernel.org/r/20230202141138.2291946-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f45374c1b2938bc9d8b6386e97c766ec8c4bc091
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Wed Feb 1 12:30:18 2023 +0800

    PCI: loongson: Prevent LS7A MRRS increases

    [ Upstream commit 8b3517f88ff2983f52698893519227c10aac90b2 ]

    Except for isochronous-configured devices, software may set
    Max_Read_Request_Size (MRRS) to any value up to 4096.  If a device issues a
    read request with size greater than the completer's Max_Payload_Size (MPS),
    the completer is required to break the response into multiple completions.

    Instead of correctly responding with multiple completions to a large read
    request, some LS7A Root Ports respond with a Completer Abort.  To prevent
    this, the MRRS must be limited to an implementation-specific value.

    The OS cannot detect that value, so rely on BIOS to configure MRRS before
    booting, and quirk the Root Ports so we never set an MRRS larger than that
    BIOS value for any downstream device.

    N.B. Hot-added devices are not configured by BIOS, and they power up with
    MRRS = 512 bytes, so these devices will be limited to 512 bytes.  If the
    LS7A limit is smaller, those hot-added devices may not work correctly, but
    per [1], hotplug is not supported with this chipset revision.
```

```
    [1] https://lore.kernel.org/r/073638a7-ae68-2847-ac3d-29e5e760d6af@loongson.cn

    [bhelgaas: commit log]
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216884
    Link: https://lore.kernel.org/r/20230201043018.778499-3-chenhuacai@loongson.cn
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 104c82d862fa83d701cd0d32e313242d329fd69d
Author: Huacai Chen <chenhuacai@kernel.org>
Date:   Wed Feb 1 12:30:17 2023 +0800

    PCI/portdrv: Prevent LS7A Bus Master clearing on shutdown

    [ Upstream commit 62b6dee1b44aa23b3935543aff7df80399ec726b ]

    After cc27b735ad3a ("PCI/portdrv: Turn off PCIe services during shutdown")
    we observe hangs during poweroff/reboot on systems with LS7A chipset.

    This happens because the portdrv .shutdown() method (pcie_portdrv_remove())
    clears PCI_COMMAND_MASTER via pci_disable_device(), which prevents bridges
    from forwarding memory or I/O Requests in the upstream direction (PCIe
    r6.0, sec 7.5.1.1.3).

    LS7A Root Ports have a hardware defect: clearing PCI_COMMAND_MASTER *also*
    prevents the bridge from forwarding CPU MMIO requests in the downstream
    direction, and these MMIO accesses to devices below the bridge happen even
    after .shutdown(), e.g., to print console messages.  LS7A neither forwards
    the requests nor sends an unsuccessful completion to the CPU, so the CPU
    waits forever, resulting in the hang.

    The purpose of .shutdown() is to disable interrupts and DMA from the
    device.  PCIe ports may generate interrupts (either MSI/MSI-X or INTx) for
    AER, DPC, PME, hotplug, etc., but they never perform DMA except MSI/MSI-X.
    Clearing PCI_COMMAND_MASTER effectively disables MSI/MSI-X, but not INTx.

    The port service driver .remove() methods clear the interrupt enables in
    PCI_ERR_ROOT_COMMAND, PCI_EXP_DPC_CTL, PCI_EXP_SLTCTL, and PCI_EXP_RTCTL,
    etc., which disables interrupts regardless of whether they are MSI/MSI-X or
    INTx.

    Add a pcie_portdrv_shutdown() method that calls all the port service driver
    .remove() methods to clear the interrupt enables for each service but does
    not clear Bus Mastering on the port itself.

    [bhelgaas: commit log]
    Link: https://lore.kernel.org/r/20230201043018.778499-2-chenhuacai@loongson.cn
    Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ab7fb29268b390a79d7efc1d775057a9cca9e034
Author: Richard Fitzgerald <rf@opensource.cirrus.com>
Date:   Mon Jan 23 17:25:20 2023 +0000

    soundwire: bus_type: Avoid lockdep assert in sdw_drv_probe()

    [ Upstream commit 3dca1f89ae3455963d7b53245ecf298ea9bae857 ]

    Don't hold sdw_dev_lock while calling the peripheral driver
    probe() and remove() callbacks.

    Holding sdw_dev_lock around the probe() and remove() calls causes
```

```
    a theoretical mutex inversion which lockdep will assert on.

    During probe() the sdw_dev_lock mutex is taken first and then
    ASoC/ALSA locks are taken by the probe() implementation.

    During normal operation ASoC can take its locks and then trigger
    a runtime resume of the component. The SoundWire resume will then
    take sdw_dev_lock. This is the reverse order compared to probe().

    It's not necessary to hold sdw_dev_lock when calling the probe()
    and remove(), it is only used to prevent the bus core calling the
    driver callbacks if there isn't a driver or the driver is removing.

    All calls to the driver callbacks are guarded by the 'probed' flag.
    So if sdw_dev_lock is held while setting and clearing the 'probed'
    flag this is sufficient to guarantee the safety of callback
    functions.

    Removing the mutex from around the call to probe() means that it
    is now possible for a bus event (PING response) to be handled in
    parallel with the probe(). But sdw_bus_probe() already has
    handling for this by calling the device update_status() after
    the probe() has completed.

    Example lockdep assert:
    [   46.098514] ======================================================
    [   46.104736] WARNING: possible circular locking dependency detected
    [   46.110961] 6.1.0-rc4-jamerson #1 Tainted: G            E
    [   46.116842] ------------------------------------------------------
    [   46.123063] mpg123/1130 is trying to acquire lock:
    [   46.127883] ffff8b445031fb80 (&slave->sdw_dev_lock){+.+.}-{3:3}, at:
 sdw_update_slave_status+0x26/0x70
    [   46.137225]
                   but task is already holding lock:
    [   46.143074] ffffffffc1455310 (&card->pcm_mutex){+.+.}-{3:3}, at:
 dpcm_fe_dai_open+0x49/0x830
    [   46.151536]
                   which lock already depends on the new lock.[   46.159732]
                   the existing dependency chain (in reverse order) is:
    [   46.167231]
                   -> #4 (&card->pcm_mutex){+.+.}-{3:3}:
    [   46.173428]        __mutex_lock+0x94/0x920
    [   46.177542]        snd_soc_dpcm_runtime_update+0x2e/0x100
    [   46.182958]        snd_soc_dapm_put_enum_double+0x1c2/0x200
    [   46.188548]        snd_ctl_elem_write+0x10c/0x1d0
    [   46.193268]        snd_ctl_ioctl+0x126/0x850
    [   46.197556]        __x64_sys_ioctl+0x87/0xc0
    [   46.201845]        do_syscall_64+0x38/0x90
    [   46.205959]        entry_SYSCALL_64_after_hwframe+0x63/0xcd
    [   46.211553]
                   -> #3 (&card->controls_rwsem){++++}-{3:3}:
    [   46.218188]        down_write+0x2b/0xd0
    [   46.222038]        snd_ctl_add_replace+0x39/0xb0
    [   46.226672]        snd_soc_add_controls+0x53/0x80
    [   46.231393]        soc_probe_component+0x1e4/0x2a0
    [   46.236202]        snd_soc_bind_card+0x51a/0xc80
    [   46.240836]        devm_snd_soc_register_card+0x43/0x90
    [   46.246079]        mc_probe+0x982/0xfe0 [snd_soc_sof_sdw]
    [   46.251500]        platform_probe+0x3c/0xa0
    [   46.255700]        really_probe+0xde/0x390
    [   46.259814]        __driver_probe_device+0x78/0x180
    [   46.264710]        driver_probe_device+0x1e/0x90
    [   46.269347]        __driver_attach+0x9f/0x1f0
    [   46.273721]        bus_for_each_dev+0x78/0xc0
    [   46.278098]        bus_add_driver+0x1ac/0x200
```

```
[   46.282473]        driver_register+0x8f/0xf0
[   46.286759]        do_one_initcall+0x58/0x310
[   46.291136]        do_init_module+0x4c/0x1f0
[   46.295422]        __do_sys_finit_module+0xb4/0x130
[   46.300321]        do_syscall_64+0x38/0x90
[   46.304434]        entry_SYSCALL_64_after_hwframe+0x63/0xcd
[   46.310027]
                -> #2 (&card->mutex){+.+.}-{3:3}:
[   46.315883]        __mutex_lock+0x94/0x920
[   46.320000]        snd_soc_bind_card+0x3e/0xc80
[   46.324551]        devm_snd_soc_register_card+0x43/0x90
[   46.329798]        mc_probe+0x982/0xfe0 [snd_soc_sof_sdw]
[   46.335219]        platform_probe+0x3c/0xa0
[   46.339420]        really_probe+0xde/0x390
[   46.343532]        __driver_probe_device+0x78/0x180
[   46.348430]        driver_probe_device+0x1e/0x90
[   46.353065]        __driver_attach+0x9f/0x1f0
[   46.357437]        bus_for_each_dev+0x78/0xc0
[   46.361812]        bus_add_driver+0x1ac/0x200
[   46.366716]        driver_register+0x8f/0xf0
[   46.371528]        do_one_initcall+0x58/0x310
[   46.376424]        do_init_module+0x4c/0x1f0
[   46.381239]        __do_sys_finit_module+0xb4/0x130
[   46.386665]        do_syscall_64+0x38/0x90
[   46.391299]        entry_SYSCALL_64_after_hwframe+0x63/0xcd
[   46.397416]
                -> #1 (client_mutex){+.+.}-{3:3}:
[   46.404307]        __mutex_lock+0x94/0x920
[   46.408941]        snd_soc_add_component+0x24/0x2c0
[   46.414345]        devm_snd_soc_register_component+0x54/0xa0
[   46.420522]        cs35l56_common_probe+0x280/0x370 [snd_soc_cs35l56]
[   46.427487]        cs35l56_sdw_probe+0xf4/0x170 [snd_soc_cs35l56_sdw]
[   46.434442]        sdw_drv_probe+0x80/0x1a0
[   46.439136]        really_probe+0xde/0x390
[   46.443738]        __driver_probe_device+0x78/0x180
[   46.449120]        driver_probe_device+0x1e/0x90
[   46.454247]        __driver_attach+0x9f/0x1f0
[   46.459106]        bus_for_each_dev+0x78/0xc0
[   46.463971]        bus_add_driver+0x1ac/0x200
[   46.468825]        driver_register+0x8f/0xf0
[   46.473592]        do_one_initcall+0x58/0x310
[   46.478441]        do_init_module+0x4c/0x1f0
[   46.483202]        __do_sys_finit_module+0xb4/0x130
[   46.488572]        do_syscall_64+0x38/0x90
[   46.493158]        entry_SYSCALL_64_after_hwframe+0x63/0xcd
[   46.499229]
                -> #0 (&slave->sdw_dev_lock){+.+.}-{3:3}:
[   46.506737]        __lock_acquire+0x1121/0x1df0
[   46.511765]        lock_acquire+0xd5/0x300
[   46.516360]        __mutex_lock+0x94/0x920
[   46.520949]        sdw_update_slave_status+0x26/0x70
[   46.526409]        sdw_clear_slave_status+0xd8/0xe0
[   46.531783]        intel_resume_runtime+0x139/0x2a0
[   46.537155]        __rpm_callback+0x41/0x120
[   46.541919]        rpm_callback+0x5d/0x70
[   46.546422]        rpm_resume+0x531/0x7e0
[   46.550920]        __pm_runtime_resume+0x4a/0x80
[   46.556024]        snd_soc_pcm_component_pm_runtime_get+0x2f/0xc0
[   46.562611]        __soc_pcm_open+0x62/0x520
[   46.567375]        dpcm_be_dai_startup+0x116/0x210
[   46.572661]        dpcm_fe_dai_open+0xf7/0x830
[   46.577597]        snd_pcm_open_substream+0x54a/0x8b0
[   46.583145]        snd_pcm_open.part.0+0xdc/0x200
[   46.588341]        snd_pcm_playback_open+0x51/0x80
[   46.593625]        chrdev_open+0xc0/0x250
```

```
     [   46.598129]        do_dentry_open+0x15f/0x430
     [   46.602981]        path_openat+0x75e/0xa80
     [   46.607575]        do_filp_open+0xb2/0x160
     [   46.612162]        do_sys_openat2+0x9a/0x160
     [   46.616922]        __x64_sys_openat+0x53/0xa0
     [   46.621767]        do_syscall_64+0x38/0x90
     [   46.626352]        entry_SYSCALL_64_after_hwframe+0x63/0xcd
     [   46.632414]
                    other info that might help us debug this:[   46.641862] Chain exists of:
                      &slave->sdw_dev_lock --> &card->controls_rwsem --> &card->pcm_mutex[
  46.655145]  Possible unsafe locking scenario:[   46.662048]        CPU0
  CPU1
     [   46.667080]        ----                    ----
     [   46.672108]   lock(&card->pcm_mutex);
     [   46.676267]                                lock(&card->controls_rwsem);
     [   46.683382]                                lock(&card->pcm_mutex);
     [   46.690063]   lock(&slave->sdw_dev_lock);
     [   46.694574]
                     *** DEADLOCK ***[   46.701942] 2 locks held by mpg123/1130:
     [   46.706356]  #0: ffff8b4457b22b90 (&pcm->open_mutex){+.+.}-{3:3}, at:
  snd_pcm_open.part.0+0xc9/0x200
     [   46.715999]  #1: ffffffffc1455310 (&card->pcm_mutex){+.+.}-{3:3}, at:
  dpcm_fe_dai_open+0x49/0x830
     [   46.725390]
                    stack backtrace:
     [   46.730752] CPU: 0 PID: 1130 Comm: mpg123 Tainted: G            E      6.1.0-rc4-
  jamerson #1
     [   46.739703] Hardware name: AAEON UP-WHL01/UP-WHL01, BIOS UPW1AM19 11/10/2020
     [   46.747270] Call Trace:
     [   46.750239]  <TASK>
     [   46.752857]  dump_stack_lvl+0x56/0x73
     [   46.757045]  check_noncircular+0x102/0x120
     [   46.761664]  __lock_acquire+0x1121/0x1df0
     [   46.766197]  lock_acquire+0xd5/0x300
     [   46.770292]  ? sdw_update_slave_status+0x26/0x70
     [   46.775432]  ? lock_is_held_type+0xe2/0x140
     [   46.780143]  __mutex_lock+0x94/0x920
     [   46.784241]  ? sdw_update_slave_status+0x26/0x70
     [   46.789387]  ? find_held_lock+0x2b/0x80
     [   46.793750]  ? sdw_update_slave_status+0x26/0x70
     [   46.798894]  ? lock_release+0x147/0x2f0
     [   46.803262]  ? lockdep_init_map_type+0x47/0x250
     [   46.808315]  ? sdw_update_slave_status+0x26/0x70
     [   46.813456]  sdw_update_slave_status+0x26/0x70
     [   46.818422]  sdw_clear_slave_status+0xd8/0xe0
     [   46.823302]  ? pm_generic_runtime_suspend+0x30/0x30
     [   46.828706]  intel_resume_runtime+0x139/0x2a0
     [   46.833583]  ? _raw_spin_unlock_irq+0x24/0x50
     [   46.838462]  ? pm_generic_runtime_suspend+0x30/0x30
     [   46.843866]  __rpm_callback+0x41/0x120
     [   46.848142]  ? pm_generic_runtime_suspend+0x30/0x30
     [   46.853550]  rpm_callback+0x5d/0x70
     [   46.857568]  rpm_resume+0x531/0x7e0
     [   46.861578]  ? _raw_spin_lock_irqsave+0x62/0x70
     [   46.866634]  __pm_runtime_resume+0x4a/0x80
     [   46.871258]  snd_soc_pcm_component_pm_runtime_get+0x2f/0xc0
     [   46.877358]  __soc_pcm_open+0x62/0x520
     [   46.881634]  ? dpcm_add_paths.isra.0+0x35d/0x4c0
     [   46.886784]  dpcm_be_dai_startup+0x116/0x210
     [   46.891592]  dpcm_fe_dai_open+0xf7/0x830
     [   46.896046]  ? debug_mutex_init+0x33/0x50
     [   46.900591]  snd_pcm_open_substream+0x54a/0x8b0
     [   46.905658]  snd_pcm_open.part.0+0xdc/0x200
     [   46.910376]  ? wake_up_q+0x90/0x90
     [   46.914312]  snd_pcm_playback_open+0x51/0x80
```

```
     [   46.919118]  chrdev_open+0xc0/0x250
     [   46.923147]  ? cdev_device_add+0x90/0x90
     [   46.927608]  do_dentry_open+0x15f/0x430
     [   46.931976]  path_openat+0x75e/0xa80
     [   46.936086]  do_filp_open+0xb2/0x160
     [   46.940194]  ? lock_release+0x147/0x2f0
     [   46.944563]  ? _raw_spin_unlock+0x29/0x50
     [   46.949101]  do_sys_openat2+0x9a/0x160
     [   46.953377]  __x64_sys_openat+0x53/0xa0
     [   46.957733]  do_syscall_64+0x38/0x90
     [   46.961829]  entry_SYSCALL_64_after_hwframe+0x63/0xcd
     [   46.967402] RIP: 0033:0x7fa6397ccd3b
     [   46.971506] Code: 25 00 00 41 00 3d 00 00 41 00 74 4b 64 8b 04 25 18 00 00 00 85 c0
 75 67 44 89 e2 48 89 ee bf 9c ff ff ff b8 01 01 00 00 0f 05 <48> 3d 00 f0 ff ff 0f 87 91 00
 00 00 48 8b 4c 24 28 64 48 33 0c 25
     [   46.991413] RSP: 002b:00007fff838e8990 EFLAGS: 00000246 ORIG_RAX: 0000000000000101
     [   46.999580] RAX: ffffffffffffffda RBX: 0000000000080802 RCX: 00007fa6397ccd3b
     [   47.007311] RDX: 0000000000080802 RSI: 00007fff838e8b50 RDI: 00000000ffffff9c
     [   47.015047] RBP: 00007fff838e8b50 R08: 0000000000000000 R09: 0000000000000011
     [   47.022787] R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000080802
     [   47.030539] R13: 0000000000000004 R14: 0000000000000000 R15: 00007fff838e8b50
     [   47.038289]  </TASK>

     Signed-off-by: Richard Fitzgerald <rf@opensource.cirrus.com>
     Reviewed-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
     Link: https://lore.kernel.org/r/20230123172520.339367-1-rf@opensource.cirrus.com
     Signed-off-by: Vinod Koul <vkoul@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 72569b5e8fffc65dd1c3a60cbc21c9011309a1a0
 Author: Marek Vasut <marex@denx.de>
 Date:   Fri Jan 27 00:14:52 2023 +0100

     media: uvcvideo: Add GUID for BGRA/X 8:8:8:8

     [ Upstream commit 015d44c2b700ba9639dd29672ba362796cc0be54 ]

     The Cypress EZUSB FX3 UVC example can be configured to report pixel
     format "e436eb7e-524f-11ce-9f53-0020af0ba770". This is its GUID for
     BGRA/X 8:8:8:8.

     The UVC 1.5 spec [1] only defines GUIDs for YUY2, NV12, M420 and I420.
     This seems to be an extension documented in the Microsoft Windows Media
     Format SDK[2]. This Media Format SDK defines this GUID as corresponding
     to `MEDIASUBTYPE_RGB32`, which is confirmed by [4] as `MEDIASUBTYPE_ARGB32`
     has different GUID.

     Note that in my case, the FX3 UVC can output either channel order,
     BGR or RGB or any other mix for that matter. Since Linux commit
     1b8dc32286a1a ("[media] uvcvideo: Add GUID for BGR 8:8:8")
     defined a GUID for `MEDIASUBTYPE_RGB24` channel order as BGR, keep
     this change consistent and define `MEDIASUBTYPE_RGB32` as BGR as well.
     Document [3] also indicates the channel order is BGR.

     [1] https://www.usb.org/document-library/video-class-v15-document-set
     [2] https://learn.microsoft.com/en-us/windows/win32/wmformat/media-type-identifiers
     [3] https://learn.microsoft.com/en-us/windows/win32/directshow/uncompressed-rgb-video-
 subtypes
     [4] https://gix.github.io/media-types/

     Signed-off-by: Marek Vasut <marex@denx.de>
     Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
     Reviewed-by: Ricardo Ribalda <ricardo@ribalda.com>
     Signed-off-by: Michael Grzeschik <m.grzeschik@pengutronix.de>
     Link: https://lore.kernel.org/r/20230126231456.3402323-2-m.grzeschik@pengutronix.de
```

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

commit 859c551e8271ddf24e02db7efe575ad4f118fafd
Author: Harshit Mogalapalli <harshit.m.mogalapalli@oracle.com>
Date:   Thu Jan 26 07:36:09 2023 -0800

    iio: accel: mma9551_core: Prevent uninitialized variable in mma9551_read_config_word()

    [ Upstream commit 64a68158738ec8f520347144352f7a09bdb9e169 ]

    Smatch Warns:
    drivers/iio/accel/mma9551_core.c:299
            mma9551_read_config_word() error: uninitialized symbol 'v'.

    When (offset >= 1 << 12) is true mma9551_transfer() will return -EINVAL
    without 'v' being initialized, so check for the error and return.

    Note: No actual bug as caller checks the return value and does not
    use the parameter in the problem case.

    Signed-off-by: Harshit Mogalapalli <harshit.m.mogalapalli@oracle.com>
    Link: https://lore.kernel.org/r/20230126153610.3586243-1-
harshit.m.mogalapalli@oracle.com
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 25aed11d3b017b5afab72a8ebdbf3a7338d86618
Author: Harshit Mogalapalli <harshit.m.mogalapalli@oracle.com>
Date:   Thu Jan 26 07:21:46 2023 -0800

    iio: accel: mma9551_core: Prevent uninitialized variable in mma9551_read_status_word()

    [ Upstream commit e56d2c34ce9dc122b1a618172ec0e05e50adb9e9 ]

    Smatch Warns: drivers/iio/accel/mma9551_core.c:357
            mma9551_read_status_word() error: uninitialized symbol 'v'.

    When (offset >= 1 << 12) is true mma9551_transfer() will return -EINVAL
    without 'v' being initialized, so check for the error and return.

    Note: Not a bug as such because the caller checks return value and
    doesn't not use this parameter in the problem case.

    Signed-off-by: Harshit Mogalapalli <harshit.m.mogalapalli@oracle.com>
    Link: https://lore.kernel.org/r/20230126152147.3585874-1-
harshit.m.mogalapalli@oracle.com
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 93b686b054ecf55dcae9a6beab2cbef2e7cf6930
Author: Manivannan Sadhasivam <mani@kernel.org>
Date:   Wed Dec 28 21:47:02 2022 +0530

    bus: mhi: ep: Fix the debug message for MHI_PKT_TYPE_RESET_CHAN_CMD cmd

    [ Upstream commit 8e697fcfdb9809634e268058ca743369c216b7ac ]

    The debug log incorrectly mentions that STOP command is received instead of
    RESET command. Fix that.

    Signed-off-by: Manivannan Sadhasivam <manivannan.sadhasivam@linaro.org>
    Reviewed-by: Jeffrey Hugo <quic_jhugo@quicinc.com>
    Link: https://lore.kernel.org/r/20221228161704.255268-5-manivannan.sadhasivam@linaro.org
    Signed-off-by: Manivannan Sadhasivam <manivannan.sadhasivam@linaro.org>
```

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

commit fa86dcdd42e65b79e693d11fa3addbcfc69d0d53
Author: Yulong Zhang <yulong.zhang@metoak.net>
Date:   Tue Jan 17 10:51:47 2023 +0800

    tools/iio/iio_utils:fix memory leak

    [ Upstream commit f2edf0c819a4823cd6c288801ce737e8d4fcde06 ]

    1. fopen sysfs without fclose.
    2. asprintf filename without free.
    3. if asprintf return error,do not need to free the buffer.

    Signed-off-by: Yulong Zhang <yulong.zhang@metoak.net>
    Link: https://lore.kernel.org/r/20230117025147.69890-1-yulong.zhang@metoak.net
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e9ba991680a57c1cefab2d79ee4057594751c115
Author: Alexander Usyskin <alexander.usyskin@intel.com>
Date:   Mon Dec 12 23:49:33 2022 +0200

    mei: bus-fixup:upon error print return values of send and receive

    [ Upstream commit 4b8659e2c258e4fdac9ccdf06cc20c0677894ef9 ]

    For easier debugging, upon error, print also return values
    from __mei_cl_recv() and __mei_cl_send() functions.

    Signed-off-by: Alexander Usyskin <alexander.usyskin@intel.com>
    Signed-off-by: Tomas Winkler <tomas.winkler@intel.com>
    Link: https://lore.kernel.org/r/20221212214933.275434-1-tomas.winkler@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b71ff206707855ce73c04794c76f7b678b2d4f72
Author: Isaac True <isaac.true@canonical.com>
Date:   Wed Nov 30 11:55:30 2022 +0100

    serial: sc16is7xx: setup GPIO controller later in probe

    [ Upstream commit c8f71b49ee4d28930c4a6798d1969fa91dc4ef3e ]

    The GPIO controller component of the sc16is7xx driver is setup too
    early, which can result in a race condition where another device tries
    to utilise the GPIO lines before the sc16is7xx device has finished
    initialising.

    This issue manifests itself as an Oops when the GPIO lines are configured:

        Unable to handle kernel read from unreadable memory at virtual address
        ...
        pc : sc16is7xx_gpio_direction_output+0x68/0x108 [sc16is7xx]
        lr : sc16is7xx_gpio_direction_output+0x4c/0x108 [sc16is7xx]
        ...
        Call trace:
        sc16is7xx_gpio_direction_output+0x68/0x108 [sc16is7xx]
        gpiod_direction_output_raw_commit+0x64/0x318
        gpiod_direction_output+0xb0/0x170
        create_gpio_led+0xec/0x198
        gpio_led_probe+0x16c/0x4f0
        platform_drv_probe+0x5c/0xb0
        really_probe+0xe8/0x448
        driver_probe_device+0xe8/0x138
```

```
          __device_attach_driver+0x94/0x118
          bus_for_each_drv+0x8c/0xe0
          __device_attach+0x100/0x1b8
          device_initial_probe+0x28/0x38
          bus_probe_device+0xa4/0xb0
          deferred_probe_work_func+0x90/0xe0
          process_one_work+0x1c4/0x480
          worker_thread+0x54/0x430
          kthread+0x138/0x150
          ret_from_fork+0x10/0x1c

      This patch moves the setup of the GPIO controller functions to later in the
      probe function, ensuring the sc16is7xx device has already finished
      initialising by the time other devices try to make use of the GPIO lines.
      The error handling has also been reordered to reflect the new
      initialisation order.

      Co-developed-by: Wen-chien Jesse Sung <jesse.sung@canonical.com>
      Signed-off-by: Wen-chien Jesse Sung <jesse.sung@canonical.com>
      Signed-off-by: Isaac True <isaac.true@canonical.com>
      Link: https://lore.kernel.org/r/20221130105529.698385-1-isaac.true@canonical.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 5c0f48eb59c12632d6ac4e9ebebe928bd1ad3f1d
  Author: Sherry Sun <sherry.sun@nxp.com>
  Date:   Wed Dec 14 11:11:35 2022 +0800

      tty: serial: fsl_lpuart: disable the CTS when send break signal

      [ Upstream commit c4c81db5cf8bc53d6160c3abf26d382c841aa434 ]

      LPUART IP has a bug that it treats the CTS as higher priority than the
      break signal, which cause the break signal sending through UARTCTRL_SBK
      may impacted by the CTS input if the HW flow control is enabled.

      Add this workaround patch to fix the IP bug, we can disable CTS before
      asserting SBK to avoid any interference from CTS, and re-enable it when
      break off.

      Such as for the bluetooth chip power save feature, host can let the BT
      chip get into sleep state by sending a UART break signal, and wake it up
      by turning off the UART break. If the BT chip enters the sleep mode
      successfully, it will pull up the CTS line, if the BT chip is woken up,
      it will pull down the CTS line. If without this workaround patch, the
      UART TX pin cannot send the break signal successfully as it affected by
      the BT CTS pin. After adding this patch, the BT power save feature can
      work well.

      Signed-off-by: Sherry Sun <sherry.sun@nxp.com>
      Link: https://lore.kernel.org/r/20221214031137.28815-2-sherry.sun@nxp.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit fcfeaa570f7a5c2d5f4f14931909531ff18b7fde
  Author: Sven Schnelle <svens@linux.ibm.com>
  Date:   Fri Dec 9 12:27:36 2022 +0100

      tty: fix out-of-bounds access in tty_driver_lookup_tty()

      [ Upstream commit db4df8e9d79e7d37732c1a1b560958e8dadfefa1 ]

      When specifying an invalid console= device like console=tty3270,
      tty_driver_lookup_tty() returns the tty struct without checking
      whether index is a valid number.
```

```
    To reproduce:

    qemu-system-x86_64 -enable-kvm -nographic -serial mon:stdio \
    -kernel ../linux-build-x86/arch/x86/boot/bzImage \
    -append "console=ttyS0 console=tty3270"

    This crashes with:

    [    0.770599] BUG: kernel NULL pointer dereference, address: 00000000000000ef
    [    0.771265] #PF: supervisor read access in kernel mode
    [    0.771773] #PF: error_code(0x0000) - not-present page
    [    0.772609] Oops: 0000 [#1] PREEMPT SMP PTI
    [    0.774878] RIP: 0010:tty_open+0x268/0x6f0
    [    0.784013]  chrdev_open+0xbd/0x230
    [    0.784444]  ? cdev_device_add+0x80/0x80
    [    0.784920]  do_dentry_open+0x1e0/0x410
    [    0.785389]  path_openat+0xca9/0x1050
    [    0.785813]  do_filp_open+0xaa/0x150
    [    0.786240]  file_open_name+0x133/0x1b0
    [    0.786746]  filp_open+0x27/0x50
    [    0.787244]  console_on_rootfs+0x14/0x4d
    [    0.787800]  kernel_init_freeable+0x1e4/0x20d
    [    0.788383]  ? rest_init+0xc0/0xc0
    [    0.788881]  kernel_init+0x11/0x120
    [    0.789356]  ret_from_fork+0x22/0x30

    Signed-off-by: Sven Schnelle <svens@linux.ibm.com>
    Reviewed-by: Jiri Slaby <jirislaby@kernel.org>
    Link: https://lore.kernel.org/r/20221209112737.3222509-2-svens@linux.ibm.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e8a0c4ee3e52dd16183f015a32ca201f5b473a7
Author: Yuan Can <yuancan@huawei.com>
Date:   Thu Jan 19 08:31:19 2023 +0000

    staging: emxx_udc: Add checks for dma_alloc_coherent()

    [ Upstream commit f6510a93cfd8c6c79b4dda0f2967cdc6df42eff4 ]

    As the dma_alloc_coherent may return NULL, the return value needs to be
    checked to avoid NULL poineter dereference.

    Signed-off-by: Yuan Can <yuancan@huawei.com>
    Reviewed-by: Simon Horman <horms@verge.net.au>
    Link: https://lore.kernel.org/r/20230119083119.16956-1-yuancan@huawei.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 39a8d416559c69fd22f53ac42b5124d7fa1078ec
Author: Anand Moon <linux.amoon@gmail.com>
Date:   Wed Jan 18 04:44:09 2023 +0000

    dt-bindings: usb: Add device id for Genesys Logic hub controller

    [ Upstream commit b72654148e34c181f532275d03ef6f37de288f24 ]

    Add usb hub device id for Genesys Logic, Inc. GL852G Hub USB 2.0
    root hub.

    Signed-off-by: Anand Moon <linux.amoon@gmail.com>
    Acked-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Link: https://lore.kernel.org/r/20230118044418.875-2-linux.amoon@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

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

commit 381c672990aee1b0728a258e7fb3df2caff30bbf
Author: Linus Walleij <linus.walleij@linaro.org>
Date:   Wed Jan 18 08:09:16 2023 +0100

    usb: fotg210: List different variants

    [ Upstream commit 170da81aab077c9e85fc2b786413ca07942774a0 ]

    There are at least two variants of the FOTG: FOTG200 and
    FOTG210. Handle them in this driver and let's add
    more quirks as we go along.

    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Link: https://lore.kernel.org/r/20230103-gemini-fotg210-usb-v2-2-100388af9810@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dea49f2993f57d8a2df2cacb0bf649ef49b28879
Author: Yong-Xuan Wang <yongxuan.wang@sifive.com>
Date:   Tue Jan 17 10:51:33 2023 +0000

    cacheinfo: Fix shared_cpu_map to handle shared caches at different levels

    [ Upstream commit 198102c9103fc78d8478495971947af77edb05c1 ]

    The cacheinfo sets up the shared_cpu_map by checking whether the caches
    with the same index are shared between CPUs. However, this will trigger
    slab-out-of-bounds access if the CPUs do not have the same cache hierarchy.
    Another problem is the mismatched shared_cpu_map when the shared cache does
    not have the same index between CPUs.

    CPU0    I       D       L3
    index   0       1       2       x
            ^       ^       ^       ^
    index   0       1       2       3
    CPU1    I       D       L2      L3

    This patch checks each cache is shared with all caches on other CPUs.

    Reviewed-by: Pierre Gondois <pierre.gondois@arm.com>
    Signed-off-by: Yong-Xuan Wang <yongxuan.wang@sifive.com>
    Link: https://lore.kernel.org/r/20230117105133.4445-2-yongxuan.wang@sifive.com
    Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cc00340fb1226a2a3a5cf15473ac417da3c952f1
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Jan 6 16:28:28 2023 +0100

    USB: fix memory leak with using debugfs_lookup()

    [ Upstream commit 30374434edab20e25776f8ecb4bc9d1e54309487 ]

    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic at
    once.

    Cc: Alan Stern <stern@rowland.harvard.edu>
    Cc: Jilin Yuan <yuanjilin@cdjrlc.com>
    Link: https://lore.kernel.org/r/20230106152828.3790902-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 0b962b034551b93fe97f55a199e677d702dddd3b
Author: Kees Cook <keescook@chromium.org>
Date:   Thu Jan 5 22:17:04 2023 -0800

    media: uvcvideo: Silence memcpy() run-time false positive warnings

    [ Upstream commit b839212988575c701aab4d3d9ca15e44c87e383c ]

    The memcpy() in uvc_video_decode_meta() intentionally copies across the
    length and flags members and into the trailing buf flexible array.
    Split the copy so that the compiler can better reason about (the lack
    of) buffer overflows here. Avoid the run-time false positive warning:

      memcpy: detected field-spanning write (size 12) of single field "&meta->length" at
drivers/media/usb/uvc/uvc_video.c:1355 (size 1)

    Additionally fix a typo in the documentation for struct uvc_meta_buf.

    Reported-by: ionut_n2001@yahoo.com
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216810
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa50ff54f13381ab45bf611f80dee4a5696b0264
Author: Ricardo Ribalda <ribalda@chromium.org>
Date:   Wed Jan 4 11:45:23 2023 +0100

    media: uvcvideo: Quirk for autosuspend in Logitech B910 and C910

    [ Upstream commit 136effa754b57632f99574fc4a3433e0cfc031d9 ]

    Logitech B910 and C910 firmware are unable to recover from a USB
    autosuspend. When it resumes, the device is in a state where it only
    produces invalid frames. Eg:

    $ echo 0xFFFF > /sys/module/uvcvideo/parameters/trace # enable verbose log
    $ yavta -c1 -n1 --file='frame#.jpg' --format MJPEG --size=1920x1080 /dev/video1
    [350438.435219] uvcvideo: uvc_v4l2_open
    [350438.529794] uvcvideo: Resuming interface 2
    [350438.529801] uvcvideo: Resuming interface 3
    [350438.529991] uvcvideo: Trying format 0x47504a4d (MJPG): 1920x1080.
    [350438.529996] uvcvideo: Using default frame interval 33333.3 us (30.0 fps).
    [350438.551496] uvcvideo: uvc_v4l2_mmap
    [350438.555890] uvcvideo: Device requested 3060 B/frame bandwidth.
    [350438.555896] uvcvideo: Selecting alternate setting 11 (3060 B/frame bandwidth).
    [350438.556362] uvcvideo: Allocated 5 URB buffers of 32x3060 bytes each.
    [350439.316468] uvcvideo: Marking buffer as bad (error bit set).
    [350439.316475] uvcvideo: Frame complete (EOF found).
    [350439.316477] uvcvideo: EOF in empty payload.
    [350439.316484] uvcvideo: frame 1 stats: 149/261/417 packets, 1/149/417 pts (early
initial), 416/417 scr, last pts/stc/sof 2976325734/2978107243/249
    [350439.384510] uvcvideo: Marking buffer as bad (error bit set).
    [350439.384516] uvcvideo: Frame complete (EOF found).
    [350439.384518] uvcvideo: EOF in empty payload.
    [350439.384525] uvcvideo: frame 2 stats: 265/379/533 packets, 1/265/533 pts (early
initial), 532/533 scr, last pts/stc/sof 2979524454/2981305193/316
    [350439.448472] uvcvideo: Marking buffer as bad (error bit set).
    [350439.448478] uvcvideo: Frame complete (EOF found).
    [350439.448480] uvcvideo: EOF in empty payload.
    [350439.448487] uvcvideo: frame 3 stats: 265/377/533 packets, 1/265/533 pts (early
initial), 532/533 scr, last pts/stc/sof 2982723174/2984503144/382
    ...(loop)...
```

```
    The devices can leave this invalid state if the alternate setting of
    the streaming interface is toggled.

    This patch adds a quirk for this device so it can be autosuspended
    properly.

    lsusb -v:
    Bus 001 Device 049: ID 046d:0821 Logitech, Inc. HD Webcam C910
    Device Descriptor:
      bLength                18
      bDescriptorType         1
      bcdUSB               2.00
      bDeviceClass          239 Miscellaneous Device
      bDeviceSubClass         2
      bDeviceProtocol         1 Interface Association
      bMaxPacketSize0        64
      idVendor           0x046d Logitech, Inc.
      idProduct          0x0821 HD Webcam C910
      bcdDevice            0.10
      iManufacturer           0
      iProduct                0
      iSerial                 1 390022B0
      bNumConfigurations      1

    Signed-off-by: Ricardo Ribalda <ribalda@chromium.org>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e1c6b92c2399b154485db05b631d5268d9bbb04e
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Tue Oct 25 16:41:01 2022 +0200

    media: uvcvideo: Handle errors from calls to usb_string

    [ Upstream commit 4867bb590ae445bcfaa711a86b603c97e94574b3 ]

    On a Webcam from Quanta, we see the following error.

    usb 3-5: New USB device found, idVendor=0408, idProduct=30d2, bcdDevice= 0.03
    usb 3-5: New USB device strings: Mfr=3, Product=1, SerialNumber=2
    usb 3-5: Product: USB2.0 HD UVC WebCam
    usb 3-5: Manufacturer: Quanta
    usb 3-5: SerialNumber: 0x0001
    ...
    uvcvideo: Found UVC 1.10 device USB2.0 HD UVC WebCam (0408:30d2)
    uvcvideo: Failed to initialize entity for entity 5
    uvcvideo: Failed to register entities (-22).

    The Webcam reports an entity of type UVC_VC_EXTENSION_UNIT. It reports a
    string index of '7' associated with that entity. The attempt to read that
    string from the camera fails with error -32 (-EPIPE). usb_string() returns
    that error, but it is ignored. As result, the entity name is empty. This
    later causes v4l2_device_register_subdev() to return -EINVAL, and no
    entities are registered as result.

    While this appears to be a firmware problem with the camera, the kernel
    should still handle the situation gracefully. To do that, check the return
    value from usb_string(). If it reports an error, assign the entity's
    default name.

    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
```

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

commit 1a76cfc388cf105d3e04ac592670a52a3864b1ba
Author: Ricardo Ribalda <ribalda@chromium.org>
Date:   Tue Sep 20 16:04:55 2022 +0200

    media: uvcvideo: Handle cameras with invalid descriptors

    [ Upstream commit 41ddb251c68ac75c101d3a50a68c4629c9055e4c ]

    If the source entity does not contain any pads, do not create a link.

    Reported-by: syzbot <syzkaller@googlegroups.com>
    Signed-off-by: Ricardo Ribalda <ribalda@chromium.org>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 90374683dd46d57108e57d52f983bdfb7d789dd2
Author: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
Date:   Tue Nov 15 18:44:29 2016 +0200

    media: uvcvideo: Remove format descriptions

    [ Upstream commit 50459f103edfe47c9a599d766a850ef6014936c5 ]

    The V4L2 core overwrites format descriptions in v4l_fill_fmtdesc(),
    there's no need to manually set the descriptions in the driver. This
    prepares for removal of the format descriptions from the uvc_fmts table.

    Unlike V4L2, UVC makes a distinction between the SD-DV, SDL-DV and HD-DV
    formats. It also indicates whether the DV format uses 50Hz or 60Hz. This
    information is parsed by the driver to construct a format name string
    that is printed in a debug message, but serves no other purpose as V4L2
    has a single V4L2_PIX_FMT_DV pixel format that covers all those cases.

    As the information is available in the UVC descriptors, and thus
    accessible to users with lsusb if they really care, don't log it in a
    debug message and drop the format name string to simplify the code.

    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Reviewed-by: Ricardo Ribalda <ribalda@chromium.org>
    Reviewed-by: Michael Grzeschik <m.grzeschik@pengutronix.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 131d91b5d34ef3a79e2d5e31d73844edf206dfad
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Thu Jan 12 21:51:24 2023 +0100

    PCI/ACPI: Account for _S0W of the target bridge in acpi_pci_bridge_d3()

    [ Upstream commit 8133844a8f2434be9576850c6978179d7cca5c81 ]

    It is questionable to allow a PCI bridge to go into D3 if it has _S0W
    returning D2 or a shallower power state, so modify acpi_pci_bridge_d3(() to
    always take the return value of _S0W for the target bridge into account.
    That is, make it return 'false' if _S0W returns D2 or a shallower power
    state for the target bridge regardless of its ancestor Root Port
    properties.  Of course, this also causes 'false' to be returned if the Root
    Port itself is the target and its _S0W returns D2 or a shallower power
    state.

    However, still allow bridges without _S0W that are power-manageable via
    ACPI to enter D3 to retain the current code behavior in that case.
```

```
    This fixes problems where a hotplug notification is missed because a bridge
    is in D3.  That means hot-added devices such as USB4 docks (and the devices
    they contain) and Thunderbolt 3 devices may not work.

    Link: https://lore.kernel.org/linux-pci/20221031223356.32570-1-
mario.limonciello@amd.com/
    Link: https://lore.kernel.org/r/12155458.O9o76ZdvQC@kreacher
    Reported-by: Mario Limonciello <mario.limonciello@amd.com>
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c8ca988b23d31ec3c8e8d9a4507d84b04d6bf5e2
Author: Jason Gunthorpe <jgg@ziepe.ca>
Date:   Tue Jan 10 10:54:07 2023 +0800

    iommu: Remove deferred attach check from __iommu_detach_device()

    [ Upstream commit dd8a25c557e109f868430bd2e3e8f394cb40eaa7 ]

    At the current moment, __iommu_detach_device() is only called via call
    chains that are after the device driver is attached - eg via explicit
    attach APIs called by the device driver.

    Commit bd421264ed30 ("iommu: Fix deferred domain attachment") has removed
    deferred domain attachment check from __iommu_attach_device() path, so it
    should just unconditionally work in the __iommu_detach_device() path.

    It actually looks like a bug that we were blocking detach on these paths
    since the attach was unconditional and the caller is going to free the
    (probably) UNAMANGED domain once this returns.

    The only place we should be testing for deferred attach is during the
    initial point the dma device is linked to the group, and then again
    during the dma api calls.

    Signed-off-by: Jason Gunthorpe <jgg@nvidia.com>
    Signed-off-by: Lu Baolu <baolu.lu@linux.intel.com>
    Link: https://lore.kernel.org/r/20230110025408.667767-5-baolu.lu@linux.intel.com
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e4c579d0c48ac7e5901ff32d3f8832fd523af8b5
Author: Dean Luick <dean.luick@cornelisnetworks.com>
Date:   Mon Jan 9 14:04:29 2023 -0500

    IB/hfi1: Update RMT size calculation

    [ Upstream commit 892ede5a77f337831609fb9c248ac60948061894 ]

    Fix possible RMT overflow:  Use the correct netdev size.
    Don't allow adjusted user contexts to go negative.

    Fix QOS calculation: Send kernel context count as an argument since
    dd->n_krcv_queues is not yet set up in earliest call.  Do not include
    the control context in the QOS calculation.  Use the same sized
    variable to find the max of krcvq[] entries.

    Update the RMT count explanation to make more sense.

    Signed-off-by: Dean Luick <dean.luick@cornelisnetworks.com>
    Signed-off-by: Dennis Dalessandro <dennis.dalessandro@cornelisnetworks.com>
    Link: https://lore.kernel.org/r/167329106946.1472990.18385495251650939054.stgit@awfm-
02.cornelisnetworks.com
    Signed-off-by: Leon Romanovsky <leon@kernel.org>
```

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

commit dc9437e9889c3dacf1f320e3cf08da74127573fe
Author: Liang He <windhl@126.com>
Date:   Thu Jan 5 14:10:55 2023 +0800

    mfd: arizona: Use pm_runtime_resume_and_get() to prevent refcnt leak

    [ Upstream commit 4414a7ab80cebf715045e3c4d465feefbad21139 ]

    In arizona_clk32k_enable(), we should use pm_runtime_resume_and_get()
    as pm_runtime_get_sync() will increase the refcnt even when it
    returns an error.

    Signed-off-by: Liang He <windhl@126.com>
    Acked-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Signed-off-by: Lee Jones <lee@kernel.org>
    Link: https://lore.kernel.org/r/20230105061055.1509261-1-windhl@126.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c967e4e8d5e22270cd0a7478af9908aaa3d7dfed
Author: Souradeep Chowdhury <quic_schowdhu@quicinc.com>
Date:   Wed Feb 22 08:27:49 2023 +0900

    bootconfig: Increase max nodes of bootconfig from 1024 to 8192 for DCC support

    [ Upstream commit 6c40624930c58529185a257380442547580ed837 ]

    The Data Capture and Compare(DCC) is a debugging tool that uses the bootconfig
    for configuring the register values during boot-time. Increase the max nodes
    supported by bootconfig to cater to the requirements of the Data Capture and
    Compare Driver.

    Link: https://lore.kernel.org/all/1674536682-18404-1-git-send-email-
quic_schowdhu@quicinc.com/

    Signed-off-by: Souradeep Chowdhury <quic_schowdhu@quicinc.com>
    Acked-by: Masami Hiramatsu (Google) <mhiramat@kernel.org>
    Signed-off-by: Masami Hiramatsu (Google) <mhiramat@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 953abd8271a59396f9df184835c46982885c1e27
Author: Darrell Kavanagh <darrell.kavanagh@gmail.com>
Date:   Wed Feb 15 11:50:45 2023 +0000

    firmware/efi sysfb_efi: Add quirk for Lenovo IdeaPad Duet 3

    [ Upstream commit e1d447157f232c650e6f32c9fb89ff3d0207c69a ]

    Another Lenovo convertable which reports a landscape resolution of
    1920x1200 with a pitch of (1920 * 4) bytes, while the actual framebuffer
    has a resolution of 1200x1920 with a pitch of (1200 * 4) bytes.

    Signed-off-by: Darrell Kavanagh <darrell.kavanagh@gmail.com>
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Ard Biesheuvel <ardb@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 13969236b6900b5a3625ad2193569588e978f1cc
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 2 16:14:11 2023 +0100

    kernel/printk/index.c: fix memory leak with using debugfs_lookup()

    [ Upstream commit 55bf243c514553e907efcf2bda92ba090eca8c64 ]
```

```
    When calling debugfs_lookup() the result must have dput() called on it,
    otherwise the memory will leak over time.  To make things simpler, just
    call debugfs_lookup_and_remove() instead which handles all of the logic
    at once.

    Cc: Chris Down <chris@chrisdown.name>
    Cc: Petr Mladek <pmladek@suse.com>
    Cc: Sergey Senozhatsky <senozhatsky@chromium.org>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: John Ogness <john.ogness@linutronix.de>
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Reviewed-by: Sergey Senozhatsky <senozhatsky@chromium.org>
    Reviewed-by: John Ogness <john.ogness@linutronix.de>
    Reviewed-by: Petr Mladek <pmladek@suse.com>
    Signed-off-by: Petr Mladek <pmladek@suse.com>
    Link: https://lore.kernel.org/r/20230202151411.2308576-1-gregkh@linuxfoundation.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e54bb56a01c29f6862d99488e72bf64deeb5afbc
Author: Jia-Ju Bai <baijiaju1990@gmail.com>
Date:   Fri Jan 13 20:55:01 2023 +0800

    tracing: Add NULL checks for buffer in ring_buffer_free_read_page()

    [ Upstream commit 3e4272b9954094907f16861199728f14002fcaf6 ]

    In a previous commit 7433632c9ff6, buffer, buffer->buffers and
    buffer->buffers[cpu] in ring_buffer_wake_waiters() can be NULL,
    and thus the related checks are added.

    However, in the same call stack, these variables are also used in
    ring_buffer_free_read_page():

    tracing_buffers_release()
      ring_buffer_wake_waiters(iter->array_buffer->buffer)
        cpu_buffer = buffer->buffers[cpu] -> Add checks by previous commit
      ring_buffer_free_read_page(iter->array_buffer->buffer)
        cpu_buffer = buffer->buffers[cpu] -> No check

    Thus, to avod possible null-pointer derefernces, the related checks
    should be added.

    These results are reported by a static tool designed by myself.

    Link: https://lkml.kernel.org/r/20230113125501.760324-1-baijiaju1990@gmail.com

    Reported-by: TOTE Robot <oslab@tsinghua.edu.cn>
    Signed-off-by: Jia-Ju Bai <baijiaju1990@gmail.com>
    Signed-off-by: Steven Rostedt (Google) <rostedt@goodmis.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6d2128d2ee60e8f7977b893a9cd248e2280f2608
Author: Dan Carpenter <error27@gmail.com>
Date:   Mon Feb 27 13:07:09 2023 +0300

    cpufreq: apple-soc: Fix an IS_ERR() vs NULL check

    [ Upstream commit f43523620f646c89ffd8ada840a0068290e51266 ]

    The of_iomap() function returns NULL if it fails.  It never returns
    error pointers.  Fix the check accordingly.

    Fixes: 6286bbb40576 ("cpufreq: apple-soc: Add new driver to control Apple SoC CPU P-
```

```
 states")
     Signed-off-by: Dan Carpenter <error27@gmail.com>
     Reviewed-by: Eric Curtin <ecurtin@redhat.com>
     Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 014b4e1c4d096011dd776a87e1b022ac01a144bd
 Author: Randy Dunlap <rdunlap@infradead.org>
 Date:   Sat Feb 25 21:39:52 2023 -0800

     thermal: intel: BXT_PMIC: select REGMAP instead of depending on it

     [ Upstream commit 1467fb960349dfa5e300658f1a409dde2cfb0c51 ]

     REGMAP is a hidden (not user visible) symbol. Users cannot set it
     directly thru "make *config", so drivers should select it instead of
     depending on it if they need it.

     Consistently using "select" or "depends on" can also help reduce
     Kconfig circular dependency issues.

     Therefore, change the use of "depends on REGMAP" to "select REGMAP".

     Fixes: b474303ffd57 ("thermal: add Intel BXT WhiskeyCove PMIC thermal driver")
     Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
     Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 24c221b11c2894e1a5f07b93362d9bc91c6d8be7
 Author: Dan Carpenter <error27@gmail.com>
 Date:   Mon Feb 27 13:06:50 2023 +0300

     thermal: intel: quark_dts: fix error pointer dereference

     [ Upstream commit f1b930e740811d416de4d2074da48b6633a672c8 ]

     If alloc_soc_dts() fails, then we can just return.  Trying to free
     "soc_dts" will lead to an Oops.

     Fixes: 8c1876939663 ("thermal: intel Quark SoC X1000 DTS thermal driver")
     Signed-off-by: Dan Carpenter <error27@gmail.com>
     Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 6f8490a74da5584a4f8abbf9c5d3552165cc5a88
 Author: Trevor Wu <trevor.wu@mediatek.com>
 Date:   Wed Mar 1 19:02:00 2023 +0800

     ASoC: mediatek: mt8195: add missing initialization

     [ Upstream commit b56ec2992a2e43bc3e60d6db86849d31640e791f ]

     In etdm dai driver, dai_etdm_parse_of() function is used to parse dts
     properties to get parameters. There are two for-loops which are
     sepearately for all etdm and etdm input only cases. In etdm in only
     loop, dai_id is not initialized, so it keeps the value intiliazed in
     another loop.

     In the patch, add the missing initialization to fix the unexpected
     parsing problem.

     Fixes: 1de9a54acafb ("ASoC: mediatek: mt8195: support etdm in platform driver")
     Signed-off-by: Trevor Wu <trevor.wu@mediatek.com>
     Reviewed-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
     Link: https://lore.kernel.org/r/20230301110200.26177-3-trevor.wu@mediatek.com
```

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

commit 4eeb622546d1b36a652d402fc834e05f637f2e55
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Feb 27 09:58:26 2023 +0100

    ASoC: zl38060 add gpiolib dependency

    [ Upstream commit 0de2cc3707b6b6e2ad40bd24ce09a5c1f65d01e1 ]

    Without gpiolib, this driver fails to link:

    arm-linux-gnueabi-ld: sound/soc/codecs/zl38060.o: in function `chip_gpio_get':
    zl38060.c:(.text+0x30): undefined reference to `gpiochip_get_data'
    arm-linux-gnueabi-ld: sound/soc/codecs/zl38060.o: in function `zl38_spi_probe':
    zl38060.c:(.text+0xa18): undefined reference to `devm_gpiochip_add_data_with_key'

    This appears to have been in the driver since the start, but is hard to
    hit in randconfig testing since gpiolib is almost always selected by something
    else.

    Fixes: 52e8a94baf90 ("ASoC: Add initial ZL38060 driver")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Link: https://lore.kernel.org/r/20230227085850.2503725-1-arnd@kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 38427ae36fe861a2ebcd2978d6fc126e2d98ba3f
Author: Daniel Wagner <dwagner@suse.de>
Date:   Tue Feb 21 17:51:06 2023 +0100

    nvme-fabrics: show well known discovery name

    [ Upstream commit 26a57cb35548ae67c14871cccbf50da3edb01ea4 ]

    The kernel always logs the unique subsystem name for a discovery
    controller, even in the case user space asked for the well known.

    This has lead to confusion as the logs of nvme-cli and the kernel
    logs didn't match.

    First, nvme-cli connects to the well known discovery controller to
    figure out if it supports TP8013. If so then nvme-cli disconnects and
    connects to the unique discovery controller. Currently, the kernel show
    that user space connected twice to the unique one.

    To avoid further confusion, show the well known discovery controller if
    user space asked for it:

      $ nvme connect-all -v -t tcp -a 192.168.0.1
      nvme0: nqn.2014-08.org.nvmexpress.discovery connected
      nvme0: nqn.2014-08.org.nvmexpress.discovery disconnected
      nvme0: nqn.discovery connected

      kernel log:
      nvme nvme0: new ctrl: NQN "nqn.2014-08.org.nvmexpress.discovery", addr
192.168.0.1:8009
      nvme nvme0: Removing ctrl: NQN "nqn.2014-08.org.nvmexpress.discovery"
      nvme nvme0: new ctrl: NQN "nqn.discovery", addr 192.168.0.1:8009

    Fixes: e5ea42faa773 ("nvme: display correct subsystem NQN")
    Signed-off-by: Daniel Wagner <dwagner@suse.de>
    Reviewed-by: Hannes Reinecke <hare@suse.de>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
```

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

commit d82f762db4776fa11de88018f0f5de2d5db72a72
Author: Akinobu Mita <akinobu.mita@gmail.com>
Date:   Sun Feb 26 21:42:54 2023 +0900

    nvme-tcp: don't access released socket during error recovery

    [ Upstream commit 76d54bf20cdcc1ed7569a89885e09636e9a8d71d ]

    While the error recovery work is temporarily failing reconnect attempts,
    running the 'nvme list' command causes a kernel NULL pointer dereference
    by calling getsockname() with a released socket.

    During error recovery work, the nvme tcp socket is released and a new one
    created, so it is not safe to access the socket without proper check.

    Signed-off-by: Akinobu Mita <akinobu.mita@gmail.com>
    Fixes: 02c57a82c008 ("nvme-tcp: print actual source IP address through sysfs "address"
attr")
    Reviewed-by: Martin Belanger <martin.belanger@dell.com>
    Reviewed-by: Hannes Reinecke <hare@suse.de>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 571047a40c28e43cac5ab99c93110e86bf5047f1
Author: Christoph Hellwig <hch@lst.de>
Date:   Tue Feb 21 14:02:25 2023 -0800

    nvme: bring back auto-removal of deleted namespaces during sequential scan

    [ Upstream commit 0dd6fff2aad4e35633fef1ea72838bec5b47559a ]

    Bring back the check of the Identify Namespace return value for the
    legacy NVMe 1.0-style sequential scanning.  While NVMe 1.0 does not
    support namespace management, there are "modern" cloud solutions like
    Google Cloud Platform that claim the obsolete 1.0 compliance for no
    good reason while supporting proprietary sideband namespace management.

    Fixes: 1a893c2bfef4 ("nvme: refactor namespace probing")
    Reported-by: Nils Hanke <nh@edgeless.systems>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Keith Busch <kbusch@kernel.org>
    Reviewed-by: Sagi Grimberg <sagi@grimberg.me>
    Tested-by: Nils Hanke <nh@edgeless.systems>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit acb1ad743e136c262e0d0d6e2fde866d8e395b09
Author: Martin Povišer <povik+lin@cutebit.org>
Date:   Fri Feb 24 16:33:02 2023 +0100

    ASoC: apple: mca: Improve handling of unavailable DMA channels

    [ Upstream commit fb1847cc460c127b12720119eae5f438ffc62e85 ]

    When we fail to obtain a DMA channel, don't return a blanket -EINVAL,
    instead return the original error code if there's one. This makes
    deferring work as it should. Also don't print an error message for
    -EPROBE_DEFER.

    Fixes: 4ec8179c212f ("ASoC: apple: mca: Postpone requesting of DMA channels")
    Signed-off-by: Martin Povišer <povik+lin@cutebit.org>
    Link: https://lore.kernel.org/r/20230224153302.45365-3-povik+lin@cutebit.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 0f18cd1d77e2e57fbba531169e997bedbf1f2d94
Author: Martin Povišer <povik+lin@cutebit.org>
Date:   Fri Feb 24 16:33:01 2023 +0100

    ASoC: apple: mca: Fix SERDES reset sequence

    [ Upstream commit d8b3e396088d787771f19fd3b7949e080dc31d6f ]

    Fix the reset sequence of reads and writes that we invoke from within
    the early trigger. It looks like there never was a SERDES_CONF_SOME_RST
    bit that should be involved in the reset sequence, and its presence in
    the driver code is a mistake from earlier.

    Instead, the reset sequence should go as follows: We should switch the
    the SERDES unit's SYNC_SEL mux to the value of 7 (so outside the range
    of 1...6 representing cluster's SYNCGEN units), then raise the RST bit
    in SERDES_STATUS and wait for it to clear.

    Properly resetting the SERDES unit fixes frame desynchronization hazard
    in case of long frames (longer than 4 used slots). The desynchronization
    manifests itself by rotating the PCM channels.

    Fixes: 3df5d0d97289 ("ASoC: apple: mca: Start new platform driver")
    Signed-off-by: Martin Povišer <povik+lin@cutebit.org>
    Link: https://lore.kernel.org/r/20230224153302.45365-2-povik+lin@cutebit.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 255fe434092fc9ed9f770e81b6b5e9ae63ed0aff
Author: Martin Povišer <povik+lin@cutebit.org>
Date:   Fri Feb 24 16:33:00 2023 +0100

    ASoC: apple: mca: Fix final status read on SERDES reset

    [ Upstream commit aaf5f0d76b6e1870e3674408de2b13a92a4d4059 ]

    From within the early trigger we are doing a reset of the SERDES unit,
    but the final status read is on a bad address. Add the missing SERDES
    unit offset in calculation of the address.

    Fixes: 3df5d0d97289 ("ASoC: apple: mca: Start new platform driver")
    Signed-off-by: Martin Povišer <povik+lin@cutebit.org>
    Link: https://lore.kernel.org/r/20230224153302.45365-1-povik+lin@cutebit.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 756d0c7bfabe23dbfb2e0f10c7431217d9606786
Author: Nuno Sá <nuno.sa@analog.com>
Date:   Fri Feb 24 11:45:51 2023 +0100

    ASoC: adau7118: don't disable regulators on device unbind

    [ Upstream commit b5bfa7277ee7d944421e0ef193586c6e34d7492c ]

    The regulators are supposed to be controlled through the
    set_bias_level() component callback. Moreover, the regulators are not
    enabled during probe and so, this would lead to a regulator unbalanced
    use count.

    Fixes: ca514c0f12b02 ("ASOC: Add ADAU7118 8 Channel PDM-to-I2S/TDM Converter driver")
    Signed-off-by: Nuno Sá <nuno.sa@analog.com>
    Link: https://lore.kernel.org/r/20230224104551.1139981-1-nuno.sa@analog.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 258809bf22bf71d53247856f374f2b1d055f2fd4
Author: Zhong Jinghua <zhongjinghua@huawei.com>
Date:   Tue Feb 21 17:50:27 2023 +0800

    loop: loop_set_status_from_info() check before assignment

    [ Upstream commit 9f6ad5d533d1c71e51bdd06a5712c4fbc8768dfa ]

    In loop_set_status_from_info(), lo->lo_offset and lo->lo_sizelimit should
    be checked before reassignment, because if an overflow error occurs, the
    original correct value will be changed to the wrong value, and it will not
    be changed back.

    More, the original patch did not solve the problem, the value was set and
    ioctl returned an error, but the subsequent io used the value in the loop
    driver, which still caused an alarm:

    loop_handle_cmd
     do_req_filebacked
      loff_t pos = ((loff_t) blk_rq_pos(rq) << 9) + lo->lo_offset;
      lo_rw_aio
       cmd->iocb.ki_pos = pos

    Fixes: c490a0b5a4f3 ("loop: Check for overflow while configuring loop")
    Signed-off-by: Zhong Jinghua <zhongjinghua@huawei.com>
    Reviewed-by: Chaitanya Kulkarni <kch@nvidia.com>
    Link: https://lore.kernel.org/r/20230221095027.3656193-1-zhongjinghua@huaweicloud.com
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94f7ac92d1f054ce1a998677012f138c875ddcb0
Author: Wojciech Lukowicz <wlukowicz01@gmail.com>
Date:   Sat Feb 18 18:41:41 2023 +0000

    io_uring: fix size calculation when registering buf ring

    [ Upstream commit 48ba08374e779421ca34bd14b4834aae19fc3e6a ]

    Using struct_size() to calculate the size of io_uring_buf_ring will sum
    the size of the struct and of the bufs array. However, the struct's fields
    are overlaid with the array making the calculated size larger than it
    should be.

    When registering a ring with N * PAGE_SIZE / sizeof(struct io_uring_buf)
    entries, i.e. with fully filled pages, the calculated size will span one
    more page than it should and io_uring will try to pin the following page.
    Depending on how the application allocated the ring, it might succeed
    using an unrelated page or fail returning EFAULT.

    The size of the ring should be the product of ring_entries and the size
    of io_uring_buf, i.e. the size of the bufs array only.

    Fixes: c7fb19428d67 ("io_uring: add support for ring mapped supplied buffers")
    Signed-off-by: Wojciech Lukowicz <wlukowicz01@gmail.com>
    Reviewed-by: Gabriel Krisman Bertazi <krisman@suse.de>
    Link: https://lore.kernel.org/r/20230218184141.70891-1-wlukowicz01@gmail.com
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2091b980f13ee0d17155e11ccfdc53794dd4525d
Author: Alexandre Belloni <alexandre.belloni@bootlin.com>
Date:   Tue Feb 14 23:27:53 2023 +0100

    rtc: allow rtc_read_alarm without read_alarm callback
```

```
    [ Upstream commit a783c962619271a8b905efad1d89adfec11ae0c8 ]

    .read_alarm is not necessary to read the current alarm because it is
    recorded in the aie_timer and so rtc_read_alarm() will never call
    rtc_read_alarm_internal() which is the only function calling the callback.

    Reported-by: Zhipeng Wang <zhipeng.wang_1@nxp.com>
    Reported-by: Marcel Ziswiler <marcel.ziswiler@toradex.com>
    Fixes: 7ae41220ef58 ("rtc: introduce features bitfield")
    Tested-by: Philippe Schenker <philippe.schenker@toradex.com>
    Link: https://lore.kernel.org/r/20230214222754.582582-1-alexandre.belloni@bootlin.com
    Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8ac713d2e9845e9234bb12ae5903040685d5aff9
Author: Shin'ichiro Kawasaki <shinichiro.kawasaki@wdc.com>
Date:   Tue Feb 14 09:50:18 2023 +0900

    scsi: mpi3mr: Use number of bits to manage bitmap sizes

    [ Upstream commit 339e61565f81a6534afdc18fd854b2e2628bf5db ]

    To allocate bitmaps, the mpi3mr driver calculates sizes of bitmaps using
    byte as unit. However, bitmap helper functions assume that bitmaps are
    allocated using unsigned long as unit. This gap causes memory access beyond
    the bitmap sizes and results in "BUG: KASAN: slab-out-of-bounds".  The BUG
    was observed at firmware download to eHBA-9600. Call trace indicated that
    the out-of-bounds access happened in find_first_zero_bit() called from
    mpi3mr_send_event_ack() for miroc->evtack_cmds_bitmap.

    To fix the BUG, do not use bytes to manage bitmap sizes. Instead, use
    number of bits, and call bitmap helper functions which take number of bits
    as arguments. For memory allocation, call bitmap_zalloc() instead of
    kzalloc() and krealloc(). For memory free, call bitmap_free() instead of
    kfree(). For zero clear, call bitmap_clear() instead of memset().

    Remove three fields for bitmap byte sizes in struct scmd_priv which are no
    longer required. Replace the field dev_handle_bitmap_sz with
    dev_handle_bitmap_bits to keep number of bits of removepend_bitmap across
    resize.

    Link: https://lore.kernel.org/r/20230214005019.1897251-4-shinichiro.kawasaki@wdc.com
    Fixes: c5758fc72b92 ("scsi: mpi3mr: Gracefully handle online FW update operation")
    Fixes: e844adb1fbdc ("scsi: mpi3mr: Implement SCSI error handler hooks")
    Fixes: c1af985d27da ("scsi: mpi3mr: Add Event acknowledgment logic")
    Fixes: 824a156633df ("scsi: mpi3mr: Base driver code")
    Signed-off-by: Shin'ichiro Kawasaki <shinichiro.kawasaki@wdc.com>
    Acked-by: Sathya Prakash Veerichetty <sathya.prakash@broadcom.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c8755f913a2fc9c168d108ea8c5af04716e8c4a5
Author: Tomas Henzl <thenzl@redhat.com>
Date:   Mon Feb 13 20:37:52 2023 +0100

    scsi: mpi3mr: Fix an issue found by KASAN

    [ Upstream commit ae7d45f5283d30274039b95d3e6d53d33c66e991 ]

    Write only correct size (32 instead of 64 bytes).

    Link: https://lore.kernel.org/r/20230213193752.6859-1-thenzl@redhat.com
    Fixes: 42fc9fee116f ("scsi: mpi3mr: Add helper functions to manage device's port")
    Signed-off-by: Tomas Henzl <thenzl@redhat.com>
```

```
    Acked-by: Sathya Prakash Veerichetty <sathya.prakash@broadcom.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 47088e61202ee15ecb00ba2f62749fdc9bbd7202
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Feb 14 14:28:08 2023 +0100

    scsi: ipr: Work around fortify-string warning

    [ Upstream commit ee4e7dfe4ffc9ca50c6875757bd119abfe22b5c5 ]

    The ipr_log_vpd_compact() function triggers a fortified memcpy() warning
    about a potential string overflow with all versions of clang:

    In file included from drivers/scsi/ipr.c:43:
    In file included from include/linux/string.h:254:
    include/linux/fortify-string.h:520:4: error: call to '__write_overflow_field' declared
with 'warning' attribute: detected write beyond size of field (1st parameter); maybe use
struct_group()? [-Werror,-Wattribute-warning]
                            __write_overflow_field(p_size_field, size);
                            ^
    include/linux/fortify-string.h:520:4: error: call to '__write_overflow_field' declared
with 'warning' attribute: detected write beyond size of field (1st parameter); maybe use
struct_group()? [-Werror,-Wattribute-warning]
    2 errors generated.

    I don't see anything actually wrong with the function, but this is the only
    instance I can reproduce of the fortification going wrong in the kernel at
    the moment, so the easiest solution may be to rewrite the function into
    something that does not trigger the warning.

    Instead of having a combined buffer for vendor/device/serial strings, use
    three separate local variables and just truncate the whitespace
    individually.

    Link: https://lore.kernel.org/r/20230214132831.2118392-1-arnd@kernel.org
    Cc: Kees Cook <keescook@chromium.org>
    Fixes: 8cf093e275d0 ("[SCSI] ipr: Improved dual adapter errors")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Damien Le Moal <damien.lemoal@opensource.wdc.com>
    Reviewed-by: Kees Cook <keescook@chromium.org>
    Acked-by: Brian King <brking@linux.vnet.ibm.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7448c73d64075051f50caed2c62f46553b69ab8a
Author: Sergey Shtylyov <s.shtylyov@omp.ru>
Date:   Wed Aug 17 23:00:45 2022 +0300

    genirq/ipi: Fix NULL pointer deref in irq_data_get_affinity_mask()

    [ Upstream commit feabecaff5902f896531dde90646ca5dfa9d4f7d ]

    If ipi_send_{mask|single}() is called with an invalid interrupt number, all
    the local variables there will be NULL. ipi_send_verify() which is invoked
    from these functions does verify its 'data' parameter, resulting in a
    kernel oops in irq_data_get_affinity_mask() as the passed NULL pointer gets
    dereferenced.

    Add a missing NULL pointer check in ipi_send_verify()...

    Found by Linux Verification Center (linuxtesting.org) with the SVACE static
    analysis tool.
```

```
    Fixes: 3b8e29a82dd1 ("genirq: Implement ipi_send_mask/single()")
    Signed-off-by: Sergey Shtylyov <s.shtylyov@omp.ru>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lore.kernel.org/r/b541232d-c2b6-1fe9-79b4-a7129459e4d0@omp.ru
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a76ffcd7996f696b9ed0962567c96e99c01dfd7f
Author: Samuel Holland <samuel@sholland.org>
Date:   Thu Dec 29 15:53:19 2022 -0600

    rtc: sun6i: Always export the internal oscillator

    [ Upstream commit 344f4030f6c50a9db2d03021884c4bf36191b53a ]

    On all variants of the hardware, the internal oscillator is one possible
    parent for the AR100 clock. It needs to be exported so we can model that
    relationship correctly in the devicetree.

    Fixes: c56afc1844d6 ("rtc: sun6i: Expose internal oscillator through device tree")
    Signed-off-by: Samuel Holland <samuel@sholland.org>
    Acked-by: Jernej Skrabec <jernej.skrabec@gmail.com>
    Link: https://lore.kernel.org/r/20221229215319.14145-1-samuel@sholland.org
    Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f527a17f4fbcab43a3c054cbea1939fb51f6838c
Author: Krishna Yarlagadda <kyarlagadda@nvidia.com>
Date:   Tue Feb 28 01:34:28 2023 +0530

    spi: tegra210-quad: Fix iterator outside loop

    [ Upstream commit 2449d436681d40bc63ec2c766fd51b632270d8a7 ]

    Fix warn: iterator used outside loop: 'xfer'. 'xfer' variable contain
    invalid value in few conditions. Complete transfer within DATA phase
    in successful case and at the end for failed transfer.

    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Link:https://lore.kernel.org/all/202210191211.46FkzKmv-lkp@intel.com/
    Fixes: 8777dd9dff40 ("spi: tegra210-quad: Fix combined sequence")

    Signed-off-by: Krishna Yarlagadda <kyarlagadda@nvidia.com>
    Link: https://lore.kernel.org/r/20230227200428.45832-1-kyarlagadda@nvidia.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f58b4378e1f37021173f0c551fa4869b3a12415c
Author: George Kennedy <george.kennedy@oracle.com>
Date:   Mon Feb 27 15:21:41 2023 -0500

    vc_screen: modify vcs_size() handling in vcs_read()

    [ Upstream commit 46d733d0efc79bc8430d63b57ab88011806d5180 ]

    Restore the vcs_size() handling in vcs_read() to what
    it had been in previous version.

    Fixes: 226fae124b2d ("vc_screen: move load of struct vc_data pointer in vcs_read() to
avoid UAF")
    Suggested-by: Jiri Slaby <jirislaby@kernel.org>
    Signed-off-by: George Kennedy <george.kennedy@oracle.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea13db527988137ef80d4366dfc8af4ffec4fca9
```

```
Author: Eric Dumazet <edumazet@google.com>
Date:   Mon Feb 27 08:33:36 2023 +0000

    tcp: tcp_check_req() can be called from process context

    [ Upstream commit 580f98cc33a260bb8c6a39ae2921b29586b84fdf ]

    This is a follow up of commit 0a375c822497 ("tcp: tcp_rtx_synack()
    can be called from process context").

    Frederick Lawler reported another "__this_cpu_add() in preemptible"
    warning caused by the same reason.

    In my former patch I took care of tcp_rtx_synack()
    but forgot that tcp_check_req() also contained some SNMP updates.

    Note that some parts of tcp_check_req() always run in BH context,
    I added a comment to clarify this.

    Fixes: 8336886f786f ("tcp: TCP Fast Open Server - support TFO listeners")
    Link: https://lore.kernel.org/netdev/8cd33923-a21d-397c-e46b-
2a068c287b03@cloudflare.com/T/
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reported-by: Frederick Lawler <fred@cloudflare.com>
    Tested-by: Frederick Lawler <fred@cloudflare.com>
    Link: https://lore.kernel.org/r/20230227083336.4153089-1-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a2f09b491bf36d2319a2bcac70a9c6218f006a65
Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
Date:   Sat Feb 25 17:22:37 2023 +0100

    ARM: dts: spear320-hmi: correct STMPE GPIO compatible

    [ Upstream commit 33a0c1b850c8c85f400531dab3a0b022cdb164b1 ]

    The compatible is st,stmpe-gpio.

    Fixes: e2eb69183ec4 ("ARM: SPEAr320: DT: Add SPEAr 320 HMI board support")
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Acked-by: Viresh Kumar <viresh.kumar@linaro.org>
    Link: https://lore.kernel.org/r/20230225162237.40242-1-krzysztof.kozlowski@linaro.org
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 16ff58cf5f285273a1ce6e6a6126148b481b13f9
Author: Eddie James <eajames@linux.ibm.com>
Date:   Tue Feb 21 11:03:52 2023 +1030

    ARM: dts: aspeed: p10bmc: Update battery node name

    [ Upstream commit a8cef541dd5ef9445130660008c029205c4c5aa5 ]

    The ADC sensor for the battery needs to be named "iio-hwmon" for
    compatibility with user space applications.

    Signed-off-by: Eddie James <eajames@linux.ibm.com>
    Link: https://lore.kernel.org/r/20230202152759.67069-1-eajames@linux.ibm.com
    Fixes: bf1914e2cfed ("ARM: dts: aspeed: p10bmc: Fix ADC iio-hwmon battery node name")
    Signed-off-by: Joel Stanley <joel@jms.id.au>
    Link: https://lore.kernel.org/r/20230221003352.1218797-1-joel@jms.id.au
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 0429d2a414eaff79bd0f05d81ca5c2a84ff65ccc
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Feb 24 17:52:34 2023 +0200

    net: dsa: felix: fix internal MDIO controller resource length

    [ Upstream commit 940af261321307cd1dd0fe8f9c34a6129f9d4bdc ]

    The blamed commit did not properly convert the resource start/end format
    into the DEFINE_RES_MEM_NAMED() start/length format, resulting in a
    resource for vsc9959_imdio_res which is much longer than expected:

    $ cat /proc/iomem
    1f8000000-1f815ffff : pcie@1f0000000
      1f8140000-1f815ffff : 0000:00:00.5
        1f8148030-1f815006f : imdio

    vs (correct)

    $ cat /proc/iomem
    1f8000000-1f815ffff : pcie@1f0000000
      1f8140000-1f815ffff : 0000:00:00.5
        1f8148030-1f814803f : imdio

    Luckily it's not big enough to exceed the size of the parent resource
    (pci_resource_end(pdev, VSC9959_IMDIO_PCI_BAR)), and it doesn't overlap
    with anything else that the Linux driver uses currently, so the larger
    than expected size isn't a practical problem that I can see. Although it
    is clearly wrong in the /proc/iomem output.

    Fixes: 044d447a801f ("net: dsa: felix: use DEFINE_RES_MEM_NAMED for resources")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 257f174eb9ba4e8424ca7ff7e4d5629f86d6f4fc
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Fri Feb 24 17:52:33 2023 +0200

    net: dsa: seville: ignore mscc-miim read errors from Lynx PCS

    [ Upstream commit 0322ef49c1ac6f0e2ef37b146c0bf8440873072c ]

    During the refactoring in the commit below, vsc9953_mdio_read() was
    replaced with mscc_miim_read(), which has one extra step: it checks for
    the MSCC_MIIM_DATA_ERROR bits before returning the result.

    On T1040RDB, there are 8 QSGMII PCSes belonging to the switch, and they
    are organized in 2 groups. First group responds to MDIO addresses 4-7
    because QSGMIIACR1[MDEV_PORT] is 1, and the second group responds to
    MDIO addresses 8-11 because QSGMIIBCR1[MDEV_PORT] is 2. I have double
    checked that these values are correctly set in the SERDES, as well as
    PCCR1[QSGMA_CFG] and PCCR1[QSGMB_CFG] are both 0b01.

    mscc_miim_read: phyad 8 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 8 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 8 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 8 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 9 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 9 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 9 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 9 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 10 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 10 reg 0x5 MIIM_DATA 0x5801
```

```
    mscc_miim_read: phyad 10 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 10 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 11 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 11 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 11 reg 0x1 MIIM_DATA 0x2d
    mscc_miim_read: phyad 11 reg 0x5 MIIM_DATA 0x5801
    mscc_miim_read: phyad 4 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 4 reg 0x5 MIIM_DATA 0x3da01, ERROR
    mscc_miim_read: phyad 5 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 5 reg 0x5 MIIM_DATA 0x35801, ERROR
    mscc_miim_read: phyad 5 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 5 reg 0x5 MIIM_DATA 0x35801, ERROR
    mscc_miim_read: phyad 6 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 6 reg 0x5 MIIM_DATA 0x35801, ERROR
    mscc_miim_read: phyad 6 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 6 reg 0x5 MIIM_DATA 0x35801, ERROR
    mscc_miim_read: phyad 7 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 7 reg 0x5 MIIM_DATA 0x35801, ERROR
    mscc_miim_read: phyad 7 reg 0x1 MIIM_DATA 0x3002d, ERROR
    mscc_miim_read: phyad 7 reg 0x5 MIIM_DATA 0x35801, ERROR

    As can be seen, the data in MIIM_DATA is still valid despite having the
    MSCC_MIIM_DATA_ERROR bits set. The driver as introduced in commit
    84705fc16552 ("net: dsa: felix: introduce support for Seville VSC9953
    switch") was ignoring these bits, perhaps deliberately (although
    unbeknownst to me).

    This is an old IP and the hardware team cannot seem to be able to help
    me track down a plausible reason for these failures. I'll keep
    investigating, but in the meantime, this is a direct regression which
    must be restored to a working state.

    The only thing I can do is keep ignoring the errors as before.

    Fixes: b99658452355 ("net: dsa: ocelot: felix: utilize shared mscc-miim driver for
indirect MDIO access")
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6d86a8691d44537cd6e9c10fe41a4644bfedf24f
Author: Pedro Tammela <pctammela@mojatatu.com>
Date:   Fri Feb 24 12:00:58 2023 -0300

    net/sched: act_sample: fix action bind logic

    [ Upstream commit 4a20056a49a18549665622241922f68197f950539 ]

    The TC architecture allows filters and actions to be created independently.
    In filters the user can reference action objects using:
    tc action add action sample ... index 1
    tc filter add ... action pedit index 1

    In the current code for act_sample this is broken as it checks netlink
    attributes for create/update before actually checking if we are binding to an
    existing action.

    tdc results:
    1..29
    ok 1 9784 - Add valid sample action with mandatory arguments
    ok 2 5c91 - Add valid sample action with mandatory arguments and continue control action
    ok 3 334b - Add valid sample action with mandatory arguments and drop control action
    ok 4 da69 - Add valid sample action with mandatory arguments and reclassify control
action
```

```
    ok 5 13ce - Add valid sample action with mandatory arguments and pipe control action
    ok 6 1886 - Add valid sample action with mandatory arguments and jump control action
    ok 7 7571 - Add sample action with invalid rate
    ok 8 b6d4 - Add sample action with mandatory arguments and invalid control action
    ok 9 a874 - Add invalid sample action without mandatory arguments
    ok 10 ac01 - Add invalid sample action without mandatory argument rate
    ok 11 4203 - Add invalid sample action without mandatory argument group
    ok 12 14a7 - Add invalid sample action without mandatory argument group
    ok 13 8f2e - Add valid sample action with trunc argument
    ok 14 45f8 - Add sample action with maximum rate argument
    ok 15 ad0c - Add sample action with maximum trunc argument
    ok 16 83a9 - Add sample action with maximum group argument
    ok 17 ed27 - Add sample action with invalid rate argument
    ok 18 2eae - Add sample action with invalid group argument
    ok 19 6ff3 - Add sample action with invalid trunc size
    ok 20 2b2a - Add sample action with invalid index
    ok 21 dee2 - Add sample action with maximum allowed index
    ok 22 560e - Add sample action with cookie
    ok 23 704a - Replace existing sample action with new rate argument
    ok 24 60eb - Replace existing sample action with new group argument
    ok 25 2cce - Replace existing sample action with new trunc argument
    ok 26 59d1 - Replace existing sample action with new control argument
    ok 27 0a6e - Replace sample action with invalid goto chain control
    ok 28 3872 - Delete sample action with valid index
    ok 29 a394 - Delete sample action with invalid index

    Fixes: 5c5670fae430 ("net/sched: Introduce sample tc action")
    Reviewed-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Signed-off-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f620f312969ce7085061b2952a56fddfb07f8032
Author: Pedro Tammela <pctammela@mojatatu.com>
Date:   Fri Feb 24 12:00:57 2023 -0300

    net/sched: act_mpls: fix action bind logic

    [ Upstream commit e88d78a773cb5242e933930c8855bf4b2e8c2397 ]

    The TC architecture allows filters and actions to be created independently.
    In filters the user can reference action objects using:
    tc action add action mpls ... index 1
    tc filter add ... action mpls index 1

    In the current code for act_mpls this is broken as it checks netlink
    attributes for create/update before actually checking if we are binding to an
    existing action.

    tdc results:
    1..53
    ok 1 a933 - Add MPLS dec_ttl action with pipe opcode
    ok 2 08d1 - Add mpls dec_ttl action with pass opcode
    ok 3 d786 - Add mpls dec_ttl action with drop opcode
    ok 4 f334 - Add mpls dec_ttl action with reclassify opcode
    ok 5 29bd - Add mpls dec_ttl action with continue opcode
    ok 6 48df - Add mpls dec_ttl action with jump opcode
    ok 7 62eb - Add mpls dec_ttl action with trap opcode
    ok 8 09d2 - Add mpls dec_ttl action with opcode and cookie
    ok 9 c170 - Add mpls dec_ttl action with opcode and cookie of max length
    ok 10 9118 - Add mpls dec_ttl action with invalid opcode
    ok 11 6ce1 - Add mpls dec_ttl action with label (invalid)
    ok 12 352f - Add mpls dec_ttl action with tc (invalid)
    ok 13 fa1c - Add mpls dec_ttl action with ttl (invalid)
```

```
    ok 14 6b79 - Add mpls dec_ttl action with bos (invalid)
    ok 15 d4c4 - Add mpls pop action with ip proto
    ok 16 91fb - Add mpls pop action with ip proto and cookie
    ok 17 92fe - Add mpls pop action with mpls proto
    ok 18 7e23 - Add mpls pop action with no protocol (invalid)
    ok 19 6182 - Add mpls pop action with label (invalid)
    ok 20 6475 - Add mpls pop action with tc (invalid)
    ok 21 067b - Add mpls pop action with ttl (invalid)
    ok 22 7316 - Add mpls pop action with bos (invalid)
    ok 23 38cc - Add mpls push action with label
    ok 24 c281 - Add mpls push action with mpls_mc protocol
    ok 25 5db4 - Add mpls push action with label, tc and ttl
    ok 26 7c34 - Add mpls push action with label, tc ttl and cookie of max length
    ok 27 16eb - Add mpls push action with label and bos
    ok 28 d69d - Add mpls push action with no label (invalid)
    ok 29 e8e4 - Add mpls push action with ipv4 protocol (invalid)
    ok 30 ecd0 - Add mpls push action with out of range label (invalid)
    ok 31 d303 - Add mpls push action with out of range tc (invalid)
    ok 32 fd6e - Add mpls push action with ttl of 0 (invalid)
    ok 33 19e9 - Add mpls mod action with mpls label
    ok 34 1fde - Add mpls mod action with max mpls label
    ok 35 0c50 - Add mpls mod action with mpls label exceeding max (invalid)
    ok 36 10b6 - Add mpls mod action with mpls label of MPLS_LABEL_IMPLNULL (invalid)
    ok 37 57c9 - Add mpls mod action with mpls min tc
    ok 38 6872 - Add mpls mod action with mpls max tc
    ok 39 a70a - Add mpls mod action with mpls tc exceeding max (invalid)
    ok 40 6ed5 - Add mpls mod action with mpls ttl
    ok 41 77c1 - Add mpls mod action with mpls ttl and cookie
    ok 42 b80f - Add mpls mod action with mpls max ttl
    ok 43 8864 - Add mpls mod action with mpls min ttl
    ok 44 6c06 - Add mpls mod action with mpls ttl of 0 (invalid)
    ok 45 b5d8 - Add mpls mod action with mpls ttl exceeding max (invalid)
    ok 46 451f - Add mpls mod action with mpls max bos
    ok 47 a1ed - Add mpls mod action with mpls min bos
    ok 48 3dcf - Add mpls mod action with mpls bos exceeding max (invalid)
    ok 49 db7c - Add mpls mod action with protocol (invalid)
    ok 50 b070 - Replace existing mpls push action with new ID
    ok 51 95a9 - Replace existing mpls push action with new label, tc, ttl and cookie
    ok 52 6cce - Delete mpls pop action
    ok 53 d138 - Flush mpls actions

    Fixes: 2a2ea50870ba ("net: sched: add mpls manipulation actions to TC")
    Reviewed-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Signed-off-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 44e3f98192884b2ac0067144f0dbe0a9f9e9b38c
Author: Pedro Tammela <pctammela@mojatatu.com>
Date:   Fri Feb 24 12:00:56 2023 -0300

    net/sched: act_pedit: fix action bind logic

    [ Upstream commit e9e42292ea76a8358b0c02ffd530d78e133a1b73 ]

    The TC architecture allows filters and actions to be created independently.
    In filters the user can reference action objects using:
    tc action add action pedit ... index 1
    tc filter add ... action pedit index 1

    In the current code for act_pedit this is broken as it checks netlink
    attributes for create/update before actually checking if we are binding to an
    existing action.
```

```
    tdc results:
    1..69
    ok 1 319a - Add pedit action that mangles IP TTL
    ok 2 7e67 - Replace pedit action with invalid goto chain
    ok 3 377e - Add pedit action with RAW_OP offset u32
    ok 4 a0ca - Add pedit action with RAW_OP offset u32 (INVALID)
    ok 5 dd8a - Add pedit action with RAW_OP offset u16 u16
    ok 6 53db - Add pedit action with RAW_OP offset u16 (INVALID)
    ok 7 5c7e - Add pedit action with RAW_OP offset u8 add value
    ok 8 2893 - Add pedit action with RAW_OP offset u8 quad
    ok 9 3a07 - Add pedit action with RAW_OP offset u8-u16-u8
    ok 10 ab0f - Add pedit action with RAW_OP offset u16-u8-u8
    ok 11 9d12 - Add pedit action with RAW_OP offset u32 set u16 clear u8 invert
    ok 12 ebfa - Add pedit action with RAW_OP offset overflow u32 (INVALID)
    ok 13 f512 - Add pedit action with RAW_OP offset u16 at offmask shift set
    ok 14 c2cb - Add pedit action with RAW_OP offset u32 retain value
    ok 15 1762 - Add pedit action with RAW_OP offset u8 clear value
    ok 16 bcee - Add pedit action with RAW_OP offset u8 retain value
    ok 17 e89f - Add pedit action with RAW_OP offset u16 retain value
    ok 18 c282 - Add pedit action with RAW_OP offset u32 clear value
    ok 19 c422 - Add pedit action with RAW_OP offset u16 invert value
    ok 20 d3d3 - Add pedit action with RAW_OP offset u32 invert value
    ok 21 57e5 - Add pedit action with RAW_OP offset u8 preserve value
    ok 22 99e0 - Add pedit action with RAW_OP offset u16 preserve value
    ok 23 1892 - Add pedit action with RAW_OP offset u32 preserve value
    ok 24 4b60 - Add pedit action with RAW_OP negative offset u16/u32 set value
    ok 25 a5a7 - Add pedit action with LAYERED_OP eth set src
    ok 26 86d4 - Add pedit action with LAYERED_OP eth set src & dst
    ok 27 f8a9 - Add pedit action with LAYERED_OP eth set dst
    ok 28 c715 - Add pedit action with LAYERED_OP eth set src (INVALID)
    ok 29 8131 - Add pedit action with LAYERED_OP eth set dst (INVALID)
    ok 30 ba22 - Add pedit action with LAYERED_OP eth type set/clear sequence
    ok 31 dec4 - Add pedit action with LAYERED_OP eth set type (INVALID)
    ok 32 ab06 - Add pedit action with LAYERED_OP eth add type
    ok 33 918d - Add pedit action with LAYERED_OP eth invert src
    ok 34 a8d4 - Add pedit action with LAYERED_OP eth invert dst
    ok 35 ee13 - Add pedit action with LAYERED_OP eth invert type
    ok 36 7588 - Add pedit action with LAYERED_OP ip set src
    ok 37 0fa7 - Add pedit action with LAYERED_OP ip set dst
    ok 38 5810 - Add pedit action with LAYERED_OP ip set src & dst
    ok 39 1092 - Add pedit action with LAYERED_OP ip set ihl & dsfield
    ok 40 02d8 - Add pedit action with LAYERED_OP ip set ttl & protocol
    ok 41 3e2d - Add pedit action with LAYERED_OP ip set ttl (INVALID)
    ok 42 31ae - Add pedit action with LAYERED_OP ip ttl clear/set
    ok 43 486f - Add pedit action with LAYERED_OP ip set duplicate fields
    ok 44 e790 - Add pedit action with LAYERED_OP ip set ce, df, mf, firstfrag, nofrag
 fields
    ok 45 cc8a - Add pedit action with LAYERED_OP ip set tos
    ok 46 7a17 - Add pedit action with LAYERED_OP ip set precedence
    ok 47 c3b6 - Add pedit action with LAYERED_OP ip add tos
    ok 48 43d3 - Add pedit action with LAYERED_OP ip add precedence
    ok 49 438e - Add pedit action with LAYERED_OP ip clear tos
    ok 50 6b1b - Add pedit action with LAYERED_OP ip clear precedence
    ok 51 824a - Add pedit action with LAYERED_OP ip invert tos
    ok 52 106f - Add pedit action with LAYERED_OP ip invert precedence
    ok 53 6829 - Add pedit action with LAYERED_OP beyond ip set dport & sport
    ok 54 afd8 - Add pedit action with LAYERED_OP beyond ip set icmp_type & icmp_code
    ok 55 3143 - Add pedit action with LAYERED_OP beyond ip set dport (INVALID)
    ok 56 815c - Add pedit action with LAYERED_OP ip6 set src
    ok 57 4dae - Add pedit action with LAYERED_OP ip6 set dst
    ok 58 fc1f - Add pedit action with LAYERED_OP ip6 set src & dst
    ok 59 6d34 - Add pedit action with LAYERED_OP ip6 dst retain value (INVALID)
    ok 60 94bb - Add pedit action with LAYERED_OP ip6 traffic_class
    ok 61 6f5e - Add pedit action with LAYERED_OP ip6 flow_lbl
    ok 62 6795 - Add pedit action with LAYERED_OP ip6 set payload_len, nexthdr, hoplimit
```

```
    ok 63 1442 - Add pedit action with LAYERED_OP tcp set dport & sport
    ok 64 b7ac - Add pedit action with LAYERED_OP tcp sport set (INVALID)
    ok 65 cfcc - Add pedit action with LAYERED_OP tcp flags set
    ok 66 3bc4 - Add pedit action with LAYERED_OP tcp set dport, sport & flags fields
    ok 67 f1c8 - Add pedit action with LAYERED_OP udp set dport & sport
    ok 68 d784 - Add pedit action with mixed RAW/LAYERED_OP #1
    ok 69 70ca - Add pedit action with mixed RAW/LAYERED_OP #2

    Fixes: 71d0ed7079df ("net/act_pedit: Support using offset relative to the conventional
network headers")
    Fixes: f67169fef8db ("net/sched: act_pedit: fix WARN() in the traffic path")
    Reviewed-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Signed-off-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cf6830fc2502b8db0a4674dff5524173d2278dad
Author: Pedro Tammela <pctammela@mojatatu.com>
Date:   Tue Jan 31 16:05:11 2023 -0300

    net/sched: transition act_pedit to rcu and percpu stats

    [ Upstream commit 52cf89f78c01bf39973f3e70d366921d70faff7a ]

    The software pedit action didn't get the same love as some of the
    other actions and it's still using spinlocks and shared stats in the
    datapath.
    Transition the action to rcu and percpu stats as this improves the
    action's performance dramatically on multiple cpu deployments.

    Reviewed-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Signed-off-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Stable-dep-of: e9e42292ea76 ("net/sched: act_pedit: fix action bind logic")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ba98db08895748c12e5ded52cd1598dce2c79e55
Author: Fedor Pchelkin <pchelkin@ispras.ru>
Date:   Sat Feb 25 13:56:14 2023 +0300

    nfc: fix memory leak of se_io context in nfc_genl_se_io

    [ Upstream commit 25ff6f8a5a3b8dc48e8abda6f013e8cc4b14ffea ]

    The callback context for sending/receiving APDUs to/from the selected
    secure element is allocated inside nfc_genl_se_io and supposed to be
    eventually freed in se_io_cb callback function. However, there are several
    error paths where the bwi_timer is not charged to call se_io_cb later, and
    the cb_context is leaked.

    The patch proposes to free the cb_context explicitly on those error paths.

    At the moment we can't simply check 'dev->ops->se_io()' return value as it
    may be negative in both cases: when the timer was charged and was not.

    Fixes: 5ce3f32b5264 ("NFC: netlink: SE API implementation")
    Reported-by: syzbot+df64c0a2e8d68e78a4fa@syzkaller.appspotmail.com
    Signed-off-by: Fedor Pchelkin <pchelkin@ispras.ru>
    Signed-off-by: Alexey Khoroshilov <khoroshilov@ispras.ru>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e7ff7500bb2563b0016c30845f84d68a90e9b98b
```

```
Author: Zhang Yi <yi.zhang@huawei.com>
Date:   Sun Jan 29 11:49:39 2023 +0800

    ext4: fix incorrect options show of original mount_opt and extend mount_opt2

    [ Upstream commit e3645d72f8865ffe36f9dc811540d40aa3c848d3 ]

    Current _ext4_show_options() do not distinguish MOPT_2 flag, so it mixed
    extend sbi->s_mount_opt2 options with sbi->s_mount_opt, it could lead to
    show incorrect options, e.g. show fc_debug_force if we mount with
    errors=continue mode and miss it if we set.

      $ mkfs.ext4 /dev/pmem0
      $ mount -o errors=remount-ro /dev/pmem0 /mnt
      $ cat /proc/fs/ext4/pmem0/options | grep fc_debug_force
        #empty
      $ mount -o remount,errors=continue /mnt
      $ cat /proc/fs/ext4/pmem0/options | grep fc_debug_force
        fc_debug_force
      $ mount -o remount,errors=remount-ro,fc_debug_force /mnt
      $ cat /proc/fs/ext4/pmem0/options | grep fc_debug_force
        #empty

    Fixes: 995a3ed67fc8 ("ext4: add fast_commit feature and handling for extended mount
options")
    Signed-off-by: Zhang Yi <yi.zhang@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20230129034939.3702550-1-yi.zhang@huaweicloud.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c4462b29eaadb0766e8ee39594bd550f33b0687a
Author: Maor Dickman <maord@nvidia.com>
Date:   Wed Feb 8 17:44:06 2023 +0200

    net/mlx5: Geneve, Fix handling of Geneve object id as error code

    [ Upstream commit d28a06d7dbedc598a06bd1e53a28125f87ca5d0c ]

    On success, mlx5_geneve_tlv_option_create returns non negative
    Geneve object id. In case the object id is positive value the
    caller functions will handle it as an error (non zero) and
    will fail to offload the Geneve rule.

    Fix this by changing caller function ,mlx5_geneve_tlv_option_add,
    to return 0 in case valid non negative object id was provided.

    Fixes: 0ccc171ea6a2 ("net/mlx5: Geneve, Manage Geneve TLV options")
    Signed-off-by: Maor Dickman <maord@nvidia.com>
    Reviewed-by: Raed Salem <raeds@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a21afd2a717ac56e73ff95d6cc6061007f913013
Author: Roi Dayan <roid@nvidia.com>
Date:   Tue Jan 31 12:04:30 2023 +0200

    net/mlx5e: Verify flow_source cap before using it

    [ Upstream commit 1bf8b0dae8dde6f02520a5ea34fdaa3b39342e69 ]

    When adding send to vport rule verify flow_source matching is
    supported by checking the flow_source cap.

    Fixes: d04442540372 ("net/mlx5: E-Switch, set flow source for send to uplink rule")
```

```
    Signed-off-by: Roi Dayan <roid@nvidia.com>
    Reviewed-by: Maor Dickman <maord@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c7b2f2bbf2be12d389701d84f2ed5715dd69e904
Author: Maher Sanalla <msanalla@nvidia.com>
Date:   Wed Feb 15 20:12:05 2023 +0200

    net/mlx5: ECPF, wait for VF pages only after disabling host PFs

    [ Upstream commit e1ed30c8c09abc85a01c897845bdbd08c0333353 ]

    Currently,  during the early stages of their unloading, particularly
    during SRIOV disablement, PFs/ECPFs wait on the release of all of
    their VFs memory pages. Furthermore, ECPFs are considered the page
    supplier for host VFs, hence the host VFs memory pages are freed only
    during ECPF cleanup when host interfaces get disabled.

    Thus, disabling SRIOV early in unload timeline causes the DPU ECPF
    to stall on driver unload while waiting on the release of host VF pages
    that won't be freed before host interfaces get disabled later on.

    Therefore, for ECPFs, wait on the release of VFs pages only after the
    disablement of host PFs during ECPF cleanup flow. Then, host PFs and VFs
    are disabled and their memory shall be freed accordingly.

    Fixes: 143a41d7623d ("net/mlx5: Disable SRIOV before PF removal")
    Signed-off-by: Maher Sanalla <msanalla@nvidia.com>
    Reviewed-by: Moshe Shemesh <moshe@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6afdedc4e66e3846ce497744f01b95c34bf39d21
Author: Vadim Fedorenko <vadfed@meta.com>
Date:   Thu Feb 2 09:13:55 2023 -0800

    mlx5: fix possible ptp queue fifo use-after-free

    [ Upstream commit 3a50cf1e8e5157b82268eee7e330dbe5736a0948 ]

    Fifo indexes are not checked during pop operations and it leads to
    potential use-after-free when poping from empty queue. Such case was
    possible during re-sync action. WARN_ON_ONCE covers future cases.

    There were out-of-order cqe spotted which lead to drain of the queue and
    use-after-free because of lack of fifo pointers check. Special check and
    counter are added to avoid resync operation if SKB could not exist in the
    fifo because of OOO cqe (skb_id must be between consumer and producer
    index).

    Fixes: 58a518948f60 ("net/mlx5e: Add resiliency for PTP TX port timestamp")
    Signed-off-by: Vadim Fedorenko <vadfed@meta.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 68504c66d08c70fb92799722e25a932d311d74fd
Author: Vadim Fedorenko <vadfed@meta.com>
Date:   Thu Feb 2 09:13:54 2023 -0800

    mlx5: fix skb leak while fifo resync and push

    [ Upstream commit e435941b1da1a0be4ff8a7ae425774c76a5ac514 ]

    During ptp resync operation SKBs were poped from the fifo but were never
```

```
    freed neither by napi_consume nor by dev_kfree_skb_any. Add call to
    napi_consume_skb to properly free SKBs.

    Another leak was happening because mlx5e_skb_fifo_has_room() had an error
    in the check. Comparing free running counters works well unless C promotes
    the types to something wider than the counter. In this case counters are
    u16 but the result of the substraction is promouted to int and it causes
    wrong result (negative value) of the check when producer have already
    overlapped but consumer haven't yet. Explicit cast to u16 fixes the issue.

    Fixes: 58a518948f60 ("net/mlx5e: Add resiliency for PTP TX port timestamp")
    Reviewed-by: Gal Pressman <gal@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Vadim Fedorenko <vadfed@meta.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 999d0a88f3bc6dbf7a574362ba63e8740a46f301
Author: Krishna Yarlagadda <kyarlagadda@nvidia.com>
Date:   Fri Feb 24 22:10:34 2023 +0530

    spi: tegra210-quad: Fix validate combined sequence

    [ Upstream commit 047ee71ae4f412d8819e39e4b08c588fa299cfc2 ]

    Check for non dma transfers that do not fit in FIFO has issue and skips
    combined sequence for Tegra234 & Tegra241 which does not have GPCDMA.

    Fixes: 1b8342cc4a38 ("spi: tegra210-quad: combined sequence mode")

    Signed-off-by: Krishna Yarlagadda <kyarlagadda@nvidia.com>
    Link: https://lore.kernel.org/r/20230224164034.56933-1-kyarlagadda@nvidia.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bd206052de13104a42249af0d9f61121e8e25d82
Author: Zhengchao Shao <shaozhengchao@huawei.com>
Date:   Wed Jan 4 10:04:24 2023 +0800

    9p/rdma: unmap receive dma buffer in rdma_request()/post_recv()

    [ Upstream commit 74a25e6e916cb57dab4267a96fbe8864ed21abdb ]

    When down_interruptible() or ib_post_send() failed in rdma_request(),
    receive dma buffer is not unmapped. Add unmap action to error path.
    Also if ib_post_recv() failed in post_recv(), dma buffer is not unmapped.
    Add unmap action to error path.

    Link: https://lkml.kernel.org/r/20230104020424.611926-1-shaozhengchao@huawei.com
    Fixes: fc79d4b104f0 ("9p: rdma: RDMA Transport Support for 9P")
    Signed-off-by: Zhengchao Shao <shaozhengchao@huawei.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Signed-off-by: Dominique Martinet <asmadeus@codewreck.org>
    Signed-off-by: Eric Van Hensbergen <ericvh@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5f6a8974e9ef317fe63f88bab1f33070195dd147
Author: Juergen Gross <jgross@suse.com>
Date:   Mon Jan 30 12:30:36 2023 +0100

    9p/xen: fix connection sequence

    [ Upstream commit c15fe55d14b3b4ded5af2a3260877460a6ffb8ad ]

    Today the connection sequence of the Xen 9pfs frontend doesn't match
```

```
    the documented sequence. It can work reliably only for a PV 9pfs device
    having been added at boot time already, as the frontend is not waiting
    for the backend to have set its state to "XenbusStateInitWait" before
    reading the backend properties from Xenstore.

    Fix that by following the documented sequence [1] (the documentation
    has a bug, so the reference is for the patch fixing that).

    [1]: https://lore.kernel.org/xen-devel/20230130090937.31623-1-jgross@suse.com/T/#u

    Link: https://lkml.kernel.org/r/20230130113036.7087-3-jgross@suse.com
    Fixes: 868eb122739a ("xen/9pfs: introduce Xen 9pfs transport driver")
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: Dominique Martinet <asmadeus@codewreck.org>
    Signed-off-by: Eric Van Hensbergen <ericvh@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 39a46b392c4c5892f3da60a9f50b7505ea73d2fc
Author: Juergen Gross <jgross@suse.com>
Date:   Mon Jan 30 12:30:35 2023 +0100

    9p/xen: fix version parsing

    [ Upstream commit f1956f4ec15195ec60976d9b5625326285ab102e ]

    When connecting the Xen 9pfs frontend to the backend, the "versions"
    Xenstore entry written by the backend is parsed in a wrong way.

    The "versions" entry is defined to contain the versions supported by
    the backend separated by commas (e.g. "1,2"). Today only version "1"
    is defined. Unfortunately the frontend doesn't look for "1" being
    listed in the entry, but it is expecting the entry to have the value
    "1".

    This will result in failure as soon as the backend will support e.g.
    versions "1" and "2".

    Fix that by scanning the entry correctly.

    Link: https://lkml.kernel.org/r/20230130113036.7087-2-jgross@suse.com
    Fixes: 71ebd71921e4 ("xen/9pfs: connect to the backend")
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: Dominique Martinet <asmadeus@codewreck.org>
    Signed-off-by: Eric Van Hensbergen <ericvh@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d3495ec81b97fd4c9c3a325cbd083d041021b3c3
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Feb 23 08:38:45 2023 +0000

    net: fix __dev_kfree_skb_any() vs drop monitor

    [ Upstream commit ac3ad19584b26fae9ac86e4faebe790becc74491 ]

    dev_kfree_skb() is aliased to consume_skb().

    When a driver is dropping a packet by calling dev_kfree_skb_any()
    we should propagate the drop reason instead of pretending
    the packet was consumed.

    Note: Now we have enum skb_drop_reason we could remove
    enum skb_free_reason (for linux-6.4)
```

```
    v2: added an unlikely(), suggested by Yunsheng Lin.

    Fixes: e6247027e517 ("net: introduce dev_consume_skb_any()")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Yunsheng Lin <linyunsheng@huawei.com>
    Reviewed-by: Yunsheng Lin <linyunsheng@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 666f15bc1f844359c16d362cb931ba70650d5c98
Author: Deepak R Varma <drv@mailo.com>
Date:   Wed Feb 22 18:58:48 2023 +0530

    octeontx2-pf: Use correct struct reference in test condition

    [ Upstream commit 3acd9db9293f3b33ac04e8d44ed05b604ad1ac26 ]

    Fix the typo/copy-paste error by replacing struct variable ah_esp_mask name
    by ah_esp_hdr.
    Issue identified using doublebitand.cocci Coccinelle semantic patch.

    Fixes: b7cf966126eb ("octeontx2-pf: Add flow classification using IP next level
protocol")
    Link: https://lore.kernel.org/all/20210111112537.3277-1-naveenm@marvell.com/
    Signed-off-by: Deepak R Varma <drv@mailo.com>
    Link: https://lore.kernel.org/r/Y/YYkKddeHOt80cO@ubun2204.myguest.virtualbox.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6d529928ea212127851a2df8c40d822237ca946b
Author: Xin Long <lucien.xin@gmail.com>
Date:   Wed Feb 22 12:07:21 2023 -0500

    sctp: add a refcnt in sctp_stream_priorities to avoid a nested loop

    [ Upstream commit 68ba44639537de6f91fe32783766322d41848127 ]

    With this refcnt added in sctp_stream_priorities, we don't need to
    traverse all streams to check if the prio is used by other streams
    when freeing one stream's prio in sctp_sched_prio_free_sid(). This
    can avoid a nested loop (up to 65535 * 65535), which may cause a
    stuck as Ying reported:

        watchdog: BUG: soft lockup - CPU#23 stuck for 26s! [ksoftirqd/23:136]
        Call Trace:
         <TASK>
         sctp_sched_prio_free_sid+0xab/0x100 [sctp]
         sctp_stream_free_ext+0x64/0xa0 [sctp]
         sctp_stream_free+0x31/0x50 [sctp]
         sctp_association_free+0xa5/0x200 [sctp]

    Note that it doesn't need to use refcount_t type for this counter,
    as its accessing is always protected under the sock lock.

    v1->v2:
     - add a check in sctp_sched_prio_set to avoid the possible prio_head
       refcnt overflow.

    Fixes: 9ed7bfc79542 ("sctp: fix memory leak in sctp_stream_outq_migrate()")
    Reported-by: Ying Xu <yinxu@redhat.com>
    Acked-by: Marcelo Ricardo Leitner <marcelo.leitner@gmail.com>
    Signed-off-by: Xin Long <lucien.xin@gmail.com>
    Link:
https://lore.kernel.org/r/825eb0c905cb864991eba335f4a2b780e543f06b.1677085641.git.lucien.xin
@gmail.com
```

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

commit 1f468c3a430e88b951d99557f594fb271b91358b
Author: Sean Anderson <seanga2@gmail.com>
Date:   Wed Feb 22 15:42:41 2023 -0500

    net: sunhme: Fix region request

    [ Upstream commit ee0a735fd97ccde766ab557d1fc722c92cebacda ]

    devm_request_region is for I/O regions. Use devm_request_mem_region
    instead.  This fixes the driver failing to probe since 99df45c9e0a4
    ("sunhme: fix an IS_ERR() vs NULL check in probe"), which checked the
    result.

    Fixes: 914d9b2711dd ("sunhme: switch to devres")
    Signed-off-by: Sean Anderson <seanga2@gmail.com>
    Reviewed-by: Pavan Chebbi <pavan.chebbi@broadcom.com>
    Link: https://lore.kernel.org/r/20230222204242.2658247-1-seanga2@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bf90cf76e8fa18cc05e7bb8b464238c941421e04
Author: Geetha sowjanya <gakula@marvell.com>
Date:   Wed Feb 22 17:06:00 2023 +0530

    octeontx2-pf: Recalculate UDP checksum for ptp 1-step sync packet

    [ Upstream commit edea0c5a994b7829c9ada8f5bc762c4e32f4f797 ]

    When checksum offload is disabled in the driver via ethtool,
    the PTP 1-step sync packets contain incorrect checksum, since
    the stack calculates the checksum before driver updates
    PTP timestamp field in the packet. This results in PTP packets
    getting dropped at the other end. This patch fixes the issue by
    re-calculating the UDP checksum after updating PTP
    timestamp field in the driver.

    Fixes: 2958d17a8984 ("octeontx2-pf: Add support for ptp 1-step mode on CN10K silicon")
    Signed-off-by: Geetha sowjanya <gakula@marvell.com>
    Signed-off-by: Hariprasad Kelam <hkelam@marvell.com>
    Signed-off-by: Sunil Kovvuri Goutham <sgoutham@marvell.com>
    Signed-off-by: Sai Krishna <saikrishnag@marvell.com>
    Link: https://lore.kernel.org/r/20230222113600.1965116-1-saikrishnag@marvell.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c35ec29e92d949f75e7494189e8323960ffc5138
Author: Matt Roper <matthew.d.roper@intel.com>
Date:   Mon Feb 13 16:19:06 2023 -0800

    drm/i915/xelpmp: Consider GSI offset when doing MCR lookups

    [ Upstream commit 33c25354939099b76ecb6c82d1c7c50400fbcca6 ]

    MCR range tables use the final MMIO offset of a register (including the
    0x380000 GSI offset when applicable).  Since the i915_mcr_reg_t passed
    as a parameter during steering lookup does not include the GSI offset,
    we need to add it back in for GSI registers before searching the tables.

    Fixes: a7ec65fc7e83 ("drm/i915/xelpmp: Add multicast steering for media GT")
    Signed-off-by: Matt Roper <matthew.d.roper@intel.com>
    Reviewed-by: Radhakrishna Sripada <radhakrishna.sripada@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230214001906.1477370-1-
```

```
 matthew.d.roper@intel.com
    (cherry picked from commit d6683bbe70d4cdbf3da6acecf7d569cc6f0b4382)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit aa75d826c221e8d48607aef33836cf872a159cf1
 Author: Lu Wei <luwei32@huawei.com>
 Date:   Wed Feb 22 16:36:28 2023 +0800

    ipv6: Add lwtunnel encap size of all siblings in nexthop calculation

    [ Upstream commit 4cc59f386991ec9374cb4bc83dbe1c0b5a95033f ]

    In function rt6_nlmsg_size(), the length of nexthop is calculated
    by multipling the nexthop length of fib6_info and the number of
    siblings. However if the fib6_info has no lwtunnel but the siblings
    have lwtunnels, the nexthop length is less than it should be, and
    it will trigger a warning in inet6_rt_notify() as follows:

    WARNING: CPU: 0 PID: 6082 at net/ipv6/route.c:6180 inet6_rt_notify+0x120/0x130
    ......
    Call Trace:
     <TASK>
     fib6_add_rt2node+0x685/0xa30
     fib6_add+0x96/0x1b0
     ip6_route_add+0x50/0xd0
     inet6_rtm_newroute+0x97/0xa0
     rtnetlink_rcv_msg+0x156/0x3d0
     netlink_rcv_skb+0x5a/0x110
     netlink_unicast+0x246/0x350
     netlink_sendmsg+0x250/0x4c0
     sock_sendmsg+0x66/0x70
     ___sys_sendmsg+0x7c/0xd0
     __sys_sendmsg+0x5d/0xb0
     do_syscall_64+0x3f/0x90
     entry_SYSCALL_64_after_hwframe+0x72/0xdc

    This bug can be reproduced by script:

    ip -6 addr add 2002::2/64 dev ens2
    ip -6 route add 100::/64 via 2002::1 dev ens2 metric 100

    for i in 10 20 30 40 50 60 70;
    do
            ip link add link ens2 name ipv_$i type ipvlan
            ip -6 addr add 2002::$i/64 dev ipv_$i
            ifconfig ipv_$i up
    done

    for i in 10 20 30 40 50 60;
    do
            ip -6 route append 100::/64 encap ip6 dst 2002::$i via 2002::1
    dev ipv_$i metric 100
    done

    ip -6 route append 100::/64 via 2002::1 dev ipv_70 metric 100

    This patch fixes it by adding nexthop_len of every siblings using
    rt6_nh_nlmsg_size().

    Fixes: beb1afac518d ("net: ipv6: Add support to dump multipath routes via RTA_MULTIPATH
 attribute")
    Signed-off-by: Lu Wei <luwei32@huawei.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Link: https://lore.kernel.org/r/20230222083629.335683-2-luwei32@huawei.com
```

```
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e0386eebcaedd1c26534f0add4f7e6211cdb267
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Tue Feb 14 20:45:33 2023 -0800

    drm/i915: move a Kconfig symbol to unbreak the menu presentation

    [ Upstream commit 0b93efca3659f6d55ed31cff6722dca5f6e4d6e2 ]

    Inserting a Kconfig symbol that does not have a dependency (DRM_I915_GVT)
    into a list of other symbols that do have a dependency (on DRM_I915)
    breaks the driver menu presentation in 'make *config'.

    Relocate the DRM_I915_GVT symbol so that it does not cause this
    problem.

    Fixes: 8b750bf74418 ("drm/i915/gvt: move the gvt code into kvmgt.ko")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: Zhi Wang <zhi.a.wang@intel.com>
    Cc: Jani Nikula <jani.nikula@linux.intel.com>
    Cc: Joonas Lahtinen <joonas.lahtinen@linux.intel.com>
    Cc: Rodrigo Vivi <rodrigo.vivi@intel.com>
    Cc: Tvrtko Ursulin <tvrtko.ursulin@linux.intel.com>
    Cc: Zhenyu Wang <zhenyuw@linux.intel.com>
    Cc: intel-gfx@lists.freedesktop.org
    Cc: intel-gvt-dev@lists.freedesktop.org
    Cc: dri-devel@lists.freedesktop.org
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Acked-by: Zhenyu Wang <zhenyuw@linux.intel.com>
    Signed-off-by: Zhenyu Wang <zhenyuw@linux.intel.com>
    Link: http://patchwork.freedesktop.org/patch/msgid/20230215044533.4847-1-
rdunlap@infradead.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 20c809eca5cf347f0137d98b218cb3e6c5596f84
Author: Íñigo Huguet <ihuguet@redhat.com>
Date:   Tue Feb 21 14:06:16 2023 +0100

    ptp: vclock: use mutex to fix "sleep on atomic" bug

    [ Upstream commit 67d93ffc0f3c47094750bde6d62e7c5765dc47a6 ]

    vclocks were using spinlocks to protect access to its timecounter and
    cyclecounter. Access to timecounter/cyclecounter is backed by the same
    driver callbacks that are used for non-virtual PHCs, but the usage of
    the spinlock imposes a new limitation that didn't exist previously: now
    they're called in atomic context so they mustn't sleep.

    Some drivers like sfc or ice may sleep on these callbacks, causing
    errors like "BUG: scheduling while atomic: ptp5/25223/0x00000002"

    Fix it replacing the vclock's spinlock by a mutex. It fix the mentioned
    bug and it doesn't introduce longer delays.

    I've tested synchronizing various different combinations of clocks:
    - vclock->sysclock
    - sysclock->vclock
    - vclock->vclock
    - hardware PHC in different NIC -> vclock
    - created 4 vclocks and launch 4 parallel phc2sys processes with
      lockdep enabled
```

```
    In all cases, comparing the delays reported by phc2sys, they are in the
    same range of values than before applying the patch.

    Link: https://lore.kernel.org/netdev/69d0ff33-bd32-6aa5-d36c-fbdc3c01337c@redhat.com/
    Fixes: 5d43f951b1ac ("ptp: add ptp virtual clock driver framework")
    Reported-by: Yalin Li <yalli@redhat.com>
    Suggested-by: Richard Cochran <richardcochran@gmail.com>
    Tested-by: Miroslav Lichvar <mlichvar@redhat.com>
    Signed-off-by: Íñigo Huguet <ihuguet@redhat.com>
    Acked-by: Richard Cochran <richardcochran@gmail.com>
    Link: https://lore.kernel.org/r/20230221130616.21837-1-ihuguet@redhat.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 84412f3a8e8d46893ed42d9fe6638cc81d440ae1
Author: Randy Dunlap <rdunlap@infradead.org>
Date:   Tue Feb 21 23:04:11 2023 -0800

    swiotlb: mark swiotlb_memblock_alloc() as __init

    [ Upstream commit 9b07d27d0fbb7f7441aa986859a0f53ec93a0335 ]

    swiotlb_memblock_alloc() calls memblock_alloc(), which calls
    (__init) memblock_alloc_try_nid(). However, swiotlb_membloc_alloc()
    can be marked as __init since it is only called by swiotlb_init_remap(),
    which is already marked as __init. This prevents a modpost build
    warning/error:

    WARNING: modpost: vmlinux.o: section mismatch in reference: swiotlb_memblock_alloc
(section: .text) -> memblock_alloc_try_nid (section: .init.text)
    WARNING: modpost: vmlinux.o: section mismatch in reference: swiotlb_memblock_alloc
(section: .text) -> memblock_alloc_try_nid (section: .init.text)

    This fixes the build warning/error seen on ARM64, PPC64, S390, i386,
    and x86_64.

    Fixes: 8d58aa484920 ("swiotlb: reduce the swiotlb buffer size on allocation failure")
    Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
    Cc: Alexey Kardashevskiy <aik@amd.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: iommu@lists.linux.dev
    Cc: Mike Rapoport <rppt@kernel.org>
    Cc: linux-mm@kvack.org
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3cc9610a87b7dde82f7360dd4eb6c2c27940ed57
Author: Pavel Tikhomirov <ptikhomirov@virtuozzo.com>
Date:   Mon Feb 13 12:25:05 2023 +0800

    netfilter: x_tables: fix percpu counter block leak on error path when creating new netns

    [ Upstream commit 0af8c09c896810879387decfba8c942994bb61f5 ]

    Here is the stack where we allocate percpu counter block:

      +-< __alloc_percpu
        +-< xt_percpu_counter_alloc
          +-< find_check_entry # {arp,ip,ip6}_tables.c
            +-< translate_table

    And it can be leaked on this code path:

      +-> ip6t_register_table
        +-> translate_table # allocates percpu counter block
```

```
        +-> xt_register_table # fails

    there is no freeing of the counter block on xt_register_table fail.
    Note: xt_percpu_counter_free should be called to free it like we do in
    do_replace through cleanup_entry helper (or in __ip6t_unregister_table).

    Probability of hitting this error path is low AFAICS (xt_register_table
    can only return ENOMEM here, as it is not replacing anything, as we are
    creating new netns, and it is hard to imagine that all previous
    allocations succeeded and after that one in xt_register_table failed).
    But it's worth fixing even the rare leak.

    Fixes: 71ae0dff02d7 ("netfilter: xtables: use percpu rule counters")
    Signed-off-by: Pavel Tikhomirov <ptikhomirov@virtuozzo.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e2de561ebb79028257e260eb9181b8fd034756bb
Author: Florian Westphal <fw@strlen.de>
Date:   Mon Feb 20 17:24:00 2023 +0100

    netfilter: ctnetlink: make event listener tracking global

    [ Upstream commit fdf6491193e411087ae77bcbc6468e3e1cff99ed ]

    pernet tracking doesn't work correctly because other netns might have
    set NETLINK_LISTEN_ALL_NSID on its event socket.

    In this case its expected that events originating in other net
    namespaces are also received.

    Making pernet-tracking work while also honoring NETLINK_LISTEN_ALL_NSID
    requires much more intrusive changes both in netlink and nfnetlink,
    f.e. adding a 'setsockopt' callback that lets nfnetlink know that the
    event socket entered (or left) ALL_NSID mode.

    Move to global tracking instead: if there is an event socket anywhere
    on the system, all net namespaces which have conntrack enabled and
    use autobind mode will allocate the ecache extension.

    netlink_has_listeners() returns false only if the given group has no
    subscribers in any net namespace, the 'net' argument passed to
    nfnetlink_has_listeners is only used to derive the protocol (nfnetlink),
    it has no other effect.

    For proper NETLINK_LISTEN_ALL_NSID-aware pernet tracking of event
    listeners a new netlink_has_net_listeners() is also needed.

    Fixes: 90d1daa45849 ("netfilter: conntrack: add nf_conntrack_events autodetect mode")
    Reported-by: Bryce Kahle <bryce.kahle@datadoghq.com>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ba97e4e9268a4f885a33ffab6a3aa862714c5a02
Author: Xin Long <lucien.xin@gmail.com>
Date:   Fri Feb 17 18:22:57 2023 -0500

    netfilter: xt_length: use skb len to match in length_mt6

    [ Upstream commit 05c07c0c6cc8ec2278ace9871618c41f1365d1f5 ]

    For IPv6 Jumbo packets, the ipv6_hdr(skb)->payload_len is always 0,
    and its real payload_len ( > 65535) is saved in hbh exthdr. With 0
    length for the jumbo packets, it may mismatch.
```

```
    To fix this, we can just use skb->len instead of parsing exthdrs, as
    the hbh exthdr parsing has been done before coming to length_mt6 in
    ip6_rcv_core() and br_validate_ipv6() and also the packet has been
    trimmed according to the correct IPv6 (ext)hdr length there, and skb
    len is trustable in length_mt6().

    Note that this patch is especially needed after the IPv6 BIG TCP was
    supported in kernel, which is using IPv6 Jumbo packets. Besides, to
    match the packets greater than 65535 more properly, a v1 revision of
    xt_length may be needed to extend "min, max" to u32 in the future,
    and for now the IPv6 Jumbo packets can be matched by:

      # ip6tables -m length ! --length 0:65535

    Fixes: 7c4e983c4f3c ("net: allow gso_max_size to exceed 65536")
    Fixes: 0fe79f28bfaf ("net: allow gro_max_size to exceed 65536")
    Signed-off-by: Xin Long <lucien.xin@gmail.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cda0e0243bd3c04008fcd37a46b0269fb3c49249
Author: Florian Westphal <fw@strlen.de>
Date:   Fri Feb 17 23:20:06 2023 +0100

    netfilter: ebtables: fix table blob use-after-free

    [ Upstream commit e58a171d35e32e6e8c37cfe0e8a94406732a331f ]

    We are not allowed to return an error at this point.
    Looking at the code it looks like ret is always 0 at this
    point, but its not.

    t = find_table_lock(net, repl->name, &ret, &ebt_mutex);

    ... this can return a valid table, with ret != 0.

    This bug causes update of table->private with the new
    blob, but then frees the blob right away in the caller.

    Syzbot report:

    BUG: KASAN: vmalloc-out-of-bounds in __ebt_unregister_table+0xc00/0xcd0
net/bridge/netfilter/ebtables.c:1168
    Read of size 4 at addr ffffc90005425000 by task kworker/u4:4/74
    Workqueue: netns cleanup_net
    Call Trace:
     kasan_report+0xbf/0x1f0 mm/kasan/report.c:517
     __ebt_unregister_table+0xc00/0xcd0 net/bridge/netfilter/ebtables.c:1168
     ebt_unregister_table+0x35/0x40 net/bridge/netfilter/ebtables.c:1372
     ops_exit_list+0xb0/0x170 net/core/net_namespace.c:169
     cleanup_net+0x4ee/0xb10 net/core/net_namespace.c:613
    ...

    ip(6)tables appears to be ok (ret should be 0 at this point) but make
    this more obvious.

    Fixes: c58dd2dd443c ("netfilter: Can't fail and free after table replacement")
    Reported-by: syzbot+f61594de72d6705aea03@syzkaller.appspotmail.com
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ec9c4c412e3a2f58e8f89c6e6fd6e2fb355fd805
Author: Phil Sutter <phil@nwl.cc>
```

```
Date:   Thu Feb 16 17:05:36 2023 +0100

    netfilter: ip6t_rpfilter: Fix regression with VRF interfaces

    [ Upstream commit efb056e5f1f0036179b2f92c1c15f5ea7a891d70 ]

    When calling ip6_route_lookup() for the packet arriving on the VRF
    interface, the result is always the real (slave) interface. Expect this
    when validating the result.

    Fixes: acc641ab95b66 ("netfilter: rpfilter/fib: Populate flowic_l3mdev field")
    Signed-off-by: Phil Sutter <phil@nwl.cc>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f89d2650bb90df75ac8b81e111646cdce636477e
Author: Florian Westphal <fw@strlen.de>
Date:   Tue Feb 14 17:23:59 2023 +0100

    netfilter: conntrack: fix rmmod double-free race

    [ Upstream commit e6d57e9ff0aec323717ee36fc9ea34ad89217151 ]

    nf_conntrack_hash_check_insert() callers free the ct entry directly, via
    nf_conntrack_free.

    This isn't safe anymore because
    nf_conntrack_hash_check_insert() might place the entry into the conntrack
    table and then delteted the entry again because it found that a conntrack
    extension has been removed at the same time.

    In this case, the just-added entry is removed again and an error is
    returned to the caller.

    Problem is that another cpu might have picked up this entry and
    incremented its reference count.

    This results in a use-after-free/double-free, once by the other cpu and
    once by the caller of nf_conntrack_hash_check_insert().

    Fix this by making nf_conntrack_hash_check_insert() not fail anymore
    after the insertion, just like before the 'Fixes' commit.

    This is safe because a racing nf_ct_iterate() has to wait for us
    to release the conntrack hash spinlocks.

    While at it, make the function return -EAGAIN in the rmmod (genid
    changed) case, this makes nfnetlink replay the command (suggested
    by Pablo Neira).

    Fixes: c56716c69ce1 ("netfilter: extensions: introduce extension genid count")
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 033ac6ea4b513f9a4a20882f431f68cea307ba87
Author: Hangyu Hua <hbh25y@gmail.com>
Date:   Fri Feb 10 15:17:30 2023 +0800

    netfilter: ctnetlink: fix possible refcount leak in ctnetlink_create_conntrack()

    [ Upstream commit ac4893980bbe79ce383daf9a0885666a30fe4c83 ]

    nf_ct_put() needs to be called to put the refcount got by
    nf_conntrack_find_get() to avoid refcount leak when
```

```
    nf_conntrack_hash_check_insert() fails.

    Fixes: 7d367e06688d ("netfilter: ctnetlink: fix soft lockup when netlink adds new
entries (v2)")
    Signed-off-by: Hangyu Hua <hbh25y@gmail.com>
    Acked-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 44083720a0f044b3c2f1cdaf64477412f93f5029
Author: George Cherian <george.cherian@marvell.com>
Date:   Thu Feb 9 02:11:17 2023 +0000

    watchdog: sbsa_wdog: Make sure the timeout programming is within the limits

    [ Upstream commit 000987a38b53c172f435142a4026dd71378ca464 ]

    Make sure to honour the max_hw_heartbeat_ms while programming the timeout
    value to WOR. Clamp the timeout passed to sbsa_gwdt_set_timeout() to
    make sure the programmed value is within the permissible range.

    Fixes: abd3ac7902fb ("watchdog: sbsa: Support architecture version 1")

    Signed-off-by: George Cherian <george.cherian@marvell.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20230209021117.1512097-1-george.cherian@marvell.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 114e9767ab59c5b4fc41e812fdc95c0ee4691bc2
Author: Li Hua <hucool.lihua@huawei.com>
Date:   Wed Nov 16 10:07:06 2022 +0800

    watchdog: pcwd_usb: Fix attempting to access uninitialized memory

    [ Upstream commit 7d06c07c67100fd0f8e6b3ab7145ce789f788117 ]

    The stack variable msb and lsb may be used uninitialized in function
    usb_pcwd_get_temperature and usb_pcwd_get_timeleft when usb card no response.

    The build waring is:
    drivers/watchdog/pcwd_usb.c:336:22: error: ‘lsb’ is used uninitialized in this function
[-Werror=uninitialized]
      *temperature = (lsb * 9 / 5) + 32;
                     ~~~~^~~
    drivers/watchdog/pcwd_usb.c:328:21: note: ‘lsb’ was declared here
      unsigned char msb, lsb;
                         ^~~
    cc1: all warnings being treated as errors
    scripts/Makefile.build:250: recipe for target 'drivers/watchdog/pcwd_usb.o' failed
    make[3]: *** [drivers/watchdog/pcwd_usb.o] Error 1

    Fixes: b7e04f8c61a4 ("mv watchdog tree under drivers")
    Signed-off-by: Li Hua <hucool.lihua@huawei.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20221116020706.70847-1-hucool.lihua@huawei.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 50808d034e199fe3ff7a9d2068a4eebeb6b4098a
Author: Chen Jun <chenjun102@huawei.com>
Date:   Wed Nov 16 01:27:14 2022 +0000
```

```
    watchdog: Fix kmemleak in watchdog_cdev_register

    [ Upstream commit 13721a2ac66b246f5802ba1b75ad8637e53eeecc ]

    kmemleak reports memory leaks in watchdog_dev_register, as follows:
    unreferenced object 0xffff888116233000 (size 2048):
      comm ""modprobe"", pid 28147, jiffies 4353426116 (age 61.741s)
      hex dump (first 32 bytes):
        80 fa b9 05 81 88 ff ff 08 30 23 16 81 88 ff ff  .........0#.....
        08 30 23 16 81 88 ff ff 00 00 00 00 00 00 00 00  .0#.............
      backtrace:
        [<000000007f001ffd>] __kmem_cache_alloc_node+0x157/0x220
        [<000000006a389304>] kmalloc_trace+0x21/0x110
        [<000000008d640eea>] watchdog_dev_register+0x4e/0x780 [watchdog]
        [<0000000053c9f248>] __watchdog_register_device+0x4f0/0x680 [watchdog]
        [<00000000b2979824>] watchdog_register_device+0xd2/0x110 [watchdog]
        [<000000001f730178>] 0xffffffffc10880ae
        [<000000007a1a8bcc>] do_one_initcall+0xcb/0x4d0
        [<00000000b98be325>] do_init_module+0x1ca/0x5f0
        [<0000000046d08e7c>] load_module+0x6133/0x70f0
        ...

    unreferenced object 0xffff888105b9fa80 (size 16):
      comm ""modprobe"", pid 28147, jiffies 4353426116 (age 61.741s)
      hex dump (first 16 bytes):
        77 61 74 63 68 64 6f 67 31 00 b9 05 81 88 ff ff  watchdog1.......
      backtrace:
        [<000000007f001ffd>] __kmem_cache_alloc_node+0x157/0x220
        [<00000000486ab89b>] __kmalloc_node_track_caller+0x44/0x1b0
        [<000000005a39aab0>] kvasprintf+0xb5/0x140
        [<0000000024806f85>] kvasprintf_const+0x55/0x180
        [<000000009276cb7f>] kobject_set_name_vargs+0x56/0x150
        [<00000000a92e820b>] dev_set_name+0xab/0xe0
        [<00000000cec812c6>] watchdog_dev_register+0x285/0x780 [watchdog]
        [<0000000053c9f248>] __watchdog_register_device+0x4f0/0x680 [watchdog]
        [<00000000b2979824>] watchdog_register_device+0xd2/0x110 [watchdog]
        [<000000001f730178>] 0xffffffffc10880ae
        [<000000007a1a8bcc>] do_one_initcall+0xcb/0x4d0
        [<00000000b98be325>] do_init_module+0x1ca/0x5f0
        [<0000000046d08e7c>] load_module+0x6133/0x70f0
        ...

    The reason is that put_device is not be called if cdev_device_add fails
    and wdd->id != 0.

    watchdog_cdev_register
      wd_data = kzalloc                             [1]
      err = dev_set_name                            [2]
      ..
      err = cdev_device_add
      if (err) {
        if (wdd->id == 0) {  // wdd->id != 0
          ..
        }
        return err;  // [1],[2] would be leaked

    To fix it, call put_device in all wdd->id cases.

    Fixes: 72139dfa2464 ("watchdog: Fix the race between the release of watchdog_core_data
  and cdev")
    Signed-off-by: Chen Jun <chenjun102@huawei.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20221116012714.102066-1-chenjun102@huawei.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
```

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

commit 755a5c8c9242647278c4e15814ec790acc7463d4
Author: ruanjinjie <ruanjinjie@huawei.com>
Date:   Wed Nov 16 17:49:50 2022 +0800

    watchdog: at91sam9_wdt: use devm_request_irq to avoid missing free_irq() in error path

    [ Upstream commit 07bec0e09c1afbab4c5674fd2341f4f52d594f30 ]

    free_irq() is missing in case of error in at91_wdt_init(), use
    devm_request_irq to fix that.

    Fixes: 5161b31dc39a ("watchdog: at91sam9_wdt: better watchdog support")
    Signed-off-by: ruanjinjie <ruanjinjie@huawei.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Link: https://lore.kernel.org/r/20221116094950.3141943-1-ruanjinjie@huawei.com
    [groeck: Adjust multi-line alignment]
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4d4a5282a15f237b943844c50ae583faa91d94f9
Author: Fabrizio Castro <fabrizio.castro.jz@renesas.com>
Date:   Thu Nov 17 11:49:07 2022 +0000

    watchdog: rzg2l_wdt: Handle TYPE-B reset for RZ/V2M

    [ Upstream commit f769f97917c1e756e12ff042a93f6e3167254b5b ]

    As per section 48.4 of the HW User Manual, IPs in the RZ/V2M
    SoC need either a TYPE-A reset sequence or a TYPE-B reset
    sequence. More specifically, the watchdog IP needs a TYPE-B
    reset sequence.

    If the proper reset sequence isn't implemented, then resetting
    IPs may lead to undesired behaviour. In the restart callback of
    the watchdog driver the reset has basically no effect on the
    desired funcionality, as the register writes following the reset
    happen before the IP manages to come out of reset.

    Implement the TYPE-B reset sequence in the watchdog driver to
    address the issues with the restart callback on RZ/V2M.

    Fixes: ec122fd94eeb ("watchdog: rzg2l_wdt: Add rzv2m support")
    Signed-off-by: Fabrizio Castro <fabrizio.castro.jz@renesas.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Link: https://lore.kernel.org/r/20221117114907.138583-3-fabrizio.castro.jz@renesas.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fa9e1c9faa2fcad4ab7a41ce855110d7c8b65a47
Author: Lad Prabhakar <prabhakar.mahadev-lad.rj@bp.renesas.com>
Date:   Thu Nov 17 11:49:06 2022 +0000

    watchdog: rzg2l_wdt: Issue a reset before we put the PM clocks

    [ Upstream commit 6ba6f0f5910d5916539268c0ad55657bb8940616 ]

    On RZ/Five SoC it was observed that setting timeout (to say 1 sec) wouldn't
    reset the system.

    The procedure described in the HW manual (Procedure for Activating Modules)
```

```
    for activating the target module states we need to start supply of the
    clock module before applying the reset signal. This patch makes sure we
    follow the same procedure to clear the registers of the WDT module, fixing
    the issues seen on RZ/Five SoC.

    While at it re-used rzg2l_wdt_stop() in rzg2l_wdt_set_timeout() as it has
    the same function calls.

    Fixes: 4055ee81009e ("watchdog: rzg2l_wdt: Add set_timeout callback")
    Signed-off-by: Lad Prabhakar <prabhakar.mahadev-lad.rj@bp.renesas.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Reviewed-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Biju Das <biju.das.jz@bp.renesas.com>
    Link: https://lore.kernel.org/r/20221117114907.138583-2-fabrizio.castro.jz@renesas.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b7724360714642099cec907f54f42e55f5325453
Author: Daeho Jeong <daehojeong@google.com>
Date:   Thu Feb 9 10:18:19 2023 -0800

    f2fs: synchronize atomic write aborts

    [ Upstream commit a46bebd502fe1a3bd1d22f64cedd93e7e7702693 ]

    To fix a race condition between atomic write aborts, I use the inode
    lock and make COW inode to be re-usable thoroughout the whole
    atomic file inode lifetime.

    Reported-by: syzbot+823000d23b3400619f7c@syzkaller.appspotmail.com
    Fixes: 3db1de0e582c ("f2fs: change the current atomic write way")
    Signed-off-by: Daeho Jeong <daehojeong@google.com>
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3015314490af0d124b489e060f7dc09e5f55d81
Author: Benjamin Berg <benjamin.berg@intel.com>
Date:   Thu Feb 9 10:00:05 2023 +0100

    um: virt-pci: properly remove PCI device from bus

    [ Upstream commit 339b84dcd7113dd076419ea2a47128cc53450305 ]

    Triggering a bus rescan will not cause the PCI device to be removed. It
    is required to explicitly stop and remove the device from the bus.

    Fixes: 68f5d3f3b654 ("um: add PCI over virtio emulation driver")
    Signed-off-by: Benjamin Berg <benjamin.berg@intel.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb2d169e6339ee8d59f93c5a5737577811545298
Author: Benjamin Berg <benjamin.berg@intel.com>
Date:   Thu Feb 9 10:00:04 2023 +0100

    um: virtio_uml: move device breaking into workqueue

    [ Upstream commit abdeb4fa5e1b5b4918034f02236fd886f40c20c1 ]

    We should not be calling virtio_break_device from an IRQ context.
    Move breaking the device into the workqueue so that it is done from
    a reasonable context.
```

```
    Fixes: af9fb41ed315 ("um: virtio_uml: Fix broken device handling in time-travel")
    Signed-off-by: Benjamin Berg <benjamin.berg@intel.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9f63b3340aa31ad9182f588abb76d7b5aca45484
Author: Benjamin Berg <benjamin.berg@intel.com>
Date:   Thu Feb 9 10:00:03 2023 +0100

    um: virtio_uml: mark device as unregistered when breaking it

    [ Upstream commit 8e9cd85139a2149d5a7c121b05e0cdb8287311f9 ]

    Mark the device as not registered anymore when scheduling the work to
    remove it. Otherwise we could end up scheduling the work multiple times
    in a row, including scheduling it while it is already running.

    Fixes: af9fb41ed315 ("um: virtio_uml: Fix broken device handling in time-travel")
    Signed-off-by: Benjamin Berg <benjamin.berg@intel.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a6fbcf92f10461d4f9c5b635483f31e2d0347191
Author: Benjamin Berg <benjamin.berg@intel.com>
Date:   Thu Feb 9 10:00:02 2023 +0100

    um: virtio_uml: free command if adding to virtqueue failed

    [ Upstream commit 8a6ca543646f2940832665dbf4e04105262505e2 ]

    If adding the command fails (i.e. the virtqueue is broken) then free it
    again if the function allocated a new buffer for it.

    Fixes: 68f5d3f3b654 ("um: add PCI over virtio emulation driver")
    Signed-off-by: Benjamin Berg <benjamin.berg@intel.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc23256a0bdef1dcf7ea63d8a772c3dcac461424
Author: Ammar Faizi <ammarfaizi2@gnuweeb.org>
Date:   Sat Dec 24 00:23:38 2022 +0700

    x86: um: vdso: Add '%rcx' and '%r11' to the syscall clobber list

    [ Upstream commit 5541992e512de8c9133110809f767bd1b54ee10d ]

    The 'syscall' instruction clobbers '%rcx' and '%r11', but they are not
    listed in the inline Assembly that performs the syscall instruction.

    No real bug is found. It wasn't buggy by luck because '%rcx' and '%r11'
    are caller-saved registers, and not used in the functions, and the
    functions are never inlined.

    Add them to the clobber list for code correctness.

    Fixes: f1c2bb8b9964ed31de988910f8b1cfb586d30091 ("um: implement a x86_64 vDSO")
    Signed-off-by: Ammar Faizi <ammarfaizi2@gnuweeb.org>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eee45d9ae088bda094747713ef20f38f881b1224
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Wed Feb 8 11:34:27 2023 +0100

    netfilter: nf_tables: allow to fetch set elements when table has an owner
```

```
    [ Upstream commit 92f3e96d642f5e05b9dc710c06fedc669f1b4f00 ]

    NFT_MSG_GETSETELEM returns -EPERM when fetching set elements that belong
    to table that has an owner. This results in empty set/map listing from
    userspace.

    Fixes: 6001a930ce03 ("netfilter: nftables: introduce table ownership")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae6d1de80b12510c7566b5f6eab8548f1e989c3c
Author: Wang Jianjian <wangjianjian3@huawei.com>
Date:   Mon Dec 19 09:51:28 2022 +0800

    ext4: don't show commit interval if it is zero

    [ Upstream commit 934b0de1e9fdea93c4c7f2e18915c54fae67bdc6 ]

    If commit interval is 0, it means using default value.

    Fixes: 6e47a3cc68fc ("ext4: get rid of super block and sbi from handle_mount_ops()")
    Signed-off-by: Wang Jianjian <wangjianjian3@huawei.com>
    Link: https://lore.kernel.org/r/20221219015128.876717-1-wangjianjian3@huawei.com
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8e9e78088fa8e7bf16d1a0a3ebc93d67060a3bb7
Author: Eric Biggers <ebiggers@google.com>
Date:   Fri Dec 16 21:02:12 2022 -0800

    ext4: use ext4_fc_tl_mem in fast-commit replay path

    [ Upstream commit 11768cfd98136dd8399480c60b7a5d3d3c7b109b ]

    To avoid 'sparse' warnings about missing endianness conversions, don't
    store native endianness values into struct ext4_fc_tl.  Instead, use a
    separate struct type, ext4_fc_tl_mem.

    Fixes: dcc5827484d6 ("ext4: factor out ext4_fc_get_tl()")
    Cc: Ye Bin <yebin10@huawei.com>
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/r/20221217050212.150665-1-ebiggers@kernel.org
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 273c118fb0adf8466b751335253ac798feadd078
Author: Yangtao Li <frank.li@vivo.com>
Date:   Mon Feb 6 22:43:08 2023 +0800

    f2fs: fix to set ipu policy

    [ Upstream commit c5bf83483382600988d7db5ffe9fcd1936b491fd ]

    For LFS mode, it should update outplace and no need inplace update.
    When using LFS mode for small-volume devices, IPU will not be used,
    and the OPU writing method is actually used, but F2FS_IPU_FORCE can
    be read from the ipu_policy node, which is different from the actual
    situation. And remount to lfs mode should be disallowed when
    f2fs ipu is enabled, let's fix it.

    Fixes: 84b89e5d943d ("f2fs: add auto tuning for small devices")
    Signed-off-by: Yangtao Li <frank.li@vivo.com>
    Reviewed-by: Chao Yu <chao@kernel.org>
```

```
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2f633a4c130b6090a9dd82caba9f02f8143dd9f
Author: Yangtao Li <frank.li@vivo.com>
Date:   Sat Nov 19 03:18:39 2022 +0800

    f2fs: introduce IS_F2FS_IPU_* macro

    [ Upstream commit fdb7ccc3f9cb316c399b072c7a75a106678eb421 ]

    IS_F2FS_IPU_* macro can be used to identify whether
    f2fs ipu related policies are enabled.

    BTW, convert to use BIT() instead of open code.

    Signed-off-by: Yangtao Li <frank.li@vivo.com>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Stable-dep-of: c5bf83483382 ("f2fs: fix to set ipu policy")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 37b382970207da05d27c9157fc951190b735cd95
Author: Stephen Boyd <swboyd@chromium.org>
Date:   Wed Jan 18 19:23:29 2023 -0800

    soc: qcom: stats: Populate all subsystem debugfs files

    [ Upstream commit acdbf5f9b2c492505145f6e50c65418521a547c4 ]

    This driver relies on SMEM to populate items for each subsystem before
    the device probes. The items in SMEM that are being looked for are
    populated by the subsystems lazily, and therefore may not exist until
    the device has booted. For example, if I build this driver into the
    kernel on Trogdor Lazor and boot up, I don't see a 'modem' debugfs file
    populated, because the modem boots and populates the SMEM item after
    this driver probes.

    Always populate the files for the subsystems if they're in SMEM, and
    make the qcom_subsystem_sleep_stats_show() function return 0 if the SMEM
    items still isn't there. This way we can run a simple command like

            grep ^ /sys/kernel/debug/qcom_stats/*

    and collect the subsystem sleep stats without interspersed errors or
    missing details entirely because this driver probed first.

    Fixes: 1d7724690344 ("soc: qcom: Add Sleep stats driver")
    Cc: Matthias Kaehlcke <mka@chromium.org>
    Signed-off-by: Stephen Boyd <swboyd@chromium.org>
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Link: https://lore.kernel.org/r/20230119032329.2909383-1-swboyd@chromium.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b083b4f61a7dd683d801354a96999c503f568082
Author: Chao Yu <chao@kernel.org>
Date:   Tue Jan 31 22:47:01 2023 +0800

    f2fs: fix to update age extent in f2fs_do_zero_range()

    [ Upstream commit a84153f939808102dfa10904aa0f743e734a3e1d ]

    We should update age extent in f2fs_do_zero_range() like we
    did in f2fs_truncate_data_blocks_range().

    Fixes: 71644dff4811 ("f2fs: add block_age-based extent cache")
```

```
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94674b5ba6cf515a65fed9487a653ca5c293257a
Author: Chao Yu <chao@kernel.org>
Date:   Tue Jan 31 22:47:00 2023 +0800

    f2fs: fix to update age extent correctly during truncation

    [ Upstream commit 8c0ed062ce27f6b7f0a568cb241e2b4dd2d9e6a6 ]

    nr_free may be less than len, we should update age extent cache
    w/ range [fofs, len] rather than [fofs, nr_free].

    Fixes: 71644dff4811 ("f2fs: add block_age-based extent cache")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 20b4f3de0f3932f71b4a8daf0671e517a8d98022
Author: Yangtao Li <frank.li@vivo.com>
Date:   Sat Jan 21 00:16:55 2023 +0800

    f2fs: fix to avoid potential memory corruption in __update_iostat_latency()

    [ Upstream commit 0dbbf0fb38d5ec5d4138d1aeaeb43d9217b9a592 ]

    Add iotype sanity check to avoid potential memory corruption.
    This is to fix the compile error below:

    fs/f2fs/iostat.c:231 __update_iostat_latency() error: buffer overflow
    'io_lat->peak_lat[type]' 3 <= 3

    vim +228 fs/f2fs/iostat.c

      211  static inline void __update_iostat_latency(struct bio_iostat_ctx
             *iostat_ctx,
      212                                     enum iostat_lat_type type)
      213  {
      214          unsigned long ts_diff;
      215          unsigned int page_type = iostat_ctx->type;
      216          struct f2fs_sb_info *sbi = iostat_ctx->sbi;
      217          struct iostat_lat_info *io_lat = sbi->iostat_io_lat;
      218          unsigned long flags;
      219
      220          if (!sbi->iostat_enable)
      221                  return;
      222
      223          ts_diff = jiffies - iostat_ctx->submit_ts;
      224          if (page_type >= META_FLUSH)
                                 ^^^^^^^^^^

      225                  page_type = META;
      226
      227          spin_lock_irqsave(&sbi->iostat_lat_lock, flags);
     @228          io_lat->sum_lat[type][page_type] += ts_diff;
                                       ^^^^^^^^^^
    Mixup between META_FLUSH and NR_PAGE_TYPE leads to memory corruption.

    Fixes: a4b6817625e7 ("f2fs: introduce periodic iostat io latency traces")
    Reported-by: kernel test robot <lkp@intel.com>
    Reported-by: Dan Carpenter <error27@gmail.com>
    Suggested-by: Chao Yu <chao@kernel.org>
    Suggested-by: Jaegeuk Kim <jaegeuk@kernel.org>
```

```
    Signed-off-by: Yangtao Li <frank.li@vivo.com>
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d856180ce8b08412e5595a1dcc8d64787d566c43
Author: Chao Yu <chao@kernel.org>
Date:   Sat Jan 28 18:32:26 2023 +0800

    f2fs: fix to handle F2FS_IOC_START_ATOMIC_REPLACE in f2fs_compat_ioctl()

    [ Upstream commit 933141e4eb49d8b48721e2377835063a1e8fb823 ]

    Otherwise, 32-bits binary call ioctl(F2FS_IOC_START_ATOMIC_REPLACE) will
    fail in 64-bits kernel.

    Fixes: 41e8f85a75fc ("f2fs: introduce F2FS_IOC_START_ATOMIC_REPLACE")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5af1c643184a5d09ff5b3f334077a4d0a163c677
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Mon Jun 13 14:59:04 2022 +0800

    ubi: ubi_wl_put_peb: Fix infinite loop when wear-leveling work failed

    [ Upstream commit 4d57a7333e26040f2b583983e1970d9d460e56b0 ]

    Following process will trigger an infinite loop in ubi_wl_put_peb():

            ubifs_bgt               ubi_bgt
    ubifs_leb_unmap
      ubi_leb_unmap
        ubi_eba_unmap_leb
          ubi_wl_put_peb    wear_leveling_worker
                              e1 = rb_entry(rb_first(&ubi->used)
                              e2 = get_peb_for_wl(ubi)
                              ubi_io_read_vid_hdr  // return err (flash fault)
                              out_error:
                                ubi->move_from = ubi->move_to = NULL
                                wl_entry_destroy(ubi, e1)
                                  ubi->lookuptbl[e->pnum] = NULL
          retry:
            e = ubi->lookuptbl[pnum];       // return NULL
            if (e == ubi->move_from) {      // NULL == NULL gets true
              goto retry;                   // infinite loop !!!

    $ top
      PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     COMMAND
      7676 root     20   0       0      0      0 R 100.0  0.0  ubifs_bgt0_0

    Fix it by:
     1) Letting ubi_wl_put_peb() returns directly if wearl leveling entry has
        been removed from 'ubi->lookuptbl'.
     2) Using 'ubi->wl_lock' protecting wl entry deletion to preventing an
        use-after-free problem for wl entry in ubi_wl_put_peb().

    Fetch a reproducer in [Link].

    Fixes: 43f9b25a9cdd7b1 ("UBI: bugfix: protect from volume removal")
    Fixes: ee59ba8b064f692 ("UBI: Fix stale pointers in ubi->lookuptbl")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216111
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
```

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

commit a100de2974d208cfca032179b02ed4d1a0a7f143
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Sat Jul 30 19:28:37 2022 +0800

    ubi: Fix UAF wear-leveling entry in eraseblk_count_seq_show()

    [ Upstream commit a240bc5c43130c6aa50831d7caaa02a1d84e1bce ]

    Wear-leveling entry could be freed in error path, which may be accessed
    again in eraseblk_count_seq_show(), for example:

    __erase_worker                eraseblk_count_seq_show
                                    wl = ubi->lookuptbl[*block_number]
                                    if (wl)
      wl_entry_destroy
        ubi->lookuptbl[e->pnum] = NULL
        kmem_cache_free(ubi_wl_entry_slab, e)
                                       erase_count = wl->ec  // UAF!

    Wear-leveling entry updating/accessing in ubi->lookuptbl should be
    protected by ubi->wl_lock, fix it by adding ubi->wl_lock to serialize
    wl entry accessing between wl_entry_destroy() and
    eraseblk_count_seq_show().

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216305
    Fixes: 7bccd12d27b7e3 ("ubi: Add debugfs file for tracking PEB state")
    Fixes: 801c135ce73d5d ("UBI: Unsorted Block Images")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit def4399071fc518d9f77c5d0f22d1c357261fb57
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Aug 9 15:06:19 2022 +0800

    ubi: fastmap: Fix missed fm_anchor PEB in wear-leveling after disabling fastmap

    [ Upstream commit 76f9476ece445a07aeb72df9d896cd563fb5b50f ]

    After disabling fastmap(ubi->fm_disabled = 1), fastmap won't be updated,
    fm_anchor PEB is missed being scheduled for erasing. Besides, fm_anchor
    PEB may have smallest erase count, it doesn't participate wear-leveling.
    The difference of erase count between fm_anchor PEB and other PEBs will
    be larger and larger later on.

    In which situation fastmap can be disabled? Initially, we have an UBI
    image with fastmap. Then the image will be atttached without module
    parameter 'fm_autoconvert', ubi turns to full scanning mode in one
    random attaching process(eg. bad fastmap caused by powercut), ubi
    fastmap is disabled since then.

    Fix it by not getting fm_anchor if fastmap is disabled in
    ubi_refill_pools().

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216341
    Fixes: 4b68bf9a69d22d ("ubi: Select fastmap anchor PEBs considering ...")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit bd188ff1c8a1935c93a1e3cacf3be62667fdf762
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Wed Jun 1 11:00:00 2022 +0800

    ubifs: ubifs_releasepage: Remove ubifs_assert(0) to valid this process

    [ Upstream commit 66f4742e93523ab2f062d9d9828b3e590bc61536 ]

    There are two states for ubifs writing pages:
    1. Dirty, Private
    2. Not Dirty, Not Private

    The normal process cannot go to ubifs_releasepage() which means there
    exists pages being private but not dirty. Reproducer[1] shows that it
    could occur (which maybe related to [2]) with following process:

         PA                      PB                    PC
    lock(page)[PA]
    ubifs_write_end
      attach_page_private         // set Private
      __set_page_dirty_nobuffers  // set Dirty
    unlock(page)

    write_cache_pages[PA]
      lock(page)
      clear_page_dirty_for_io(page) // clear Dirty
      ubifs_writepage

                            do_truncation[PB]
                              truncate_setsize
                                i_size_write(inode, newsize) // newsize = 0

        i_size = i_size_read(inode) // i_size = 0
        end_index = i_size >> PAGE_SHIFT
        if (page->index > end_index)
          goto out // jump
    out:
    unlock(page)   // Private, Not Dirty

                                                    generic_fadvise[PC]
                                                      lock(page)
                                                      invalidate_inode_page
                                                        try_to_release_page
                                                          ubifs_releasepage
                                                            ubifs_assert(c, 0)
                                                            // bad assertion!
                                                      unlock(page)
                              truncate_pagecache[PB]

    Then we may get following assertion failed:
      UBIFS error (ubi0:0 pid 1683): ubifs_assert_failed [ubifs]:
      UBIFS assert failed: 0, in fs/ubifs/file.c:1513
      UBIFS warning (ubi0:0 pid 1683): ubifs_ro_mode [ubifs]:
      switched to read-only mode, error -22
      CPU: 2 PID: 1683 Comm: aa Not tainted 5.16.0-rc5-00184-g0bca5994cacc-dirty #308
      Call Trace:
        dump_stack+0x13/0x1b
        ubifs_ro_mode+0x54/0x60 [ubifs]
        ubifs_assert_failed+0x4b/0x80 [ubifs]
        ubifs_releasepage+0x67/0x1d0 [ubifs]
        try_to_release_page+0x57/0xe0
        invalidate_inode_page+0xfb/0x130
        __invalidate_mapping_pages+0xb9/0x280
        invalidate_mapping_pagevec+0x12/0x20
```

```
         generic_fadvise+0x303/0x3c0
         ksys_fadvise64_64+0x4c/0xb0

    [1] https://bugzilla.kernel.org/show_bug.cgi?id=215373
    [2] https://linux-mtd.infradead.narkive.com/NQoBeT1u/patch-rfc-ubifs-fix-assert-failed-
in-ubifs-set-page-dirty

    Fixes: 1e51764a3c2ac0 ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1d7d02e9cb01622f3c2d95c95cc75833aa51d788
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Wed Jun 1 10:59:59 2022 +0800

    ubifs: ubifs_writepage: Mark page dirty after writing inode failed

    [ Upstream commit fb8bc4c74ae4526d9489362ab2793a936d072b84 ]

    There are two states for ubifs writing pages:
    1. Dirty, Private
    2. Not Dirty, Not Private

    There is a third possibility which maybe related to [1] that page is
    private but not dirty caused by following process:

              PA
    lock(page)
    ubifs_write_end
      attach_page_private           // set Private
        __set_page_dirty_nobuffers  // set Dirty
    unlock(page)

    write_cache_pages
      lock(page)
      clear_page_dirty_for_io(page) // clear Dirty
      ubifs_writepage
        write_inode
        // fail, goto out, following codes are not executed
        // do_writepage
        //   set_page_writeback     // set Writeback
        //   detach_page_private    // clear Private
        //   end_page_writeback     // clear Writeback
        out:
        unlock(page)                // Private, Not Dirty

                                          PB
                                    ksys_fadvise64_64
                                      generic_fadvise
                                        invalidate_inode_page
                                        // page is neither Dirty nor Writeback
                                          invalidate_complete_page
                                          // page_has_private is true
                                            try_to_release_page
                                              ubifs_releasepage
                                                ubifs_assert(c, 0) !!!

    Then we may get following assertion failed:
      UBIFS error (ubi0:0 pid 1492): ubifs_assert_failed [ubifs]:
      UBIFS assert failed: 0, in fs/ubifs/file.c:1499
      UBIFS warning (ubi0:0 pid 1492): ubifs_ro_mode [ubifs]:
      switched to read-only mode, error -22
      CPU: 2 PID: 1492 Comm: aa Not tainted 5.16.0-rc2-00012-g7bb767dee0ba-dirty
      Call Trace:
```

```
         dump_stack+0x13/0x1b
         ubifs_ro_mode+0x54/0x60 [ubifs]
         ubifs_assert_failed+0x4b/0x80 [ubifs]
         ubifs_releasepage+0x7e/0x1e0 [ubifs]
         try_to_release_page+0x57/0xe0
         invalidate_inode_page+0xfb/0x130
         invalidate_mapping_pagevec+0x12/0x20
         generic_fadvise+0x303/0x3c0
         vfs_fadvise+0x35/0x40
         ksys_fadvise64_64+0x4c/0xb0

    Jump [2] to find a reproducer.

    [1] https://linux-mtd.infradead.narkive.com/NQoBeT1u/patch-rfc-ubifs-fix-assert-failed-
in-ubifs-set-page-dirty
    [2] https://bugzilla.kernel.org/show_bug.cgi?id=215357

    Fixes: 1e51764a3c2ac0 ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bbcf899036eeb98c3a938fb08946c6b52771f71c
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Fri Nov 18 17:02:36 2022 +0800

    ubifs: dirty_cow_znode: Fix memleak in error handling path

    [ Upstream commit 122deabfe1428bffe95e2bf364ff8a5059bdf089 ]

    Following process will cause a memleak for copied up znode:

    dirty_cow_znode
      zn = copy_znode(c, znode);
      err = insert_old_idx(c, zbr->lnum, zbr->offs);
      if (unlikely(err))
         return ERR_PTR(err);   // No one refers to zn.

    Fix it by adding copied znode back to tnc, then it will be freed
    by ubifs_destroy_tnc_subtree() while closing tnc.

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216705
    Fixes: 1e51764a3c2a ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 42b71f3f74cd84108cc02dc569e248c32a8d83ff
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Fri Nov 18 17:02:35 2022 +0800

    ubifs: Re-statistic cleaned znode count if commit failed

    [ Upstream commit 944e096aa24071d3fe22822f6249d3ae309e39ea ]

    Dirty znodes will be written on flash in committing process with
    following states:

                  process A                 |  znode state
    ------------------------------------------------------
    do_commit                               | DIRTY_ZNODE
      ubifs_tnc_start_commit                | DIRTY_ZNODE
       get_znodes_to_commit                 | DIRTY_ZNODE | COW_ZNODE
```

```
         layout_commit                        | DIRTY_ZNODE | COW_ZNODE
          fill_gap                            | 0
       write master                           | 0 or OBSOLETE_ZNODE

                    process B                 |  znode state
    ------------------------------------------------------
    do_commit                                 | DIRTY_ZNODE[1]
      ubifs_tnc_start_commit                  | DIRTY_ZNODE
       get_znodes_to_commit                   | DIRTY_ZNODE | COW_ZNODE
      ubifs_tnc_end_commit                    | DIRTY_ZNODE | COW_ZNODE
       write_index                            | 0
      write master                            | 0 or OBSOLETE_ZNODE[2] or
                                              | DIRTY_ZNODE[3]

    [1] znode is dirtied without concurrent committing process
    [2] znode is copied up (re-dirtied by other process) before cleaned
        up in committing process
    [3] znode is re-dirtied after cleaned up in committing process

    Currently, the clean znode count is updated in free_obsolete_znodes(),
    which is called only in normal path. If do_commit failed, clean znode
    count won't be updated, which triggers a failure ubifs assertion[4] in
    ubifs_tnc_close():
     ubifs_assert_failed [ubifs]: UBIFS assert failed: freed == n

    [4] Commit 380347e9ca7682 ("UBIFS: Add an assertion for clean_zn_cnt").

    Fix it by re-statisticing cleaned znode count in tnc_destroy_cnext().

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216704
    Fixes: 1e51764a3c2a ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9eccdb0760cbcb4427b5303a83a3007de998af51
Author: Yang Yingliang <yangyingliang@huawei.com>
Date:   Mon Nov 14 18:26:24 2022 +0800

    ubi: Fix possible null-ptr-deref in ubi_free_volume()

    [ Upstream commit c15859bfd326c10230f09cb48a17f8a35f190342 ]

    It willl cause null-ptr-deref in the following case:

    uif_init()
      ubi_add_volume()
        cdev_add() -> if it fails, call kill_volumes()
        device_register()

    kill_volumes() -> if ubi_add_volume() fails call this function
      ubi_free_volume()
        cdev_del()
        device_unregister() -> trying to delete a not added device,
                               it causes null-ptr-deref

    So in ubi_free_volume(), it delete devices whether they are added
    or not, it will causes null-ptr-deref.

    Handle the error case whlie calling ubi_add_volume() to fix this
    problem. If add volume fails, set the corresponding vol to null,
    so it can not be accessed in kill_volumes() and release the
    resource in ubi_add_volume() error path.
```

```
    Fixes: 801c135ce73d ("UBI: Unsorted Block Images")
    Suggested-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Yang Yingliang <yangyingliang@huawei.com>
    Reviewed-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e11f36d3bc4d23f620754a948fe7b82b63dcb185
Author: Li Zetao <lizetao1@huawei.com>
Date:   Sat Oct 22 19:52:11 2022 +0800

    ubifs: Fix memory leak in alloc_wbufs()

    [ Upstream commit 4a1ff3c5d04b9079b4f768d9a71b51c4af578dd2 ]

    kmemleak reported a sequence of memory leaks, and show them as following:

      unreferenced object 0xffff8881575f8400 (size 1024):
        comm "mount", pid 19625, jiffies 4297119604 (age 20.383s)
        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:
          [<ffffffff8176cecd>] __kmalloc+0x4d/0x150
          [<ffffffffa0406b2b>] ubifs_mount+0x307b/0x7170 [ubifs]
          [<ffffffff819fa8fd>] legacy_get_tree+0xed/0x1d0
          [<ffffffff81936f2d>] vfs_get_tree+0x7d/0x230
          [<ffffffff819b2bd4>] path_mount+0xdd4/0x17b0
          [<ffffffff819b37aa>] __x64_sys_mount+0x1fa/0x270
          [<ffffffff83c14295>] do_syscall_64+0x35/0x80
          [<ffffffff83e0006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0

      unreferenced object 0xffff8881798a6e00 (size 512):
        comm "mount", pid 19677, jiffies 4297121912 (age 37.816s)
        hex dump (first 32 bytes):
          6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
          6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
        backtrace:
          [<ffffffff8176cecd>] __kmalloc+0x4d/0x150
          [<ffffffffa0418342>] ubifs_wbuf_init+0x52/0x480 [ubifs]
          [<ffffffffa0406ca5>] ubifs_mount+0x31f5/0x7170 [ubifs]
          [<ffffffff819fa8fd>] legacy_get_tree+0xed/0x1d0
          [<ffffffff81936f2d>] vfs_get_tree+0x7d/0x230
          [<ffffffff819b2bd4>] path_mount+0xdd4/0x17b0
          [<ffffffff819b37aa>] __x64_sys_mount+0x1fa/0x270
          [<ffffffff83c14295>] do_syscall_64+0x35/0x80
          [<ffffffff83e0006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0

    The problem is that the ubifs_wbuf_init() returns an error in the
    loop which in the alloc_wbufs(), then the wbuf->buf and wbuf->inodes
    that were successfully alloced before are not freed.

    Fix it by adding error hanging path in alloc_wbufs() which frees
    the memory alloced before when ubifs_wbuf_init() returns an error.

    Fixes: 1e51764a3c2a ("UBIFS: add new flash file system")
    Signed-off-by: Li Zetao <lizetao1@huawei.com>
    Reviewed-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5c0c81a313492b83bd0c038b8839b0e04eb87563
Author: Li Zetao <lizetao1@huawei.com>
Date:   Fri Oct 21 18:21:57 2022 +0800
```

```
    ubi: Fix unreferenced object reported by kmemleak in ubi_resize_volume()

    [ Upstream commit 1e591ea072df7211f64542a09482b5f81cb3ad27 ]

    There is a memory leaks problem reported by kmemleak:

    unreferenced object 0xffff888102007a00 (size 128):
      comm "ubirsvol", pid 32090, jiffies 4298464136 (age 2361.231s)
      hex dump (first 32 bytes):
    ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
    ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
      backtrace:
    [<ffffffff8176cecd>] __kmalloc+0x4d/0x150
    [<ffffffffa02a9a36>] ubi_eba_create_table+0x76/0x170 [ubi]
    [<ffffffffa029764e>] ubi_resize_volume+0x1be/0xbc0 [ubi]
    [<ffffffffa02a3321>] ubi_cdev_ioctl+0x701/0x1850 [ubi]
    [<ffffffff81975d2d>] __x64_sys_ioctl+0x11d/0x170
    [<ffffffff83c142a5>] do_syscall_64+0x35/0x80
    [<ffffffff83e0006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0

    This is due to a mismatch between create and destroy interfaces, and
    in detail that "new_eba_tbl" created by ubi_eba_create_table() but
    destroyed by kfree(), while will causing "new_eba_tbl->entries" not
    freed.

    Fix it by replacing kfree(new_eba_tbl) with
    ubi_eba_destroy_table(new_eba_tbl)

    Fixes: 799dca34ac54 ("UBI: hide EBA internals")
    Signed-off-by: Li Zetao <lizetao1@huawei.com>
    Reviewed-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3d6378f7056ac7350338f941001162a8f660853c
Author: Li Zetao <lizetao1@huawei.com>
Date:   Fri Oct 21 18:21:56 2022 +0800

    ubi: Fix use-after-free when volume resizing failed

    [ Upstream commit 9af31d6ec1a4be4caab2550096c6bd2ba8fba472 ]

    There is an use-after-free problem reported by KASAN:
      ==================================================================
      BUG: KASAN: use-after-free in ubi_eba_copy_table+0x11f/0x1c0 [ubi]
      Read of size 8 at addr ffff888101eec008 by task ubirsvol/4735

      CPU: 2 PID: 4735 Comm: ubirsvol
      Not tainted 6.1.0-rc1-00003-g84fa3304a7fc-dirty #14
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996),
      BIOS 1.14.0-1.fc33 04/01/2014
      Call Trace:
       <TASK>
       dump_stack_lvl+0x34/0x44
       print_report+0x171/0x472
       kasan_report+0xad/0x130
       ubi_eba_copy_table+0x11f/0x1c0 [ubi]
       ubi_resize_volume+0x4f9/0xbc0 [ubi]
       ubi_cdev_ioctl+0x701/0x1850 [ubi]
       __x64_sys_ioctl+0x11d/0x170
       do_syscall_64+0x35/0x80
       entry_SYSCALL_64_after_hwframe+0x46/0xb0
       </TASK>
```

```
    When ubi_change_vtbl_record() returns an error in ubi_resize_volume(),
    "new_eba_tbl" will be freed on error handing path, but it is holded
    by "vol->eba_tbl" in ubi_eba_replace_table(). It means that the liftcycle
    of "vol->eba_tbl" and "vol" are different, so when resizing volume in
    next time, it causing an use-after-free fault.

    Fix it by not freeing "new_eba_tbl" after it replaced in
    ubi_eba_replace_table(), while will be freed in next volume resizing.

    Fixes: 801c135ce73d ("UBI: Unsorted Block Images")
    Signed-off-by: Li Zetao <lizetao1@huawei.com>
    Reviewed-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e6551ca100dfb3ff1c735fd839960a7609680fe
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Oct 11 11:47:32 2022 +0800

    ubifs: Reserve one leb for each journal head while doing budget

    [ Upstream commit e874dcde1cbf82c786c0e7f2899811c02630cc52 ]

    UBIFS calculates available space by c->main_bytes - c->lst.total_used
    (which means non-index lebs' free and dirty space is accounted into
    total available), then index lebs and four lebs (one for gc_lnum, one
    for deletions, two for journal heads) are deducted.
    In following situation, ubifs may get -ENOSPC from make_reservation():
     LEB 84: DATAHD   free 122880 used 1920  dirty 2176  dark 6144
     LEB 110:DELETION free 126976 used 0     dirty 0     dark 6144 (empty)
     LEB 201:gc_lnum  free 126976 used 0     dirty 0     dark 6144
     LEB 272:GCHD     free 77824  used 47672 dirty 1480  dark 6144
     LEB 356:BASEHD   free 0      used 39776 dirty 87200 dark 6144
     OTHERS: index lebs, zero-available non-index lebs

    UBIFS calculates the available bytes is 6888 (How to calculate it:
    126976 * 5[remain main bytes] - 1920[used] - 47672[used] - 39776[used] -
    126976 * 1[deletions] - 126976 * 1[gc_lnum] - 126976 * 2[journal heads]
    - 6144 * 5[dark] = 6888) after doing budget, however UBIFS cannot use
    BASEHD's dirty space(87200), because UBIFS cannot find next BASEHD to
    reclaim current BASEHD. (c->bi.min_idx_lebs equals to c->lst.idx_lebs,
    the empty leb won't be found by ubifs_find_free_space(), and dirty index
    lebs won't be picked as gced lebs. All non-index lebs has dirty space
    less then c->dead_wm, non-index lebs won't be picked as gced lebs
    either. So new free lebs won't be produced.). See more details in Link.

    To fix it, reserve one leb for each journal head while doing budget.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216562
    Fixes: 1e51764a3c2ac0 ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit af9ba369cbb9e974410a0def6bc0fd0bc5847f5b
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Oct 11 11:47:31 2022 +0800

    ubifs: do_rename: Fix wrong space budget when target inode's nlink > 1

    [ Upstream commit 25fce616a61fc2f1821e4a9ce212d0e064707093 ]

    If target inode is a special file (eg. block/char device) with nlink
    count greater than 1, the inode with ui->data will be re-written on
    disk. However, UBIFS losts target inode's data_len while doing space
```

```
    budget. Bad space budget may let make_reservation() return with -ENOSPC,
    which could turn ubifs to read-only mode in do_writepage() process.

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216494
    Fixes: 1e51764a3c2ac0 ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c37fe388b5eac9b1901775747e5c1566f31b64d7
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Oct 11 11:47:30 2022 +0800

    ubifs: Fix wrong dirty space budget for dirty inode

    [ Upstream commit b248eaf049d9cdc5eb76b59399e4d3de233f02ac ]

    Each dirty inode should reserve 'c->bi.inode_budget' bytes in space
    budget calculation. Currently, space budget for dirty inode reports
    more space than what UBIFS actually needs to write.

    Fixes: 1e51764a3c2ac0 ("UBIFS: add new flash file system")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 576facf5c34fd45150c2d073fd05acf25b7c3e1b
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Oct 11 11:47:28 2022 +0800

    ubifs: Rectify space budget for ubifs_xrename()

    [ Upstream commit 1b2ba09060e41adb356b9ae58ef94a7390928004 ]

    There is no space budget for ubifs_xrename(). It may let
    make_reservation() return with -ENOSPC, which could turn
    ubifs to read-only mode in do_writepage() process.
    Fix it by adding space budget for ubifs_xrename().

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216569
    Fixes: 9ec64962afb170 ("ubifs: Implement RENAME_EXCHANGE")
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e87eccc60f59d470e3e0ba3912ee73bff5e6e004
Author: Zhihao Cheng <chengzhihao1@huawei.com>
Date:   Tue Oct 11 11:47:27 2022 +0800

    ubifs: Rectify space budget for ubifs_symlink() if symlink is encrypted

    [ Upstream commit c2c36cc6ca23e614f9e4238d0ecf48549ee9002a ]

    Fix bad space budget when symlink file is encrypted. Bad space budget
    may let make_reservation() return with -ENOSPC, which could turn ubifs
    to read-only mode in do_writepage() process.

    Fetch a reproducer in [Link].

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216490
    Fixes: ca7f85be8d6cf9 ("ubifs: Add support for encrypted symlinks")
```

```
    Signed-off-by: Zhihao Cheng <chengzhihao1@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d42c2b18c42da7378e67b6414aafe93b65de89d1
Author: Liu Shixin <liushixin2@huawei.com>
Date:   Thu Oct 20 12:30:31 2022 +0800

    ubifs: Fix memory leak in ubifs_sysfs_init()

    [ Upstream commit 203a55f04f66eea1a1ca7e5a302a7f5c99c62327 ]

    When insmod ubifs.ko, a kmemleak reported as below:

     unreferenced object 0xffff88817fb1a780 (size 8):
       comm "insmod", pid 25265, jiffies 4295239702 (age 100.130s)
       hex dump (first 8 bytes):
         75 62 69 66 73 00 ff ff                          ubifs...
       backtrace:
         [<ffffffff81b3fc4c>] slab_post_alloc_hook+0x9c/0x3c0
         [<ffffffff81b44bf3>] __kmalloc_track_caller+0x183/0x410
         [<ffffffff8198d3da>] kstrdup+0x3a/0x80
         [<ffffffff8198d486>] kstrdup_const+0x66/0x80
         [<ffffffff83989325>] kvasprintf_const+0x155/0x190
         [<ffffffff83bf55bb>] kobject_set_name_vargs+0x5b/0x150
         [<ffffffff83bf576b>] kobject_set_name+0xbb/0xf0
         [<ffffffff8100204c>] do_one_initcall+0x14c/0x5a0
         [<ffffffff8157e380>] do_init_module+0x1f0/0x660
         [<ffffffff815857be>] load_module+0x6d7e/0x7590
         [<ffffffff8158644f>] __do_sys_finit_module+0x19f/0x230
         [<ffffffff815866b3>] __x64_sys_finit_module+0x73/0xb0
         [<ffffffff88c98e85>] do_syscall_64+0x35/0x80
         [<ffffffff88e00087>] entry_SYSCALL_64_after_hwframe+0x63/0xcd

    When kset_register() failed, we should call kset_put to cleanup it.

    Fixes: 2e3cbf425804 ("ubifs: Export filesystem error counters")
    Signed-off-by: Liu Shixin <liushixin2@huawei.com>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8b9836ea9d2c8efa70252c794a4e55788670a9c
Author: Li Hua <hucool.lihua@huawei.com>
Date:   Mon Nov 21 19:18:47 2022 +0800

    ubifs: Fix build errors as symbol undefined

    [ Upstream commit aa6d148e6d6270274e3d5a529b71c54cd329d17f ]

    With CONFIG_UBIFS_FS_AUTHENTICATION not set, the compiler can assume that
    ubifs_node_check_hash() is never true and drops the call to ubifs_bad_hash().
    Is CONFIG_CC_OPTIMIZE_FOR_SIZE enabled this optimization does not happen anymore.

    So When CONFIG_UBIFS_FS and CONFIG_CC_OPTIMIZE_FOR_SIZE is enabled but
    CONFIG_UBIFS_FS_AUTHENTICATION is not set, the build errors is as followd:
        ERROR: modpost: "ubifs_bad_hash" [fs/ubifs/ubifs.ko] undefined!

    Fix it by add no-op ubifs_bad_hash() for the CONFIG_UBIFS_FS_AUTHENTICATION=n case.

    Fixes: 16a26b20d2af ("ubifs: authentication: Add hashes to index nodes")
    Signed-off-by: Li Hua <hucool.lihua@huawei.com>
    Reviewed-by: Sascha Hauer <s.hauer@pengutronix.de>
    Signed-off-by: Richard Weinberger <richard@nod.at>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit e1b73fe4f4c6bb80755eb4bf4b867a8fd8b1a7fe
Author: George Kennedy <george.kennedy@oracle.com>
Date:   Tue Nov 15 10:14:44 2022 -0500

    ubi: ensure that VID header offset + VID header size <= alloc, size

    [ Upstream commit 1b42b1a36fc946f0d7088425b90d491b4257ca3e ]

    Ensure that the VID header offset + VID header size does not exceed
    the allocated area to avoid slab OOB.

    BUG: KASAN: slab-out-of-bounds in crc32_body lib/crc32.c:111 [inline]
    BUG: KASAN: slab-out-of-bounds in crc32_le_generic lib/crc32.c:179 [inline]
    BUG: KASAN: slab-out-of-bounds in crc32_le_base+0x58c/0x626 lib/crc32.c:197
    Read of size 4 at addr ffff88802bb36f00 by task syz-executor136/1555

    CPU: 2 PID: 1555 Comm: syz-executor136 Tainted: G        W
    6.0.0-1868 #1
    Hardware name: Red Hat KVM, BIOS 1.13.0-2.module+el8.3.0+7860+a7792d29
    04/01/2014
    Call Trace:
      <TASK>
      __dump_stack lib/dump_stack.c:88 [inline]
      dump_stack_lvl+0x85/0xad lib/dump_stack.c:106
      print_address_description mm/kasan/report.c:317 [inline]
      print_report.cold.13+0xb6/0x6bb mm/kasan/report.c:433
      kasan_report+0xa7/0x11b mm/kasan/report.c:495
      crc32_body lib/crc32.c:111 [inline]
      crc32_le_generic lib/crc32.c:179 [inline]
      crc32_le_base+0x58c/0x626 lib/crc32.c:197
      ubi_io_write_vid_hdr+0x1b7/0x472 drivers/mtd/ubi/io.c:1067
      create_vtbl+0x4d5/0x9c4 drivers/mtd/ubi/vtbl.c:317
      create_empty_lvol drivers/mtd/ubi/vtbl.c:500 [inline]
      ubi_read_volume_table+0x67b/0x288a drivers/mtd/ubi/vtbl.c:812
      ubi_attach+0xf34/0x1603 drivers/mtd/ubi/attach.c:1601
      ubi_attach_mtd_dev+0x6f3/0x185e drivers/mtd/ubi/build.c:965
      ctrl_cdev_ioctl+0x2db/0x347 drivers/mtd/ubi/cdev.c:1043
      vfs_ioctl fs/ioctl.c:51 [inline]
      __do_sys_ioctl fs/ioctl.c:870 [inline]
      __se_sys_ioctl fs/ioctl.c:856 [inline]
      __x64_sys_ioctl+0x193/0x213 fs/ioctl.c:856
      do_syscall_x64 arch/x86/entry/common.c:50 [inline]
      do_syscall_64+0x3e/0x86 arch/x86/entry/common.c:80
      entry_SYSCALL_64_after_hwframe+0x63/0x0
    RIP: 0033:0x7f96d5cf753d
    Code:
    RSP: 002b:00007fffd72206f8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010
    RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007f96d5cf753d
    RDX: 0000000020000080 RSI: 0000000040186f40 RDI: 0000000000000003
    RBP: 0000000000400cd0 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000400be0
    R13: 00007fffd72207e0 R14: 0000000000000000 R15: 0000000000000000
      </TASK>

    Allocated by task 1555:
      kasan_save_stack+0x20/0x3d mm/kasan/common.c:38
      kasan_set_track mm/kasan/common.c:45 [inline]
      set_alloc_info mm/kasan/common.c:437 [inline]
      ____kasan_kmalloc mm/kasan/common.c:516 [inline]
      __kasan_kmalloc+0x88/0xa3 mm/kasan/common.c:525
      kasan_kmalloc include/linux/kasan.h:234 [inline]
      __kmalloc+0x138/0x257 mm/slub.c:4429
      kmalloc include/linux/slab.h:605 [inline]
      ubi_alloc_vid_buf drivers/mtd/ubi/ubi.h:1093 [inline]
      create_vtbl+0xcc/0x9c4 drivers/mtd/ubi/vtbl.c:295
```

```
       create_empty_lvol drivers/mtd/ubi/vtbl.c:500 [inline]
       ubi_read_volume_table+0x67b/0x288a drivers/mtd/ubi/vtbl.c:812
       ubi_attach+0xf34/0x1603 drivers/mtd/ubi/attach.c:1601
       ubi_attach_mtd_dev+0x6f3/0x185e drivers/mtd/ubi/build.c:965
       ctrl_cdev_ioctl+0x2db/0x347 drivers/mtd/ubi/cdev.c:1043
       vfs_ioctl fs/ioctl.c:51 [inline]
       __do_sys_ioctl fs/ioctl.c:870 [inline]
       __se_sys_ioctl fs/ioctl.c:856 [inline]
       __x64_sys_ioctl+0x193/0x213 fs/ioctl.c:856
       do_syscall_x64 arch/x86/entry/common.c:50 [inline]
       do_syscall_64+0x3e/0x86 arch/x86/entry/common.c:80
       entry_SYSCALL_64_after_hwframe+0x63/0x0

     The buggy address belongs to the object at ffff88802bb36e00
       which belongs to the cache kmalloc-256 of size 256
     The buggy address is located 0 bytes to the right of
       256-byte region [ffff88802bb36e00, ffff88802bb36f00)

     The buggy address belongs to the physical page:
     page:00000000ea4d1263 refcount:1 mapcount:0 mapping:0000000000000000
     index:0x0 pfn:0x2bb36
     head:00000000ea4d1263 order:1 compound_mapcount:0 compound_pincount:0
     flags: 0xfffffc0010200(slab|head|node=0|zone=1|lastcpupid=0x1fffff)
     raw: 000fffffc0010200 ffffea000066c300 dead000000000003 ffff888100042b40
     raw: 0000000000000000 0000000000100010 00000001ffffffff 0000000000000000
     page dumped because: kasan: bad access detected

     Memory state around the buggy address:
       ffff88802bb36e00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
       ffff88802bb36e80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
     >ffff88802bb36f00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
                        ^
       ffff88802bb36f80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
       ffff88802bb37000: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     ==================================================================

     Fixes: 801c135ce73d ("UBI: Unsorted Block Images")
     Reported-by: syzkaller <syzkaller@googlegroups.com>
     Signed-off-by: George Kennedy <george.kennedy@oracle.com>
     Signed-off-by: Richard Weinberger <richard@nod.at>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 634a9c139cc1362f6a9cc6cbfe442dbb60ff9f3f
  Author: Xiang Yang <xiangyang3@huawei.com>
  Date:   Tue Nov 15 15:32:25 2022 +0800

     um: vector: Fix memory leak in vector_config

     [ Upstream commit 8f88c73afe481f93d40801596927e8c0047b6d96 ]

     If the return value of the uml_parse_vector_ifspec function is NULL,
     we should call kfree(params) to prevent memory leak.

     Fixes: 49da7e64f33e ("High Performance UML Vector Network Driver")
     Signed-off-by: Xiang Yang <xiangyang3@huawei.com>
     Acked-By: Anton Ivanov <anton.ivanov@kot-begemot.co.uk>
     Signed-off-by: Richard Weinberger <richard@nod.at>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 1028a3103a637660b1b923bb56ac4f3766046dbe
  Author: Chao Yu <chao@kernel.org>
  Date:   Mon Jan 9 11:44:51 2023 +0800

     f2fs: fix to abort atomic write only during do_exist()
```

```
    [ Upstream commit ae267fc1cfe9f941afcb90dc963ee6448dae73cf ]

    Commit 7a10f0177e11 ("f2fs: don't give partially written atomic data
    from process crash") attempted to drop atomic write data after process
    crash, however, f2fs_abort_atomic_write() may be called from noncrash
    case, fix it by adding missed PF_EXITING check condition
    f2fs_file_flush().

    - application crashs
     - do_exit
      - exit_signals -- sets PF_EXITING
      - exit_files
       - put_files_struct
        - close_files
         - filp_close
          - flush (f2fs_file_flush)
           - check atomic_write_task && PF_EXITING
           - f2fs_abort_atomic_write

    Fixes: 7a10f0177e11 ("f2fs: don't give partially written atomic data from process
crash")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7158fb10182e9cb21d64c11923caa901f50c8afb
Author: Yangtao Li <frank.li@vivo.com>
Date:   Mon Jan 23 17:46:01 2023 +0800

    f2fs: allow set compression option of files without blocks

    [ Upstream commit e6261beb0c629403dc58997294dd521bd23664af ]

    Files created by truncate have a size but no blocks, so
    they can be allowed to set compression option.

    Fixes: e1e8debec656 ("f2fs: add F2FS_IOC_SET_COMPRESS_OPTION ioctl")
    Signed-off-by: Yangtao Li <frank.li@vivo.com>
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 54e644b3d82f74a225b01900852da2f65fb00ec5
Author: Alexander Potapenko <glider@google.com>
Date:   Mon Nov 21 12:21:32 2022 +0100

    fs: f2fs: initialize fsdata in pagecache_write()

    [ Upstream commit b1b9896718bc1a212dc288ad66a5fa2fef11353d ]

    When aops->write_begin() does not initialize fsdata, KMSAN may report
    an error passing the latter to aops->write_end().

    Fix this by unconditionally initializing fsdata.

    Suggested-by: Eric Biggers <ebiggers@kernel.org>
    Fixes: 95ae251fe828 ("f2fs: add fs-verity support")
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Reviewed-by: Eric Biggers <ebiggers@google.com>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3d1fd01250ffca88c11e9bd8d19be779acc7557
Author: Chao Yu <chao@kernel.org>
Date:   Mon Jan 9 11:49:20 2023 +0800
```

```
    f2fs: fix to do sanity check on extent cache correctly

    [ Upstream commit d48a7b3a72f121655d95b5157c32c7d555e44c05 ]

    In do_read_inode(), sanity_check_inode() should be called after
    f2fs_init_read_extent_tree(), fix it.

    Fixes: 72840cccc0a1 ("f2fs: allocate the extent_cache by default")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3b6c4754df9bc7a2acbe47fb106d01f2543b165d
Author: Shang XiaoJing <shangxiaojing@huawei.com>
Date:   Wed Jan 11 15:45:18 2023 +0800

    soc: mediatek: mtk-svs: Use pm_runtime_resume_and_get() in svs_init01()

    [ Upstream commit 37fa2aff8fe490771f2229b0f2fcd15796b1bfca ]

    svs_init01() calls pm_runtime_get_sync() and added fail path as
    svs_init01_finish to put usage_counter. However, pm_runtime_get_sync()
    will increment usage_counter even it failed. Fix it by replacing it with
    pm_runtime_resume_and_get() to keep usage counter balanced.

    Fixes: 681a02e95000 ("soc: mediatek: SVS: introduce MTK SVS engine")
    Signed-off-by: Shang XiaoJing <shangxiaojing@huawei.com>
    Signed-off-by: Roger Lu <roger.lu@mediatek.com>
    Reviewed-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Link: https://lore.kernel.org/r/20230111074528.29354-5-roger.lu@mediatek.com
    Signed-off-by: Matthias Brugger <matthias.bgg@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa20dc4b1a0fc25577c5009fcbe8a67bda06ba1c
Author: Roger Lu <roger.lu@mediatek.com>
Date:   Wed Jan 11 15:45:16 2023 +0800

    soc: mediatek: mtk-svs: reset svs when svs_resume() fail

    [ Upstream commit f4f8ad204a15d57c1a3e8ea7eca62157b44cbf59 ]

    Add svs reset when svs_resume() fail.

    Fixes: a825d72f74a3 ("soc: mediatek: fix missing clk_disable_unprepare() on err in
svs_resume()")
    Signed-off-by: Roger Lu <roger.lu@mediatek.com>
    Link: https://lore.kernel.org/r/20230111074528.29354-3-roger.lu@mediatek.com
    Signed-off-by: Matthias Brugger <matthias.bgg@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5001cd159500d8ce702a7abe025bb7ad425d8c08
Author: Roger Lu <roger.lu@mediatek.com>
Date:   Wed Jan 11 15:45:15 2023 +0800

    soc: mediatek: mtk-svs: restore default voltages when svs_init02() fail

    [ Upstream commit a0674cd237fc24b08c7dcb4f8e48df3ee769293a ]

    If svs init02 fail, it means we cannot rely on svs bank voltages anymore.
    We need to disable svs function and restore DVFS opp voltages back to the
    default voltages for making sure we have enough DVFS voltages.

    Fixes: 681a02e95000 ("soc: mediatek: SVS: introduce MTK SVS engine")
    Fixes: 0bbb09b2af9d ("soc: mediatek: SVS: add mt8192 SVS GPU driver")
```

```
    Signed-off-by: Roger Lu <roger.lu@mediatek.com>
    Link: https://lore.kernel.org/r/20230111074528.29354-2-roger.lu@mediatek.com
    Signed-off-by: Matthias Brugger <matthias.bgg@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 97684cfa2aef3d823c88faa6ceb9e84f902950cd
Author: Chao Yu <chao@kernel.org>
Date:   Mon Jan 9 11:44:50 2023 +0800

    f2fs: clear atomic_write_task in f2fs_abort_atomic_write()

    [ Upstream commit 0e8d040bfa4c476d7d2a23119527c744c7de13cd ]

    Otherwise, last .atomic_write_task will be remained in structure
    f2fs_inode_info, resulting in aborting atomic_write accidentally
    in race case. Meanwhile, clear original_i_size as well.

    Fixes: 7a10f0177e11 ("f2fs: don't give partially written atomic data from process
crash")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f955d66b179679312310d42108c0df1cd0714313
Author: Chao Yu <chao@kernel.org>
Date:   Mon Jan 9 11:44:49 2023 +0800

    f2fs: introduce trace_f2fs_replace_atomic_write_block

    [ Upstream commit 2f3a9ae990a7881c9a57a073bb52ebe34fdc3160 ]

    Commit 3db1de0e582c ("f2fs: change the current atomic write way")
    removed old tracepoints, but it missed to add new one, this patch
    fixes to introduce trace_f2fs_replace_atomic_write_block to trace
    atomic_write commit flow.

    Fixes: 3db1de0e582c ("f2fs: change the current atomic write way")
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8266383ce2fc762fe216e93ebd0d00c09865f4a6
Author: Fabrice Gasnier <fabrice.gasnier@foss.st.com>
Date:   Wed Nov 23 14:36:52 2022 +0100

    pwm: stm32-lp: fix the check on arr and cmp registers update

    [ Upstream commit 3066bc2d58be31275afb51a589668f265e419c37 ]

    The ARR (auto reload register) and CMP (compare) registers are
    successively written. The status bits to check the update of these
    registers are polled together with regmap_read_poll_timeout().
    The condition to end the loop may become true, even if one of the
    register isn't correctly updated.
    So ensure both status bits are set before clearing them.

    Fixes: e70a540b4e02 ("pwm: Add STM32 LPTimer PWM driver")
    Signed-off-by: Fabrice Gasnier <fabrice.gasnier@foss.st.com>
    Acked-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 832aaf64140419c56a5bcf3b05ff0e56b0bd1fcb
Author: Emil Renner Berthing <emil.renner.berthing@canonical.com>
Date:   Wed Nov 9 12:37:24 2022 +0100
```

```
    pwm: sifive: Always let the first pwm_apply_state succeed

    [ Upstream commit 334c7b13d38321e47d1a51dba0bef9f4c403ec75 ]

    Commit 2cfe9bbec56ea579135cdd92409fff371841904f added support for the
    RGB and green PWM controlled LEDs on the HiFive Unmatched board
    managed by the leds-pwm-multicolor and leds-pwm drivers respectively.
    All three colours of the RGB LED and the green LED run from different
    lines of the same PWM, but with the same period so this works fine when
    the LED drivers are loaded one after the other.

    Unfortunately it does expose a race in the PWM driver when both LED
    drivers are loaded at roughly the same time. Here is an example:

      |          Thread A           |          Thread B           |
      |  led_pwm_mc_probe           |  led_pwm_probe              |
      |    devm_fwnode_pwm_get      |                             |
      |      pwm_sifive_request     |                             |
      |        ddata->user_count++  |                             |
      |                             |    devm_fwnode_pwm_get      |
      |                             |      pwm_sifive_request     |
      |                             |        ddata->user_count++  |
      |         ...                 |          ...                |
      |    pwm_state_apply          |    pwm_state_apply          |
      |      pwm_sifive_apply       |      pwm_sifive_apply       |

    Now both calls to pwm_sifive_apply will see that ddata->approx_period,
    initially 0, is different from the requested period and the clock needs
    to be updated. But since ddata->user_count >= 2 both calls will fail
    with -EBUSY, which will then cause both LED drivers to fail to probe.

    Fix it by letting the first call to pwm_sifive_apply update the clock
    even when ddata->user_count != 1.

    Fixes: 9e37a53eb051 ("pwm: sifive: Add a driver for SiFive SoC PWM")
    Signed-off-by: Emil Renner Berthing <emil.renner.berthing@canonical.com>
    Signed-off-by: Thierry Reding <thierry.reding@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 66ea96629bbccf1b483be506f3daff754069cdd3
Author: Ricardo Ribalda <ribalda@chromium.org>
Date:   Tue Dec 20 22:35:59 2022 +0100

    soc: mediatek: mtk-svs: Enable the IRQ later

    [ Upstream commit b74952aba6c3f47e7f2c5165abaeefa44c377140 ]

    If the system does not come from reset (like when is booted via
    kexec()), the peripheral might triger an IRQ before the data structures
    are initialised.

    Fixes:

    [    0.227710] Unable to handle kernel NULL pointer dereference at virtual address
0000000000000f08
    [    0.227913] Call trace:
    [    0.227918]  svs_isr+0x8c/0x538

    Signed-off-by: Ricardo Ribalda <ribalda@chromium.org>
    Link: https://lore.kernel.org/r/20221127-mtk-svs-v2-0-145b07663ea8@chromium.org
    Signed-off-by: Matthias Brugger <matthias.bgg@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1c0dbfc94c187b494af96e6ef874b446c13d16f7
```

```
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Wed Nov 23 15:41:18 2022 +0100

    memory: renesas-rpc-if: Move resource acquisition to .probe()

    [ Upstream commit 8b3580df15f53045fda3ffae53f74575c96aa77e ]

    While the acquired resources are tied to the lifetime of the RPC-IF core
    device (through the use of managed resource functions), the actual
    resource acquisition is triggered from the HyperBus and SPI child
    drivers.  Due to this mismatch, unbinding and rebinding the child
    drivers manually fails with -EBUSY:

        # echo rpc-if-hyperflash > /sys/bus/platform/drivers/rpc-if-hyperflash/unbind
        # echo rpc-if-hyperflash > /sys/bus/platform/drivers/rpc-if-hyperflash/bind
        rpc-if ee200000.spi: can't request region for resource [mem 0xee200000-0xee2001ff]
        rpc-if-hyperflash: probe of rpc-if-hyperflash failed with error -16

    The same is true for rpc-if-spi.

    Fix this by moving all resource acquisition to the core driver's probe
    routine.

    Fixes: ca7d8b980b67 ("memory: add Renesas RPC-IF driver")
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Acked-by: Wolfram Sang <wsa+renesas@sang-engineering.com>
    Link:
https://lore.kernel.org/r/c1012ef1de799e08a70817ab7313794e2d8d7bfb.1669213027.git.geert+rene
sas@glider.be
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dfa69d89151669dc04f2c7cb3dda6c236151c872
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Wed Nov 23 15:41:17 2022 +0100

    memory: renesas-rpc-if: Split-off private data from struct rpcif

    [ Upstream commit 51de3fc9a84d8e99dd3f02536a623f9fb95d0c0a ]

    The rpcif structure is used as a common data structure, shared by the
    RPC-IF core driver and by the HyperBus and SPI child drivers.
    This poses several problems:
      - Most structure members describe private core driver state, which
        should not be accessible by the child drivers,
      - The structure's lifetime is controlled by the child drivers,
        complicating use by the core driver.

    Fix this by moving the private core driver state to its own structure,
    managed by the RPC-IF core driver, and store it in the core driver's
    private data field.  This requires absorbing the child's platform
    device, as that was stored in the driver's private data field before.

    Fixes: ca7d8b980b67 ("memory: add Renesas RPC-IF driver")
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Acked-by: Wolfram Sang <wsa+renesas@sang-engineering.com>
    Link:
https://lore.kernel.org/r/09fbb6fa67d5a8cd48a08808c9afa2f6a499aa42.1669213027.git.geert+rene
sas@glider.be
    Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a18cf8d7144cc4bff4f9a2e51090873adc4c016f
Author: Stephan Gerhold <stephan@gerhold.net>
Date:   Wed Jan 4 12:53:45 2023 +0100
```

```
    soc: qcom: socinfo: Fix soc_id order

    [ Upstream commit 017a7c11a8a29e266aa40c6d1bf2ba83f880f719 ]

    The soc_id array is mostly ordered by the numeric "msm-id" defined in
    qcom,ids.h but some recent entries were added at the wrong place.

    While it does not make a functional difference it does make it harder
    to regenerate the entire array after adding a bunch of new IDs.

    Fixes: de320c07da3d ("soc: qcom: socinfo: Add MSM8956/76 SoC IDs to the soc_id table")
    Fixes: 147f6534b8ff ("soc: qcom: socinfo: Add SM8550 ID")
    Signed-off-by: Stephan Gerhold <stephan@gerhold.net>
    Reviewed-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
    Signed-off-by: Bjorn Andersson <andersson@kernel.org>
    Link: https://lore.kernel.org/r/20230104115348.25046-2-stephan@gerhold.net
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6ada28709322ecfb2ba522db6b15b47373e788ed
Author: Tinghan Shen <tinghan.shen@mediatek.com>
Date:   Wed Oct 12 15:54:34 2022 +0800

    soc: mediatek: mtk-pm-domains: Allow mt8186 ADSP default power on

    [ Upstream commit 0d08c56d97a614f56d74f490d693faf8038db125 ]

    In the use case of configuring the access permissions of the ADSP core,
    the mt8186 SoC ADSP power will be switched on in the bootloader because
    the permission control registers are located in the ADSP subsys.

    Signed-off-by: Tinghan Shen <tinghan.shen@mediatek.com>
    Fixes: 88590cbc1703 ("soc: mediatek: pm-domains: Add support for mt8186")
    Reviewed-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Link: https://lore.kernel.org/r/20221012075434.30009-1-tinghan.shen@mediatek.com
    Signed-off-by: Matthias Brugger <matthias.bgg@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d131718d9c45d559951f57c4b88209ca407433c4
Author: Miaoqian Lin <linmq006@gmail.com>
Date:   Mon Dec 5 12:06:42 2022 +0400

    objtool: Fix memory leak in create_static_call_sections()

    [ Upstream commit 3da73f102309fe29150e5c35acd20dd82063ff67 ]

    strdup() allocates memory for key_name. We need to release the memory in
    the following error paths. Add free() to avoid memory leak.

    Fixes: 1e7e47883830 ("x86/static_call: Add inline static call implementation for x86-
64")
    Signed-off-by: Miaoqian Lin <linmq006@gmail.com>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Link: https://lore.kernel.org/r/20221205080642.558583-1-linmq006@gmail.com
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a05603e541f0a2036b274ad0c5ac2122648916eb
Author: Chao Yu <chao@kernel.org>
Date:   Fri Dec 16 23:50:00 2022 +0800

    f2fs: fix to avoid potential deadlock

    [ Upstream commit 5eaac835f27f2de6b73412d7c24e755733b49de0 ]
```

```
    There is a potential deadlock reported by syzbot as below:

    F2FS-fs (loop2): invalid crc value
    F2FS-fs (loop2): Found nat_bits in checkpoint
    F2FS-fs (loop2): Mounted with checkpoint version = 48b305e4
    ======================================================
    WARNING: possible circular locking dependency detected
    6.1.0-rc8-syzkaller-33330-ga5541c0811a0 #0 Not tainted
    ------------------------------------------------------
    syz-executor.2/32123 is trying to acquire lock:
    ffff0000c0e1a608 (&mm->mmap_lock){++++}-{3:3}, at: __might_fault+0x54/0xb4
mm/memory.c:5644

    but task is already holding lock:
    ffff0001317c6088 (&sbi->sb_lock){++++}-{3:3}, at: f2fs_down_write fs/f2fs/f2fs.h:2205
[inline]
    ffff0001317c6088 (&sbi->sb_lock){++++}-{3:3}, at: f2fs_ioc_get_encryption_pwsalt
fs/f2fs/file.c:2334 [inline]
    ffff0001317c6088 (&sbi->sb_lock){++++}-{3:3}, at: __f2fs_ioctl+0x1370/0x3318
fs/f2fs/file.c:4151

    which lock already depends on the new lock.

    Chain exists of:
      &mm->mmap_lock --> &nm_i->nat_tree_lock --> &sbi->sb_lock

     Possible unsafe locking scenario:

           CPU0                    CPU1
           ----                    ----
      lock(&sbi->sb_lock);
                                   lock(&nm_i->nat_tree_lock);
                                   lock(&sbi->sb_lock);
      lock(&mm->mmap_lock);

    Let's try to avoid above deadlock condition by moving __might_fault()
    out of sbi->sb_lock coverage.

    Fixes: 95fa90c9e5a7 ("f2fs: support recording errors into superblock")
    Link: https://lore.kernel.org/linux-f2fs-
devel/000000000000cd5fe305ef617fe2@google.com/T/#u
    Reported-by: syzbot+4793f6096d174c90b4f7@syzkaller.appspotmail.com
    Signed-off-by: Chao Yu <chao@kernel.org>
    Reviewed-by: Eric Biggers <ebiggers@google.com>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9f28671a6cfe888907ebd4764bcf665136537b4b
Author: Christoph Hellwig <hch@lst.de>
Date:   Mon Nov 28 10:15:09 2022 +0100

    f2fs: don't rely on F2FS_MAP_* in f2fs_iomap_begin

    [ Upstream commit 8d3c1fa3fa5eacfd14f5b018eddb6c1a91c57783 ]

    When testing with a mixed zoned / convention device combination, there
    are regular but not 100% reproducible failures in xfstests generic/113
    where the __is_valid_data_blkaddr assert hits due to finding a hole.

    This seems to be because f2fs_map_blocks can set this flag on a hole
    when it was found in the extent cache.

    Rework f2fs_iomap_begin to just check the special block numbers directly.
    This has the added benefits of the WARN_ON showing which invalid block
```

```
    address we found, and being properly error out on delalloc blocks that
    are confusingly called unwritten but not actually suitable for direct
    I/O.

    Fixes: 1517c1a7a445 ("f2fs: implement iomap operations")
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9dfb6c784e385f6e61994bb4e16ce12f3e4940be
Author: Gaosheng Cui <cuigaosheng1@huawei.com>
Date:   Tue Nov 29 09:01:46 2022 +0800

    driver: soc: xilinx: fix memory leak in xlnx_add_cb_for_notify_event()

    [ Upstream commit 1bea534991b9b35c41848a397666ada436456beb ]

    The kfree() should be called when memory fails to be allocated for
    cb_data in xlnx_add_cb_for_notify_event(), otherwise there will be
    a memory leak, so add kfree() to fix it.

    Fixes: 05e5ba40ea7a ("driver: soc: xilinx: Add support of multiple callbacks for same
event in event management driver")
    Signed-off-by: Gaosheng Cui <cuigaosheng1@huawei.com>
    Acked-by: Michal Simek <michal.simek@amd.com>
    Link: https://lore.kernel.org/r/20221129010146.1026685-1-cuigaosheng1@huawei.com
    Signed-off-by: Michal Simek <michal.simek@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e6aa8b2b73c908acc0a3b7541ff860ef2b7b1fc5
Author: Liu Shixin via Jfs-discussion <jfs-discussion@lists.sourceforge.net>
Date:   Thu Nov 3 11:01:59 2022 +0800

    fs/jfs: fix shift exponent db_agl2size negative

    [ Upstream commit fad376fce0af58deebc5075b8539dc05bf639af3 ]

    As a shift exponent, db_agl2size can not be less than 0. Add the missing
    check to fix the shift-out-of-bounds bug reported by syzkaller:

     UBSAN: shift-out-of-bounds in fs/jfs/jfs_dmap.c:2227:15
     shift exponent -744642816 is negative

    Reported-by: syzbot+0be96567042453c0c820@syzkaller.appspotmail.com
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Liu Shixin <liushixin2@huawei.com>
    Signed-off-by: Dave Kleikamp <dave.kleikamp@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5d407911e605702ffcc0e97a6db546592ab27dd0
Author: Jianglei Nie <niejianglei2021@163.com>
Date:   Tue Nov 29 16:15:42 2022 +0800

    auxdisplay: hd44780: Fix potential memory leak in hd44780_remove()

    [ Upstream commit ddf75a86aba2cfb7ec4497e8692b60c8c8fe0ee7 ]

    hd44780_probe() allocates a memory chunk for hd with kzalloc() and
    makes "lcd->drvdata->hd44780" point to it. When we call hd44780_remove(),
    we should release all relevant memory and resource. But "lcd->drvdata
    ->hd44780" is not released, which will lead to a memory leak.

    We should release the "lcd->drvdata->hd44780" in hd44780_remove() to fix
    the memory leak bug.
```

```
    Fixes: 718e05ed92ec ("auxdisplay: Introduce hd44780_common.[ch]")
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Reported-by: kernel test robot <lkp@intel.com>
    Signed-off-by: Jianglei Nie <niejianglei2021@163.com>
    Signed-off-by: Miguel Ojeda <ojeda@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 372ae77cf11d11fb118cbe2d37def9dd5f826abd
Author: Jamal Hadi Salim <jhs@mojatatu.com>
Date:   Tue Feb 14 08:49:14 2023 -0500

    net/sched: Retire tcindex classifier

    commit 8c710f75256bb3cf05ac7b1672c82b92c43f3d28 upstream.

    The tcindex classifier has served us well for about a quarter of a century
    but has not been getting much TLC due to lack of known users. Most recently
    it has become easy prey to syzkaller. For this reason, we are retiring it.

    Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Acked-by: Jiri Pirko <jiri@nvidia.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```