```
commit 264e468fc201cb81c313ad50924bb46506a1b31c
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Jun 17 16:43:05 2020 +0200

    Linux 5.7.3

commit eaffdc4cebe4ba45976481953249a35db75eb3ea
Author: John Stultz <john.stultz@linaro.org>
Date:   Tue Apr 28 18:40:50 2020 +0000

    serial: amba-pl011: Make sure we initialize the port.lock spinlock

    commit 8508f4cba308f785b2fd4b8c38849c117b407297 upstream.

    Valentine reported seeing:

    [    3.626638] INFO: trying to register non-static key.
    [    3.626639] the code is fine but needs lockdep annotation.
    [    3.626640] turning off the locking correctness validator.
    [    3.626644] CPU: 7 PID: 51 Comm: kworker/7:1 Not tainted 5.7.0-rc2-00115-
g8c2e9790f196 #116
    [    3.626646] Hardware name: HiKey960 (DT)
    [    3.626656] Workqueue: events deferred_probe_work_func
    [    3.632476] sd 0:0:0:0: [sda] Optimal transfer size 8192 bytes not a multiple of
physical block size (16384 bytes)
    [    3.640220] Call trace:
    [    3.640225]  dump_backtrace+0x0/0x1b8
    [    3.640227]  show_stack+0x20/0x30
    [    3.640230]  dump_stack+0xec/0x158
    [    3.640234]  register_lock_class+0x598/0x5c0
    [    3.640235]  __lock_acquire+0x80/0x16c0
    [    3.640236]  lock_acquire+0xf4/0x4a0
    [    3.640241]  _raw_spin_lock_irqsave+0x70/0xa8
    [    3.640245]  uart_add_one_port+0x388/0x4b8
    [    3.640248]  pl011_register_port+0x70/0xf0
    [    3.640250]  pl011_probe+0x184/0x1b8
    [    3.640254]  amba_probe+0xdc/0x180
    [    3.640256]  really_probe+0xe0/0x338
    [    3.640257]  driver_probe_device+0x60/0xf8
    [    3.640259]  __device_attach_driver+0x8c/0xd0
    [    3.640260]  bus_for_each_drv+0x84/0xd8
    [    3.640261]  __device_attach+0xe4/0x140
    [    3.640263]  device_initial_probe+0x1c/0x28
    [    3.640265]  bus_probe_device+0xa4/0xb0
    [    3.640266]  deferred_probe_work_func+0x7c/0xb8
    [    3.640269]  process_one_work+0x2c0/0x768
    [    3.640271]  worker_thread+0x4c/0x498
    [    3.640272]  kthread+0x14c/0x158
    [    3.640275]  ret_from_fork+0x10/0x1c

    Which seems to be due to the fact that after allocating the uap
    structure, nothing initializes the spinlock.

    Its a little confusing, as uart_port_spin_lock_init() is one
    place where the lock is supposed to be initialized, but it has
    an exception for the case where the port is a console.

    This makes it seem like a deeper fix is needed to properly
    register the console, but I'm not sure what that entails, and
    Andy suggested that this approach is less invasive.

    Thus, this patch resolves the issue by initializing the spinlock
    in the driver, and resolves the resulting warning.
```

```
    Cc: Andy Shevchenko <andy.shevchenko@gmail.com>
    Cc: Russell King <linux@armlinux.org.uk>
    Cc: Jiri Slaby <jslaby@suse.com>
    Cc: linux-serial@vger.kernel.org
    Reported-by: Valentin Schneider <valentin.schneider@arm.com>
    Reviewed-by: Andy Shevchenko <andy.shevchenko@gmail.com>
    Signed-off-by: John Stultz <john.stultz@linaro.org>
    Reviewed-and-tested-by: Valentin Schneider <valentin.schneider@arm.com>
    Link: https://lore.kernel.org/r/20200428184050.6501-1-john.stultz@linaro.org
    Cc: Naresh Kamboju <naresh.kamboju@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dafcfb953d95c85e035117cb3455fc78e7e3fb4c
Author: Marc Zyngier <maz@kernel.org>
Date:   Tue Jun 9 08:50:29 2020 +0100

    KVM: arm64: Synchronize sysreg state on injecting an AArch32 exception

    commit 0370964dd3ff7d3d406f292cb443a927952cbd05 upstream.

    On a VHE system, the EL1 state is left in the CPU most of the time,
    and only syncronized back to memory when vcpu_put() is called (most
    of the time on preemption).

    Which means that when injecting an exception, we'd better have a way
    to either:
    (1) write directly to the EL1 sysregs
    (2) synchronize the state back to memory, and do the changes there

    For an AArch64, we already do (1), so we are safe. Unfortunately,
    doing the same thing for AArch32 would be pretty invasive. Instead,
    we can easily implement (2) by calling the put/load architectural
    backends, and keep preemption disabled. We can then reload the
    state back into EL1.

    Cc: stable@vger.kernel.org
    Reported-by: James Morse <james.morse@arm.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0a7c8851514bbb8147277db64c2bde86c427e1fe
Author: Marc Zyngier <maz@kernel.org>
Date:   Wed Jun 3 18:24:01 2020 +0100

    KVM: arm64: Save the host's PtrAuth keys in non-preemptible context

    commit ef3e40a7ea8dbe2abd0a345032cd7d5023b9684f upstream.

    When using the PtrAuth feature in a guest, we need to save the host's
    keys before allowing the guest to program them. For that, we dump
    them in a per-CPU data structure (the so called host context).

    But both call sites that do this are in preemptible context,
    which may end up in disaster should the vcpu thread get preempted
    before reentering the guest.

    Instead, save the keys eagerly on each vcpu_load(). This has an
    increased overhead, but is at least safe.

    Cc: stable@vger.kernel.org
    Reviewed-by: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e4a98f8d8dfd63bc830ba4db4c73280760790ae3
```

```
Author: Mattia Dongili <malattia@linux.it>
Date:   Fri May 8 09:14:05 2020 +0900

    platform/x86: sony-laptop: Make resuming thermal profile safer

    commit 476d60b1b4c8a2b14a53ef9b772058f35e604661 upstream.

    The thermal handle object may fail initialization when the module is
    loaded in the first place. Avoid attempting to use it on resume then.

    Fixes: 6d232b29cfce ("ACPICA: Dispatcher: always generate buffer objects for ASL
create_field() operator")
    Reported-by: Dominik Mierzejewski <dominik@greysector.net>
    Bugzilla: https://bugzilla.kernel.org/show_bug.cgi?id=207491
    Signed-off-by: Mattia Dongili <malattia@linux.it>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f5f7fec477b4cd36c782074323d583f1eadb145f
Author: Mattia Dongili <malattia@linux.it>
Date:   Fri May 8 09:14:04 2020 +0900

    platform/x86: sony-laptop: SNC calls should handle BUFFER types

    commit 47828d22539f76c8c9dcf2a55f18ea3a8039d8ef upstream.

    After commit 6d232b29cfce ("ACPICA: Dispatcher: always generate buffer
    objects for ASL create_field() operator") ACPICA creates buffers even
    when new fields are small enough to fit into an integer.
    Many SNC calls counted on the old behaviour.
    Since sony-laptop already handles the INTEGER/BUFFER case in
    sony_nc_buffer_call, switch sony_nc_int_call to use its more generic
    function instead.

    Fixes: 6d232b29cfce ("ACPICA: Dispatcher: always generate buffer objects for ASL
create_field() operator")
    Reported-by: Dominik Mierzejewski <dominik@greysector.net>
    Bugzilla: https://bugzilla.kernel.org/show_bug.cgi?id=207491
    Reported-by: William Bader <williambader@hotmail.com>
    Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=1830150
    Signed-off-by: Mattia Dongili <malattia@linux.it>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 63efec1f80bb5f112ffc7ae89faff86e067dbe42
Author: Juergen Gross <jgross@suse.com>
Date:   Mon May 11 09:42:31 2020 +0200

    xen/pvcalls-back: test for errors when calling backend_connect()

    commit c8d70a29d6bbc956013f3401f92a4431a9385a3c upstream.

    backend_connect() can fail, so switch the device to connected only if
    no error occurred.

    Fixes: 0a9c75c2c7258f2 ("xen/pvcalls: xenbus state handling")
    Cc: stable@vger.kernel.org
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Link: https://lore.kernel.org/r/20200511074231.19794-1-jgross@suse.com
    Reviewed-by: Stefano Stabellini <sstabellini@kernel.org>
    Signed-off-by: Boris Ostrovsky <boris.ostrovsky@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f8f2b599ea5210fa545634cb763ebc923187fab6
Author: Jiri Kosina <jkosina@suse.cz>
```

```
Date:   Tue May 26 11:49:18 2020 +0200

    block/floppy: fix contended case in floppy_queue_rq()

    commit 263c61581a38d0a5ad1f5f4a9143b27d68caeffd upstream.

    Since the switch of floppy driver to blk-mq, the contended (fdc_busy) case
    in floppy_queue_rq() is not handled correctly.

    In case we reach floppy_queue_rq() with fdc_busy set (i.e. with the floppy
    locked due to another request still being in-flight), we put the request
    on the list of requests and return BLK_STS_OK to the block core, without
    actually scheduling delayed work / doing further processing of the
    request. This means that processing of this request is postponed until
    another request comes and passess uncontended.

    Which in some cases might actually never happen and we keep waiting
    indefinitely. The simple testcase is

            for i in `seq 1 2000`; do echo -en $i '\r'; blkid --info /dev/fd0 2> /dev/null;
done

    run in quemu. That reliably causes blkid eventually indefinitely hanging
    in __floppy_read_block_0() waiting for completion, as the BIO callback
    never happens, and no further IO is ever submitted on the (non-existent)
    floppy device. This was observed reliably on qemu-emulated device.

    Fix that by not queuing the request in the contended case, and return
    BLK_STS_RESOURCE instead, so that blk core handles the request
    rescheduling and let it pass properly non-contended later.

    Fixes: a9f38e1dec107a ("floppy: convert to blk-mq")
    Cc: stable@vger.kernel.org
    Tested-by: Libor Pechacek <lpechacek@suse.cz>
    Signed-off-by: Jiri Kosina <jkosina@suse.cz>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c2fe1a711ac17552f895e861531b0d19f707c056
Author: Ulf Hansson <ulf.hansson@linaro.org>
Date:   Thu Apr 30 11:16:38 2020 +0200

    mmc: sdio: Fix several potential memory leaks in mmc_sdio_init_card()

    commit a94a59f43749b4f8cd81b8be87c95f9ef898d19d upstream.

    Over the years, the code in mmc_sdio_init_card() has grown to become quite
    messy. Unfortunate this has also lead to that several paths are leaking
    memory in form of an allocated struct mmc_card, which includes additional
    data, such as initialized struct device for example.

    Unfortunate, it's a too complex task find each offending commit. Therefore,
    this change fixes all memory leaks at once.

    Cc: <stable@vger.kernel.org>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Link: https://lore.kernel.org/r/20200430091640.455-3-ulf.hansson@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7724fd11ee72a15b39aa4916d5963b1175bc659a
Author: Ulf Hansson <ulf.hansson@linaro.org>
Date:   Thu Apr 30 11:16:37 2020 +0200

    mmc: sdio: Fix potential NULL pointer error in mmc_sdio_init_card()
```

```
    commit f04086c225da11ad16d7f9a2fbca6483ab16dded upstream.

    During some scenarios mmc_sdio_init_card() runs a retry path for the UHS-I
    specific initialization, which leads to removal of the previously allocated
    card. A new card is then re-allocated while retrying.

    However, in one of the corresponding error paths we may end up to remove an
    already removed card, which likely leads to a NULL pointer exception. So,
    let's fix this.

    Fixes: 5fc3d80ef496 ("mmc: sdio: don't use rocr to check if the card could support UHS
mode")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Link: https://lore.kernel.org/r/20200430091640.455-2-ulf.hansson@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3cede7940a4866ba4b1e2b4b74bb7e22ffc351d2
Author: Ludovic Desroches <ludovic.desroches@microchip.com>
Date:   Thu Apr 2 00:15:00 2020 +0200

    ARM: dts: at91: sama5d2_ptc_ek: fix sdmmc0 node description

    commit a1af7f36c70369b971ee1cf679dd68368dad23f0 upstream.

    Remove non-removable and mmc-ddr-1_8v properties from the sdmmc0
    node which come probably from an unchecked copy/paste.

    Signed-off-by: Ludovic Desroches <ludovic.desroches@microchip.com>
    Fixes:42ed535595ec "ARM: dts: at91: introduce the sama5d2 ptc ek board"
    Cc: stable@vger.kernel.org # 4.19 and later
    Link: https://lore.kernel.org/r/20200401221504.41196-1-ludovic.desroches@microchip.com
    Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 37f281c5753031801616b5d8e3a42cb72f9d7174
Author: Masahiro Yamada <yamada.masahiro@socionext.com>
Date:   Mon May 11 15:21:58 2020 +0900

    mmc: uniphier-sd: call devm_request_irq() after tmio_mmc_host_probe()

    commit 5d1f42e14b135773c0cc1d82e904c5b223783a9d upstream.

    Currently, tmio_mmc_irq() handler is registered before the host is
    fully initialized by tmio_mmc_host_probe(). I did not previously notice
    this problem.

    The boot ROM of a new Socionext SoC unmasks interrupts (CTL_IRQ_MASK)
    somehow. The handler is invoked before tmio_mmc_host_probe(), then
    emits noisy call trace.

    Move devm_request_irq() below tmio_mmc_host_probe().

    Fixes: 3fd784f745dd ("mmc: uniphier-sd: add UniPhier SD/eMMC controller driver")
    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200511062158.1790924-1-yamada.masahiro@socionext.com
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d11a02427f5a996f787a8dd337bb21f13e8c9dd8
Author: Ulf Hansson <ulf.hansson@linaro.org>
Date:   Tue May 19 17:24:34 2020 +0200

    mmc: tmio: Further fixup runtime PM management at remove
```

```
    commit 4bd784411aca022622e484eb262f5a0540ae732c upstream.

    Before calling tmio_mmc_host_probe(), the caller is required to enable
    clocks for its device, as to make it accessible when reading/writing
    registers during probe.

    Therefore, the responsibility to disable these clocks, in the error path of
    ->probe() and during ->remove(), is better managed outside
    tmio_mmc_host_remove(). As a matter of fact, callers of
    tmio_mmc_host_remove() already expects this to be the behaviour.

    However, there's a problem with tmio_mmc_host_remove() when the Kconfig
    option, CONFIG_PM, is set. More precisely, tmio_mmc_host_remove() may then
    disable the clock via runtime PM, which leads to clock enable/disable
    imbalance problems, when the caller of tmio_mmc_host_remove() also tries to
    disable the same clocks.

    To solve the problem, let's make sure tmio_mmc_host_remove() leaves the
    device with clocks enabled, but also make sure to disable the IRQs, as we
    normally do at ->runtime_suspend().

    Reported-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Wolfram Sang <wsa+renesas@sang-engineering.com>
    Tested-by: Wolfram Sang <wsa+renesas@sang-engineering.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200519152434.6867-1-ulf.hansson@linaro.org
    Tested-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fc062db0e1a30aa69aa3a6ffed0612b81f13e2c1
Author: Ludovic Barre <ludovic.barre@st.com>
Date:   Tue May 26 17:51:02 2020 +0200

    mmc: mmci_sdmmc: fix DMA API warning overlapping mappings

    commit fe8d33bd33d527dee3155d2bccd714a655f37334 upstream.

    Turning on CONFIG_DMA_API_DEBUG_SG results in the following warning:
    WARNING: CPU: 1 PID: 20 at kernel/dma/debug.c:500 add_dma_entry+0x16c/0x17c
    DMA-API: exceeded 7 overlapping mappings of cacheline 0x031d2645
    Modules linked in:
    CPU: 1 PID: 20 Comm: kworker/1:1 Not tainted 5.5.0-rc2-00021-gdeda30999c2b-dirty #49
    Hardware name: STM32 (Device Tree Support)
    Workqueue: events_freezable mmc_rescan
    [<c03138c0>] (unwind_backtrace) from [<c030d760>] (show_stack+0x10/0x14)
    [<c030d760>] (show_stack) from [<c0f2eb28>] (dump_stack+0xc0/0xd4)
    [<c0f2eb28>] (dump_stack) from [<c034a14c>] (__warn+0xd0/0xf8)
    [<c034a14c>] (__warn) from [<c034a530>] (warn_slowpath_fmt+0x94/0xb8)
    [<c034a530>] (warn_slowpath_fmt) from [<c03bca0c>] (add_dma_entry+0x16c/0x17c)
    [<c03bca0c>] (add_dma_entry) from [<c03bdf54>] (debug_dma_map_sg+0xe4/0x3d4)
    [<c03bdf54>] (debug_dma_map_sg) from [<c0d09244>] (sdmmc_idma_prep_data+0x94/0xf8)
    [<c0d09244>] (sdmmc_idma_prep_data) from [<c0d05a2c>] (mmci_prep_data+0x2c/0xb0)
    [<c0d05a2c>] (mmci_prep_data) from [<c0d073ec>] (mmci_start_data+0x134/0x2f0)
    [<c0d073ec>] (mmci_start_data) from [<c0d078d0>] (mmci_request+0xe8/0x154)
    [<c0d078d0>] (mmci_request) from [<c0cecb44>] (mmc_start_request+0x94/0xbc)

    DMA api debug brings to light leaking dma-mappings, dma_map_sg and
    dma_unmap_sg are not correctly balanced.

    If a request is prepared, the dma_map/unmap are done in asynchronous call
    pre_req (prep_data) and post_req (unprep_data). In this case the
    dma-mapping is right balanced.
```

```
    But if the request was not prepared, the data->host_cookie is define to
    zero and the dma_map/unmap must be done in the request.  The dma_map is
    called by mmci_dma_start (prep_data), but there is no dma_unmap in this
    case.

    This patch adds dma_unmap_sg when the dma is finalized and the data cookie
    is zero (request not prepared).

    Signed-off-by: Ludovic Barre <ludovic.barre@st.com>
    Link: https://lore.kernel.org/r/20200526155103.12514-2-ludovic.barre@st.com
    Fixes: 46b723dd867d ("mmc: mmci: add stm32 sdmmc variant")
    Cc: stable@vger.kernel.org
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bcc5d61fee32040a2a1fcfb7d163faee51916663
Author: Eugen Hristev <eugen.hristev@microchip.com>
Date:   Wed May 27 13:56:59 2020 +0300

    mmc: sdhci-of-at91: fix CALCR register being rewritten

    commit dbdea70f71d672c12bc4454e7c258a8f78194d74 upstream.

    When enabling calibration at reset, the CALCR register was completely
    rewritten. This may cause certain bits being deleted unintentedly.
    Fix by issuing a read-modify-write operation.

    Fixes: 727d836a375a ("mmc: sdhci-of-at91: add DT property to enable calibration on full
reset")
    Signed-off-by: Eugen Hristev <eugen.hristev@microchip.com>
    Link: https://lore.kernel.org/r/20200527105659.142560-1-eugen.hristev@microchip.com
    Cc: stable@vger.kernel.org
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d22ca081703130ba422aff0e2732c316191afa54
Author: Veerabhadrarao Badiganti <vbadigan@codeaurora.org>
Date:   Thu May 28 20:43:52 2020 +0530

    mmc: sdhci-msm: Clear tuning done flag while hs400 tuning

    commit 9253d71011c349d5f5cc0cebdf68b4a80811b92d upstream.

    Clear tuning_done flag while executing tuning to ensure vendor
    specific HS400 settings are applied properly when the controller
    is re-initialized in HS400 mode.

    Without this, re-initialization of the qcom SDHC in HS400 mode fails
    while resuming the driver from runtime-suspend or system-suspend.

    Fixes: ff06ce417828 ("mmc: sdhci-msm: Add HS400 platform support")
    Cc: stable@vger.kernel.org
    Signed-off-by: Veerabhadrarao Badiganti <vbadigan@codeaurora.org>
    Link: https://lore.kernel.org/r/1590678838-18099-1-git-send-email-
vbadigan@codeaurora.org
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4470c8a395bbb207139e854daf38f33825193e56
Author: Chris Wilson <chris@chris-wilson.co.uk>
Date:   Fri Apr 10 09:35:35 2020 +0100

    agp/intel: Reinforce the barrier after GTT updates

    commit f30d3ced9fafa03e4855508929b5b6334907f45e upstream.
```

```
    After changing the timing between GTT updates and execution on the GPU,
    we started seeing sporadic failures on Ironlake. These were narrowed
    down to being an insufficiently strong enough barrier/delay after
    updating the GTT and scheduling execution on the GPU. By forcing the
    uncached read, and adding the missing barrier for the singular
    insert_page (relocation paths), the sporadic failures go away.

    Fixes: 983d308cb8f6 ("agp/intel: Serialise after GTT updates")
    Fixes: 3497971a71d8 ("agp/intel: Flush chipset writes after updating a single PTE")
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Acked-by: Andi Shyti <andi.shyti@intel.com>
    Cc: stable@vger.kernel.org # v4.0+
    Link: https://patchwork.freedesktop.org/patch/msgid/20200410083535.25464-1-chris@chris-
wilson.co.uk
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8703c51557065abd14e78b58a6662bb78b371fa
Author: Barret Rhoden <brho@google.com>
Date:   Tue Apr 14 18:29:20 2020 -0400

    perf: Add cond_resched() to task_function_call()

    commit 2ed6edd33a214bca02bd2b45e3fc3038a059436b upstream.

    Under rare circumstances, task_function_call() can repeatedly fail and
    cause a soft lockup.

    There is a slight race where the process is no longer running on the cpu
    we targeted by the time remote_function() runs.  The code will simply
    try again.  If we are very unlucky, this will continue to fail, until a
    watchdog fires.  This can happen in a heavily loaded, multi-core virtual
    machine.

    Reported-by: syzbot+bb4935a5c09b5ff79940@syzkaller.appspotmail.com
    Signed-off-by: Barret Rhoden <brho@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Link: https://lkml.kernel.org/r/20200414222920.121401-1-brho@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1d56725ba05e150cc40e541a6e80ababa200dbd5
Author: OGAWA Hirofumi <hirofumi@mail.parknet.co.jp>
Date:   Thu Jun 4 16:50:56 2020 -0700

    fat: don't allow to mount if the FAT length == 0

    commit b1b65750b8db67834482f758fc385bfa7560d228 upstream.

    If FAT length == 0, the image doesn't have any data. And it can be the
    cause of overlapping the root dir and FAT entries.

    Also Windows treats it as invalid format.

    Reported-by: syzbot+6f1624f937d9d6911e2d@syzkaller.appspotmail.com
    Signed-off-by: OGAWA Hirofumi <hirofumi@mail.parknet.co.jp>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Cc: Marco Elver <elver@google.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Link: http://lkml.kernel.org/r/87r1wz8mrd.fsf@mail.parknet.co.jp
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5e028cc3dafbdf31cfe056f05608e50a1e20dca3
Author: Wang Hai <wanghai38@huawei.com>
Date:   Wed Jun 3 15:56:21 2020 -0700
```

```
    mm/slub: fix a memory leak in sysfs_slab_add()

    commit dde3c6b72a16c2db826f54b2d49bdea26c3534a2 upstream.

    syzkaller reports for memory leak when kobject_init_and_add() returns an
    error in the function sysfs_slab_add() [1]

    When this happened, the function kobject_put() is not called for the
    corresponding kobject, which potentially leads to memory leak.

    This patch fixes the issue by calling kobject_put() even if
    kobject_init_and_add() fails.

    [1]
      BUG: memory leak
      unreferenced object 0xffff8880a6d4be88 (size 8):
      comm "syz-executor.3", pid 946, jiffies 4295772514 (age 18.396s)
      hex dump (first 8 bytes):
        70 69 64 5f 33 00 ff ff                          pid_3...
      backtrace:
         kstrdup+0x35/0x70 mm/util.c:60
         kstrdup_const+0x3d/0x50 mm/util.c:82
         kvasprintf_const+0x112/0x170 lib/kasprintf.c:48
         kobject_set_name_vargs+0x55/0x130 lib/kobject.c:289
         kobject_add_varg lib/kobject.c:384 [inline]
         kobject_init_and_add+0xd8/0x170 lib/kobject.c:473
         sysfs_slab_add+0x1d8/0x290 mm/slub.c:5811
         __kmem_cache_create+0x50a/0x570 mm/slub.c:4384
         create_cache+0x113/0x1e0 mm/slab_common.c:407
         kmem_cache_create_usercopy+0x1a1/0x260 mm/slab_common.c:505
         kmem_cache_create+0xd/0x10 mm/slab_common.c:564
         create_pid_cachep kernel/pid_namespace.c:54 [inline]
         create_pid_namespace kernel/pid_namespace.c:96 [inline]
         copy_pid_ns+0x77c/0x8f0 kernel/pid_namespace.c:148
         create_new_namespaces+0x26b/0xa30 kernel/nsproxy.c:95
         unshare_nsproxy_namespaces+0xa7/0x1e0 kernel/nsproxy.c:229
         ksys_unshare+0x3d2/0x770 kernel/fork.c:2969
         __do_sys_unshare kernel/fork.c:3037 [inline]
         __se_sys_unshare kernel/fork.c:3035 [inline]
         __x64_sys_unshare+0x2d/0x40 kernel/fork.c:3035
         do_syscall_64+0xa1/0x530 arch/x86/entry/common.c:295

    Fixes: 80da026a8e5d ("mm/slub: fix slab double-free in case of duplicate sysfs
filename")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: Wang Hai <wanghai38@huawei.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Cc: Christoph Lameter <cl@linux.com>
    Cc: Pekka Enberg <penberg@kernel.org>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Joonsoo Kim <iamjoonsoo.kim@lge.com>
    Link: http://lkml.kernel.org/r/20200602115033.1054-1-wanghai38@huawei.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5d00faaeecfc699e10a2dce53e4805d7215be5ac
Author: Ezequiel Garcia <ezequiel@collabora.com>
Date:   Mon Apr 27 18:44:05 2020 -0300

    drm/vkms: Hold gem object while still in-use

    commit 0ea2ea42b31abc1141f2fd3911f952a97d401fcb upstream.

    We need to keep the reference to the drm_gem_object
```

```
    until the last access by vkms_dumb_create.

    Therefore, the put the object after it is used.

    This fixes a use-after-free issue reported by syzbot.

    While here, change vkms_gem_create() symbol to static.

    Reported-and-tested-by: syzbot+e3372a2afe1e7ef04bc7@syzkaller.appspotmail.com
    Signed-off-by: Ezequiel Garcia <ezequiel@collabora.com>
    Reviewed-by: Rodrigo Siqueira <Rodrigo.Siqueira@amd.com>
    Signed-off-by: Rodrigo Siqueira <rodrigosiqueiramelo@gmail.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200427214405.13069-1-
ezequiel@collabora.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1f4404f89383e325ee1e6e72e9627ee7aef74d83
Author: Casey Schaufler <casey@schaufler-ca.com>
Date:   Thu Apr 9 16:35:28 2020 -0700

    Smack: slab-out-of-bounds in vsscanf

    commit 84e99e58e8d1e26f04c097f4266e431a33987f36 upstream.

    Add barrier to soob. Return -EOVERFLOW if the buffer
    is exceeded.

    Suggested-by: Hillf Danton <hdanton@sina.com>
    Reported-by: syzbot+bfdd4a2f07be52351350@syzkaller.appspotmail.com
    Signed-off-by: Casey Schaufler <casey@schaufler-ca.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6602f080cb28745259e2fab1a4cf55eeb5894f93
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Sat Apr 4 12:18:38 2020 +0800

    ath9k: Fix general protection fault in ath9k_hif_usb_rx_cb

    commit 2bbcaaee1fcbd83272e29f31e2bb7e70d8c49e05 upstream.

    In ath9k_hif_usb_rx_cb interface number is assumed to be 0.
    usb_ifnum_to_if(urb->dev, 0)
    But it isn't always true.

    The case reported by syzbot:
    https://lore.kernel.org/linux-usb/000000000000666c9c05a1c05d12@google.com
    usb 2-1: new high-speed USB device number 2 using dummy_hcd
    usb 2-1: config 1 has an invalid interface number: 2 but max is 0
    usb 2-1: config 1 has no interface number 0
    usb 2-1: New USB device found, idVendor=0cf3, idProduct=9271, bcdDevice=
    1.08
    usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
    general protection fault, probably for non-canonical address
    0xdffffc0000000015: 0000 [#1] SMP KASAN
    KASAN: null-ptr-deref in range [0x00000000000000a8-0x00000000000000af]
    CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.6.0-rc5-syzkaller #0

    Call Trace
    __usb_hcd_giveback_urb+0x29a/0x550 drivers/usb/core/hcd.c:1650
    usb_hcd_giveback_urb+0x368/0x420 drivers/usb/core/hcd.c:1716
    dummy_timer+0x1258/0x32ae drivers/usb/gadget/udc/dummy_hcd.c:1966
    call_timer_fn+0x195/0x6f0 kernel/time/timer.c:1404
    expire_timers kernel/time/timer.c:1449 [inline]
    __run_timers kernel/time/timer.c:1773 [inline]
    __run_timers kernel/time/timer.c:1740 [inline]
```

```
    run_timer_softirq+0x5f9/0x1500 kernel/time/timer.c:1786
    __do_softirq+0x21e/0x950 kernel/softirq.c:292
    invoke_softirq kernel/softirq.c:373 [inline]
    irq_exit+0x178/0x1a0 kernel/softirq.c:413
    exiting_irq arch/x86/include/asm/apic.h:546 [inline]
    smp_apic_timer_interrupt+0x141/0x540 arch/x86/kernel/apic/apic.c:1146
    apic_timer_interrupt+0xf/0x20 arch/x86/entry/entry_64.S:829

    Reported-and-tested-by: syzbot+40d5d2e8a4680952f042@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Link: https://lore.kernel.org/r/20200404041838.10426-6-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c28bd5dd7d5c82f0d7ae8955ec1d05b8061bb599
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Sat Apr 4 12:18:37 2020 +0800

    ath9x: Fix stack-out-of-bounds Write in ath9k_hif_usb_rx_cb

    commit 19d6c375d671ce9949a864fb9a03e19f5487b4d3 upstream.

    Add barrier to accessing the stack array skb_pool.

    The case reported by syzbot:
    https://lore.kernel.org/linux-usb/0000000000003d7c1505a2168418@google.com
    BUG: KASAN: stack-out-of-bounds in ath9k_hif_usb_rx_stream
    drivers/net/wireless/ath/ath9k/hif_usb.c:626 [inline]
    BUG: KASAN: stack-out-of-bounds in ath9k_hif_usb_rx_cb+0xdf6/0xf70
    drivers/net/wireless/ath/ath9k/hif_usb.c:666
    Write of size 8 at addr ffff8881db309a28 by task swapper/1/0

    Call Trace:
    ath9k_hif_usb_rx_stream drivers/net/wireless/ath/ath9k/hif_usb.c:626
    [inline]
    ath9k_hif_usb_rx_cb+0xdf6/0xf70
    drivers/net/wireless/ath/ath9k/hif_usb.c:666
    __usb_hcd_giveback_urb+0x1f2/0x470 drivers/usb/core/hcd.c:1648
    usb_hcd_giveback_urb+0x368/0x420 drivers/usb/core/hcd.c:1713
    dummy_timer+0x1258/0x32ae drivers/usb/gadget/udc/dummy_hcd.c:1966
    call_timer_fn+0x195/0x6f0 kernel/time/timer.c:1404
    expire_timers kernel/time/timer.c:1449 [inline]
    __run_timers kernel/time/timer.c:1773 [inline]
    __run_timers kernel/time/timer.c:1740 [inline]
    run_timer_softirq+0x5f9/0x1500 kernel/time/timer.c:1786

    Reported-and-tested-by: syzbot+d403396d4df67ad0bd5f@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Link: https://lore.kernel.org/r/20200404041838.10426-5-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 16ea1406128755cefc3b4d83674b28f68657ac4e
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Sat Apr 4 12:18:36 2020 +0800

    ath9k: Fix use-after-free Write in ath9k_htc_rx_msg

    commit e4ff08a4d727146bb6717a39a8d399d834654345 upstream.

    Write out of slab bounds. We should check epid.

    The case reported by syzbot:
    https://lore.kernel.org/linux-usb/0000000000006ac55b05a1c05d72@google.com
    BUG: KASAN: use-after-free in htc_process_conn_rsp
```

```
    drivers/net/wireless/ath/ath9k/htc_hst.c:131 [inline]
    BUG: KASAN: use-after-free in ath9k_htc_rx_msg+0xa25/0xaf0
    drivers/net/wireless/ath/ath9k/htc_hst.c:443
    Write of size 2 at addr ffff8881cea291f0 by task swapper/1/0

    Call Trace:
     htc_process_conn_rsp drivers/net/wireless/ath/ath9k/htc_hst.c:131
    [inline]
    ath9k_htc_rx_msg+0xa25/0xaf0
    drivers/net/wireless/ath/ath9k/htc_hst.c:443
    ath9k_hif_usb_reg_in_cb+0x1ba/0x630
    drivers/net/wireless/ath/ath9k/hif_usb.c:718
    __usb_hcd_giveback_urb+0x29a/0x550 drivers/usb/core/hcd.c:1650
    usb_hcd_giveback_urb+0x368/0x420 drivers/usb/core/hcd.c:1716
    dummy_timer+0x1258/0x32ae drivers/usb/gadget/udc/dummy_hcd.c:1966
    call_timer_fn+0x195/0x6f0 kernel/time/timer.c:1404
    expire_timers kernel/time/timer.c:1449 [inline]
    __run_timers kernel/time/timer.c:1773 [inline]
    __run_timers kernel/time/timer.c:1740 [inline]
    run_timer_softirq+0x5f9/0x1500 kernel/time/timer.c:1786

    Reported-and-tested-by: syzbot+b1c61e5f11be5782f192@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Link: https://lore.kernel.org/r/20200404041838.10426-4-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 378d2734bf603bac4959bce2cadf5927aa2beffc
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Sat Apr 4 12:18:35 2020 +0800

    ath9k: Fix use-after-free Read in ath9k_wmi_ctrl_rx

    commit abeaa85054ff8cfe8b99aafc5c70ea067e5d0908 upstream.

    Free wmi later after cmd urb has been killed, as urb cb will access wmi.

    the case reported by syzbot:
    https://lore.kernel.org/linux-usb/0000000000000002fc05a1d61a68@google.com
    BUG: KASAN: use-after-free in ath9k_wmi_ctrl_rx+0x416/0x500
    drivers/net/wireless/ath/ath9k/wmi.c:215
    Read of size 1 at addr ffff8881cef1417c by task swapper/1/0

    Call Trace:
    <IRQ>
    ath9k_wmi_ctrl_rx+0x416/0x500 drivers/net/wireless/ath/ath9k/wmi.c:215
    ath9k_htc_rx_msg+0x2da/0xaf0
    drivers/net/wireless/ath/ath9k/htc_hst.c:459
    ath9k_hif_usb_reg_in_cb+0x1ba/0x630
    drivers/net/wireless/ath/ath9k/hif_usb.c:718
    __usb_hcd_giveback_urb+0x29a/0x550 drivers/usb/core/hcd.c:1650
    usb_hcd_giveback_urb+0x368/0x420 drivers/usb/core/hcd.c:1716
    dummy_timer+0x1258/0x32ae drivers/usb/gadget/udc/dummy_hcd.c:1966
    call_timer_fn+0x195/0x6f0 kernel/time/timer.c:1404
    expire_timers kernel/time/timer.c:1449 [inline]
    __run_timers kernel/time/timer.c:1773 [inline]
    __run_timers kernel/time/timer.c:1740 [inline]
    run_timer_softirq+0x5f9/0x1500 kernel/time/timer.c:1786

    Reported-and-tested-by: syzbot+5d338854440137ea0fef@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Link: https://lore.kernel.org/r/20200404041838.10426-3-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 9ddf89d5adad1914b15c286cc137cf54414f94c1
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Sat Apr 4 12:18:34 2020 +0800

    ath9k: Fix use-after-free Read in htc_connect_service

    commit ced21a4c726bdc60b1680c050a284b08803bc64c upstream.

    The skb is consumed by htc_send_epid, so it needn't release again.

    The case reported by syzbot:

    https://lore.kernel.org/linux-usb/000000000000590f6b05a1c05d15@google.com
    usb 1-1: ath9k_htc: Firmware ath9k_htc/htc_9271-1.4.0.fw requested
    usb 1-1: ath9k_htc: Transferred FW: ath9k_htc/htc_9271-1.4.0.fw, size:
    51008
    usb 1-1: Service connection timeout for: 256
    ==================================================================
    BUG: KASAN: use-after-free in atomic_read
    include/asm-generic/atomic-instrumented.h:26 [inline]
    BUG: KASAN: use-after-free in refcount_read include/linux/refcount.h:134
    [inline]
    BUG: KASAN: use-after-free in skb_unref include/linux/skbuff.h:1042
    [inline]
    BUG: KASAN: use-after-free in kfree_skb+0x32/0x3d0 net/core/skbuff.c:692
    Read of size 4 at addr ffff8881d0957994 by task kworker/1:2/83

    Call Trace:
    kfree_skb+0x32/0x3d0 net/core/skbuff.c:692
    htc_connect_service.cold+0xa9/0x109
    drivers/net/wireless/ath/ath9k/htc_hst.c:282
    ath9k_wmi_connect+0xd2/0x1a0 drivers/net/wireless/ath/ath9k/wmi.c:265
    ath9k_init_htc_services.constprop.0+0xb4/0x650
    drivers/net/wireless/ath/ath9k/htc_drv_init.c:146
    ath9k_htc_probe_device+0x25a/0x1d80
    drivers/net/wireless/ath/ath9k/htc_drv_init.c:959
    ath9k_htc_hw_init+0x31/0x60
    drivers/net/wireless/ath/ath9k/htc_hst.c:501
    ath9k_hif_usb_firmware_cb+0x26b/0x500
    drivers/net/wireless/ath/ath9k/hif_usb.c:1187
    request_firmware_work_func+0x126/0x242
    drivers/base/firmware_loader/main.c:976
    process_one_work+0x94b/0x1620 kernel/workqueue.c:2264
    worker_thread+0x96/0xe20 kernel/workqueue.c:2410
    kthread+0x318/0x420 kernel/kthread.c:255
    ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:352

    Allocated by task 83:
    kmem_cache_alloc_node+0xdc/0x330 mm/slub.c:2814
    __alloc_skb+0xba/0x5a0 net/core/skbuff.c:198
    alloc_skb include/linux/skbuff.h:1081 [inline]
    htc_connect_service+0x2cc/0x840
    drivers/net/wireless/ath/ath9k/htc_hst.c:257
    ath9k_wmi_connect+0xd2/0x1a0 drivers/net/wireless/ath/ath9k/wmi.c:265
    ath9k_init_htc_services.constprop.0+0xb4/0x650
    drivers/net/wireless/ath/ath9k/htc_drv_init.c:146
    ath9k_htc_probe_device+0x25a/0x1d80
    drivers/net/wireless/ath/ath9k/htc_drv_init.c:959
    ath9k_htc_hw_init+0x31/0x60
    drivers/net/wireless/ath/ath9k/htc_hst.c:501
    ath9k_hif_usb_firmware_cb+0x26b/0x500
    drivers/net/wireless/ath/ath9k/hif_usb.c:1187
    request_firmware_work_func+0x126/0x242
    drivers/base/firmware_loader/main.c:976
    process_one_work+0x94b/0x1620 kernel/workqueue.c:2264
```

```
      worker_thread+0x96/0xe20 kernel/workqueue.c:2410
      kthread+0x318/0x420 kernel/kthread.c:255
      ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:352

      Freed by task 0:
      kfree_skb+0x102/0x3d0 net/core/skbuff.c:690
      ath9k_htc_txcompletion_cb+0x1f8/0x2b0
      drivers/net/wireless/ath/ath9k/htc_hst.c:356
      hif_usb_regout_cb+0x10b/0x1b0
      drivers/net/wireless/ath/ath9k/hif_usb.c:90
      __usb_hcd_giveback_urb+0x29a/0x550 drivers/usb/core/hcd.c:1650
      usb_hcd_giveback_urb+0x368/0x420 drivers/usb/core/hcd.c:1716
      dummy_timer+0x1258/0x32ae drivers/usb/gadget/udc/dummy_hcd.c:1966
      call_timer_fn+0x195/0x6f0 kernel/time/timer.c:1404
      expire_timers kernel/time/timer.c:1449 [inline]
      __run_timers kernel/time/timer.c:1773 [inline]
      __run_timers kernel/time/timer.c:1740 [inline]
      run_timer_softirq+0x5f9/0x1500 kernel/time/timer.c:1786
      __do_softirq+0x21e/0x950 kernel/softirq.c:292

      Reported-and-tested-by: syzbot+9505af1ae303dabdc646@syzkaller.appspotmail.com
      Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
      Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
      Link: https://lore.kernel.org/r/20200404041838.10426-2-hqjagain@gmail.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 3c3acb4cba830caf17accd44ec7e11a8026f9e03
  Author: Masami Hiramatsu <mhiramat@kernel.org>
  Date:   Mon May 25 19:20:57 2020 +0900

      selftests/ftrace: Return unsupported if no error_log file

      commit 619ee76f5c9f6a1d601d1a056a454d62bf676ae4 upstream.

      Check whether error_log file exists in tracing/error_log testcase
      and return UNSUPPORTED if no error_log file.

      This can happen if we run the ftracetest on the older stable
      kernel.

      Fixes: 4eab1cc461a6 ("selftests/ftrace: Add tracing/error_log testcase")
      Cc: stable@vger.kernel.org
      Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
      Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 0b7d22d48cfe9bde5f69dda7f8195e7f1acbeeef
  Author: Shivasharan S <shivasharan.srikanteshwara@broadcom.com>
  Date:   Fri May 8 14:21:30 2020 +0530

      scsi: megaraid_sas: Replace undefined MFI_BIG_ENDIAN macro with __BIG_ENDIAN_BITFIELD
  macro

      commit b9d5e3e7f370a817c742fb089ac1a86dfe8947dc upstream.

      MFI_BIG_ENDIAN macro used in drivers structure bitfield to check the CPU
      big endianness is undefined which would break the code on big endian
      machine. __BIG_ENDIAN_BITFIELD kernel macro should be used in places of
      MFI_BIG_ENDIAN macro.

      Link: https://lore.kernel.org/r/20200508085130.23339-1-chandrakanth.patil@broadcom.com
      Fixes: a7faf81d7858 ("scsi: megaraid_sas: Set no_write_same only for Virtual Disk")
      Cc: <stable@vger.kernel.org> # v5.6+
      Signed-off-by: Shivasharan S <shivasharan.srikanteshwara@broadcom.com>
      Signed-off-by: Chandrakanth Patil <chandrakanth.patil@broadcom.com>
```

```
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 937d502f2192d25b09c3bd6c100b93e949a5581f
Author: Dick Kennedy <dick.kennedy@broadcom.com>
Date:   Fri May 1 14:43:05 2020 -0700

    scsi: lpfc: Fix negation of else clause in lpfc_prep_node_fc4type

    commit f809da6db68a8be49e317f0ccfbced1af9258839 upstream.

    Implementation of a previous patch added a condition to an if check that
    always end up with the if test being true. Execution of the else clause was
    inadvertently negated.  The additional condition check was incorrect and
    unnecessary after the other modifications had been done in that patch.

    Remove the check from the if series.

    Link: https://lore.kernel.org/r/20200501214310.91713-5-jsmart2021@gmail.com
    Fixes: b95b21193c85 ("scsi: lpfc: Fix loss of remote port after devloss due to lack of
RPIs")
    Cc: <stable@vger.kernel.org> # v5.4+
    Reviewed-by: Hannes Reinecke <hare@suse.de>
    Signed-off-by: Dick Kennedy <dick.kennedy@broadcom.com>
    Signed-off-by: James Smart <jsmart2021@gmail.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9c4eb2eb0cd984e257cc0b2ad8b42143635ee266
Author: Sumit Saxena <sumit.saxena@broadcom.com>
Date:   Fri May 8 14:22:42 2020 +0530

    scsi: megaraid_sas: TM command refire leads to controller firmware crash

    commit 6fd8525a70221c26823b1c7e912fb21f218fb0c5 upstream.

    When TM command times out, driver invokes the controller reset. Post reset,
    driver re-fires pended TM commands which leads to firmware crash.

    Post controller reset, return pended TM commands back to OS.

    Link: https://lore.kernel.org/r/20200508085242.23406-1-chandrakanth.patil@broadcom.com
    Cc: stable@vger.kernel.org
    Signed-off-by: Sumit Saxena <sumit.saxena@broadcom.com>
    Signed-off-by: Chandrakanth Patil <chandrakanth.patil@broadcom.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a51cb2fd5764880e31666982dee0cc9020158634
Author: Marc Zyngier <maz@kernel.org>
Date:   Tue Jun 9 08:40:35 2020 +0100

    KVM: arm64: Make vcpu_cp1x() work on Big Endian hosts

    commit 3204be4109ad681523e3461ce64454c79278450a upstream.

    AArch32 CP1x registers are overlayed on their AArch64 counterparts
    in the vcpu struct. This leads to an interesting problem as they
    are stored in their CPU-local format, and thus a CP1x register
    doesn't "hit" the lower 32bit portion of the AArch64 register on
    a BE host.

    To workaround this unfortunate situation, introduce a bias trick
    in the vcpu_cp1x() accessors which picks the correct half of the
    64bit register.
```

```
    Cc: stable@vger.kernel.org
    Reported-by: James Morse <james.morse@arm.com>
    Tested-by: James Morse <james.morse@arm.com>
    Acked-by: James Morse <james.morse@arm.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 47a4013ea7cbe6bf52939b55ed9ac7cdf60ac0eb
Author: James Morse <james.morse@arm.com>
Date:   Fri May 29 15:06:54 2020 +0000

    KVM: arm64: Stop writing aarch32's CSSELR into ACTLR

    commit 7c582bf4ed84f3eb58bdd1f63024a14c17551e7d upstream.

    aarch32 has pairs of registers to access the high and low parts of 64bit
    registers. KVM has a union of 64bit sys_regs[] and 32bit copro[]. The
    32bit accessors read the high or low part of the 64bit sys_reg[] value
    through the union.

    Both sys_reg_descs[] and cp15_regs[] list access_csselr() as the accessor
    for CSSELR{,_EL1}. access_csselr() is only aware of the 64bit sys_regs[],
    and expects r->reg to be 'CSSELR_EL1' in the enum, index 2 of the 64bit
    array.

    cp15_regs[] uses the 32bit copro[] alias of sys_regs[]. Here CSSELR is
    c0_CSSELR which is the same location in sys_reg[]. r->reg is 'c0_CSSELR',
    index 4 in the 32bit array.

    access_csselr() uses the 32bit r->reg value to access the 64bit array,
    so reads and write the wrong value. sys_regs[4], is ACTLR_EL1, which
    is subsequently save/restored when we enter the guest.

    ACTLR_EL1 is supposed to be read-only for the guest. This register
    only affects execution at EL1, and the host's value is restored before
    we return to host EL1.

    Convert the 32bit register index back to the 64bit version.

    Suggested-by: Marc Zyngier <maz@kernel.org>
    Signed-off-by: James Morse <james.morse@arm.com>
    Signed-off-by: Marc Zyngier <maz@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200529150656.7339-2-james.morse@arm.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1c7cb7005c4e318f196dbadb7474e05ccaa27343
Author: Xing Li <lixing@loongson.cn>
Date:   Sat May 23 15:56:29 2020 +0800

    KVM: MIPS: Fix VPN2_MASK definition for variable cpu_vmbits

    commit 5816c76dea116a458f1932eefe064e35403248eb upstream.

    If a CPU support more than 32bit vmbits (which is true for 64bit CPUs),
    VPN2_MASK set to fixed 0xffffe000 will lead to a wrong EntryHi in some
    functions such as _kvm_mips_host_tlb_inv().

    The cpu_vmbits definition of 32bit CPU in cpu-features.h is 31, so we
    still use the old definition.

    Cc: Stable <stable@vger.kernel.org>
    Reviewed-by: Aleksandar Markovic <aleksandar.qemu.devel@gmail.com>
    Signed-off-by: Xing Li <lixing@loongson.cn>
```

```
    [Huacai: Improve commit messages]
    Signed-off-by: Huacai Chen <chenhc@lemote.com>
    Message-Id: <1590220602-3547-3-git-send-email-chenhc@lemote.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2a4f0a1f214e9ca0befa7627bdb9c0703168308
Author: Xing Li <lixing@loongson.cn>
Date:   Sat May 23 15:56:28 2020 +0800

    KVM: MIPS: Define KVM_ENTRYHI_ASID to cpu_asid_mask(&boot_cpu_data)

    commit fe2b73dba47fb6d6922df1ad44e83b1754d5ed4d upstream.

    The code in decode_config4() of arch/mips/kernel/cpu-probe.c

            asid_mask = MIPS_ENTRYHI_ASID;
            if (config4 & MIPS_CONF4_AE)
                    asid_mask |= MIPS_ENTRYHI_ASIDX;
            set_cpu_asid_mask(c, asid_mask);

    set asid_mask to cpuinfo->asid_mask.

    So in order to support variable ASID_MASK, KVM_ENTRYHI_ASID should also
    be changed to cpu_asid_mask(&boot_cpu_data).

    Cc: Stable <stable@vger.kernel.org>  #4.9+
    Reviewed-by: Aleksandar Markovic <aleksandar.qemu.devel@gmail.com>
    Signed-off-by: Xing Li <lixing@loongson.cn>
    [Huacai: Change current_cpu_data to boot_cpu_data for optimization]
    Signed-off-by: Huacai Chen <chenhc@lemote.com>
    Message-Id: <1590220602-3547-2-git-send-email-chenhc@lemote.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7359437efd014bccd8d8cb57d5d3157205cfa19c
Author: Sean Christopherson <sean.j.christopherson@intel.com>
Date:   Thu Feb 27 09:44:30 2020 -0800

    KVM: nVMX: Consult only the "basic" exit reason when routing nested exit

    commit 2ebac8bb3c2d35f5135466490fc8eeaf3f3e2d37 upstream.

    Consult only the basic exit reason, i.e. bits 15:0 of vmcs.EXIT_REASON,
    when determining whether a nested VM-Exit should be reflected into L1 or
    handled by KVM in L0.

    For better or worse, the switch statement in nested_vmx_exit_reflected()
    currently defaults to "true", i.e. reflects any nested VM-Exit without
    dedicated logic.  Because the case statements only contain the basic
    exit reason, any VM-Exit with modifier bits set will be reflected to L1,
    even if KVM intended to handle it in L0.

    Practically speaking, this only affects EXIT_REASON_MCE_DURING_VMENTRY,
    i.e. a #MC that occurs on nested VM-Enter would be incorrectly routed to
    L1, as "failed VM-Entry" is the only modifier that KVM can currently
    encounter.  The SMM modifiers will never be generated as KVM doesn't
    support/employ a SMI Transfer Monitor.  Ditto for "exit from enclave",
    as KVM doesn't yet support virtualizing SGX, i.e. it's impossible to
    enter an enclave in a KVM guest (L1 or L2).

    Fixes: 644d711aa0e1 ("KVM: nVMX: Deciding if L0 or L1 should handle an L2 exit")
    Cc: Jim Mattson <jmattson@google.com>
    Cc: Xiaoyao Li <xiaoyao.li@intel.com>
    Cc: stable@vger.kernel.org
```

```
    Signed-off-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Message-Id: <20200227174430.26371-1-sean.j.christopherson@intel.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d589685262f30656c4c7bf652fb47da059c61561
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Wed May 20 08:02:17 2020 -0400

    KVM: nSVM: leave ASID aside in copy_vmcb_control_area

    commit 6c0238c4a62b3a0b1201aeb7e33a4636d552a436 upstream.

    Restoring the ASID from the hsave area on VMEXIT is wrong, because its
    value depends on the handling of TLB flushes.  Just skipping the field in
    copy_vmcb_control_area will do.

    Cc: stable@vger.kernel.org
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a34683aad894ab54f4201d21b00d7d5b4a4244e5
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Sat May 16 09:19:06 2020 -0400

    KVM: nSVM: fix condition for filtering async PF

    commit a3535be731c2a343912578465021f50937f7b099 upstream.

    Async page faults have to be trapped in the host (L1 in this case),
    since the APF reason was passed from L0 to L1 and stored in the L1 APF
    data page.  This was completely reversed: the page faults were passed
    to the guest, a L2 hypervisor.

    Cc: stable@vger.kernel.org
    Reviewed-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 02a3868d18fe639363f1cea6e6d7914513198e43
Author: Sean Christopherson <sean.j.christopherson@intel.com>
Date:   Fri May 1 09:31:17 2020 -0700

    KVM: nVMX: Skip IBPB when switching between vmcs01 and vmcs02

    commit 5c911beff20aa8639e7a1f28988736c13e03ed54 upstream.

    Skip the Indirect Branch Prediction Barrier that is triggered on a VMCS
    switch when running with spectre_v2_user=on/auto if the switch is
    between two VMCSes in the same guest, i.e. between vmcs01 and vmcs02.
    The IBPB is intended to prevent one guest from attacking another, which
    is unnecessary in the nested case as it's the same guest from KVM's
    perspective.

    This all but eliminates the overhead observed for nested VMX transitions
    when running with CONFIG_RETPOLINE=y and spectre_v2_user=on/auto, which
    can be significant, e.g. roughly 3x on current systems.

    Reported-by: Alexander Graf <graf@amazon.com>
    Cc: KarimAllah Raslan <karahmed@amazon.de>
    Cc: stable@vger.kernel.org
    Fixes: 15d45071523d ("KVM/x86: Add IBPB support")
    Signed-off-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Message-Id: <20200501163117.4655-1-sean.j.christopherson@intel.com>
    [Invert direction of bool argument. - Paolo]
```

```
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 338333e1757d5c5680f392e806dd2ac9727388f4
Author: Tomi Valkeinen <tomi.valkeinen@ti.com>
Date:   Wed May 27 10:23:34 2020 +0200

    media: videobuf2-dma-contig: fix bad kfree in vb2_dma_contig_clear_max_seg_size

    commit 0d9668721311607353d4861e6c32afeb272813dc upstream.

    Commit 9495b7e92f716ab2bd6814fab5e97ab4a39adfdd ("driver core: platform:
    Initialize dma_parms for platform devices") in v5.7-rc5 causes
    vb2_dma_contig_clear_max_seg_size() to kfree memory that was not
    allocated by vb2_dma_contig_set_max_seg_size().

    The assumption in vb2_dma_contig_set_max_seg_size() seems to be that
    dev->dma_parms is always NULL when the driver is probed, and the case
    where dev->dma_parms has bee initialized by someone else than the driver
    (by calling vb2_dma_contig_set_max_seg_size) will cause a failure.

    All the current users of these functions are platform devices, which now
    always have dma_parms set by the driver core. To fix the issue for v5.7,
    make vb2_dma_contig_set_max_seg_size() return an error if dma_parms is
    NULL to be on the safe side, and remove the kfree code from
    vb2_dma_contig_clear_max_seg_size().

    For v5.8 we should remove the two functions and move the
    dma_set_max_seg_size() calls into the drivers.

    Signed-off-by: Tomi Valkeinen <tomi.valkeinen@ti.com>
    Fixes: 9495b7e92f71 ("driver core: platform: Initialize dma_parms for platform devices")
    Cc: stable@vger.kernel.org
    Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
    Reviewed-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 921335091f4ba024cd757d2133eee9eb2d767c7e
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Wed May 6 20:19:02 2020 +0200

    video: fbdev: w100fb: Fix a potential double free.

    commit 18722d48a6bb9c2e8d046214c0a5fd19d0a7c9f6 upstream.

    Some memory is vmalloc'ed in the 'w100fb_save_vidmem' function and freed in
    the 'w100fb_restore_vidmem' function. (these functions are called
    respectively from the 'suspend' and the 'resume' functions)

    However, it is also freed in the 'remove' function.

    In order to avoid a potential double free, set the corresponding pointer
    to NULL once freed in the 'w100fb_restore_vidmem' function.

    Fixes: aac51f09d96a ("[PATCH] w100fb: Rewrite for platform independence")
    Cc: Richard Purdie <rpurdie@rpsys.net>
    Cc: Antonino Daplas <adaplas@pol.net>
    Cc: Bartlomiej Zolnierkiewicz <b.zolnierkie@samsung.com>
    Cc: <stable@vger.kernel.org> # v2.6.14+
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Signed-off-by: Sam Ravnborg <sam@ravnborg.org>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200506181902.193290-1-
christophe.jaillet@wanadoo.fr
```

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

commit 4ec5f20ec1857ca985a4d6ad3c044768af7ff719
Author: Sam Ravnborg <sam@ravnborg.org>
Date:   Sun Apr 12 22:21:43 2020 +0200

    video: vt8500lcdfb: fix fallthrough warning

    commit 1c49f35e9e9156273124a0cfd38b57f7a7d4828f upstream.

    Fix following warning:
    vt8500lcdfb.c: In function 'vt8500lcd_blank':
    vt8500lcdfb.c:229:6: warning: this statement may fall through [-Wimplicit-fallthrough=]
          if (info->fix.visual == FB_VISUAL_PSEUDOCOLOR ||
              ^
    vt8500lcdfb.c:233:2: note: here
         case FB_BLANK_UNBLANK:
         ^~~~

    Adding a simple "fallthrough;" fixed the warning.
    The fix was build tested.

    Signed-off-by: Sam Ravnborg <sam@ravnborg.org>
    Reported-by: kbuild test robot <lkp@intel.com>
    Fixes: e41f1a989408 ("fbdev: Implement simple blanking in pseudocolor modes for
vt8500lcdfb")
    Cc: Alexey Charkov <alchark@gmail.com>
    Cc: Paul Mundt <lethal@linux-sh.org>
    Cc: <stable@vger.kernel.org> # v2.6.38+
    Signed-off-by: Bartlomiej Zolnierkiewicz <b.zolnierkie@samsung.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200412202143.GA26948@ravnborg.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a706b8ecbeafc8d7e613083df6734642e57227ce
Author: Qiuxu Zhuo <qiuxu.zhuo@intel.com>
Date:   Fri May 15 20:34:06 2020 +0800

    EDAC/skx: Use the mcmtr register to retrieve close_pg/bank_xor_enable

    commit 1032095053b34d474aa20f2625d97dd306e0991b upstream.

    The skx_edac driver wrongly uses the mtr register to retrieve two fields
    close_pg and bank_xor_enable. Fix it by using the correct mcmtr register
    to get the two fields.

    Cc: <stable@vger.kernel.org>
    Signed-off-by: Qiuxu Zhuo <qiuxu.zhuo@intel.com>
    Reported-by: Matthew Riley <mattdr@google.com>
    Acked-by: Aristeu Rozanski <aris@redhat.com>
    Signed-off-by: Tony Luck <tony.luck@intel.com>
    Link: https://lore.kernel.org/r/20200515210146.1337-1-tony.luck@intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 22ce3bd338368f227c4ca588e9e337da1e040c5c
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Mon May 18 12:49:45 2020 +0200

    cpufreq: Fix up cpufreq_boost_set_sw()

    commit 552abb884e97d26589964e5a8c7e736f852f95f0 upstream.

    After commit 18c49926c4bf ("cpufreq: Add QoS requests for userspace
    constraints") the return value of freq_qos_update_request(), that can
    be 1, passed by cpufreq_boost_set_sw() to its caller sometimes
    confuses the latter, which only expects to see 0 or negative error
```

```
    codes, so notice that cpufreq_boost_set_sw() can return an error code
    (which should not be -EINVAL for that matter) as soon as the first
    policy without a frequency table is found (because either all policies
    have a frequency table or none of them have it) and rework it to meet
    its caller's expectations.

    Fixes: 18c49926c4bf ("cpufreq: Add QoS requests for userspace constraints")
    Reported-by: Serge Semin <Sergey.Semin@baikalelectronics.ru>
    Reported-by: Xiongfeng Wang <wangxiongfeng2@huawei.com>
    Acked-by: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: 5.3+ <stable@vger.kernel.org> # 5.3+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 30297fb509062d0bbf3b1f4ed677000bbdafd79e
Author: Suman Anna <s-anna@ti.com>
Date:   Mon Apr 20 11:06:00 2020 -0500

    remoteproc: Fix and restore the parenting hierarchy for vdev

    commit c774ad010873bb89dcc0cdcb1e96aef6664d8caf upstream.

    The commit 086d08725d34 ("remoteproc: create vdev subdevice with specific
    dma memory pool") has introduced a new vdev subdevice for each vdev
    declared in the firmware resource table and made it as the parent for the
    created virtio rpmsg devices instead of the previous remoteproc device.
    This changed the overall parenting hierarchy for the rpmsg devices, which
    were children of virtio devices, and does not allow the corresponding
    rpmsg drivers to retrieve the parent rproc device through the
    rproc_get_by_child() API.

    Fix this by restoring the remoteproc device as the parent. The new vdev
    subdevice can continue to inherit the DMA attributes from the remoteproc's
    parent device (actual platform device).

    Cc: stable@vger.kernel.org
    Fixes: 086d08725d34 ("remoteproc: create vdev subdevice with specific dma memory pool")
    Signed-off-by: Suman Anna <s-anna@ti.com>
    Reviewed-by: Mathieu Poirier <mathieu.poirier@linaro.org>
    Acked-by: Arnaud Pouliquen <arnaud.pouliquen@st.com>
    Link: https://lore.kernel.org/r/20200420160600.10467-3-s-anna@ti.com
    Signed-off-by: Bjorn Andersson <bjorn.andersson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7272546c2081f0f697f7aae8833cccea407ef46c
Author: Tero Kristo <t-kristo@ti.com>
Date:   Mon Apr 20 11:05:59 2020 -0500

    remoteproc: Fall back to using parent memory pool if no dedicated available

    commit db9178a4f8c4e523f824892cb8bab00961b07385 upstream.

    In some cases, like with OMAP remoteproc, we are not creating dedicated
    memory pool for the virtio device. Instead, we use the same memory pool
    for all shared memories. The current virtio memory pool handling forces
    a split between these two, as a separate device is created for it,
    causing memory to be allocated from bad location if the dedicated pool
    is not available. Fix this by falling back to using the parent device
    memory pool if dedicated is not available.

    Cc: stable@vger.kernel.org
    Reviewed-by: Mathieu Poirier <mathieu.poirier@linaro.org>
    Acked-by: Arnaud Pouliquen <arnaud.pouliquen@st.com>
    Fixes: 086d08725d34 ("remoteproc: create vdev subdevice with specific dma memory pool")
    Signed-off-by: Tero Kristo <t-kristo@ti.com>
```

```
    Signed-off-by: Suman Anna <s-anna@ti.com>
    Link: https://lore.kernel.org/r/20200420160600.10467-2-s-anna@ti.com
    Signed-off-by: Bjorn Andersson <bjorn.andersson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 476bdd28059a987b860a2c3aaade706eeac971a8
Author: Eric W. Biederman <ebiederm@xmission.com>
Date:   Fri Jun 12 09:42:03 2020 -0500

    proc: Use new_inode not new_inode_pseudo

    commit ef1548adada51a2f32ed7faef50aa465e1b4c5da upstream.

    Recently syzbot reported that unmounting proc when there is an ongoing
    inotify watch on the root directory of proc could result in a use
    after free when the watch is removed after the unmount of proc
    when the watcher exits.

    Commit 69879c01a0c3 ("proc: Remove the now unnecessary internal mount
    of proc") made it easier to unmount proc and allowed syzbot to see the
    problem, but looking at the code it has been around for a long time.

    Looking at the code the fsnotify watch should have been removed by
    fsnotify_sb_delete in generic_shutdown_super.  Unfortunately the inode
    was allocated with new_inode_pseudo instead of new_inode so the inode
    was not on the sb->s_inodes list.  Which prevented
    fsnotify_unmount_inodes from finding the inode and removing the watch
    as well as made it so the "VFS: Busy inodes after unmount" warning
    could not find the inodes to warn about them.

    Make all of the inodes in proc visible to generic_shutdown_super,
    and fsnotify_sb_delete by using new_inode instead of new_inode_pseudo.
    The only functional difference is that new_inode places the inodes
    on the sb->s_inodes list.

    I wrote a small test program and I can verify that without changes it
    can trigger this issue, and by replacing new_inode_pseudo with
    new_inode the issues goes away.

    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/000000000000d788c905a7dfa3f4@google.com
    Reported-by: syzbot+7d2debdcdb3cb93c1e5e@syzkaller.appspotmail.com
    Fixes: 0097875bd415 ("proc: Implement /proc/thread-self to point at the directory of the
current thread")
    Fixes: 021ada7dff22 ("procfs: switch /proc/self away from proc_dir_entry")
    Fixes: 51f0885e5415 ("vfs,proc: guarantee unique inodes in /proc")
    Signed-off-by: "Eric W. Biederman" <ebiederm@xmission.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 213fdef3c1c660d0b64fd8fb0b5612eef005e661
Author: Namjae Jeon <namjae.jeon@samsung.com>
Date:   Thu Jun 4 08:05:31 2020 +0900

    exfat: fix incorrect update of stream entry in __exfat_truncate()

    commit 29bbb14bfc80dd760b07d2be0a27e610562982e3 upstream.

    At truncate, there is a problem of incorrect updating in the file entry
    pointer instead of stream entry. This will cause the problem of
    overwriting the time field of the file entry to new_size. Fix it to
    update stream entry.

    Fixes: 98d917047e8b ("exfat: add file operations")
    Cc: stable@vger.kernel.org # v5.7
    Signed-off-by: Namjae Jeon <namjae.jeon@samsung.com>
```

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

commit 98c0af94c22af9622d9eb2eb484f4436ee4063a9
Author: Al Viro <viro@zeniv.linux.org.uk>
Date:   Wed Jun 3 09:48:36 2020 +0900

    exfat: fix memory leak in exfat_parse_param()

    commit f341a7d8dcc4e3d01544d7bc145633f062ef6249 upstream.

    butt3rflyh4ck reported memory leak found by syzkaller.

    A param->string held by exfat_mount_options.

    BUG: memory leak

    unreferenced object 0xffff88801972e090 (size 8):
      comm "syz-executor.2", pid 16298, jiffies 4295172466 (age 14.060s)
      hex dump (first 8 bytes):
        6b 6f 69 38 2d 75 00 00                          koi8-u..
      backtrace:
        [<000000005bfe35d6>] kstrdup+0x36/0x70 mm/util.c:60
        [<0000000018ed3277>] exfat_parse_param+0x160/0x5e0
    fs/exfat/super.c:276
        [<000000007680462b>] vfs_parse_fs_param+0x2b4/0x610
    fs/fs_context.c:147
        [<0000000097c027f2>] vfs_parse_fs_string+0xe6/0x150
    fs/fs_context.c:191
        [<00000000371bf78f>] generic_parse_monolithic+0x16f/0x1f0
    fs/fs_context.c:231
        [<000000005ce5eb1b>] do_new_mount fs/namespace.c:2812 [inline]
        [<000000005ce5eb1b>] do_mount+0x12bb/0x1b30 fs/namespace.c:3141
        [<00000000b642040c>] __do_sys_mount fs/namespace.c:3350 [inline]
        [<00000000b642040c>] __se_sys_mount fs/namespace.c:3327 [inline]
        [<00000000b642040c>] __x64_sys_mount+0x18f/0x230 fs/namespace.c:3327
        [<000000003b024e98>] do_syscall_64+0xf6/0x7d0
    arch/x86/entry/common.c:295
        [<00000000ce2b698c>] entry_SYSCALL_64_after_hwframe+0x49/0xb3

    exfat_free() should call exfat_free_iocharset(), to prevent a leak
    in case we fail after parsing iocharset= but before calling
    get_tree_bdev().

    Additionally, there's no point copying param->string in
    exfat_parse_param() - just steal it, leaving NULL in param->string.
    That's independent from the leak or fix thereof - it's simply
    avoiding an extra copy.

    Fixes: 719c1e182916 ("exfat: add super block operations")
    Cc: stable@vger.kernel.org # v5.7
    Reported-by: butt3rflyh4ck <butterflyhuangxx@gmail.com>
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
    Signed-off-by: Namjae Jeon <namjae.jeon@samsung.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 53625485952ff7adc39be535dfc385b6025f121d
Author: Yuxuan Shui <yshuiv7@gmail.com>
Date:   Wed May 27 04:08:02 2020 +0100

    ovl: initialize error in ovl_copy_xattr

    commit 520da69d265a91c6536c63851cbb8a53946974f0 upstream.

    In ovl_copy_xattr, if all the xattrs to be copied are overlayfs private
    xattrs, the copy loop will terminate without assigning anything to the
```

```
    error variable, thus returning an uninitialized value.

    If ovl_copy_xattr is called from ovl_clear_empty, this uninitialized error
    value is put into a pointer by ERR_PTR(), causing potential invalid memory
    accesses down the line.

    This commit initialize error with 0. This is the correct value because when
    there's no xattr to copy, because all xattrs are private, ovl_copy_xattr
    should succeed.

    This bug is discovered with the help of INIT_STACK_ALL and clang.

    Signed-off-by: Yuxuan Shui <yshuiv7@gmail.com>
    Link: https://bugs.chromium.org/p/chromium/issues/detail?id=1050405
    Fixes: 0956254a2d5b ("ovl: don't copy up opaqueness")
    Cc: stable@vger.kernel.org # v4.8
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Signed-off-by: Miklos Szeredi <mszeredi@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 59c79f3957450ca093f4928415fafa9e463ee030
Author: Amir Goldstein <amir73il@gmail.com>
Date:   Sat May 23 16:21:55 2020 +0300

    ovl: fix out of bounds access warning in ovl_check_fb_len()

    commit 522f6e6cba6880a038e2bd88e10390b84cd3febd upstream.

    syzbot reported out of bounds memory access from open_by_handle_at()
    with a crafted file handle that looks like this:

      { .handle_bytes = 2, .handle_type = OVL_FILEID_V1 }

    handle_bytes gets rounded down to 0 and we end up calling:
      ovl_check_fh_len(fh, 0) => ovl_check_fb_len(fh + 3, -3)

    But fh buffer is only 2 bytes long, so accessing struct ovl_fb at
    fh + 3 is illegal.

    Fixes: cbe7fba8edfc ("ovl: make sure that real fid is 32bit aligned in memory")
    Reported-and-tested-by: syzbot+61958888b1c60361a791@syzkaller.appspotmail.com
    Cc: <stable@vger.kernel.org> # v5.5
    Signed-off-by: Amir Goldstein <amir73il@gmail.com>
    Signed-off-by: Miklos Szeredi <mszeredi@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b68d196b0c6ad4738c70d7c2632c10f9aa1d03fc
Author: Oz Shlomo <ozsh@mellanox.com>
Date:   Sun Jun 7 15:40:40 2020 +0000

    net/mlx5e: CT: Fix ipv6 nat header rewrite actions

    [ Upstream commit 0d156f2deda8675c29fa2b8b5ed9b374370e47f2 ]

    Set the ipv6 word fields according to the hardware definitions.

    Fixes: ac991b48d43c ("net/mlx5e: CT: Offload established flows")
    Signed-off-by: Oz Shlomo <ozsh@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 733bcf622c8936fadd8b5beb3da95cf16f2c2a3f
Author: Valentin Longchamp <valentin@longchamp.me>
Date:   Tue Jun 9 22:11:54 2020 +0200
```

```
    net: sched: export __netdev_watchdog_up()

    [ Upstream commit 1a3db27ad9a72d033235b9673653962c02e3486e ]

    Since the quiesce/activate rework, __netdev_watchdog_up() is directly
    called in the ucc_geth driver.

    Unfortunately, this function is not available for modules and thus
    ucc_geth cannot be built as a module anymore. Fix it by exporting
    __netdev_watchdog_up().

    Since the commit introducing the regression was backported to stable
    branches, this one should ideally be as well.

    Fixes: 79dde73cf9bc ("net/ethernet/freescale: rework quiesce/activate for ucc_geth")
    Signed-off-by: Valentin Longchamp <valentin@longchamp.me>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0b8614d6e849879b537990c83cede93d4c466905
Author: Grygorii Strashko <grygorii.strashko@ti.com>
Date:   Sat Jun 13 17:52:59 2020 +0300

    net: ethernet: ti: am65-cpsw-nuss: fix ale parameters init

    [ Upstream commit 2074f9eaa58795a99e9da61c10f93180f810cfd6 ]

    The ALE parameters structure is created on stack, so it has to be reset
    before passing to cpsw_ale_create() to avoid garbage values.

    Fixes: 93a76530316a ("net: ethernet: ti: introduce am65x/j721e gigabit eth subsystem
driver")
    Signed-off-by: Grygorii Strashko <grygorii.strashko@ti.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 67160caf3eec2f15436e2cd636b4cdde164a7c32
Author: Grygorii Strashko <grygorii.strashko@ti.com>
Date:   Sat Jun 13 17:54:14 2020 +0300

    net: ethernet: ti: ale: fix allmulti for nu type ale

    [ Upstream commit bc139119a1708ae3db1ebb379630f286e28d06e8 ]

    On AM65xx MCU CPSW2G NUSS and 66AK2E/L NUSS allmulti setting does not allow
    unregistered mcast packets to pass.

    This happens, because ALE VLAN entries on these SoCs do not contain port
    masks for reg/unreg mcast packets, but instead store indexes of
    ALE_VLAN_MASK_MUXx_REG registers which intended for store port masks for
    reg/unreg mcast packets.
    This path was missed by commit 9d1f6447274f ("net: ethernet: ti: ale: fix
    seeing unreg mcast packets with promisc and allmulti disabled").

    Hence, fix it by taking into account ALE type in cpsw_ale_set_allmulti().

    Fixes: 9d1f6447274f ("net: ethernet: ti: ale: fix seeing unreg mcast packets with
promisc and allmulti disabled")
    Signed-off-by: Grygorii Strashko <grygorii.strashko@ti.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ca2dc67681633643a0166d04a9ffacdb1e3f2c60
Author: Paolo Abeni <pabeni@redhat.com>
```

```
Date:   Wed Jun 10 10:47:41 2020 +0200

    mptcp: fix races between shutdown and recvmsg

    [ Upstream commit 5969856ae8ce29c9d523a1a6145cbd9e87f7046c ]

    The msk sk_shutdown flag is set by a workqueue, possibly
    introducing some delay in user-space notification. If the last
    subflow carries some data with the fin packet, the user space
    can wake-up before RCV_SHUTDOWN is set. If it executes unblocking
    recvmsg(), it may return with an error instead of eof.

    Address the issue explicitly checking for eof in recvmsg(), when
    no data is found.

    Fixes: 59832e246515 ("mptcp: subflow: check parent mptcp socket on subflow state
change")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Reviewed-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 07f111d902a94242d823fce881d6cb437e2402b2
Author: Shannon Nelson <snelson@pensando.io>
Date:   Mon Jun 8 20:41:43 2020 -0700

    ionic: wait on queue start until after IFF_UP

    [ Upstream commit 976ee3b21119dcf5c6d96233d688a1453f29fa83 ]

    The netif_running() test looks at __LINK_STATE_START which
    gets set before ndo_open() is called, there is a window of
    time between that and when the queues are actually ready to
    be run.  If ionic_check_link_status() notices that the link is
    up very soon after netif_running() becomes true, it might try
    to run the queues before they are ready, causing all manner of
    potential issues.  Since the netdev->flags IFF_UP isn't set
    until after ndo_open() returns, we can wait for that before
    we allow ionic_check_link_status() to start the queues.

    On the way back to close, __LINK_STATE_START is cleared before
    calling ndo_stop(), and IFF_UP is cleared after.  Both of
    these need to be true in order to safely stop the queues
    from ionic_check_link_status().

    Fixes: 49d3b493673a ("ionic: disable the queues on link down")
    Signed-off-by: Shannon Nelson <snelson@pensando.io>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 513c4ffa0dd26cb8935c3cb1278d6fdecc73f886
Author: Paolo Abeni <pabeni@redhat.com>
Date:   Wed Jun 10 10:49:00 2020 +0200

    mptcp: don't leak msk in token container

    [ Upstream commit 4b5af44129d0653a4df44e5511c7d480c61c8f3c ]

    If a listening MPTCP socket has unaccepted sockets at close
    time, the related msks are freed via mptcp_sock_destruct(),
    which in turn does not invoke the proto->destroy() method
    nor the mptcp_token_destroy() function.

    Due to the above, the child msk socket is not removed from
    the token container, leading to later UaF.
```

```
    Address the issue explicitly removing the token even in the
    above error path.

    Fixes: 79c0949e9a09 ("mptcp: Add key generation and token tree")
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Reviewed-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 13f75d88f8f957f38c02a1a19467f507fde1c067
Author: Parav Pandit <parav@mellanox.com>
Date:   Thu May 14 05:12:56 2020 -0500

    net/mlx5: Disable reload while removing the device

    [ Upstream commit 60904cd349abc98cb888fc28d1ca55a8e2cf87b3 ]

    While unregistration is in progress, user might be reloading the
    interface.
    This can race with unregistration in below flow which uses the
    resources which are getting disabled by reload flow.

    Hence, disable the devlink reloading first when removing the device.

         CPU0                                    CPU1
         ----                                    ----
    local_pci_remove()                    devlink_mutex
      remove_one()                         devlink_nl_cmd_reload()
        mlx5_unregister_device()             devlink_reload()
                                             ops->reload_down()
                                               mlx5_unload_one()

    Fixes: 4383cfcc65e7 ("net/mlx5: Add devlink reload")
    Signed-off-by: Parav Pandit <parav@mellanox.com>
    Reviewed-by: Moshe Shemesh <moshe@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 89771f608f957772560d53869f123778f25b1eb4
Author: Charles Keepax <ckeepax@opensource.cirrus.com>
Date:   Mon Jun 15 14:18:54 2020 +0100

    net: macb: Only disable NAPI on the actual error path

    [ Upstream commit 939a5bf7c9b7a1ad9c5d3481c93766a522773531 ]

    A recent change added a disable to NAPI into macb_open, this was
    intended to only happen on the error path but accidentally applies
    to all paths. This causes NAPI to be disabled on the success path, which
    leads to the network to no longer functioning.

    Fixes: 014406babc1f ("net: cadence: macb: disable NAPI on error")
    Signed-off-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Tested-by: Corentin Labbe <clabbe@baylibre.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c859ce6585f4ceab82de512f64ed03d0ffbea322
Author: Corentin Labbe <clabbe@baylibre.com>
Date:   Wed Jun 10 09:53:44 2020 +0000

    net: cadence: macb: disable NAPI on error

    [ Upstream commit 014406babc1f5f887a08737566b5b356c7018242 ]
```

```
    When the PHY is not working, the macb driver crash on a second try to
    setup it.
    [   78.545994] macb e000b000.ethernet eth0: Could not attach PHY (-19)
    ifconfig: SIOCSIFFLAGS: No such device
    [   78.655457] ------------[ cut here ]------------
    [   78.656014] kernel BUG at /linux-next/include/linux/netdevice.h:521!
    [   78.656504] Internal error: Oops - BUG: 0 [#1] SMP ARM
    [   78.657079] Modules linked in:
    [   78.657795] CPU: 0 PID: 122 Comm: ifconfig Not tainted 5.7.0-next-20200609 #1
    [   78.658202] Hardware name: Xilinx Zynq Platform
    [   78.659632] PC is at macb_open+0x220/0x294
    [   78.660160] LR is at 0x0
    [   78.660373] pc : [<c0b0a634>]    lr : [<00000000>]    psr: 60000013
    [   78.660716] sp : c89ffd70  ip : c8a28800  fp : c199bac0
    [   78.661040] r10: 00000000  r9 : c8838540  r8 : c8838568
    [   78.661362] r7 : 00000001  r6 : c8838000  r5 : c883c000  r4 : 00000000
    [   78.661724] r3 : 00000010  r2 : 00000000  r1 : 00000000  r0 : 00000000
    [   78.662187] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
    [   78.662635] Control: 10c5387d  Table: 08b64059  DAC: 00000051
    [   78.663035] Process ifconfig (pid: 122, stack limit = 0x(ptrval))
    [   78.663476] Stack: (0xc89ffd70 to 0xc8a00000)
    [   78.664121] fd60:                                     00000000 c89fe000 c8838000
c89fe000
    [   78.664866] fd80: 00000000 c11ff9ac c8838028 00000000 00000000 c0de6f2c 00000001
c1804eec
    [   78.665579] fda0: c19b8178 c8838000 00000000 ca760866 c8838000 00000001 00001043
c89fe000
    [   78.666355] fdc0: 00001002 c0de72f4 c89fe000 c0de8dc0 00008914 c89fe000 c199bac0
ca760866
    [   78.667111] fde0: c89ffddc c8838000 00001002 00000000 c8838138 c881010c 00008914
c0de7364
    [   78.667862] fe00: 00000000 c89ffe70 c89fe000 ffffffff c881010c c0e8bd48 00000003
00000000
    [   78.668601] fe20: c8838000 c8810100 39c1118f 00039c11 c89a0960 00001043 00000000
000a26d0
    [   78.669343] fe40: b6f43000 ca760866 c89a0960 00000051 befe6c50 00008914 c8b2a3c0
befe6c50
    [   78.670086] fe60: 00000003 ee610500 00000000 c0e8ef58 30687465 00000000 00000000
00000000
    [   78.670865] fe80: 00001043 00000000 000a26d0 b6f43000 c89a0600 ee40ae7c c8870d00
c0ddabf4
    [   78.671593] fea0: c89ffeec c0ddabf4 c89ffeec c199bac0 00008913 c0ddac48 c89ffeec
c89fe000
    [   78.672324] fec0: befe6c50 ca760866 befe6c50 00008914 c89fe000 befe6c50 c8b2a3c0
c0dc00e4
    [   78.673088] fee0: c89a0480 00000201 00000cc0 30687465 00000000 00000000 00000000
00001002
    [   78.673822] ff00: 00000000 000a26d0 b6f43000 ca760866 00008914 c8b2a3c0 000a0ec4
c8b2a3c0
    [   78.674576] ff20: befe6c50 c04b21bc 000d5004 00000817 c89a0480 c0315f94 00000000
00000003
    [   78.675415] ff40: c19a2bc8 c8a3cc00 c89fe000 00000255 00000000 00000000 00000000
000d5000
    [   78.676182] ff60: 000f6000 c180b2a0 00000817 c0315e64 000d5004 c89fffb0 b6ec0c30
ca760866
    [   78.676928] ff80: 00000000 000b609b befe6c50 000a0ec4 00000036 c03002c4 c89fe000
00000036
    [   78.677673] ffa0: 00000000 c03000c0 000b609b befe6c50 00000003 00008914 befe6c50
000b609b
    [   78.678415] ffc0: 000b609b befe6c50 000a0ec4 00000036 befe6e0c befe6f1a 000d5150
00000000
    [   78.679154] ffe0: 000d41e4 befe6bf4 00019648 b6e4509c 20000010 00000003 00000000
00000000
    [   78.681059] [<c0b0a634>] (macb_open) from [<c0de6f2c>] (__dev_open+0xd0/0x154)
```

```
    [   78.681571] [<c0de6f2c>] (__dev_open) from [<c0de72f4>]
(__dev_change_flags+0x16c/0x1c4)
    [   78.682015] [<c0de72f4>] (__dev_change_flags) from [<c0de7364>]
(dev_change_flags+0x18/0x48)
    [   78.682493] [<c0de7364>] (dev_change_flags) from [<c0e8bd48>]
(devinet_ioctl+0x5e4/0x75c)
    [   78.682945] [<c0e8bd48>] (devinet_ioctl) from [<c0e8ef58>] (inet_ioctl+0x1f0/0x3b4)
    [   78.683381] [<c0e8ef58>] (inet_ioctl) from [<c0dc00e4>] (sock_ioctl+0x39c/0x664)
    [   78.683818] [<c0dc00e4>] (sock_ioctl) from [<c04b21bc>] (ksys_ioctl+0x2d8/0x9c0)
    [   78.684343] [<c04b21bc>] (ksys_ioctl) from [<c03000c0>] (ret_fast_syscall+0x0/0x54)
    [   78.684789] Exception stack(0xc89fffa8 to 0xc89ffff0)
    [   78.685346] ffa0:                   000b609b befe6c50 00000003 00008914 befe6c50
000b609b
    [   78.686106] ffc0: 000b609b befe6c50 000a0ec4 00000036 befe6e0c befe6f1a 000d5150
00000000
    [   78.686710] ffe0: 000d41e4 befe6bf4 00019648 b6e4509c
    [   78.687582] Code: 9a000003 e5983078 e3130001 1affffef (e7f001f2)
    [   78.688788] ---[ end trace e3f2f6ab69754eae ]---

    This is due to NAPI left enabled if macb_phylink_connect() fail.

    Fixes: 7897b071ac3b ("net: macb: convert to phylink")
    Signed-off-by: Corentin Labbe <clabbe@baylibre.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a76c8834f7e9cb369e5151ea9195f3f7c473b6a
Author: Maxim Mikityanskiy <maximmi@mellanox.com>
Date:   Mon Jun 1 16:03:44 2020 +0300

    net/mlx5e: Fix repeated XSK usage on one channel

    [ Upstream commit 36d45fb9d2fdf348d778bfe73f0427db1c6f9bc7 ]

    After an XSK is closed, the relevant structures in the channel are not
    zeroed. If an XSK is opened the second time on the same channel without
    recreating channels, the stray values in the structures will lead to
    incorrect operation of queues, which causes CQE errors, and the new
    socket doesn't work at all.

    This patch fixes the issue by explicitly zeroing XSK-related structs in
    the channel on XSK close. Note that those structs are zeroed on channel
    creation, and usually a configuration change (XDP program is set)
    happens on XSK open, which leads to recreating channels, so typical XSK
    usecases don't suffer from this issue. However, if XSKs are opened and
    closed on the same channel without removing the XDP program, this bug
    reproduces.

    Fixes: db05815b36cb ("net/mlx5e: Add XSK zero-copy support")
    Signed-off-by: Maxim Mikityanskiy <maximmi@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 69d0c4bfbb6093c17d2395725fc67996d8fc55e2
Author: Shay Drory <shayd@mellanox.com>
Date:   Thu May 7 09:32:53 2020 +0300

    net/mlx5: Fix fatal error handling during device load

    [ Upstream commit b6e0b6bebe0732d5cac51f0791f269d2413b8980 ]

    Currently, in case of fatal error during mlx5_load_one(), we cannot
    enter error state until mlx5_load_one() is finished, what can take
    several minutes until commands will get timeouts, because these commands
    can't be processed due to the fatal error.
```

```
    Fix it by setting dev->state as MLX5_DEVICE_STATE_INTERNAL_ERROR before
    requesting the lock.

    Fixes: c1d4d2e92ad6 ("net/mlx5: Avoid calling sleeping function by the health poll
thread")
    Signed-off-by: Shay Drory <shayd@mellanox.com>
    Reviewed-by: Moshe Shemesh <moshe@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b393a071736ab898fc64780598f06ce5da3bf1ed
Author: Shay Drory <shayd@mellanox.com>
Date:   Wed May 6 15:59:48 2020 +0300

    net/mlx5: drain health workqueue in case of driver load error

    [ Upstream commit 42ea9f1b5c625fad225d4ac96a7e757dd4199d9c ]

    In case there is a work in the health WQ when we teardown the driver,
    in driver load error flow, the health work will try to read dev->iseg,
    which was already unmap in mlx5_pci_close().
    Fix it by draining the health workqueue first thing in mlx5_pci_close().

    Trace of the error:
    BUG: unable to handle page fault for address: ffffb5b141c18014
    PF: supervisor read access in kernel mode
    PF: error_code(0x0000) - not-present page
    PGD 1fe95d067 P4D 1fe95d067 PUD 1fe95e067 PMD 1b7823067 PTE 0
    Oops: 0000 [#1] SMP PTI
    CPU: 3 PID: 6755 Comm: kworker/u128:2 Not tainted 5.2.0-net-next-mlx5-hv_stats-over-
last-worked-hyperv #1
    Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS 090006
04/28/2016
    Workqueue: mlx5_healtha050:00:02.0 mlx5_fw_fatal_reporter_err_work [mlx5_core]
    RIP: 0010:ioread32be+0x30/0x40
    Code: 00 77 27 48 81 ff 00 00 01 00 76 07 0f b7 d7 ed 0f c8 c3 55 48 c7 c6 3b ee d5 9f
48 89 e5 e8 67 fc ff ff b8 ff ff ff ff 5d c3 <8b> 07 0f c8 c3 66 66 2e 0f 1f 84 00 00 00 00
00 48 81 fe ff ff 03
    RSP: 0018:ffffb5b14c56fd78 EFLAGS: 00010292
    RAX: ffffb5b141c18000 RBX: ffff8e9f78a801c0 RCX: 0000000000000000
    RDX: 0000000000000001 RSI: ffff8e9f7ecd7628 RDI: ffffb5b141c18014
    RBP: ffffb5b14c56fd90 R08: 0000000000000001 R09: 0000000000000000
    R10: ffff8e9f372a2c30 R11: ffff8e9f87f4bc40 R12: ffff8e9f372a1fc0
    R13: ffff8e9f78a80000 R14: ffffffffc07136a0 R15: ffff8e9f78ae6f20
    FS:  0000000000000000(0000) GS:ffff8e9f7ecc0000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: ffffb5b141c18014 CR3: 00000001c8f82006 CR4: 00000000003606e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     ? mlx5_health_try_recover+0x4d/0x270 [mlx5_core]
     mlx5_fw_fatal_reporter_recover+0x16/0x20 [mlx5_core]
     devlink_health_reporter_recover+0x1c/0x50
     devlink_health_report+0xfb/0x240
     mlx5_fw_fatal_reporter_err_work+0x65/0xd0 [mlx5_core]
     process_one_work+0x1fb/0x4e0
     ? process_one_work+0x16b/0x4e0
     worker_thread+0x4f/0x3d0
     kthread+0x10d/0x140
     ? process_one_work+0x4e0/0x4e0
     ? kthread_cancel_delayed_work_sync+0x20/0x20
     ret_from_fork+0x1f/0x30
    Modules linked in: nfsv3 rpcsec_gss_krb5 nfsv4 nfs fscache 8021q garp mrp stp llc
ipmi_devintf ipmi_msghandler rpcrdma rdma_ucm ib_iser rdma_cm ib_umad iw_cm ib_ipoib
libiscsi scsi_transport_iscsi ib_cm mlx5_ib ib_uverbs ib_core mlx5_core sb_edac
```

```
 crct10dif_pclmul crc32_pclmul ghash_clmulni_intel aesni_intel aes_x86_64 mlxfw crypto_simd
 cryptd glue_helper input_leds hyperv_fb intel_rapl_perf joydev serio_raw pci_hyperv
 pci_hyperv_mini mac_hid hv_balloon nfsd auth_rpcgss nfs_acl lockd grace sunrpc sch_fq_codel
 ip_tables x_tables autofs4 hv_utils hid_generic hv_storvsc ptp hid_hyperv hid hv_netvsc
 hyperv_keyboard pps_core scsi_transport_fc psmouse hv_vmbus i2c_piix4 floppy pata_acpi
     CR2: ffffb5b141c18014
     ---[ end trace b12c5503157cad24 ]---
     RIP: 0010:ioread32be+0x30/0x40
     Code: 00 77 27 48 81 ff 00 00 01 00 76 07 0f b7 d7 ed 0f c8 c3 55 48 c7 c6 3b ee d5 9f
 48 89 e5 e8 67 fc ff ff b8 ff ff ff ff 5d c3 <8b> 07 0f c8 c3 66 66 2e 0f 1f 84 00 00 00 00
 00 48 81 fe ff ff 03
     RSP: 0018:ffffb5b14c56fd78 EFLAGS: 00010292
     RAX: ffffb5b141c18000 RBX: ffff8e9f78a801c0 RCX: 0000000000000000
     RDX: 0000000000000001 RSI: ffff8e9f7ecd7628 RDI: ffffb5b141c18014
     RBP: ffffb5b14c56fd90 R08: 0000000000000001 R09: 0000000000000000
     R10: ffff8e9f372a2c30 R11: ffff8e9f87f4bc40 R12: ffff8e9f372a1fc0
     R13: ffff8e9f78a80000 R14: ffffffffc07136a0 R15: ffff8e9f78ae6f20
     FS:  0000000000000000(0000) GS:ffff8e9f7ecc0000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     CR2: ffffb5b141c18014 CR3: 00000001c8f82006 CR4: 00000000003606e0
     DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     BUG: sleeping function called from invalid context at ./include/linux/percpu-rwsem.h:38
     in_atomic(): 0, irqs_disabled(): 1, pid: 6755, name: kworker/u128:2
     INFO: lockdep is turned off.
     CPU: 3 PID: 6755 Comm: kworker/u128:2 Tainted: G      D           5.2.0-net-next-mlx5-
 hv_stats-over-last-worked-hyperv #1
     Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS 090006
 04/28/2016
     Workqueue: mlx5_healtha050:00:02.0 mlx5_fw_fatal_reporter_err_work [mlx5_core]
     Call Trace:
      dump_stack+0x63/0x88
      ___might_sleep+0x10a/0x130
      __might_sleep+0x4a/0x80
      exit_signals+0x33/0x230
      ? blocking_notifier_call_chain+0x16/0x20
      do_exit+0xb1/0xc30
      ? kthread+0x10d/0x140
      ? process_one_work+0x4e0/0x4e0

     Fixes: 52c368dc3da7 ("net/mlx5: Move health and page alloc init to mdev_init")
     Signed-off-by: Shay Drory <shayd@mellanox.com>
     Reviewed-by: Moshe Shemesh <moshe@mellanox.com>
     Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit f00d2d705bdd09e10078bcb77c8e162b2edfb755
 Author: tannerlove <tannerlove@google.com>
 Date:   Tue Jun 9 17:21:32 2020 -0400

     selftests/net: in rxtimestamp getopt_long needs terminating null entry

     [ Upstream commit 865a6cbb2288f8af7f9dc3b153c61b7014fdcf1e ]

     getopt_long requires the last element to be filled with zeros.
     Otherwise, passing an unrecognized option can cause a segfault.

     Fixes: 16e781224198 ("selftests/net: Add a test to validate behavior of rx timestamps")
     Signed-off-by: Tanner Love <tannerlove@google.com>
     Acked-by: Willem de Bruijn <willemb@google.com>
     Signed-off-by: David S. Miller <davem@davemloft.net>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 17a74b08054866708886e00aee140567e8352cca
 Author: Lorenzo Bianconi <lorenzo@kernel.org>
```

```
Date:   Tue Jun 9 00:02:39 2020 +0200

    net: mvneta: do not redirect frames during reconfiguration

    [ Upstream commit 62a502cc91f97e3ffd312d9b42e8d01a137c63ff ]

    Disable frames injection in mvneta_xdp_xmit routine during hw
    re-configuration in order to avoid hardware hangs

    Fixes: b0a43db9087a ("net: mvneta: add XDP_TX support")
    Signed-off-by: Lorenzo Bianconi <lorenzo@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a74374c8b01346fa892a12d0129855462c416764
Author: Wang Hai <wanghai38@huawei.com>
Date:   Tue Jun 9 22:18:16 2020 +0800

    dccp: Fix possible memleak in dccp_init and dccp_fini

    [ Upstream commit c96b6acc8f89a4a7f6258dfe1d077654c11415be ]

    There are some memory leaks in dccp_init() and dccp_fini().

    In dccp_fini() and the error handling path in dccp_init(), free lhash2
    is missing. Add inet_hashinfo2_free_mod() to do it.

    If inet_hashinfo2_init_mod() failed in dccp_init(),
    percpu_counter_destroy() should be called to destroy dccp_orphan_count.
    It need to goto out_free_percpu when inet_hashinfo2_init_mod() failed.

    Fixes: c92c81df93df ("net: dccp: fix kernel crash on module load")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: Wang Hai <wanghai38@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3e62100a178702387041ea48ba8b2bfe89e2e517
Author: Franck LENORMAND <franck.lenormand@nxp.com>
Date:   Thu Mar 26 00:00:05 2020 +0200

    firmware: imx: scu: Fix corruption of header

    [ Upstream commit f5f27b79eab80de0287c243a22169e4876b08d5e ]

    The header of the message to send can be changed if the
    response is longer than the request:
     - 1st word, the header is sent
     - the remaining words of the message are sent
     - the response is received asynchronously during the
       execution of the loop, changing the size field in
       the header
     - the for loop test the termination condition using
       the corrupted header

    It is the case for the API build_info which has just a
    header as request but 3 words in response.

    This issue is fixed storing the header locally instead of
    using a pointer on it.

    Fixes: edbee095fafb (firmware: imx: add SCU firmware driver support)

    Signed-off-by: Franck LENORMAND <franck.lenormand@nxp.com>
    Reviewed-by: Leonard Crestez <leonard.crestez@nxp.com>
```

```
    Signed-off-by: Leonard Crestez <leonard.crestez@nxp.com>
    Cc: stable@vger.kernel.org
    Reviewed-by: Dong Aisheng <aisheng.dong@nxp.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d74ae7e578c0c45a68e6a2c3302e429df109fd76
Author: Peng Fan <peng.fan@nxp.com>
Date:   Thu Mar 19 15:49:53 2020 +0800

    firmware: imx-scu: Support one TX and one RX

    [ Upstream commit f25a066d1a07affb7bea4e5d9c179c3338338e23 ]

    Current imx-scu requires four TX and four RX to communicate with
    SCU. This is low efficient and causes lots of mailbox interrupts.

    With imx-mailbox driver could support one TX to use all four transmit
    registers and one RX to use all four receive registers, imx-scu
    could use one TX and one RX.

    Signed-off-by: Peng Fan <peng.fan@nxp.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3486eedc5cc84065f79667e678ff6f081677c7db
Author: Tony Luck <tony.luck@intel.com>
Date:   Wed May 20 09:35:46 2020 -0700

    x86/{mce,mm}: Unmap the entire page if the whole page is affected and poisoned

    commit 17fae1294ad9d711b2c3dd0edef479d40c76a5e8 upstream.

    An interesting thing happened when a guest Linux instance took a machine
    check. The VMM unmapped the bad page from guest physical space and
    passed the machine check to the guest.

    Linux took all the normal actions to offline the page from the process
    that was using it. But then guest Linux crashed because it said there
    was a second machine check inside the kernel with this stack trace:

    do_memory_failure
        set_mce_nospec
             set_memory_uc
                  _set_memory_uc
                       change_page_attr_set_clr
                            cpa_flush
                                 clflush_cache_range_opt

    This was odd, because a CLFLUSH instruction shouldn't raise a machine
    check (it isn't consuming the data). Further investigation showed that
    the VMM had passed in another machine check because is appeared that the
    guest was accessing the bad page.

    Fix is to check the scope of the poison by checking the MCi_MISC register.
    If the entire page is affected, then unmap the page. If only part of the
    page is affected, then mark the page as uncacheable.

    This assumes that VMMs will do the logical thing and pass in the "whole
    page scope" via the MCi_MISC register (since they unmapped the entire
    page).

      [ bp: Adjust to x86/entry changes. ]

    Fixes: 284ce4011ba6 ("x86/memory_failure: Introduce {set, clear}_mce_nospec()")
```

```
    Reported-by: Jue Wang <juew@google.com>
    Signed-off-by: Tony Luck <tony.luck@intel.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Tested-by: Jue Wang <juew@google.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lkml.kernel.org/r/20200520163546.GA7977@agluck-desk2.amr.corp.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 986fd581f5ddf6546ef82dcbc96d055a13461bd5
Author: Longpeng(Mike) <longpeng2@huawei.com>
Date:   Tue Jun 2 15:04:59 2020 +0800

    crypto: virtio: Fix src/dst scatterlist calculation in __virtio_crypto_skcipher_do_req()

    commit b02989f37fc5e865ceeee9070907e4493b3a21e2 upstream.

    The system will crash when the users insmod crypto/tcrypt.ko with mode=38
    ( testing "cts(cbc(aes))" ).

    Usually the next entry of one sg will be @sg@ + 1, but if this sg element
    is part of a chained scatterlist, it could jump to the start of a new
    scatterlist array. Fix it by sg_next() on calculation of src/dst
    scatterlist.

    Fixes: dbaf0624ffa5 ("crypto: add virtio-crypto driver")
    Reported-by: LABBE Corentin <clabbe@baylibre.com>
    Cc: Herbert Xu <herbert@gondor.apana.org.au>
    Cc: "Michael S. Tsirkin" <mst@redhat.com>
    Cc: Jason Wang <jasowang@redhat.com>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: virtualization@lists.linux-foundation.org
    Cc: linux-kernel@vger.kernel.org
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200123101000.GB24255@Red
    Signed-off-by: Gonglei <arei.gonglei@huawei.com>
    Signed-off-by: Longpeng(Mike) <longpeng2@huawei.com>
    Link: https://lore.kernel.org/r/20200602070501.2023-2-longpeng2@huawei.com
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b336d9bb00e60d81dbda15046e0856c01dd42707
Author: Longpeng(Mike) <longpeng2@huawei.com>
Date:   Tue Jun 2 15:05:00 2020 +0800

    crypto: virtio: Fix use-after-free in virtio_crypto_skcipher_finalize_req()

    commit 8c855f0720ff006d75d0a2512c7f6c4f60ff60ee upstream.

    The system'll crash when the users insmod crypto/tcrypto.ko with mode=155
    ( testing "authenc(hmac(sha1),cbc(aes))" ). It's caused by reuse the memory
    of request structure.

    In crypto_authenc_init_tfm(), the reqsize is set to:
      [PART 1] sizeof(authenc_request_ctx) +
      [PART 2] ictx->reqoff +
      [PART 3] MAX(ahash part, skcipher part)
    and the 'PART 3' is used by both ahash and skcipher in turn.

    When the virtio_crypto driver finish skcipher req, it'll call ->complete
    callback(in crypto_finalize_skcipher_request) and then free its
    resources whose pointers are recorded in 'skcipher parts'.

    However, the ->complete is 'crypto_authenc_encrypt_done' in this case,
    it will use the 'ahash part' of the request and change its content,
```

```
    so virtio_crypto driver will get the wrong pointer after ->complete
    finish and mistakenly free some other's memory. So the system will crash
    when these memory will be used again.

    The resources which need to be cleaned up are not used any more. But the
    pointers of these resources may be changed in the function
    "crypto_finalize_skcipher_request". Thus release specific resources before
    calling this function.

    Fixes: dbaf0624ffa5 ("crypto: add virtio-crypto driver")
    Reported-by: LABBE Corentin <clabbe@baylibre.com>
    Cc: Gonglei <arei.gonglei@huawei.com>
    Cc: Herbert Xu <herbert@gondor.apana.org.au>
    Cc: "Michael S. Tsirkin" <mst@redhat.com>
    Cc: Jason Wang <jasowang@redhat.com>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: virtualization@lists.linux-foundation.org
    Cc: linux-kernel@vger.kernel.org
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200123101000.GB24255@Red
    Acked-by: Gonglei <arei.gonglei@huawei.com>
    Signed-off-by: Longpeng(Mike) <longpeng2@huawei.com>
    Link: https://lore.kernel.org/r/20200602070501.2023-3-longpeng2@huawei.com
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bae7e52681f081e9c20a42e8352a571c2e793b95
Author: Longpeng(Mike) <longpeng2@huawei.com>
Date:   Tue Jun 2 15:05:01 2020 +0800

    crypto: virtio: Fix dest length calculation in __virtio_crypto_skcipher_do_req()

    commit d90ca42012db2863a9a30b564a2ace6016594bda upstream.

    The src/dst length is not aligned with AES_BLOCK_SIZE(which is 16) in some
    testcases in tcrypto.ko.

    For example, the src/dst length of one of cts(cbc(aes))'s testcase is 17, the
    crypto_virtio driver will set @src_data_len=16 but @dst_data_len=17 in this
    case and get a wrong at then end.

      SRC: pp pp pp pp pp pp pp pp pp pp pp pp pp pp pp pp pp (17 bytes)
      EXP: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc pp (17 bytes)
      DST: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc 00 (pollute the last bytes)
      (pp: plaintext  cc:ciphertext)

    Fix this issue by limit the length of dest buffer.

    Fixes: dbaf0624ffa5 ("crypto: add virtio-crypto driver")
    Cc: Gonglei <arei.gonglei@huawei.com>
    Cc: Herbert Xu <herbert@gondor.apana.org.au>
    Cc: "Michael S. Tsirkin" <mst@redhat.com>
    Cc: Jason Wang <jasowang@redhat.com>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: virtualization@lists.linux-foundation.org
    Cc: linux-kernel@vger.kernel.org
    Cc: stable@vger.kernel.org
    Signed-off-by: Longpeng(Mike) <longpeng2@huawei.com>
    Link: https://lore.kernel.org/r/20200602070501.2023-4-longpeng2@huawei.com
    Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 29cebadac3157215b191f7b82a1a9ea766924cac
Author: Wei Yongjun <weiyongjun1@huawei.com>
Date:   Thu Apr 30 08:13:53 2020 +0000
```

```
    crypto: drbg - fix error return code in drbg_alloc_state()

    commit e0664ebcea6ac5e16da703409fb4bd61f8cd37d9 upstream.

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

    Reported-by: Xiumei Mu <xmu@redhat.com>
    Fixes: db07cd26ac6a ("crypto: drbg - add FIPS 140-2 CTRNG for noise source")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Wei Yongjun <weiyongjun1@huawei.com>
    Reviewed-by: Stephan Mueller <smueller@chronox.de>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5e5aef0f25cbf63c23bdbfcfec54ff569f3c414d
Author: Eric Biggers <ebiggers@google.com>
Date:   Mon Apr 6 23:02:40 2020 -0700

    crypto: algapi - Avoid spurious modprobe on LOADED

    commit beeb460cd12ac9b91640b484b6a52dcba9d9fc8f upstream.

    Currently after any algorithm is registered and tested, there's an
    unnecessary request_module("cryptomgr") even if it's already loaded.
    Also, CRYPTO_MSG_ALG_LOADED is sent twice, and thus if the algorithm is
    "crct10dif", lib/crc-t10dif.c replaces the tfm twice rather than once.

    This occurs because CRYPTO_MSG_ALG_LOADED is sent using
    crypto_probing_notify(), which tries to load "cryptomgr" if the
    notification is not handled (NOTIFY_DONE).  This doesn't make sense
    because "cryptomgr" doesn't handle this notification.

    Fix this by using crypto_notify() instead of crypto_probing_notify().

    Fixes: dd8b083f9a5e ("crypto: api - Introduce notifier for new crypto algorithms")
    Cc: <stable@vger.kernel.org> # v4.20+
    Cc: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Reviewed-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0c82e4f4f44a2f0b1b1e7b17f0f39de276d435c3
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Sat May 30 15:35:37 2020 +0200

    crypto: cavium/nitrox - Fix 'nitrox_get_first_device()' when ndevlist is fully iterated

    commit 320bdbd816156f9ca07e5fed7bfb449f2908dda7 upstream.

    When a list is completely iterated with 'list_for_each_entry(x, ...)', x is
    not NULL at the end.

    While at it, remove a useless initialization of the ndev variable. It
    is overridden by 'list_for_each_entry'.

    Fixes: f2663872f073 ("crypto: cavium - Register the CNN55XX supported crypto
algorithms.")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 8e45fdafdecc8436c5b6e1620c30726056e6b29c
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Wed May 27 18:29:34 2020 -0700

    gup: document and work around "COW can break either way" issue

    commit 17839856fd588f4ab6b789f482ed3ffd7c403e1f upstream.

    Doing a "get_user_pages()" on a copy-on-write page for reading can be
    ambiguous: the page can be COW'ed at any time afterwards, and the
    direction of a COW event isn't defined.

    Yes, whoever writes to it will generally do the COW, but if the thread
    that did the get_user_pages() unmapped the page before the write (and
    that could happen due to memory pressure in addition to any outright
    action), the writer could also just take over the old page instead.

    End result: the get_user_pages() call might result in a page pointer
    that is no longer associated with the original VM, and is associated
    with - and controlled by - another VM having taken it over instead.

    So when doing a get_user_pages() on a COW mapping, the only really safe
    thing to do would be to break the COW when getting the page, even when
    only getting it for reading.

    At the same time, some users simply don't even care.

    For example, the perf code wants to look up the page not because it
    cares about the page, but because the code simply wants to look up the
    physical address of the access for informational purposes, and doesn't
    really care about races when a page might be unmapped and remapped
    elsewhere.

    This adds logic to force a COW event by setting FOLL_WRITE on any
    copy-on-write mapping when FOLL_GET (or FOLL_PIN) is used to get a page
    pointer as a result.

    The current semantics end up being:

     - __get_user_pages_fast(): no change. If you don't ask for a write,
       you won't break COW. You'd better know what you're doing.

     - get_user_pages_fast(): the fast-case "look it up in the page tables
       without anything getting mmap_sem" now refuses to follow a read-only
       page, since it might need COW breaking.  Which happens in the slow
       path - the fast path doesn't know if the memory might be COW or not.

     - get_user_pages() (including the slow-path fallback for gup_fast()):
       for a COW mapping, turn on FOLL_WRITE for FOLL_GET/FOLL_PIN, with
       very similar semantics to FOLL_FORCE.

    If it turns out that we want finer granularity (ie "only break COW when
    it might actually matter" - things like the zero page are special and
    don't need to be broken) we might need to push these semantics deeper
    into the lookup fault path.  So if people care enough, it's possible
    that we might end up adding a new internal FOLL_BREAK_COW flag to go
    with the internal FOLL_COW flag we already have for tracking "I had a
    COW".

    Alternatively, if it turns out that different callers might want to
    explicitly control the forced COW break behavior, we might even want to
    make such a flag visible to the users of get_user_pages() instead of
    using the above default semantics.

    But for now, this is mostly commentary on the issue (this commit message
```

```
    being a lot bigger than the patch, and that patch in turn is almost all
    comments), with that minimal "enable COW breaking early" logic using the
    existing FOLL_WRITE behavior.

    [ It might be worth noting that we've always had this ambiguity, and it
      could arguably be seen as a user-space issue.

      You only get private COW mappings that could break either way in
      situations where user space is doing cooperative things (ie fork()
      before an execve() etc), but it _is_ surprising and very subtle, and
      fork() is supposed to give you independent address spaces.

      So let's treat this as a kernel issue and make the semantics of
      get_user_pages() easier to understand. Note that obviously a true
      shared mapping will still get a page that can change under us, so this
      does _not_ mean that get_user_pages() somehow returns any "stable"
      page ]

    Reported-by: Jann Horn <jannh@google.com>
    Tested-by: Christoph Hellwig <hch@lst.de>
    Acked-by: Oleg Nesterov <oleg@redhat.com>
    Acked-by: Kirill Shutemov <kirill@shutemov.name>
    Acked-by: Jan Kara <jack@suse.cz>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 593c0da9a7754caabc7316ab2b64d25c642e5a59
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Thu May 21 19:08:09 2020 +0200

    PM: runtime: clk: Fix clk_pm_runtime_get() error path

    commit 64c7d7ea22d86cacb65d0c097cc447bc0e6d8abd upstream.

    clk_pm_runtime_get() assumes that the PM-runtime usage counter will
    be dropped by pm_runtime_get_sync() on errors, which is not the case,
    so PM-runtime references to devices acquired by the former are leaked
    on errors returned by the latter.

    Fix this by modifying clk_pm_runtime_get() to drop the reference if
    pm_runtime_get_sync() returns an error.

    Fixes: 9a34b45397e5 clk: Add support for runtime PM
    Cc: 4.15+ <stable@vger.kernel.org> # 4.15+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Reviewed-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db018fa589c5054ac788b13f3cb4052d4e85fe4a
Author: Justin Chen <justinpopo6@gmail.com>
Date:   Mon Apr 20 15:08:49 2020 -0400

    spi: bcm-qspi: when tx/rx buffer is NULL set to 0

    commit 4df3bea7f9d2ddd9ac2c29ba945c7c4db2def29c upstream.

    Currently we set the tx/rx buffer to 0xff when NULL. This causes
    problems with some spi slaves where 0xff is a valid command. Looking
    at other drivers, the tx/rx buffer is usually set to 0x00 when NULL.
    Following this convention solves the issue.

    Fixes: fa236a7ef240 ("spi: bcm-qspi: Add Broadcom MSPI driver")
    Signed-off-by: Justin Chen <justinpopo6@gmail.com>
```

```
    Signed-off-by: Kamal Dasu <kdasu.kdev@gmail.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200420190853.45614-6-kdasu.kdev@gmail.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 90378a37388e9837a8c98d65c1425442f85fb9fc
Author: Florian Fainelli <f.fainelli@gmail.com>
Date:   Mon Apr 20 15:08:45 2020 -0400

    spi: bcm-qspi: Handle clock probe deferral

    commit 0392727c261bab65a35cd4f82ee9459bc237591d upstream.

    The clock provider may not be ready by the time spi-bcm-qspi gets
    probed, handle probe deferral using devm_clk_get_optional().

    Signed-off-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: Kamal Dasu <kdasu.kdev@gmail.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200420190853.45614-2-kdasu.kdev@gmail.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f38afdec6c6029ded84d975f5cb0345748f63414
Author: Lukas Wunner <lukas@wunner.de>
Date:   Fri May 15 17:58:03 2020 +0200

    spi: bcm2835aux: Fix controller unregister order

    commit b9dd3f6d417258ad0beeb292a1bc74200149f15d upstream.

    The BCM2835aux SPI driver uses devm_spi_register_master() on bind.
    As a consequence, on unbind, __device_release_driver() first invokes
    bcm2835aux_spi_remove() before unregistering the SPI controller via
    devres_release_all().

    This order is incorrect:  bcm2835aux_spi_remove() turns off the SPI
    controller, including its interrupts and clock.  The SPI controller
    is thus no longer usable.

    When the SPI controller is subsequently unregistered, it unbinds all
    its slave devices.  If their drivers need to access the SPI bus,
    e.g. to quiesce their interrupts, unbinding will fail.

    As a rule, devm_spi_register_master() must not be used if the
    ->remove() hook performs teardown steps which shall be performed
    after unbinding of slaves.

    Fix by using the non-devm variant spi_register_master().  Note that the
    struct spi_master as well as the driver-private data are not freed until
    after bcm2835aux_spi_remove() has finished, so accessing them is safe.

    Fixes: 1ea29b39f4c8 ("spi: bcm2835aux: add bcm2835 auxiliary spi device driver")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Cc: stable@vger.kernel.org # v4.4+
    Cc: Martin Sperl <kernel@martin.sperl.org>
    Link:
https://lore.kernel.org/r/32f27f4d8242e4d75f9a53f7e8f1f77483b08669.1589557526.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit abb29b74f376c39801f53b2c3251996b6ae97170
Author: Lukas Wunner <lukas@wunner.de>
```

```
Date:   Fri May 15 17:58:02 2020 +0200

    spi: bcm2835: Fix controller unregister order

    commit 9dd277ff92d06f6aa95b39936ad83981d781f49b upstream.

    The BCM2835 SPI driver uses devm_spi_register_controller() on bind.
    As a consequence, on unbind, __device_release_driver() first invokes
    bcm2835_spi_remove() before unregistering the SPI controller via
    devres_release_all().

    This order is incorrect:  bcm2835_spi_remove() tears down the DMA
    channels and turns off the SPI controller, including its interrupts
    and clock.  The SPI controller is thus no longer usable.

    When the SPI controller is subsequently unregistered, it unbinds all
    its slave devices.  If their drivers need to access the SPI bus,
    e.g. to quiesce their interrupts, unbinding will fail.

    As a rule, devm_spi_register_controller() must not be used if the
    ->remove() hook performs teardown steps which shall be performed
    after unbinding of slaves.

    Fix by using the non-devm variant spi_register_controller().  Note that
    the struct spi_controller as well as the driver-private data are not
    freed until after bcm2835_spi_remove() has finished, so accessing them
    is safe.

    Fixes: 247263dba208 ("spi: bcm2835: use devm_spi_register_master()")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Cc: stable@vger.kernel.org # v3.13+
    Link:
https://lore.kernel.org/r/2397dd70cdbe95e0bc4da2b9fca0f31cb94e5aed.1589557526.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3e75f0124102c6307a8a5bb0036d8ce10366cd7b
Author: Lukas Wunner <lukas@wunner.de>
Date:   Mon May 25 14:25:03 2020 +0200

    spi: pxa2xx: Fix runtime PM ref imbalance on probe error

    commit 65e318e17358a3fd4fcb5a69d89b14016dee2f06 upstream.

    The PXA2xx SPI driver releases a runtime PM ref in the probe error path
    even though it hasn't acquired a ref earlier.

    Apparently commit e2b714afee32 ("spi: pxa2xx: Disable runtime PM if
    controller registration fails") sought to copy-paste the invocation of
    pm_runtime_disable() from pxa2xx_spi_remove(), but erroneously copied
    the call to pm_runtime_put_noidle() as well.  Drop it.

    Fixes: e2b714afee32 ("spi: pxa2xx: Disable runtime PM if controller registration fails")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Reviewed-by: Jarkko Nikula <jarkko.nikula@linux.intel.com>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Cc: stable@vger.kernel.org # v4.17+
    Cc: Jarkko Nikula <jarkko.nikula@linux.intel.com>
    Link:
https://lore.kernel.org/r/58b2ac6942ca1f91aaeeafe512144bc5343e1d84.1590408496.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 67877ea1219479d87173034e11c327bd03fe4277
Author: Lukas Wunner <lukas@wunner.de>
Date:   Mon May 25 14:25:02 2020 +0200

    spi: pxa2xx: Fix controller unregister order

    commit 32e5b57232c0411e7dea96625c415510430ac079 upstream.

    The PXA2xx SPI driver uses devm_spi_register_controller() on bind.
    As a consequence, on unbind, __device_release_driver() first invokes
    pxa2xx_spi_remove() before unregistering the SPI controller via
    devres_release_all().

    This order is incorrect:  pxa2xx_spi_remove() disables the chip,
    rendering the SPI bus inaccessible even though the SPI controller is
    still registered.  When the SPI controller is subsequently unregistered,
    it unbinds all its slave devices.  Because their drivers cannot access
    the SPI bus, e.g. to quiesce interrupts, the slave devices may be left
    in an improper state.

    As a rule, devm_spi_register_controller() must not be used if the
    ->remove() hook performs teardown steps which shall be performed after
    unregistering the controller and specifically after unbinding of slaves.

    Fix by reverting to the non-devm variant of spi_register_controller().

    An alternative approach would be to use device-managed functions for all
    steps in pxa2xx_spi_remove(), e.g. by calling devm_add_action_or_reset()
    on probe.  However that approach would add more LoC to the driver and
    it wouldn't lend itself as well to backporting to stable.

    The improper use of devm_spi_register_controller() was introduced in 2013
    by commit a807fcd090d6 ("spi: pxa2xx: use devm_spi_register_master()"),
    but all earlier versions of the driver going back to 2006 were likewise
    broken because they invoked spi_unregister_master() at the end of
    pxa2xx_spi_remove(), rather than at the beginning.

    Fixes: e0c9905e87ac ("[PATCH] SPI: add PXA2xx SSP SPI Driver")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Cc: stable@vger.kernel.org # v2.6.17+
    Cc: Tsuchiya Yuto <kitakar@gmail.com>
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206403#c1
    Link:
https://lore.kernel.org/r/834c446b1cf3284d2660f1bee1ebe3e737cd02a9.1590408496.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2d83862dca3e9f5c119740c2f8d4433b6f337683
Author: Lukas Wunner <lukas@wunner.de>
Date:   Fri May 15 17:58:01 2020 +0200

    spi: Fix controller unregister order

    commit 84855678add8aba927faf76bc2f130a40f94b6f7 upstream.

    When an SPI controller unregisters, it unbinds all its slave devices.
    For this, their drivers may need to access the SPI bus, e.g. to quiesce
    interrupts.

    However since commit ffbbdd21329f ("spi: create a message queueing
    infrastructure"), spi_destroy_queue() is executed before unbinding the
    slaves.  It sets ctlr->running = false, thereby preventing SPI bus
    access and causing unbinding of slave devices to fail.
```

```
    Fix by unbinding slaves before calling spi_destroy_queue().

    Fixes: ffbbdd21329f ("spi: create a message queueing infrastructure")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Cc: stable@vger.kernel.org # v3.4+
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Link:
https://lore.kernel.org/r/8aaf9d44c153fe233b17bc2dec4eb679898d7e7b.1589557526.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dda74862a7e422305919109df09f9d1dd5171b2c
Author: Lukas Wunner <lukas@wunner.de>
Date:   Mon May 25 14:25:01 2020 +0200

    spi: dw: Fix controller unregister order

    commit ca8b19d61e3fce5d2d7790cde27a0b57bcb3f341 upstream.

    The Designware SPI driver uses devm_spi_register_controller() on bind.
    As a consequence, on unbind, __device_release_driver() first invokes
    dw_spi_remove_host() before unregistering the SPI controller via
    devres_release_all().

    This order is incorrect:  dw_spi_remove_host() shuts down the chip,
    rendering the SPI bus inaccessible even though the SPI controller is
    still registered.  When the SPI controller is subsequently unregistered,
    it unbinds all its slave devices.  Because their drivers cannot access
    the SPI bus, e.g. to quiesce interrupts, the slave devices may be left
    in an improper state.

    As a rule, devm_spi_register_controller() must not be used if the
    ->remove() hook performs teardown steps which shall be performed after
    unregistering the controller and specifically after unbinding of slaves.

    Fix by reverting to the non-devm variant of spi_register_controller().

    An alternative approach would be to use device-managed functions for all
    steps in dw_spi_remove_host(), e.g. by calling devm_add_action_or_reset()
    on probe.  However that approach would add more LoC to the driver and
    it wouldn't lend itself as well to backporting to stable.

    Fixes: 04f421e7b0b1 ("spi: dw: use managed resources")
    Signed-off-by: Lukas Wunner <lukas@wunner.de>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Cc: stable@vger.kernel.org # v3.14+
    Cc: Baruch Siach <baruch@tkos.co.il>
    Link:
https://lore.kernel.org/r/3fff8cb8ae44a9893840d0688be15bb88c090a14.1590408496.git.lukas@wunn
er.de
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2fe9522054a3707a75a2d9c4df816ff81eddac4b
Author: Alexander Gordeev <agordeev@linux.ibm.com>
Date:   Wed Jun 10 18:41:41 2020 -0700

    lib: fix bitmap_parse() on 64-bit big endian archs

    commit 81c4f4d924d5d009b5ed785a3e22b18d0f7b831f upstream.

    Commit 2d6261583be0 ("lib: rework bitmap_parse()") does not take into
    account order of halfwords on 64-bit big endian architectures.  As
```

```
    result (at least) Receive Packet Steering, IRQ affinity masks and
    runtime kernel test "test_bitmap" get broken on s390.

    [andriy.shevchenko@linux.intel.com: convert infinite while loop to a for loop]
      Link: http://lkml.kernel.org/r/20200609140535.87160-1-
andriy.shevchenko@linux.intel.com

    Fixes: 2d6261583be0 ("lib: rework bitmap_parse()")
    Signed-off-by: Alexander Gordeev <agordeev@linux.ibm.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Reviewed-by: Andy Shevchenko <andy.shevchenko@gmail.com>
    Cc: Yury Norov <yury.norov@gmail.com>
    Cc: Amritha Nambiar <amritha.nambiar@intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: Kees Cook <keescook@chromium.org>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Miklos Szeredi <mszeredi@redhat.com>
    Cc: Rasmus Villemoes <linux@rasmusvillemoes.dk>
    Cc: Steffen Klassert <steffen.klassert@secunet.com>
    Cc: "Tobin C . Harding" <tobin@kernel.org>
    Cc: Vineet Gupta <vineet.gupta1@synopsys.com>
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: Willem de Bruijn <willemb@google.com>
    Cc: <stable@vger.kernel.org>
    Link: http://lkml.kernel.org/r/1591634471-17647-1-git-send-email-agordeev@linux.ibm.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 341b4138bb3819fda80e8b2b2007c7ff1bfc37ae
Author: Ryusuke Konishi <konishi.ryusuke@gmail.com>
Date:   Wed Jun 10 18:41:35 2020 -0700

    nilfs2: fix null pointer dereference at nilfs_segctor_do_construct()

    commit 8301c719a2bd131436438e49130ee381d30933f5 upstream.

    After commit c3aab9a0bd91 ("mm/filemap.c: don't initiate writeback if
    mapping has no dirty pages"), the following null pointer dereference has
    been reported on nilfs2:

      BUG: kernel NULL pointer dereference, address: 00000000000000a8
      #PF: supervisor read access in kernel mode
      #PF: error_code(0x0000) - not-present page
      PGD 0 P4D 0
      Oops: 0000 [#1] SMP PTI
      ...
      RIP: 0010:percpu_counter_add_batch+0xa/0x60
      ...
      Call Trace:
        __test_set_page_writeback+0x2d3/0x330
        nilfs_segctor_do_construct+0x10d3/0x2110 [nilfs2]
        nilfs_segctor_construct+0x168/0x260 [nilfs2]
        nilfs_segctor_thread+0x127/0x3b0 [nilfs2]
        kthread+0xf8/0x130
        ...

    This crash turned out to be caused by set_page_writeback() call for
    segment summary buffers at nilfs_segctor_prepare_write().

    set_page_writeback() can call inc_wb_stat(inode_to_wb(inode),
    WB_WRITEBACK) where inode_to_wb(inode) is NULL if the inode of
    underlying block device does not have an associated wb.
```

```
    This fixes the issue by calling inode_attach_wb() in advance to ensure
    to associate the bdev inode with its wb.

    Fixes: c3aab9a0bd91 ("mm/filemap.c: don't initiate writeback if mapping has no dirty
pages")
    Reported-by: Walton Hoops <me@waltonhoops.com>
    Reported-by: Tomas Hlavaty <tom@logand.com>
    Reported-by: ARAI Shun-ichi <hermes@ceres.dti.ne.jp>
    Reported-by: Hideki EIRAKU <hdk1983@gmail.com>
    Signed-off-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Tested-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
    Cc: <stable@vger.kernel.org>    [5.4+]
    Link:
http://lkml.kernel.org/r/20200608.011819.1399059588922299158.konishi.ryusuke@gmail.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2cc48ffeccf419cd2ac50743e1f2966203958d94
Author: Dave Rodgman <dave.rodgman@arm.com>
Date:   Thu Jun 11 17:34:54 2020 -0700

    lib/lzo: fix ambiguous encoding bug in lzo-rle

    commit b5265c813ce4efbfa2e46fd27cdf9a7f44a35d2e upstream.

    In some rare cases, for input data over 32 KB, lzo-rle could encode two
    different inputs to the same compressed representation, so that
    decompression is then ambiguous (i.e.  data may be corrupted - although
    zram is not affected because it operates over 4 KB pages).

    This modifies the compressor without changing the decompressor or the
    bitstream format, such that:

     - there is no change to how data produced by the old compressor is
       decompressed

     - an old decompressor will correctly decode data from the updated
       compressor

     - performance and compression ratio are not affected

     - we avoid introducing a new bitstream format

    In testing over 12.8M real-world files totalling 903 GB, three files
    were affected by this bug.  I also constructed 37M semi-random 64 KB
    files totalling 2.27 TB, and saw no affected files.  Finally I tested
    over files constructed to contain each of the ~1024 possible bad input
    sequences; for all of these cases, updated lzo-rle worked correctly.

    There is no significant impact to performance or compression ratio.

    Signed-off-by: Dave Rodgman <dave.rodgman@arm.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Dave Rodgman <dave.rodgman@arm.com>
    Cc: Willy Tarreau <w@1wt.eu>
    Cc: Sergey Senozhatsky <sergey.senozhatsky.work@gmail.com>
    Cc: Markus F.X.J. Oberhumer <markus@oberhumer.com>
    Cc: Minchan Kim <minchan@kernel.org>
    Cc: Nitin Gupta <ngupta@vflare.org>
    Cc: Chao Yu <yuchao0@huawei.com>
    Cc: <stable@vger.kernel.org>
    Link: http://lkml.kernel.org/r/20200507100203.29785-1-dave.rodgman@arm.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
```

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

commit e5a9a3732e0c367431f8d3a831036edf5dfdd2e8
Author: Nick Desaulniers <ndesaulniers@google.com>
Date:   Mon Jun 8 13:38:17 2020 -0700

    arm64: acpi: fix UBSAN warning

    commit a194c33f45f83068ef13bf1d16e26d4ca3ecc098 upstream.

    Will reported a UBSAN warning:

    UBSAN: null-ptr-deref in arch/arm64/kernel/smp.c:596:6
    member access within null pointer of type 'struct acpi_madt_generic_interrupt'
    CPU: 0 PID: 0 Comm: swapper Not tainted 5.7.0-rc6-00124-g96bc42ff0a82 #1
    Call trace:
     dump_backtrace+0x0/0x384
     show_stack+0x28/0x38
     dump_stack+0xec/0x174
     handle_null_ptr_deref+0x134/0x174
     __ubsan_handle_type_mismatch_v1+0x84/0xa4
     acpi_parse_gic_cpu_interface+0x60/0xe8
     acpi_parse_entries_array+0x288/0x498
     acpi_table_parse_entries_array+0x178/0x1b4
     acpi_table_parse_madt+0xa4/0x110
     acpi_parse_and_init_cpus+0x38/0x100
     smp_init_cpus+0x74/0x258
     setup_arch+0x350/0x3ec
     start_kernel+0x98/0x6f4

    This is from the use of the ACPI_OFFSET in
    arch/arm64/include/asm/acpi.h. Replace its use with offsetof from
    include/linux/stddef.h which should implement the same logic using
    __builtin_offsetof, so that UBSAN wont warn.

    Reported-by: Will Deacon <will@kernel.org>
    Suggested-by: Ard Biesheuvel <ardb@kernel.org>
    Signed-off-by: Nick Desaulniers <ndesaulniers@google.com>
    Reviewed-by: Jeremy Linton <jeremy.linton@arm.com>
    Acked-by: Lorenzo Pieralisi <lorenzo.pieralisi@arm.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/lkml/20200521100952.GA5360@willie-the-truck/
    Link: https://lore.kernel.org/r/20200608203818.189423-1-ndesaulniers@google.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7732e041f0d0b710afeaf41caec13d093200d9e1
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Thu Jun 4 19:22:26 2020 +0200

    ACPI: PM: Avoid using power resources if there are none for D0

    commit 956ad9d98b73f59e442cc119c98ba1e04e94fe6d upstream.

    As recently reported, some platforms provide a list of power
    resources for device power state D3hot, through the _PR3 object,
    but they do not provide a list of power resources for device power
    state D0.

    Among other things, this causes acpi_device_get_power() to return
    D3hot as the current state of the device in question if all of the
    D3hot power resources are "on", because it sees the power_resources
    flag set and calls acpi_power_get_inferred_state() which finds that
    D3hot is the shallowest power state with all of the associated power
    resources turned "on", so that's what it returns.  Moreover, that
```

```
    value takes precedence over the acpi_dev_pm_explicit_get() return
    value, because it means a deeper power state.  The device may very
    well be in D0 physically at that point, however.

    Moreover, the presence of _PR3 without _PR0 for a given device
    means that only one D3-level power state can be supported by it.
    Namely, because there are no power resources to turn "off" when
    transitioning the device from D0 into D3cold (which should be
    supported since _PR3 is present), the evaluation of _PS3 should
    be sufficient to put it straight into D3cold, but this means that
    the effect of turning "on" the _PR3 power resources is unclear,
    so it is better to avoid doing that altogether.  Consequently,
    there is no practical way do distinguish D3cold from D3hot for
    the device in question and the power states of it can be labeled
    so that D3hot is the deepest supported one (and Linux assumes
    that putting a device into D3hot via ACPI may cause power to be
    removed from it anyway, for legacy reasons).

    To work around the problem described above modify the ACPI
    enumeration of devices so that power resources are only used
    for device power management if the list of D0 power resources
    is not empty and make it mart D3cold as supported only if that
    is the case and the D3hot list of power resources is not empty
    too.

    Fixes: ef85bdbec444 ("ACPI / scan: Consolidate extraction of power resources lists")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=205057
    Link: https://lore.kernel.org/linux-acpi/20200603194659.185757-1-hdegoede@redhat.com/
    Reported-by: Hans de Goede <hdegoede@redhat.com>
    Tested-by: Hans de Goede <hdegoede@redhat.com>
    Tested-by: youling257@gmail.com
    Cc: 3.10+ <stable@vger.kernel.org> # 3.10+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9b50dc00f0f58dd729ac212040f2ecdd68291add
Author: Ard Biesheuvel <ardb@kernel.org>
Date:   Fri May 15 11:36:13 2020 +0200

    ACPI: GED: add support for _Exx / _Lxx handler methods

    commit ea6f3af4c5e63f6981c0b0ab8ebec438e2d5ef40 upstream.

    Per the ACPI spec, interrupts in the range [0, 255] may be handled
    in AML using individual methods whose naming is based on the format
    _Exx or _Lxx, where xx is the hex representation of the interrupt
    index.

    Add support for this missing feature to our ACPI GED driver.

    Cc: v4.9+ <stable@vger.kernel.org> # v4.9+
    Signed-off-by: Ard Biesheuvel <ardb@kernel.org>
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 977fd29853fc7ec3b8b00e410790a1f8d0d08488
Author: Qiushi Wu <wu000273@umn.edu>
Date:   Wed May 27 17:35:51 2020 -0500

    ACPI: CPPC: Fix reference count leak in acpi_cppc_processor_probe()

    commit 4d8be4bc94f74bb7d096e1c2e44457b530d5a170 upstream.

    kobject_init_and_add() takes reference even when it fails.
```

```
    If this function returns an error, kobject_put() must be called to
    properly clean up the memory associated with the object. Previous
    commit "b8eb718348b8" fixed a similar problem.

    Fixes: 158c998ea44b ("ACPI / CPPC: add sysfs support to compute delivered performance")
    Signed-off-by: Qiushi Wu <wu000273@umn.edu>
    Cc: 4.10+ <stable@vger.kernel.org> # 4.10+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4e29f7e2e4b19143cd7f01b53bc3037df1465ab8
Author: Qiushi Wu <wu000273@umn.edu>
Date:   Wed May 27 16:17:17 2020 -0500

    ACPI: sysfs: Fix reference count leak in acpi_sysfs_add_hotplug_profile()

    commit 6e6c25283dff866308c87b49434c7dbad4774cc0 upstream.

    kobject_init_and_add() takes reference even when it fails.
    Thus, when kobject_init_and_add() returns an error,
    kobject_put() must be called to properly clean up the kobject.

    Fixes: 3f8055c35836 ("ACPI / hotplug: Introduce user space interface for hotplug
profiles")
    Signed-off-by: Qiushi Wu <wu000273@umn.edu>
    Cc: 3.10+ <stable@vger.kernel.org> # 3.10+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6e929230c81070f246af5dba5f3133e46dfa662a
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Mon Jun 8 14:26:28 2020 +0800

    ALSA: usb-audio: Add vendor, product and profile name for HP Thunderbolt Dock

    commit 0c5086f5699906ec8e31ea6509239489f060f2dc upstream.

    The HP Thunderbolt Dock has two separate USB devices, one is for speaker
    and one is for headset. Add names for them so userspace can apply UCM
    settings.

    Signed-off-by: Kai-Heng Feng <kai.heng.feng@canonical.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200608062630.10806-1-kai.heng.feng@canonical.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1220ac4e35490b300953bf4b58046d046131511e
Author: Takashi Iwai <tiwai@suse.de>
Date:   Wed Jun 3 17:37:08 2020 +0200

    ALSA: usb-audio: Fix inconsistent card PM state after resume

    commit 862b2509d157c629dd26d7ac6c6cdbf043d332eb upstream.

    When a USB-audio interface gets runtime-suspended via auto-pm feature,
    the driver suspends all functionality and increment
    chip->num_suspended_intf.  Later on, when the system gets suspended to
    S3, the driver increments chip->num_suspended_intf again, skips the
    device changes, and sets the card power state to
    SNDRV_CTL_POWER_D3hot.  In return, when the system gets resumed from
    S3, the resume callback decrements chip->num_suspended_intf.  Since
    this refcount is still not zero (it's been runtime-suspended), the
    whole resume is skipped.  But there is a small pitfall here.
```

```
    The problem is that the driver doesn't restore the card power state
    after this resume call, leaving it as SNDRV_CTL_POWER_D3hot.  So,
    even after the system resume finishes, the card instance still appears
    as if it were system-suspended, and this confuses many ioctl accesses
    that are blocked unexpectedly.

    In details, we have two issues behind the scene: one is that the card
    power state is changed only when the refcount becomes zero, and
    another is that the prior auto-suspend check is kept in a boolean
    flag.  Although the latter problem is almost negligible since the
    auto-pm feature is imposed only on the primary interface, but this can
    be a potential problem on the devices with multiple interfaces.

    This patch addresses those issues by the following:

    - Replace chip->autosuspended boolean flag with chip->system_suspend
      counter

    - At the first system-suspend, chip->num_suspended_intf is recorded to
      chip->system_suspend

    - At system-resume, the card power state is restored when the
      chip->num_suspended_intf refcount reaches to chip->system_suspend,
      i.e. the state returns to the auto-suspended

    Also, the patch fixes yet another hidden problem by the code
    refactoring along with the fixes above: namely, when some resume
    procedure failed, the driver left chip->num_suspended_intf that was
    already decreased, and it might lead to the refcount unbalance.
    In the new code, the refcount decrement is done after the whole resume
    procedure, and the problem is avoided as well.

    Fixes: 0662292aec05 ("ALSA: usb-audio: Handle normal and auto-suspend equally")
    Reported-and-tested-by: Macpaul Lin <macpaul.lin@mediatek.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200603153709.6293-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4f55a74b0e958d9ee10530eb239834334f2be688
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Mon Jun 8 12:06:32 2020 +0200

    ALSA: pcm: fix snd_pcm_link() lockdep splat

    commit e18035cf5cb3d2bf8e4f4d350a23608bd208b934 upstream.

    Add and use snd_pcm_stream_lock_nested() in snd_pcm_link/unlink
    implementation.  The code is fine, but generates a lockdep complaint:

    ============================================
    WARNING: possible recursive locking detected
    5.7.1mq+ #381 Tainted: G           O
    --------------------------------------------
    pulseaudio/4180 is trying to acquire lock:
    ffff888402d6f508 (&group->lock){-...}-{2:2}, at: snd_pcm_common_ioctl+0xda8/0xee0
[snd_pcm]

    but task is already holding lock:
    ffff8883f7a8cf18 (&group->lock){-...}-{2:2}, at: snd_pcm_common_ioctl+0xe4e/0xee0
[snd_pcm]

    other info that might help us debug this:
     Possible unsafe locking scenario:
```

```
            CPU0
            ----
       lock(&group->lock);
       lock(&group->lock);

      *** DEADLOCK ***

      May be due to missing lock nesting notation

     2 locks held by pulseaudio/4180:
      #0: ffffffffa1a05190 (snd_pcm_link_rwsem){++++}-{3:3}, at:
 snd_pcm_common_ioctl+0xca0/0xee0 [snd_pcm]
      #1: ffff8883f7a8cf18 (&group->lock){-...}-{2:2}, at: snd_pcm_common_ioctl+0xe4e/0xee0
 [snd_pcm]
     [...]

     Cc: stable@vger.kernel.org
     Fixes: f57f3df03a8e ("ALSA: pcm: More fine-grained PCM link locking")
     Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
     Link:
 https://lore.kernel.org/r/37252c65941e58473b1219ca9fab03d48f47e3e3.1591610330.git.mirq-
 linux@rere.qmqm.pl
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

     Signed-off-by: Takashi Iwai <tiwai@suse.de>

 commit b75524c5a8b5dfcd508dfdc7a070f714e7928614
 Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
 Date:   Mon Jun 8 18:50:39 2020 +0200

     ALSA: pcm: disallow linking stream to itself

     commit 951e2736f4b11b58dc44d41964fa17c3527d882a upstream.

     Prevent SNDRV_PCM_IOCTL_LINK linking stream to itself - the code
     can't handle it. Fixed commit is not where bug was introduced, but
     changes the context significantly.

     Cc: stable@vger.kernel.org
     Fixes: 0888c321de70 ("pcm_native: switch to fdget()/fdput()")
     Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
     Link:
 https://lore.kernel.org/r/89c4a2487609a0ed6af3ecf01cc972bdc59a7a2d.1591634956.git.mirq-
 linux@rere.qmqm.pl
     Signed-off-by: Takashi Iwai <tiwai@suse.de>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 8521f0e5e5120b0e8457f89f46aae6c48d2838ff
 Author: Hui Wang <hui.wang@canonical.com>
 Date:   Mon Jun 8 19:55:41 2020 +0800

     ALSA: hda/realtek - add a pintbl quirk for several Lenovo machines

     commit 573fcbfd319ccef26caa3700320242accea7fd5c upstream.

     A couple of Lenovo ThinkCentre machines all have 2 front mics and they
     use the same codec alc623 and have the same pin config, so add a
     pintbl entry for those machines to apply the fixup
     ALC283_FIXUP_HEADSET_MIC.

     Cc: <stable@vger.kernel.org>
     Signed-off-by: Hui Wang <hui.wang@canonical.com>
     Link: https://lore.kernel.org/r/20200608115541.9531-1-hui.wang@canonical.com
     Signed-off-by: Takashi Iwai <tiwai@suse.de>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit e451e125847520b8df2c1a574ce2af5a597afcb0
Author: Takashi Sakamoto <o-takashi@sakamocchi.jp>
Date:   Sun May 10 16:42:57 2020 +0900

    ALSA: fireface: start IR context immediately

    commit f4588cc425beb62e355bc2a5de5d5c83e26a74ca upstream.

    In the latter models of RME Fireface series, device start to transfer
    packets several dozens of milliseconds. On the other hand, ALSA fireface
    driver starts IR context 2 milliseconds after the start. This results
    in loss to handle incoming packets on the context.

    This commit changes to start IR context immediately instead of
    postponement. For Fireface 800, this affects nothing because the device
    transfer packets 100 milliseconds or so after the start and this is
    within wait timeout.

    Cc: <stable@vger.kernel.org>
    Fixes: acfedcbe1ce4 ("ALSA: firewire-lib: postpone to start IR context")
    Signed-off-by: Takashi Sakamoto <o-takashi@sakamocchi.jp>
    Link: https://lore.kernel.org/r/20200510074301.116224-3-o-takashi@sakamocchi.jp
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit da185f35470c3c74a51d72aad2e7b5e1adbb389e
Author: Takashi Sakamoto <o-takashi@sakamocchi.jp>
Date:   Sun May 10 16:42:56 2020 +0900

    ALSA: fireface: fix configuration error for nominal sampling transfer frequency

    commit bbd6aac3ae15bef762af03bf62e35ace5c4292bd upstream.

    128000 and 192000 are congruence modulo 32000, thus it's wrong to
    distinguish them as multiple of 32000 and 48000 by modulo 32000 at
    first.

    Additionally, used condition statement to detect quadruple speed can
    cause missing bit flag.

    Furthermore, counter to ensure the configuration is wrong and it
    causes false positive.

    This commit fixes the above three bugs.

    Cc: <stable@vger.kernel.org>
    Fixes: 60aec494b389 ("ALSA: fireface: support allocate_resources operation in latter
protocol")
    Signed-off-by: Takashi Sakamoto <o-takashi@sakamocchi.jp>
    Link: https://lore.kernel.org/r/20200510074301.116224-2-o-takashi@sakamocchi.jp
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 30e7f8d501c421fba249e1d7bb4c176a5ae47b9b
Author: Hersen Wu <hersenxs.wu@amd.com>
Date:   Tue Jun 2 21:31:37 2020 -0400

    ALSA: hda: add sienna_cichlid audio asic id for sienna_cichlid up

    commit 27a7c67012cfa6d79f87fbb51afa13c6c0e24e34 upstream.

    dp/hdmi ati hda is not shown in audio settings

    [ rearranged to a more appropriate place per device number order
```

```
       -- tiwai ]

    Signed-off-by: Hersen Wu <hersenxs.wu@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200603013137.1849404-1-alexander.deucher@amd.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 41b87ff6dda3b3a4a4c75ce50d42888dd6f1225b
Author: Chuhong Yuan <hslester96@gmail.com>
Date:   Wed Jun 3 17:24:59 2020 +0800

    ALSA: es1688: Add the missed snd_card_free()

    commit d9b8fbf15d05350b36081eddafcf7b15aa1add50 upstream.

    snd_es968_pnp_detect() misses a snd_card_free() in a failed path.
    Add the missed function call to fix it.

    Fixes: a20971b201ac ("ALSA: Merge es1688 and es968 drivers")
    Signed-off-by: Chuhong Yuan <hslester96@gmail.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200603092459.1424093-1-hslester96@gmail.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bb8ab2343740a70d2982929e9367b6dfe411ac0c
Author: Fabio Estevam <festevam@gmail.com>
Date:   Sun Apr 12 20:01:22 2020 -0300

    watchdog: imx_sc_wdt: Fix reboot on crash

    commit e56d48e92b1017b6a8dbe64923a889283733fd96 upstream.

    Currently when running the samples/watchdog/watchdog-simple.c
    application and forcing a kernel crash by doing:

    # ./watchdog-simple &
    # echo c > /proc/sysrq-trigger

    The system does not reboot as expected.

    Fix it by calling imx_sc_wdt_set_timeout() to configure the i.MX8QXP
    watchdog with a proper timeout.

    Cc: <stable@vger.kernel.org>
    Fixes: 986857acbc9a ("watchdog: imx_sc: Add i.MX system controller watchdog support")
    Reported-by: Breno Lima <breno.lima@nxp.com>
    Signed-off-by: Fabio Estevam <festevam@gmail.com>
    Reviewed-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Breno Lima <breno.lima@nxp.com>
    Link: https://lore.kernel.org/r/20200412230122.5601-1-festevam@gmail.com
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Wim Van Sebroeck <wim@linux-watchdog.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit acba9bd0d0a9757f148e72f84d38d4cd3b7e390d
Author: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
Date:   Mon May 25 13:59:52 2020 +0300

    serial: imx: Initialize lock for non-registered console

    commit 8f065acec7573672dd15916e31d1e9b2e785566c upstream.
```

```
    The commit a3cb39d258ef
    ("serial: core: Allow detach and attach serial device for console")
    changed a bit logic behind lock initialization since for most of the console
    driver it's supposed to have lock already initialized even if console is not
    enabled. However, it's not the case for Freescale IMX console.

    Initialize lock explicitly in the ->probe().

    Note, there is still an open question should or shouldn't not this driver
    register console properly.

    Fixes: a3cb39d258ef ("serial: core: Allow detach and attach serial device for console")
    Reported-by: Guenter Roeck <linux@roeck-us.net>
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Link: https://lore.kernel.org/r/20200525105952.13744-1-andriy.shevchenko@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5acd7d78daa790671e9437414921b867d04504c7
Author: Steve French <stfrench@microsoft.com>
Date:   Tue Jun 9 19:50:40 2020 -0500

    smb3: fix typo in mount options displayed in /proc/mounts

    commit 7866c177a03b18be3d83175014c643546e5b53c6 upstream.

    Missing the final 's' in "max_channels" mount option when displayed in
    /proc/mounts (or by mount command)

    CC: Stable <stable@vger.kernel.org>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Reviewed-by: Shyam Prasad N <nspmangalore@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f20d4ab17fe99e0d537febac66bc129e6002d5b9
Author: Namjae Jeon <namjae.jeon@samsung.com>
Date:   Thu Jun 11 11:21:19 2020 +0900

    smb3: add indatalen that can be a non-zero value to calculation of credit charge in smb2
ioctl

    commit ebf57440ec59a36e1fc5fe91e31d66ae0d1662d0 upstream.

    Some of tests in xfstests failed with cifsd kernel server since commit
    e80ddeb2f70e. cifsd kernel server validates credit charge from client
    by calculating it base on max((InputCount + OutputCount) and
    (MaxInputResponse + MaxOutputResponse)) according to specification.

    MS-SMB2 specification describe credit charge calculation of smb2 ioctl :

    If Connection.SupportsMultiCredit is TRUE, the server MUST validate
    CreditCharge based on the maximum of (InputCount + OutputCount) and
    (MaxInputResponse + MaxOutputResponse), as specified in section 3.3.5.2.5.
    If the validation fails, it MUST fail the IOCTL request with
    STATUS_INVALID_PARAMETER.

    This patch add indatalen that can be a non-zero value to calculation of
    credit charge in SMB2_ioctl_init().

    Fixes: e80ddeb2f70e ("smb3: fix incorrect number of credits when ioctl
    MaxOutputResponse > 64K")
    Cc: Stable <stable@vger.kernel.org>
    Reviewed-by: Aurelien Aptel <aaptel@suse.com>
    Cc: Steve French <smfrench@gmail.com>
```

```
    Signed-off-by: Namjae Jeon <namjae.jeon@samsung.com>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 297d265e1073463d2e8de8021ad3083b49048f58
Author: Steve French <stfrench@microsoft.com>
Date:   Wed Jun 3 01:33:58 2020 -0500

    smb3: fix incorrect number of credits when ioctl MaxOutputResponse > 64K

    commit e80ddeb2f70ebd0786aa7cdba3e58bc931fa0bb5 upstream.

    We were not checking to see if ioctl requests asked for more than
    64K (ie when CIFSMaxBufSize was > 64K) so when setting larger
    CIFSMaxBufSize then ioctls would fail with invalid parameter errors.
    When requests ask for more than 64K in MaxOutputResponse then we
    need to ask for more than 1 credit.

    Signed-off-by: Steve French <stfrench@microsoft.com>
    CC: Stable <stable@vger.kernel.org>
    Reviewed-by: Aurelien Aptel <aaptel@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9a93bccfad9f7cf378e81f26a453b156d9ee7a82
Author: Ard Biesheuvel <ardb@kernel.org>
Date:   Fri May 22 18:15:49 2020 +0200

    efi/efivars: Add missing kobject_put() in sysfs entry creation error path

    commit d8bd8c6e2cfab8b78b537715255be8d7557791c0 upstream.

    The documentation provided by kobject_init_and_add() clearly spells out
    the need to call kobject_put() on the kobject if an error is returned.
    Add this missing call to the error path.

    Cc: <stable@vger.kernel.org>
    Reported-by: 亿一 <teroincn@gmail.com>
    Signed-off-by: Ard Biesheuvel <ardb@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5f8f15c983865ccea52702e91aab0a4bd3979722
Author: Jens Axboe <axboe@kernel.dk>
Date:   Tue Jun 9 19:23:05 2020 -0600

    io_uring: allow O_NONBLOCK async retry

    commit c5b856255cbc3b664d686a83fa9397a835e063de upstream.

    We can assume that O_NONBLOCK is always honored, even if we don't
    have a ->read/write_iter() for the file type. Also unify the read/write
    checking for allowing async punt, having the write side factoring in the
    REQ_F_NOWAIT flag as well.

    Cc: stable@vger.kernel.org
    Fixes: 490e89676a52 ("io_uring: only force async punt if poll based retry can't handle
it")
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f049c6245141990f6719661ea4374cfc01f235ad
Author: Denis Efremov <efremov@linux.com>
Date:   Fri Jun 5 12:32:03 2020 +0300

    io_uring: use kvfree() in io_sqe_buffer_register()
```

```
    commit a8c73c1a614f6da6c0b04c393f87447e28cb6de4 upstream.

    Use kvfree() to free the pages and vmas, since they are allocated by
    kvmalloc_array() in a loop.

    Fixes: d4ef647510b1 ("io_uring: avoid page allocation warnings")
    Signed-off-by: Denis Efremov <efremov@linux.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200605093203.40087-1-efremov@linux.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit df32293edeb582cd9ad975e9ab49059b43e1e7cf
Author: Jens Axboe <axboe@kernel.dk>
Date:   Thu Jun 4 11:27:01 2020 -0600

    io_uring: re-set iov base/len for buffer select retry

    commit dddb3e26f6d88c5344d28cb5ff9d3d6fa05c4f7a upstream.

    We already have the buffer selected, but we should set the iter list
    again.

    Cc: stable@vger.kernel.org # v5.7
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8cdf8e270d7c0bde149cfcf037664b662f00b3eb
Author: Pavel Begunkov <asml.silence@gmail.com>
Date:   Tue May 26 20:34:02 2020 +0300

    io_uring: fix flush req->refs underflow

    commit 4518a3cc273cf82efdd36522fb1f13baad173c70 upstream.

    In io_uring_cancel_files(), after refcount_sub_and_test() leaves 0
    req->refs, it calls io_put_req(), which would also put a ref. Call
    io_free_req() instead.

    Cc: stable@vger.kernel.org
    Fixes: 2ca10259b418 ("io_uring: prune request from overflow list on flush")
    Signed-off-by: Pavel Begunkov <asml.silence@gmail.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f4147ddd46cbc276adea7134e8a23053e31a341a
Author: Pavel Dobias <dobias@2n.cz>
Date:   Fri May 15 14:07:57 2020 +0200

    ASoC: max9867: fix volume controls

    commit 8ba4dc3cff8cbe2c571063a5fd7116e8bde563ca upstream.

    The xmax values for Master Playback Volume and Mic Boost
    Capture Volume are specified incorrectly (one greater)
    which results in the wrong dB gain being shown to the user
    in the case of Master Playback Volume.

    Signed-off-by: Pavel Dobias <dobias@2n.cz>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200515120757.24669-1-dobias@2n.cz
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5f6b487bdb90abb7a5dd8dee02a8d04eca28bc82
```

```
Author: Dan Murphy <dmurphy@ti.com>
Date:   Mon Apr 27 15:36:08 2020 -0500

    ASoC: tlv320adcx140: Fix mic gain registers

    commit be8499c48f115b912f5747c420f66a5e2c31defe upstream.

    Fix the mic gain registers for channels 2-4.
    The incorret register was being set as it was touching the CH1 config
    registers.

    Fixes: 37bde5acf040 ("ASoC: tlv320adcx140: Add the tlv320adcx140 codec driver family")
    Signed-off-by: Dan Murphy <dmurphy@ti.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200427203608.7031-1-dmurphy@ti.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 67922330913718a7ff48be33d536a722db588908
Author: Christophe Leroy <christophe.leroy@csgroup.eu>
Date:   Tue May 19 05:48:54 2020 +0000

    powerpc/ptdump: Properly handle non standard page size

    commit b00ff6d8c1c3898b0f768cbb38ef722d25bd2f39 upstream.

    In order to properly display information regardless of the page size,
    it is necessary to take into account real page size.

    Fixes: cabe8138b23c ("powerpc: dump as a single line areas mapping a single physical
page.")
    Cc: stable@vger.kernel.org
    Signed-off-by: Christophe Leroy <christophe.leroy@csgroup.eu>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
https://lore.kernel.org/r/a53b2a0ffd042a8d85464bf90d55bc5b970e00a1.1589866984.git.christophe
.leroy@csgroup.eu
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d6dcf203691c8a31871dce6d6086fff4bb0e57c5
Author: Eiichi Tsukata <eiichi.tsukata@nutanix.com>
Date:   Sat Jun 6 13:26:27 2020 +0900

    KVM: x86: Fix APIC page invalidation race

    commit e649b3f0188f8fd34dd0dde8d43fd3312b902fb2 upstream.

    Commit b1394e745b94 ("KVM: x86: fix APIC page invalidation") tried
    to fix inappropriate APIC page invalidation by re-introducing arch
    specific kvm_arch_mmu_notifier_invalidate_range() and calling it from
    kvm_mmu_notifier_invalidate_range_start. However, the patch left a
    possible race where the VMCS APIC address cache is updated *before*
    it is unmapped:

      (Invalidator) kvm_mmu_notifier_invalidate_range_start()
      (Invalidator) kvm_make_all_cpus_request(kvm, KVM_REQ_APIC_PAGE_RELOAD)
      (KVM VCPU) vcpu_enter_guest()
      (KVM VCPU) kvm_vcpu_reload_apic_access_page()
      (Invalidator) actually unmap page

    Because of the above race, there can be a mismatch between the
    host physical address stored in the APIC_ACCESS_PAGE VMCS field and
    the host physical address stored in the EPT entry for the APIC GPA
    (0xfee0000).  When this happens, the processor will not trap APIC
    accesses, and will instead show the raw contents of the APIC-access page.
```

```
    Because Windows OS periodically checks for unexpected modifications to
    the LAPIC register, this will show up as a BSOD crash with BugCheck
    CRITICAL_STRUCTURE_CORRUPTION (109) we are currently seeing in
    https://bugzilla.redhat.com/show_bug.cgi?id=1751017.

    The root cause of the issue is that kvm_arch_mmu_notifier_invalidate_range()
    cannot guarantee that no additional references are taken to the pages in
    the range before kvm_mmu_notifier_invalidate_range_end().  Fortunately,
    this case is supported by the MMU notifier API, as documented in
    include/linux/mmu_notifier.h:

            * If the subsystem
            * can't guarantee that no additional references are taken to
            * the pages in the range, it has to implement the
            * invalidate_range() notifier to remove any references taken
            * after invalidate_range_start().

    The fix therefore is to reload the APIC-access page field in the VMCS
    from kvm_mmu_notifier_invalidate_range() instead of ..._range_start().

    Cc: stable@vger.kernel.org
    Fixes: b1394e745b94 ("KVM: x86: fix APIC page invalidation")
    Fixes: https://bugzilla.kernel.org/show_bug.cgi?id=197951
    Signed-off-by: Eiichi Tsukata <eiichi.tsukata@nutanix.com>
    Message-Id: <20200606042627.61070-1-eiichi.tsukata@nutanix.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d95db354a8e4163e36d1139ac2c3dadaa411e021
Author: Felipe Franciosi <felipe@nutanix.com>
Date:   Tue May 19 08:11:22 2020 +0000

    KVM: x86: respect singlestep when emulating instruction

    commit 384dea1c9183880be183cfaae161d99aafd16df6 upstream.

    When userspace configures KVM_GUESTDBG_SINGLESTEP, KVM will manage the
    presence of X86_EFLAGS_TF via kvm_set/get_rflags on vcpus. The actual
    rflag bit is therefore hidden from callers.

    That includes init_emulate_ctxt() which uses the value returned from
    kvm_get_flags() to set ctxt->tf. As a result, x86_emulate_instruction()
    will skip a single step, leaving singlestep_rip stale and not returning
    to userspace.

    This resolves the issue by observing the vcpu guest_debug configuration
    alongside ctxt->tf in x86_emulate_instruction(), performing the single
    step if set.

    Cc: stable@vger.kernel.org
    Signed-off-by: Felipe Franciosi <felipe@nutanix.com>
    Message-Id: <20200519081048.8204-1-felipe@nutanix.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0c81a7be550821e2b44c4745031c5c5aadb6a814
Author: Sean Christopherson <sean.j.christopherson@intel.com>
Date:   Wed May 27 01:49:09 2020 -0700

    KVM: x86/mmu: Set mmio_value to '0' if reserved #PF can't be generated

    commit 6129ed877d409037b79866327102c9dc59a302fe upstream.

    Set the mmio_value to '0' instead of simply clearing the present bit to
    squash a benign warning in kvm_mmu_set_mmio_spte_mask() that complains
```

```
    about the mmio_value overlapping the lower GFN mask on systems with 52
    bits of PA space.

    Opportunistically clean up the code and comments.

    Cc: stable@vger.kernel.org
    Fixes: d43e2675e96fc ("KVM: x86: only do L1TF workaround on affected processors")
    Signed-off-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Message-Id: <20200527084909.23492-1-sean.j.christopherson@intel.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a34084836c34eabfa3a1aa4ab867d09e7c572541
Author: Maxim Levitsky <mlevitsk@redhat.com>
Date:   Sat May 23 19:14:54 2020 +0300

    KVM: VMX: enable X86_FEATURE_WAITPKG in KVM capabilities

    commit 0abcc8f65cc23b65bc8d1614cc64b02b1641ed7c upstream.

    Even though we might not allow the guest to use WAITPKG's new
    instructions, we should tell KVM that the feature is supported by the
    host CPU.

    Note that vmx_waitpkg_supported checks that WAITPKG _can_ be set in
    secondary execution controls as specified by VMX capability MSR, rather
    that we actually enable it for a guest.

    Cc: stable@vger.kernel.org
    Fixes: e69e72faa3a0 ("KVM: x86: Add support for user wait instructions")
    Suggested-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Maxim Levitsky <mlevitsk@redhat.com>
    Message-Id: <20200523161455.3940-2-mlevitsk@redhat.com>
    Reviewed-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0e1894df27b4c731538c3b3a14d6dcde9bd8b409
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Tue May 19 12:51:32 2020 -0400

    KVM: x86: allow KVM_STATE_NESTED_MTF_PENDING in kvm_state flags

    commit df2a69af85bef169ab6810cc57f6b6b943941e7e upstream.

    The migration functionality was left incomplete in commit 5ef8acbdd687
    ("KVM: nVMX: Emulate MTF when performing instruction emulation", 2020-02-23),
    fix it.

    Fixes: 5ef8acbdd687 ("KVM: nVMX: Emulate MTF when performing instruction emulation")
    Cc: stable@vger.kernel.org
    Reviewed-by: Oliver Upton <oupton@google.com>
    Reviewed-by: Vitaly Kuznetsov <vkuznets@redhat.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit effc6048854384727abbadc0298e6bbbd0a89e55
Author: Maxim Levitsky <mlevitsk@redhat.com>
Date:   Sat May 23 19:14:55 2020 +0300

    KVM: x86: don't expose MSR_IA32_UMWAIT_CONTROL unconditionally

    commit f4cfcd2d5aea4e96c5d483c476f3057b6b7baf6a upstream.

    This msr is only available when the host supports WAITPKG feature.
```

```
    This breaks a nested guest, if the L1 hypervisor is set to ignore
    unknown msrs, because the only other safety check that the
    kernel does is that it attempts to read the msr and
    rejects it if it gets an exception.

    Cc: stable@vger.kernel.org
    Fixes: 6e3ba4abce ("KVM: vmx: Emulate MSR IA32_UMWAIT_CONTROL")
    Signed-off-by: Maxim Levitsky <mlevitsk@redhat.com>
    Message-Id: <20200523161455.3940-3-mlevitsk@redhat.com>
    Reviewed-by: Sean Christopherson <sean.j.christopherson@intel.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a25ee62b42b123228450a5486cc3bbcb61fcd4f0
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Fri May 1 05:54:42 2020 -0700

    perf/x86/intel: Add more available bits for OFFCORE_RESPONSE of Intel Tremont

    commit 0813c40556fce1eeefb996e020cc5339e0b84137 upstream.

    The mask in the extra_regs for Intel Tremont need to be extended to
    allow more defined bits.

    "Outstanding Requests" (bit 63) is only available on MSR_OFFCORE_RSP0;

    Fixes: 6daeb8737f8a ("perf/x86/intel: Add Tremont core PMU support")
    Reported-by: Stephane Eranian <eranian@google.com>
    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20200501125442.7030-1-kan.liang@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d7d3023c220d40472adb5789322a91bffb9f6401
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Sat Jun 6 23:51:17 2020 +0200

    x86/vdso: Unbreak paravirt VDSO clocks

    commit 7778d8417b74aded842eeb372961cfc460417fa0 upstream.

    The conversion of x86 VDSO to the generic clock mode storage broke the
    paravirt and hyperv clocksource logic. These clock sources have their own
    internal sequence counter to validate the clocksource at the point of
    reading it. This is necessary because the hypervisor can invalidate the
    clocksource asynchronously so a check during the VDSO data update is not
    sufficient. If the internal check during read invalidates the clocksource
    the read return U64_MAX. The original code checked this efficiently by
    testing whether the result (casted to signed) is negative, i.e. bit 63 is
    set. This was done that way because an extra indicator for the validity had
    more overhead.

    The conversion broke this check because the check was replaced by a check
    for a valid VDSO clock mode.

    The wreckage manifests itself when the paravirt clock is installed as a
    valid VDSO clock and during runtime invalidated by the hypervisor,
    e.g. after a host suspend/resume cycle. After the invalidation the read
    function returns U64_MAX which is used as cycles and makes the clock jump
    by ~2200 seconds, and become stale until the 2200 seconds have elapsed
    where it starts to jump again. The period of this effect depends on the
    shift/mult pair of the clocksource and the jumps and staleness are an
    artifact of undefined but reproducible behaviour of math overflow.
```

```
    Implement an x86 version of the new vdso_cycles_ok() inline which adds this
    check back and a variant of vdso_clocksource_ok() which lets the compiler
    optimize it out to avoid the extra conditional. That's suboptimal when the
    system does not have a VDSO capable clocksource, but that's not the case
    which is optimized for.

    Fixes: 5d51bee725cc ("clocksource: Add common vdso clock mode storage")
    Reported-by: Miklos Szeredi <miklos@szeredi.hu>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Tested-by: Miklos Szeredi <mszeredi@redhat.com>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20200606221532.080560273@linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 40ed16e6e4df762692ad4403fade31a63c6f4209
Author: Hill Ma <maahiuzeon@gmail.com>
Date:   Sat Apr 25 13:06:41 2020 -0700

    x86/reboot/quirks: Add MacBook6,1 reboot quirk

    commit 140fd4ac78d385e6c8e6a5757585f6c707085f87 upstream.

    On MacBook6,1 reboot would hang unless parameter reboot=pci is added.
    Make it automatic.

    Signed-off-by: Hill Ma <maahiuzeon@gmail.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20200425200641.GA1554@cslab.localdomain
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 69e93896809da49f0946044bd31daf9a7482b440
Author: Anthony Steinhauser <asteinhauser@google.com>
Date:   Sun Jun 7 05:44:19 2020 -0700

    x86/speculation: PR_SPEC_FORCE_DISABLE enforcement for indirect branches.

    commit 4d8df8cbb9156b0a0ab3f802b80cb5db57acc0bf upstream.

    Currently, it is possible to enable indirect branch speculation even after
    it was force-disabled using the PR_SPEC_FORCE_DISABLE option. Moreover, the
    PR_GET_SPECULATION_CTRL command gives afterwards an incorrect result
    (force-disabled when it is in fact enabled). This also is inconsistent
    vs. STIBP and the documention which cleary states that
    PR_SPEC_FORCE_DISABLE cannot be undone.

    Fix this by actually enforcing force-disabled indirect branch
    speculation. PR_SPEC_ENABLE called after PR_SPEC_FORCE_DISABLE now fails
    with -EPERM as described in the documentation.

    Fixes: 9137bb27e60e ("x86/speculation: Add prctl() control for indirect branch
speculation")
    Signed-off-by: Anthony Steinhauser <asteinhauser@google.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 862442343c016befe654c1f3f8d9d5791071df4c
Author: Anthony Steinhauser <asteinhauser@google.com>
Date:   Tue May 19 06:40:42 2020 -0700

    x86/speculation: Avoid force-disabling IBPB based on STIBP and enhanced IBRS.

    commit 21998a351512eba4ed5969006f0c55882d995ada upstream.
```

```
    When STIBP is unavailable or enhanced IBRS is available, Linux
    force-disables the IBPB mitigation of Spectre-BTB even when simultaneous
    multithreading is disabled. While attempts to enable IBPB using
    prctl(PR_SET_SPECULATION_CTRL, PR_SPEC_INDIRECT_BRANCH, ...) fail with
    EPERM, the seccomp syscall (or its prctl(PR_SET_SECCOMP, ...) equivalent)
    which are used e.g. by Chromium or OpenSSH succeed with no errors but the
    application remains silently vulnerable to cross-process Spectre v2 attacks
    (classical BTB poisoning). At the same time the SYSFS reporting
    (/sys/devices/system/cpu/vulnerabilities/spectre_v2) displays that IBPB is
    conditionally enabled when in fact it is unconditionally disabled.

    STIBP is useful only when SMT is enabled. When SMT is disabled and STIBP is
    unavailable, it makes no sense to force-disable also IBPB, because IBPB
    protects against cross-process Spectre-BTB attacks regardless of the SMT
    state. At the same time since missing STIBP was only observed on AMD CPUs,
    AMD does not recommend using STIBP, but recommends using IBPB, so disabling
    IBPB because of missing STIBP goes directly against AMD's advice:
    https://developer.amd.com/wp-
content/resources/Architecture_Guidelines_Update_Indirect_Branch_Control.pdf

    Similarly, enhanced IBRS is designed to protect cross-core BTB poisoning
    and BTB-poisoning attacks from user space against kernel (and
    BTB-poisoning attacks from guest against hypervisor), it is not designed
    to prevent cross-process (or cross-VM) BTB poisoning between processes (or
    VMs) running on the same core. Therefore, even with enhanced IBRS it is
    necessary to flush the BTB during context-switches, so there is no reason
    to force disable IBPB when enhanced IBRS is available.

    Enable the prctl control of IBPB even when STIBP is unavailable or enhanced
    IBRS is available.

    Fixes: 7cc765a67d8e ("x86/speculation: Enable prctl mode for spectre_v2_user")
    Signed-off-by: Anthony Steinhauser <asteinhauser@google.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 18f82da06ec6653646fd2670765aac24275f4833
Author: Anthony Steinhauser <asteinhauser@google.com>
Date:   Sun Jan 5 12:19:43 2020 -0800

    x86/speculation: Prevent rogue cross-process SSBD shutdown

    commit dbbe2ad02e9df26e372f38cc3e70dab9222c832e upstream.

    On context switch the change of TIF_SSBD and TIF_SPEC_IB are evaluated
    to adjust the mitigations accordingly. This is optimized to avoid the
    expensive MSR write if not needed.

    This optimization is buggy and allows an attacker to shutdown the SSBD
    protection of a victim process.

    The update logic reads the cached base value for the speculation control
    MSR which has neither the SSBD nor the STIBP bit set. It then OR's the
    SSBD bit only when TIF_SSBD is different and requests the MSR update.

    That means if TIF_SSBD of the previous and next task are the same, then
    the base value is not updated, even if TIF_SSBD is set. The MSR write is
    not requested.

    Subsequently if the TIF_STIBP bit differs then the STIBP bit is updated
    in the base value and the MSR is written with a wrong SSBD value.

    This was introduced when the per task/process conditional STIPB
```

```
    switching was added on top of the existing SSBD switching.

    It is exploitable if the attacker creates a process which enforces SSBD
    and has the contrary value of STIBP than the victim process (i.e. if the
    victim process enforces STIBP, the attacker process must not enforce it;
    if the victim process does not enforce STIBP, the attacker process must
    enforce it) and schedule it on the same core as the victim process. If
    the victim runs after the attacker the victim becomes vulnerable to
    Spectre V4.

    To fix this, update the MSR value independent of the TIF_SSBD difference
    and dependent on the SSBD mitigation method available. This ensures that
    a subsequent STIPB initiated MSR write has the correct state of SSBD.

    [ tglx: Handle X86_FEATURE_VIRT_SSBD & X86_FEATURE_VIRT_SSBD correctly
            and massaged changelog ]

    Fixes: 5bfbe3ad5840 ("x86/speculation: Prepare for per task indirect branch speculation
control")
    Signed-off-by: Anthony Steinhauser <asteinhauser@google.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6c8a0112fc896df03951898d5452263ead978429
Author: Xiaochun Lee <lixc17@lenovo.com>
Date:   Thu May 14 23:31:07 2020 -0400

    x86/PCI: Mark Intel C620 MROMs as having non-compliant BARs

    commit 1574051e52cb4b5b7f7509cfd729b76ca1117808 upstream.

    The Intel C620 Platform Controller Hub has MROM functions that have non-PCI
    registers (undocumented in the public spec) where BAR 0 is supposed to be,
    which results in messages like this:

      pci 0000:00:11.0: [Firmware Bug]: reg 0x30: invalid BAR (can't size)

    Mark these MROM functions as having non-compliant BARs so we don't try to
    probe any of them.  There are no other BARs on these devices.

    See the Intel C620 Series Chipset Platform Controller Hub Datasheet,
    May 2019, Document Number 336067-007US, sec 2.1, 35.5, 35.6.

    [bhelgaas: commit log, add 0xa26d]
    Link: https://lore.kernel.org/r/1589513467-17070-1-git-send-email-
lixiaochun.2888@163.com
    Signed-off-by: Xiaochun Lee <lixc17@lenovo.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 81845dd732fc5fbac446c7e834bd477fb34596b9
Author: Steven Price <steven.price@arm.com>
Date:   Mon Jun 1 21:49:58 2020 -0700

    x86: mm: ptdump: calculate effective permissions correctly

    commit 1494e0c38ee903e83aefb58caf54a9217273d49a upstream.

    Patch series "Fix W+X debug feature on x86"

    Jan alerted me[1] that the W+X detection debug feature was broken in x86
    by my change[2] to switch x86 to use the generic ptdump infrastructure.
```

```
    Fundamentally the approach of trying to move the calculation of
    effective permissions into note_page() was broken because note_page() is
    only called for 'leaf' entries and the effective permissions are passed
    down via the internal nodes of the page tree.  The solution I've taken
    here is to create a new (optional) callback which is called for all
    nodes of the page tree and therefore can calculate the effective
    permissions.

    Secondly on some configurations (32 bit with PAE) "unsigned long" is not
    large enough to store the table entries.  The fix here is simple - let's
    just use a u64.

    [1] https://lore.kernel.org/lkml/d573dc7e-e742-84de-473d-f971142fa319@suse.com/
    [2] 2ae27137b2db ("x86: mm: convert dump_pagetables to use walk_page_range")

    This patch (of 2):

    By switching the x86 page table dump code to use the generic code the
    effective permissions are no longer calculated correctly because the
    note_page() function is only called for *leaf* entries.  To calculate
    the actual effective permissions it is necessary to observe the full
    hierarchy of the page tree.

    Introduce a new callback for ptdump which is called for every entry and
    can therefore update the prot_levels array correctly.  note_page() can
    then simply access the appropriate element in the array.

    [steven.price@arm.com: make the assignment conditional on val != 0]
      Link: http://lkml.kernel.org/r/430c8ab4-e7cd-6933-dde6-087fac6db872@arm.com
    Fixes: 2ae27137b2db ("x86: mm: convert dump_pagetables to use walk_page_range")
    Reported-by: Jan Beulich <jbeulich@suse.com>
    Signed-off-by: Steven Price <steven.price@arm.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Cc: Qian Cai <cai@lca.pw>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Dave Hansen <dave.hansen@linux.intel.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: <stable@vger.kernel.org>
    Link: http://lkml.kernel.org/r/20200521152308.33096-1-steven.price@arm.com
    Link: http://lkml.kernel.org/r/20200521152308.33096-2-steven.price@arm.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6dfcb285c39eb5d9bb7cb5dc279f2c5ed2d42393
Author: Bob Haarman <inglorion@google.com>
Date:   Tue Jun 2 12:30:59 2020 -0700

    x86_64: Fix jiffies ODR violation

    commit d8ad6d39c35d2b44b3d48b787df7f3359381dcbf upstream.

    'jiffies' and 'jiffies_64' are meant to alias (two different symbols that
    share the same address).  Most architectures make the symbols alias to the
    same address via a linker script assignment in their
    arch/<arch>/kernel/vmlinux.lds.S:

    jiffies = jiffies_64;

    which is effectively a definition of jiffies.

    jiffies and jiffies_64 are both forward declared for all architectures in
    include/linux/jiffies.h. jiffies_64 is defined in kernel/time/timer.c.
```

```
    x86_64 was peculiar in that it wasn't doing the above linker script
    assignment, but rather was:
    1. defining jiffies in arch/x86/kernel/time.c instead via the linker script.
    2. overriding the symbol jiffies_64 from kernel/time/timer.c in
    arch/x86/kernel/vmlinux.lds.s via 'jiffies_64 = jiffies;'.

    As Fangrui notes:

      In LLD, symbol assignments in linker scripts override definitions in
      object files. GNU ld appears to have the same behavior. It would
      probably make sense for LLD to error "duplicate symbol" but GNU ld
      is unlikely to adopt for compatibility reasons.

    This results in an ODR violation (UB), which seems to have survived
    thus far. Where it becomes harmful is when;

    1. -fno-semantic-interposition is used:

    As Fangrui notes:

      Clang after LLVM commit 5b22bcc2b70d
      ("[X86][ELF] Prefer to lower MC_GlobalAddress operands to .Lfoo$local")
      defaults to -fno-semantic-interposition similar semantics which help
      -fpic/-fPIC code avoid GOT/PLT when the referenced symbol is defined
      within the same translation unit. Unlike GCC
      -fno-semantic-interposition, Clang emits such relocations referencing
      local symbols for non-pic code as well.

    This causes references to jiffies to refer to '.Ljiffies$local' when
    jiffies is defined in the same translation unit. Likewise, references to
    jiffies_64 become references to '.Ljiffies_64$local' in translation units
    that define jiffies_64.  Because these differ from the names used in the
    linker script, they will not be rewritten to alias one another.

    2. Full LTO

    Full LTO effectively treats all source files as one translation
    unit, causing these local references to be produced everywhere.  When
    the linker processes the linker script, there are no longer any
    references to jiffies_64' anywhere to replace with 'jiffies'.  And
    thus '.Ljiffies$local' and '.Ljiffies_64$local' no longer alias
    at all.

    In the process of porting patches enabling Full LTO from arm64 to x86_64,
    spooky bugs have been observed where the kernel appeared to boot, but init
    doesn't get scheduled.

    Avoid the ODR violation by matching other architectures and define jiffies
    only by linker script.  For -fno-semantic-interposition + Full LTO, there
    is no longer a global definition of jiffies for the compiler to produce a
    local symbol which the linker script won't ensure aliases to jiffies_64.

    Fixes: 40747ffa5aa8 ("asmlinkage: Make jiffies visible")
    Reported-by: Nathan Chancellor <natechancellor@gmail.com>
    Reported-by: Alistair Delva <adelva@google.com>
    Debugged-by: Nick Desaulniers <ndesaulniers@google.com>
    Debugged-by: Sami Tolvanen <samitolvanen@google.com>
    Suggested-by: Fangrui Song <maskray@google.com>
    Signed-off-by: Bob Haarman <inglorion@google.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Tested-by: Sedat Dilek <sedat.dilek@gmail.com> # build+boot on
    Reviewed-by: Andi Kleen <ak@linux.intel.com>
    Reviewed-by: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: stable@vger.kernel.org
```

```
    Link: https://github.com/ClangBuiltLinux/linux/issues/852
    Link: https://lkml.kernel.org/r/20200602193100.229287-1-inglorion@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b33e28d64d1e62ac8c6ae03613cf3d7281d71d6a
Author: Vlastimil Babka <vbabka@suse.cz>
Date:   Mon Jun 1 21:45:43 2020 -0700

    usercopy: mark dma-kmalloc caches as usercopy caches

    commit 49f2d2419d60a103752e5fbaf158cf8d07c0d884 upstream.

    We have seen a "usercopy: Kernel memory overwrite attempt detected to
    SLUB object 'dma-kmalloc-1 k' (offset 0, size 11)!" error on s390x, as
    IUCV uses kmalloc() with __GFP_DMA because of memory address
    restrictions.  The issue has been discussed [2] and it has been noted
    that if all the kmalloc caches are marked as usercopy, there's little
    reason not to mark dma-kmalloc caches too.  The 'dma' part merely means
    that __GFP_DMA is used to restrict memory address range.

    As Jann Horn put it [3]:
     "I think dma-kmalloc slabs should be handled the same way as normal
      kmalloc slabs. When a dma-kmalloc allocation is freshly created, it is
      just normal kernel memory - even if it might later be used for DMA -,
      and it should be perfectly fine to copy_from_user() into such
      allocations at that point, and to copy_to_user() out of them at the
      end. If you look at the places where such allocations are created, you
      can see things like kmemdup(), memcpy() and so on - all normal
      operations that shouldn't conceptually be different from usercopy in
      any relevant way."

    Thus this patch marks the dma-kmalloc-* caches as usercopy.

    [1] https://bugzilla.suse.com/show_bug.cgi?id=1156053
    [2] https://lore.kernel.org/kernel-hardening/bfca96db-bbd0-d958-7732-
76e36c667c68@suse.cz/
    [3] https://lore.kernel.org/kernel-
hardening/CAG48ez1a4waGk9kB0WLaSbs4muSoK0AYAVk8=XYaKj4_+6e6Hg@mail.gmail.com/

    Signed-off-by: Vlastimil Babka <vbabka@suse.cz>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Acked-by: Christian Borntraeger <borntraeger@de.ibm.com>
    Acked-by: Jiri Slaby <jslaby@suse.cz>
    Cc: Jann Horn <jannh@google.com>
    Cc: Christoph Hellwig <hch@infradead.org>
    Cc: Christopher Lameter <cl@linux.com>
    Cc: Julian Wiedmann <jwi@linux.ibm.com>
    Cc: Ursula Braun <ubraun@linux.ibm.com>
    Cc: Alexander Viro <viro@zeniv.linux.org.uk>
    Cc: David Windsor <dave@nullcore.net>
    Cc: Pekka Enberg <penberg@kernel.org>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Joonsoo Kim <iamjoonsoo.kim@lge.com>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: "David S. Miller" <davem@davemloft.net>
    Cc: Laura Abbott <labbott@redhat.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: "Martin K. Petersen" <martin.petersen@oracle.com>
    Cc: Paolo Bonzini <pbonzini@redhat.com>
    Cc: Christoffer Dall <christoffer.dall@linaro.org>
    Cc: Dave Kleikamp <dave.kleikamp@oracle.com>
    Cc: Jan Kara <jack@suse.cz>
    Cc: Luis de Bethencourt <luisbg@kernel.org>
    Cc: Marc Zyngier <marc.zyngier@arm.com>
    Cc: Rik van Riel <riel@surriel.com>
```

```
    Cc: Matthew Garrett <mjg59@google.com>
    Cc: Michal Kubecek <mkubecek@suse.cz>
    Link: http://lkml.kernel.org/r/7d810f6d-8085-ea2f-7805-47ba3842dc50@suse.cz
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c27f7272c5064519c0fd2a0f381756bdab44d722
Author: Miklos Szeredi <mszeredi@redhat.com>
Date:   Thu May 14 16:44:24 2020 +0200

    aio: fix async fsync creds

    commit 530f32fc370fd1431ea9802dbc53ab5601dfccdb upstream.

    Avi Kivity reports that on fuse filesystems running in a user namespace
    asyncronous fsync fails with EOVERFLOW.

    The reason is that f_ops->fsync() is called with the creds of the kthread
    performing aio work instead of the creds of the process originally
    submitting IOCB_CMD_FSYNC.

    Fuse sends the creds of the caller in the request header and it needs to
    translate the uid and gid into the server's user namespace.  Since the
    kthread is running in init_user_ns, the translation will fail and the
    operation returns an error.

    It can be argued that fsync doesn't actually need any creds, but just
    zeroing out those fields in the header (as with requests that currently
    don't take creds) is a backward compatibility risk.

    Instead of working around this issue in fuse, solve the core of the problem
    by calling the filesystem with the proper creds.

    Reported-by: Avi Kivity <avi@scylladb.com>
    Tested-by: Giuseppe Scrivano <gscrivan@redhat.com>
    Fixes: c9582eb0ff7d ("fuse: Fail all requests with invalid uids or gids")
    Cc: stable@vger.kernel.org  # 4.18+
    Signed-off-by: Miklos Szeredi <mszeredi@redhat.com>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 22d65fe6d7b85d63c4aebdddbad58f4e864a1517
Author: Bjorn Helgaas <bhelgaas@google.com>
Date:   Fri May 15 14:31:16 2020 -0500

    PCI/PM: Adjust pcie_wait_for_link_delay() for caller delay

    [ Upstream commit f044baaff1eb7ae5aa7a36f1b7ad5bd8eeb672c4 ]

    The caller of pcie_wait_for_link_delay() specifies the time to wait after
    the link becomes active.  When the downstream port doesn't support link
    active reporting, obviously we can't tell when the link becomes active, so
    we waited the worst-case time (1000 ms) plus 100 ms, ignoring the delay
    from the caller.

    Instead, wait for 1000 ms + the delay from the caller.

    Fixes: 4827d63891b6 ("PCI/PM: Add pcie_wait_for_link_delay()")
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit df4366395bc98e6d977d84aecdfee7a69b6b4dbb
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Tue May 19 05:34:41 2020 -0400
```

```
    KVM: x86: only do L1TF workaround on affected processors

    commit d43e2675e96fc6ae1a633b6a69d296394448cc32 upstream.

    KVM stores the gfn in MMIO SPTEs as a caching optimization.  These are split
    in two parts, as in "[high 11111 low]", to thwart any attempt to use these bits
    in an L1TF attack.  This works as long as there are 5 free bits between
    MAXPHYADDR and bit 50 (inclusive), leaving bit 51 free so that the MMIO
    access triggers a reserved-bit-set page fault.

    The bit positions however were computed wrongly for AMD processors that have
    encryption support.  In this case, x86_phys_bits is reduced (for example
    from 48 to 43, to account for the C bit at position 47 and four bits used
    internally to store the SEV ASID and other stuff) while x86_cache_bits in
    would remain set to 48, and _all_ bits between the reduced MAXPHYADDR
    and bit 51 are set.  Then low_phys_bits would also cover some of the
    bits that are set in the shadow_mmio_value, terribly confusing the gfn
    caching mechanism.

    To fix this, avoid splitting gfns as long as the processor does not have
    the L1TF bug (which includes all AMD processors).  When there is no
    splitting, low_phys_bits can be set to the reduced MAXPHYADDR removing
    the overlap.  This fixes "npt=0" operation on EPYC processors.

    Thanks to Maxim Levitsky for bisecting this bug.

    Cc: stable@vger.kernel.org
    Fixes: 52918ed5fcf0 ("KVM: SVM: Override default MMIO mask if memory encryption is
enabled")
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 790a14b5193ca1558e3ee73fe93806c70769ebb7
Author: Daniel Jordan <daniel.m.jordan@oracle.com>
Date:   Tue Apr 21 12:34:55 2020 -0400

    padata: add separate cpuhp node for CPUHP_PADATA_DEAD

    [ Upstream commit 3c2214b6027ff37945799de717c417212e1a8c54 ]

    Removing the pcrypt module triggers this:

      general protection fault, probably for non-canonical
        address 0xdead000000000122
      CPU: 5 PID: 264 Comm: modprobe Not tainted 5.6.0+ #2
      Hardware name: QEMU Standard PC
      RIP: 0010:__cpuhp_state_remove_instance+0xcc/0x120
      Call Trace:
       padata_sysfs_release+0x74/0xce
       kobject_put+0x81/0xd0
       padata_free+0x12/0x20
       pcrypt_exit+0x43/0x8ee [pcrypt]

    padata instances wrongly use the same hlist node for the online and dead
    states, so __padata_free()'s second cpuhp remove call chokes on the node
    that the first poisoned.

    cpuhp multi-instance callbacks only walk forward in cpuhp_step->list and
    the same node is linked in both the online and dead lists, so the list
    corruption that results from padata_alloc() adding the node to a second
    list without removing it from the first doesn't cause problems as long
    as no instances are freed.

    Avoid the issue by giving each state its own node.
```

```
    Fixes: 894c9ef9780c ("padata: validate cpumask without removed CPU during offline")
    Signed-off-by: Daniel Jordan <daniel.m.jordan@oracle.com>
    Cc: Herbert Xu <herbert@gondor.apana.org.au>
    Cc: Steffen Klassert <steffen.klassert@secunet.com>
    Cc: linux-crypto@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a9ed450969608cc3bb73b72b563947b26903eedb
Author: Jason Gunthorpe <jgg@ziepe.ca>
Date:   Mon Apr 6 21:44:26 2020 -0300

    RDMA/uverbs: Make the event_queue fds return POLLERR when disassociated

    [ Upstream commit eb356e6dc15a30af604f052cd0e170450193c254 ]

    If is_closed is set, and the event list is empty, then read() will return
    -EIO without blocking. After setting is_closed in
    ib_uverbs_free_event_queue(), we do trigger a wake_up on the poll_wait,
    but the fops->poll() function does not check it, so poll will continue to
    sleep on an empty list.

    Fixes: 14e23bd6d221 ("RDMA/core: Fix locking in ib_uverbs_event_read")
    Link: https://lore.kernel.org/r/0-v1-ace813388969+48859-
uverbs_poll_fix%25jgg@mellanox.com
    Reviewed-by: Leon Romanovsky <leonro@mellanox.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5fe730762d7c45ad0b315618c7b84c146ba49217
Author: Kim Phillips <kim.phillips@amd.com>
Date:   Fri Apr 17 09:33:56 2020 -0500

    x86/cpu/amd: Make erratum #1054 a legacy erratum

    [ Upstream commit e2abfc0448a46d8a137505aa180caf14070ec535 ]

    Commit

      21b5ee59ef18 ("x86/cpu/amd: Enable the fixed Instructions Retired
                    counter IRPERF")

    mistakenly added erratum #1054 as an OS Visible Workaround (OSVW) ID 0.
    Erratum #1054 is not OSVW ID 0 [1], so make it a legacy erratum.

    There would never have been a false positive on older hardware that
    has OSVW bit 0 set, since the IRPERF feature was not available.

    However, save a couple of RDMSR executions per thread, on modern
    system configurations that correctly set non-zero values in their
    OSVW_ID_Length MSRs.

    [1] Revision Guide for AMD Family 17h Models 00h-0Fh Processors. The
    revision guide is available from the bugzilla link below.

    Fixes: 21b5ee59ef18 ("x86/cpu/amd: Enable the fixed Instructions Retired counter
IRPERF")
    Reported-by: Andrew Cooper <andrew.cooper3@citrix.com>
    Signed-off-by: Kim Phillips <kim.phillips@amd.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Link: https://lkml.kernel.org/r/20200417143356.26054-1-kim.phillips@amd.com
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206537
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 241c45f3907a03d795347f6c0e31ccb1f48230cc
Author: Petr Tesarik <ptesarik@suse.com>
Date:   Fri May 22 20:39:22 2020 +0200

    s390/pci: Log new handle in clp_disable_fh()

    [ Upstream commit e1750a3d9abbea2ece29cac8dc5a6f5bc19c1492 ]

    After disabling a function, the original handle is logged instead of
    the disabled handle.

    Link: https://lkml.kernel.org/r/20200522183922.5253-1-ptesarik@suse.com
    Fixes: 17cdec960cf7 ("s390/pci: Recover handle in clp_set_pci_fn()")
    Reviewed-by: Pierre Morel <pmorel@linux.ibm.com>
    Signed-off-by: Petr Tesarik <ptesarik@suse.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 862b865247465f8c9355fddeda5eaf2844de818b
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Wed Apr 8 21:04:31 2020 +0200

    smack: avoid unused 'sip' variable warning

    [ Upstream commit 00720f0e7f288d29681d265c23b22bb0f0f4e5b4 ]

    The mix of IS_ENABLED() and #ifdef checks has left a combination
    that causes a warning about an unused variable:

    security/smack/smack_lsm.c: In function 'smack_socket_connect':
    security/smack/smack_lsm.c:2838:24: error: unused variable 'sip' [-Werror=unused-
variable]
     2838 |   struct sockaddr_in6 *sip = (struct sockaddr_in6 *)sap;

    Change the code to use C-style checks consistently so the compiler
    can handle it correctly.

    Fixes: 87fbfffcc89b ("broken ping to ipv6 linklocal addresses on debian buster")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Casey Schaufler <casey@schaufler-ca.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eebb7f685c12d9e5067e24d24f757ef874a15ab7
Author: Masashi Honma <masashi.honma@gmail.com>
Date:   Tue May 5 06:44:43 2020 +0900

    ath9k_htc: Silence undersized packet warnings

    [ Upstream commit 450edd2805982d14ed79733a82927d2857b27cac ]

    Some devices like TP-Link TL-WN722N produces this kind of messages
    frequently.

    kernel: ath: phy0: Short RX data len, dropping (dlen: 4)

    This warning is useful for developers to recognize that the device
    (Wi-Fi dongle or USB hub etc) is noisy but not for general users. So
    this patch make this warning to debug message.

    Reported-By: Denis <pro.denis@protonmail.com>
    Ref: https://bugzilla.kernel.org/show_bug.cgi?id=207539
    Fixes: cd486e627e67 ("ath9k_htc: Discard undersized packets")
    Signed-off-by: Masashi Honma <masashi.honma@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
```

```
    Link: https://lore.kernel.org/r/20200504214443.4485-1-masashi.honma@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3877a343bdd7f7c3e9ffc2efd6860b743335bcf
Author: Sasha Levin <sashal@kernel.org>
Date:   Thu Jun 11 20:17:21 2020 -0400

    spi: dw: Fix native CS being unset

    [ Upstream commit 9aea644ca17b94f82ad7fa767cbc4509642f4420 ]

    Commit 6e0a32d6f376 ("spi: dw: Fix default polarity of native
    chipselect") attempted to fix the problem when GPIO active-high
    chip-select is utilized to communicate with some SPI slave. It fixed
    the problem, but broke the normal native CS support. At the same time
    the reversion commit ada9e3fcc175 ("spi: dw: Correct handling of native
    chipselect") didn't solve the problem either, since it just inverted
    the set_cs() polarity perception without taking into account that
    CS-high might be applicable. Here is what is done to finally fix the
    problem.

    DW SPI controller demands any native CS being set in order to proceed
    with data transfer. So in order to activate the SPI communications we
    must set any bit in the Slave Select DW SPI controller register no
    matter whether the platform requests the GPIO- or native CS. Preferably
    it should be the bit corresponding to the SPI slave CS number. But
    currently the dw_spi_set_cs() method activates the chip-select
    only if the second argument is false. Since the second argument of the
    set_cs callback is expected to be a boolean with "is-high" semantics
    (actual chip-select pin state value), the bit in the DW SPI Slave
    Select register will be set only if SPI core requests the driver
    to set the CS in the low state. So this will work for active-low
    GPIO-based CS case, and won't work for active-high CS setting
    the bit when SPI core actually needs to deactivate the CS.

    This commit fixes the problem for all described cases. So no matter
    whether an SPI slave needs GPIO- or native-based CS with active-high
    or low signal the corresponding bit will be set in SER.

    Signed-off-by: Serge Semin <Sergey.Semin@baikalelectronics.ru>
    Fixes: ada9e3fcc175 ("spi: dw: Correct handling of native chipselect")
    Fixes: 6e0a32d6f376 ("spi: dw: Fix default polarity of native chipselect")
    Reviewed-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Acked-by: Linus Walleij <linus.walleij@linaro.org>

    Link: https://lore.kernel.org/r/20200515104758.6934-5-Sergey.Semin@baikalelectronics.ru
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c4777ed636154e46be2a8f527ba282ae7045662a
Author: Cédric Le Goater <clg@kaod.org>
Date:   Wed Apr 29 09:51:20 2020 +0200

    powerpc/xive: Clear the page tables for the ESB IO mapping

    [ Upstream commit a101950fcb78b0ba20cd487be6627dea58d55c2b ]

    Commit 1ca3dec2b2df ("powerpc/xive: Prevent page fault issues in the
    machine crash handler") fixed an issue in the FW assisted dump of
    machines using hash MMU and the XIVE interrupt mode under the POWER
    hypervisor. It forced the mapping of the ESB page of interrupts being
    mapped in the Linux IRQ number space to make sure the 'crash kexec'
    sequence worked during such an event. But it didn't handle the
    un-mapping.
```

```
    This mapping is now blocking the removal of a passthrough IO adapter
    under the POWER hypervisor because it expects the guest OS to have
    cleared all page table entries related to the adapter. If some are
    still present, the RTAS call which isolates the PCI slot returns error
    9001 "valid outstanding translations".

    Remove these mapping in the IRQ data cleanup routine.

    Under KVM, this cleanup is not required because the ESB pages for the
    adapter interrupts are un-mapped from the guest by the hypervisor in
    the KVM XIVE native device. This is now redundant but it's harmless.

    Fixes: 1ca3dec2b2df ("powerpc/xive: Prevent page fault issues in the machine crash
handler")
    Cc: stable@vger.kernel.org # v5.5+
    Signed-off-by: Cédric Le Goater <clg@kaod.org>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/20200429075122.1216388-2-clg@kaod.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fffd9981950c5f34f1c1548a4babe8445407f847
Author: Amir Goldstein <amir73il@gmail.com>
Date:   Sun May 24 10:24:41 2020 +0300

    fanotify: fix ignore mask logic for events on child and on dir

    [ Upstream commit 2f02fd3fa13e51713b630164f8a8e5b42de8283b ]

    The comments in fanotify_group_event_mask() say:

      "If the event is on dir/child and this mark doesn't care about
       events on dir/child, don't send it!"

    Specifically, mount and filesystem marks do not care about events
    on child, but they can still specify an ignore mask for those events.
    For example, a group that has:
    - A mount mark with mask 0 and ignore_mask FAN_OPEN
    - An inode mark on a directory with mask FAN_OPEN | FAN_OPEN_EXEC
      with flag FAN_EVENT_ON_CHILD

    A child file open for exec would be reported to group with the FAN_OPEN
    event despite the fact that FAN_OPEN is in ignore mask of mount mark,
    because the mark iteration loop skips over non-inode marks for events
    on child when calculating the ignore mask.

    Move ignore mask calculation to the top of the iteration loop block
    before excluding marks for events on dir/child.

    Link: https://lore.kernel.org/r/20200524072441.18258-1-amir73il@gmail.com
    Reported-by: Jan Kara <jack@suse.cz>
    Link: https://lore.kernel.org/linux-fsdevel/20200521162443.GA26052@quack2.suse.cz/
    Fixes: 55bf882c7f13 "fanotify: fix merging marks masks with FAN_ONDIR"
    Fixes: b469e7e47c8a "fanotify: fix handling of events on child..."
    Signed-off-by: Amir Goldstein <amir73il@gmail.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 70154bb515a325c007bcf64971099e1e17f16f28
Author: Saravana Kannan <saravanak@google.com>
Date:   Tue May 26 15:09:27 2020 -0700

    driver core: Update device link status correctly for SYNC_STATE_ONLY links

    [ Upstream commit 8c3e315d4296421cd26b3300ee0ac117f0877f20 ]
```

```
    When SYNC_STATE_ONLY support was added in commit 05ef983e0d65 ("driver
    core: Add device link support for SYNC_STATE_ONLY flag"),
    SYNC_STATE_ONLY links were treated similar to STATELESS links in terms
    of not blocking consumer probe if the supplier hasn't probed yet.

    That caused a SYNC_STATE_ONLY device link's status to not get updated.
    Since SYNC_STATE_ONLY device link is no longer useful once the
    consumer probes, commit 21c27f06587d ("driver core: Fix
    SYNC_STATE_ONLY device link implementation") addresses the status
    update issue by deleting the SYNC_STATE_ONLY device link instead of
    complicating the status update code.

    However, there are still some cases where we need to update the status
    of a SYNC_STATE_ONLY device link. This is because a SYNC_STATE_ONLY
    device link can later get converted into a normal MANAGED device link
    when a normal MANAGED device link is created between a supplier and
    consumer that already have a SYNC_STATE_ONLY device link between them.

    If a SYNC_STATE_ONLY device link's status isn't maintained correctly
    till it's converted to a normal MANAGED device link, then the normal
    MANAGED device link will end up with a wrong link status. This can cause
    a warning stack trace[1] when the consumer device probes successfully.

    This commit fixes the SYNC_STATE_ONLY device link status update issue
    where it wouldn't transition correctly from DL_STATE_DORMANT or
    DL_STATE_AVAILABLE to DL_STATE_CONSUMER_PROBE. It also resets the status
    back to DL_STATE_DORMANT or DL_STATE_AVAILABLE if the consumer probe
    fails.

    [1] - https://lore.kernel.org/lkml/20200522204120.3b3c9ed6@apollo/
    Fixes: 05ef983e0d65 ("driver core: Add device link support for SYNC_STATE_ONLY flag")
    Fixes: 21c27f06587d ("driver core: Fix SYNC_STATE_ONLY device link implementation")
    Reported-by: Michael Walle <michael@walle.cc>
    Tested-by: Michael Walle <michael@walle.cc>
    Signed-off-by: Saravana Kannan <saravanak@google.com>
    Reviewed-by: Rafael J. Wysocki <rrafael.j.wysocki@intel.com>
    Link: https://lore.kernel.org/r/20200526220928.49939-1-saravanak@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b60b8ebe2afe994a62459e5cbdf782c4559aa0e6
Author: Masami Hiramatsu <mhiramat@kernel.org>
Date:   Wed May 6 23:29:12 2020 +0900

    perf probe: Accept the instance number of kretprobe event

    [ Upstream commit c6aab66a728b6518772c74bd9dff66e1a1c652fd ]

    Since the commit 6a13a0d7b4d1 ("ftrace/kprobe: Show the maxactive number
    on kprobe_events") introduced to show the instance number of kretprobe
    events, the length of the 1st format of the kprobe event will not 1, but
    it can be longer.  This caused a parser error in perf-probe.

    Skip the length check the 1st format of the kprobe event to accept this
    instance number.

    Without this fix:

      # perf probe -a vfs_read%return
      Added new event:
        probe:vfs_read__return (on vfs_read%return)

      You can now use it in all perf tools, such as:
```

```
             perf record -e probe:vfs_read__return -aR sleep 1

       # perf probe -l
       Semantic error :Failed to parse event name: r16:probe/vfs_read__return
         Error: Failed to show event list.

    And with this fixes:

       # perf probe -a vfs_read%return
       ...
       # perf probe -l
         probe:vfs_read__return (on vfs_read%return)

    Fixes: 6a13a0d7b4d1 ("ftrace/kprobe: Show the maxactive number on kprobe_events")
    Reported-by: Yuxuan Shui <yshuiv7@gmail.com>
    Signed-off-by: Masami Hiramatsu <mhiramat@kernel.org>
    Tested-by: Yuxuan Shui <yshuiv7@gmail.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: stable@vger.kernel.org
    Bugzilla: https://bugzilla.kernel.org/show_bug.cgi?id=207587
    Link: http://lore.kernel.org/lkml/158877535215.26469.1113127926699134067.stgit@devnote2
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f492c6b6da1fce315d7d5a7b912ffeaa8a29fb2c
Author: Waiman Long <longman@redhat.com>
Date:   Thu Jun 4 16:48:21 2020 -0700

    mm: add kvfree_sensitive() for freeing sensitive data objects

    [ Upstream commit d4eaa2837851db2bfed572898bfc17f9a9f9151e ]

    For kvmalloc'ed data object that contains sensitive information like
    cryptographic keys, we need to make sure that the buffer is always cleared
    before freeing it.  Using memset() alone for buffer clearing may not
    provide certainty as the compiler may compile it away.  To be sure, the
    special memzero_explicit() has to be used.

    This patch introduces a new kvfree_sensitive() for freeing those sensitive
    data objects allocated by kvmalloc().  The relevant places where
    kvfree_sensitive() can be used are modified to use it.

    Fixes: 4f0882491a14 ("KEYS: Avoid false positive ENOMEM error on key read")
    Suggested-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Waiman Long <longman@redhat.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Reviewed-by: Eric Biggers <ebiggers@google.com>
    Acked-by: David Howells <dhowells@redhat.com>
    Cc: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Cc: James Morris <jmorris@namei.org>
    Cc: "Serge E. Hallyn" <serge@hallyn.com>
    Cc: Joe Perches <joe@perches.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: David Rientjes <rientjes@google.com>
    Cc: Uladzislau Rezki <urezki@gmail.com>
    Link: http://lkml.kernel.org/r/20200407200318.11711-1-longman@redhat.com
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d5e7105e690f60ac4cbb0b252c2e975bb909dec0
Author: Vlad Buslov <vladbu@mellanox.com>
Date:   Thu May 14 09:35:52 2020 +0300

    selftests: fix flower parent qdisc
```

```
    [ Upstream commit 0531b0357ba37464e5c0033e1b7c69bbf5ecd8fb ]

    Flower tests used to create ingress filter with specified parent qdisc
    "parent ffff:" but dump them on "ingress". With recent commit that fixed
    tcm_parent handling in dump those are not considered same parent anymore,
    which causes iproute2 tc to emit additional "parent ffff:" in first line of
    filter dump output. The change in output causes filter match in tests to
    fail.

    Prevent parent qdisc output when dumping filters in flower tests by always
    correctly specifying "ingress" parent both when creating and dumping
    filters.

    Fixes: a7df4870d79b ("net_sched: fix tcm_parent in tc filter dump")
    Signed-off-by: Vlad Buslov <vladbu@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 012fa49c2e4e1d8be3d4be25c2ee77e6a21ae69a
Author: Jérôme Pouiller <jerome.pouiller@silabs.com>
Date:   Tue May 5 14:37:45 2020 +0200

    staging: wfx: fix double free

    [ Upstream commit 832cc98141b4b93acbb9231ca9e36f7fbe347f47 ]

    In case of error in wfx_probe(), wdev->hw is freed. Since an error
    occurred, wfx_free_common() is called, then wdev->hw is freed again.

    Signed-off-by: Jérôme Pouiller <jerome.pouiller@silabs.com>
    Reviewed-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Fixes: 4033714d6cbe ("staging: wfx: fix init/remove vs IRQ race")
    Link: https://lore.kernel.org/r/20200505123757.39506-4-Jerome.Pouiller@silabs.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 83d936da919a6b1611aa3f2ce6c2a22a6f3bb63c
Author: Sergio Paracuellos <sergio.paracuellos@gmail.com>
Date:   Thu Apr 9 13:16:52 2020 +0200

    staging: mt7621-pci: properly power off dual-ported pcie phy

    [ Upstream commit 5fcded5e857cf66c9592e4be28c4dab4520c9177 ]

    Pcie phy for pcie0 and pcie1 is shared using a dual ported
    one. Current code was assuming that if nothing is connected
    in pcie0 it won't be also nothing connected in pcie1. This
    assumtion is wrong for some devices such us 'Mikrotik rbm33g'
    and 'ZyXEL LTE3301-PLUS' where only connecting a card to the
    second bus on the phy is possible. For such devices kernel
    hangs in the same point because of the wrong poweroff of the
    phy getting the following trace:

    mt7621-pci-phy 1e149000.pcie-phy: PHY for 0xbe149000 (dual port = 1)
    mt7621-pci-phy 1e14a000.pcie-phy: PHY for 0xbe14a000 (dual port = 0)
    mt7621-pci-phy 1e149000.pcie-phy: Xtal is 40MHz
    mt7621-pci-phy 1e14a000.pcie-phy: Xtal is 40MHz
    mt7621-pci 1e140000.pcie: pcie0 no card, disable it (RST & CLK)
    [hangs]

    The wrong assumption is located in the 'mt7621_pcie_init_ports'
    function where we are just making a power off of the phy for
    slots 0 and 2 if nothing is connected in them. Hence, only
    poweroff the phy if nothing is connected in both slot 0 and
```

```
    slot 1 avoiding the kernel to hang.

    Fixes: 5737cfe87a9c ("staging: mt7621-pci: avoid to poweroff the phy for slot one")
    Signed-off-by: Sergio Paracuellos <sergio.paracuellos@gmail.com>
    Link: https://lore.kernel.org/r/20200409111652.30964-1-sergio.paracuellos@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 03f8898c93f2892577b28e5088434589ba95ebd4
Author: Nick Desaulniers <ndesaulniers@google.com>
Date:   Thu Jun 4 16:50:49 2020 -0700

    elfnote: mark all .note sections SHF_ALLOC

    commit 51da9dfb7f20911ae4e79e9b412a9c2d4c373d4b upstream.

    ELFNOTE_START allows callers to specify flags for .pushsection assembler
    directives.  All callsites but ELF_NOTE use "a" for SHF_ALLOC.  For vdso's
    that explicitly use ELF_NOTE_START and BUILD_SALT, the same section is
    specified twice after preprocessing, once with "a" flag, once without.
    Example:

    .pushsection .note.Linux, "a", @note ;
    .pushsection .note.Linux, "", @note ;

    While GNU as allows this ordering, it warns for the opposite ordering,
    making these directives position dependent.  We'd prefer not to precisely
    match this behavior in Clang's integrated assembler.  Instead, the non
    __ASSEMBLY__ definition of ELF_NOTE uses
    __attribute__((section(".note.Linux"))) which is created with SHF_ALLOC,
    so let's make the __ASSEMBLY__ definition of ELF_NOTE consistent with C
    and just always use "a" flag.

    This allows Clang to assemble a working mainline (5.6) kernel via:
    $ make CC=clang AS=clang

    Signed-off-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Reviewed-by: Fangrui Song <maskray@google.com>
    Cc: Jeremy Fitzhardinge <jeremy@goop.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vincenzo Frascino <vincenzo.frascino@arm.com>
    Link: https://github.com/ClangBuiltLinux/linux/issues/913
    Link: http://lkml.kernel.org/r/20200325231250.99205-1-ndesaulniers@google.com
    Debugged-by: Ilie Halip <ilie.halip@gmail.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Jian Cai <jiancai@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 379efd7d060a5222fd437446b788a51d68cc86e2
Author: Tuong Lien <tuong.t.lien@dektech.com.au>
Date:   Wed Jun 3 12:06:01 2020 +0700

    tipc: fix NULL pointer dereference in streaming

    [ Upstream commit 5e9eeccc58f3e6bcc99b929670665d2ce047e9c9 ]

    syzbot found the following crash:

    general protection fault, probably for non-canonical address 0xdffffc0000000019: 0000
[#1] PREEMPT SMP KASAN
    KASAN: null-ptr-deref in range [0x00000000000000c8-0x00000000000000cf]
    CPU: 1 PID: 7060 Comm: syz-executor394 Not tainted 5.7.0-rc6-syzkaller #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
```

```
 01/01/2011
    RIP: 0010:__tipc_sendstream+0xbde/0x11f0 net/tipc/socket.c:1591
    Code: 00 00 00 00 48 39 5c 24 28 48 0f 44 d8 e8 fa 3e db f9 48 b8 00 00 00 00 00 fc ff
 df 48 8d bb c8 00 00 00 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 e2 04 00 00 48 8b 9b c8 00
 00 00 48 b8 00 00 00
    RSP: 0018:ffffc90003ef7818 EFLAGS: 00010202
    RAX: dfffc0000000000 RBX: 0000000000000000 RCX: ffffffff8797fd9d
    RDX: 0000000000000019 RSI: ffffffff8797fde6 RDI: 00000000000000c8
    RBP: ffff888099848040 R08: ffff88809a5f6440 R09: fffffbfff1860b4c
    R10: ffffffff8c305a5f R11: fffffbfff1860b4b R12: ffff88809984857e
    R13: 0000000000000000 R14: ffff888086aa4000 R15: 0000000000000000
    FS:  00000000009b4880(0000) GS:ffff8880ae700000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 0000000020000140 CR3: 00000000a7fdf000 CR4: 00000000001406e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     tipc_sendstream+0x4c/0x70 net/tipc/socket.c:1533
     sock_sendmsg_nosec net/socket.c:652 [inline]
     sock_sendmsg+0xcf/0x120 net/socket.c:672
     ____sys_sendmsg+0x32f/0x810 net/socket.c:2352
     ___sys_sendmsg+0x100/0x170 net/socket.c:2406
     __sys_sendmmsg+0x195/0x480 net/socket.c:2496
     __do_sys_sendmmsg net/socket.c:2525 [inline]
     __se_sys_sendmmsg net/socket.c:2522 [inline]
     __x64_sys_sendmmsg+0x99/0x100 net/socket.c:2522
     do_syscall_64+0xf6/0x7d0 arch/x86/entry/common.c:295
     entry_SYSCALL_64_after_hwframe+0x49/0xb3
    RIP: 0033:0x440199
    ...

    This bug was bisected to commit 0a3e060f340d ("tipc: add test for Nagle
    algorithm effectiveness"). However, it is not the case, the trouble was
    from the base in the case of zero data length message sending, we would
    unexpectedly make an empty 'txq' queue after the 'tipc_msg_append()' in
    Nagle mode.

    A similar crash can be generated even without the bisected patch but at
    the link layer when it accesses the empty queue.

    We solve the issues by building at least one buffer to go with socket's
    header and an optional data section that may be empty like what we had
    with the 'tipc_msg_build()'.

    Note: the previous commit 4c21daae3dbc ("tipc: Fix NULL pointer
    dereference in __tipc_sendstream()") is obsoleted by this one since the
    'txq' will be never empty and the check of 'skb != NULL' is unnecessary
    but it is safe anyway.

    Reported-by: syzbot+8eac6d030e7807c21d32@syzkaller.appspotmail.com
    Fixes: c0bceb97db9e ("tipc: add smart nagle feature")
    Acked-by: Jon Maloy <jmaloy@redhat.com>
    Signed-off-by: Tuong Lien <tuong.t.lien@dektech.com.au>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 6e2e663b0dc2acf61bd1653635ae9cfa404d5dae
 Author: Michal Vokáč <michal.vokac@ysoft.com>
 Date:   Wed Jun 3 13:31:39 2020 +0200

    net: dsa: qca8k: Fix "Unexpected gfp" kernel exception

    [ Upstream commit 67122a7910bf2135dc7f7ececfcf16a5bdb362c1 ]

    Commit 7e99e3470172 ("net: dsa: remove dsa_switch_alloc helper")
```

```
    replaced the dsa_switch_alloc helper by devm_kzalloc in all DSA
    drivers. Unfortunately it introduced a typo in qca8k.c driver and
    wrong argument is passed to the devm_kzalloc function.

    This fix mitigates the following kernel exception:

      Unexpected gfp: 0x6 (__GFP_HIGHMEM|GFP_DMA32). Fixing up to gfp: 0x101
(GFP_DMA|__GFP_ZERO). Fix your code!
      CPU: 1 PID: 44 Comm: kworker/1:1 Not tainted 5.5.9-yocto-ua #1
      Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
      Workqueue: events deferred_probe_work_func
      [<c0014924>] (unwind_backtrace) from [<c00123bc>] (show_stack+0x10/0x14)
      [<c00123bc>] (show_stack) from [<c04c8fb4>] (dump_stack+0x90/0xa4)
      [<c04c8fb4>] (dump_stack) from [<c00e1b10>] (new_slab+0x20c/0x214)
      [<c00e1b10>] (new_slab) from [<c00e1cd0>] (___slab_alloc.constprop.0+0x1b8/0x540)
      [<c00e1cd0>] (___slab_alloc.constprop.0) from [<c00e2074>]
(__slab_alloc.constprop.0+0x1c/0x24)
      [<c00e2074>] (__slab_alloc.constprop.0) from [<c00e4538>]
(__kmalloc_track_caller+0x1b0/0x298)
      [<c00e4538>] (__kmalloc_track_caller) from [<c02cccac>] (devm_kmalloc+0x24/0x70)
      [<c02cccac>] (devm_kmalloc) from [<c030d888>] (qca8k_sw_probe+0x94/0x1ac)
      [<c030d888>] (qca8k_sw_probe) from [<c0304788>] (mdio_probe+0x30/0x54)
      [<c0304788>] (mdio_probe) from [<c02c93bc>] (really_probe+0x1e0/0x348)
      [<c02c93bc>] (really_probe) from [<c02c9884>] (driver_probe_device+0x60/0x16c)
      [<c02c9884>] (driver_probe_device) from [<c02c7fb0>] (bus_for_each_drv+0x70/0x94)
      [<c02c7fb0>] (bus_for_each_drv) from [<c02c9708>] (__device_attach+0xb4/0x11c)
      [<c02c9708>] (__device_attach) from [<c02c8148>] (bus_probe_device+0x84/0x8c)
      [<c02c8148>] (bus_probe_device) from [<c02c8cec>] (deferred_probe_work_func+0x64/0x90)
      [<c02c8cec>] (deferred_probe_work_func) from [<c0033c14>]
(process_one_work+0x1d4/0x41c)
      [<c0033c14>] (process_one_work) from [<c00340a4>] (worker_thread+0x248/0x528)
      [<c00340a4>] (worker_thread) from [<c0039148>] (kthread+0x124/0x150)
      [<c0039148>] (kthread) from [<c00090d8>] (ret_from_fork+0x14/0x3c)
      Exception stack(0xee1b5fb0 to 0xee1b5ff8)
      5fa0:                                     00000000 00000000 00000000 00000000
      5fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      5fe0: 00000000 00000000 00000000 00000000 00000013 00000000
      qca8k 2188000.ethernet-1:0a: Using legacy PHYLIB callbacks. Please migrate to PHYLINK!
      qca8k 2188000.ethernet-1:0a eth2 (uninitialized): PHY [2188000.ethernet-1:01] driver
[Generic PHY]
      qca8k 2188000.ethernet-1:0a eth1 (uninitialized): PHY [2188000.ethernet-1:02] driver
[Generic PHY]

    Fixes: 7e99e3470172 ("net: dsa: remove dsa_switch_alloc helper")
    Signed-off-by: Michal Vokáč <michal.vokac@ysoft.com>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 321b063c64e6a87997ae08dc0836464e0fd9add1
Author: Cong Wang <xiyou.wangcong@gmail.com>
Date:   Tue Jun 2 21:49:10 2020 -0700

    genetlink: fix memory leaks in genl_family_rcv_msg_dumpit()

    [ Upstream commit c36f05559104b66bcd7f617e931e38c680227b74 ]

    There are two kinds of memory leaks in genl_family_rcv_msg_dumpit():

    1. Before we call ops->start(), whenever an error happens, we forget
       to free the memory allocated in genl_family_rcv_msg_dumpit().

    2. When ops->start() fails, the 'info' has been already installed on
       the per socket control block, so we should not free it here. More
       importantly, nlk->cb_running is still false at this point, so
```

```
        netlink_sock_destruct() cannot free it either.

    The first kind of memory leaks is easier to resolve, but the second
    one requires some deeper thoughts.

    After reviewing how netfilter handles this, the most elegant solution
    I find is just to use a similar way to allocate the memory, that is,
    moving memory allocations from caller into ops->start(). With this,
    we can solve both kinds of memory leaks: for 1), no memory allocation
    happens before ops->start(); for 2), ops->start() handles its own
    failures and 'info' is installed to the socket control block only
    when success. The only ugliness here is we have to pass all local
    variables on stack via a struct, but this is not hard to understand.

    Alternatively, we can introduce a ops->free() to solve this too,
    but it is overkill as only genetlink has this problem so far.

    Fixes: 1927f41a22a0 ("net: genetlink: introduce dump info struct to be available during
dumpit op")
    Reported-by: syzbot+21f04f481f449c8db840@syzkaller.appspotmail.com
    Cc: "Jason A. Donenfeld" <Jason@zx2c4.com>
    Cc: Florian Westphal <fw@strlen.de>
    Cc: Pablo Neira Ayuso <pablo@netfilter.org>
    Cc: Jiri Pirko <jiri@mellanox.com>
    Cc: YueHaibing <yuehaibing@huawei.com>
    Cc: Shaochun Chen <cscnull@gmail.com>
    Signed-off-by: Cong Wang <xiyou.wangcong@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a5e0fee8ae617520f53626f5848ab0973e5e40ae
Author: Geliang Tang <geliangtang@gmail.com>
Date:   Mon Jun 8 18:47:54 2020 +0800

    mptcp: bugfix for RM_ADDR option parsing

    [ Upstream commit 8e60eed6b38e464e8c9d68f9caecafaa554dffe0 ]

    In MPTCPOPT_RM_ADDR option parsing, the pointer "ptr" pointed to the
    "Subtype" octet, the pointer "ptr+1" pointed to the "Address ID" octet:

      +-------+-------+---------------+
      |Subtype|(resvd)|   Address ID  |
      +-------+-------+---------------+
      |               |
     ptr            ptr+1

    We should set mp_opt->rm_id to the value of "ptr+1", not "ptr". This patch
    will fix this bug.

    Fixes: 3df523ab582c ("mptcp: Add ADD_ADDR handling")
    Signed-off-by: Geliang Tang <geliangtang@gmail.com>
    Reviewed-by: Matthieu Baerts <matthieu.baerts@tessares.net>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fe8ff6d927076e224aaf7810ea9b40be05b13d73
Author: Sameeh Jubran <sameehj@amazon.com>
Date:   Wed Jun 3 08:50:23 2020 +0000

    net: ena: xdp: update napi budget for DROP and ABORTED

    [ Upstream commit 3921a81c31df6057183aeb7f7d204003bf699d6f ]

    This patch fixes two issues with XDP:
```

```
    1. If the XDP verdict is XDP_ABORTED we break the loop, which results in
       us handling one buffer per napi cycle instead of the total budget
       (usually 64). To overcome this simply change the xdp_verdict check to
       != XDP_PASS. When the verdict is XDP_PASS, the skb is not expected to
       be NULL.

    2. Update the residual budget for XDP_DROP and XDP_ABORTED, since
       packets are handled in these cases.

    Fixes: 548c4940b9f1 ("net: ena: Implement XDP_TX action")
    Signed-off-by: Sameeh Jubran <sameehj@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 711f2c6c4969c093ec3bfccf9bc45e0d819c51f5
Author: Sameeh Jubran <sameehj@amazon.com>
Date:   Wed Jun 3 08:50:22 2020 +0000

    net: ena: xdp: XDP_TX: fix memory leak

    [ Upstream commit cd07ecccba13b8bd5023ffe7be57363d07e3105f ]

    When sending very high packet rate, the XDP tx queues can get full and
    start dropping packets. In this case we don't free the pages which
    results in ena driver draining the system memory.

    Fix:
    Simply free the pages when necessary.

    Fixes: 548c4940b9f1 ("net: ena: Implement XDP_TX action")
    Signed-off-by: Sameeh Jubran <sameehj@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ecf0b3c5a6684fcf27073365d576164390bc000e
Author: Ido Schimmel <idosch@mellanox.com>
Date:   Mon Jun 1 15:58:55 2020 +0300

    vxlan: Avoid infinite loop when suppressing NS messages with invalid options

    [ Upstream commit 8066e6b449e050675df48e7c4b16c29f00507ff0 ]

    When proxy mode is enabled the vxlan device might reply to Neighbor
    Solicitation (NS) messages on behalf of remote hosts.

    In case the NS message includes the "Source link-layer address" option
    [1], the vxlan device will use the specified address as the link-layer
    destination address in its reply.

    To avoid an infinite loop, break out of the options parsing loop when
    encountering an option with length zero and disregard the NS message.

    This is consistent with the IPv6 ndisc code and RFC 4886 which states
    that "Nodes MUST silently discard an ND packet that contains an option
    with length zero" [2].

    [1] https://tools.ietf.org/html/rfc4861#section-4.3
    [2] https://tools.ietf.org/html/rfc4861#section-4.6

    Fixes: 4b29dba9c085 ("vxlan: fix nonfunctional neigh_reduce()")
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Acked-by: Nikolay Aleksandrov <nikolay@cumulusnetworks.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit f3f4183f6d36df54f5a867653c30852ec6b5ab9d
Author: Ido Schimmel <idosch@mellanox.com>
Date:   Mon Jun 1 15:58:54 2020 +0300

    bridge: Avoid infinite loop when suppressing NS messages with invalid options

    [ Upstream commit 53fc685243bd6fb90d90305cea54598b78d3cbfc ]

    When neighbor suppression is enabled the bridge device might reply to
    Neighbor Solicitation (NS) messages on behalf of remote hosts.

    In case the NS message includes the "Source link-layer address" option
    [1], the bridge device will use the specified address as the link-layer
    destination address in its reply.

    To avoid an infinite loop, break out of the options parsing loop when
    encountering an option with length zero and disregard the NS message.

    This is consistent with the IPv6 ndisc code and RFC 4886 which states
    that "Nodes MUST silently discard an ND packet that contains an option
    with length zero" [2].

    [1] https://tools.ietf.org/html/rfc4861#section-4.3
    [2] https://tools.ietf.org/html/rfc4861#section-4.6

    Fixes: ed842faeb2bd ("bridge: suppress nd pkts on BR_NEIGH_SUPPRESS ports")
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Reported-by: Alla Segal <allas@mellanox.com>
    Tested-by: Alla Segal <allas@mellanox.com>
    Acked-by: Nikolay Aleksandrov <nikolay@cumulusnetworks.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ab5e1d8d91872d6d80119a560255ff549985cff9
Author: Willem de Bruijn <willemb@google.com>
Date:   Sat May 30 15:41:31 2020 -0400

    tun: correct header offsets in napi frags mode

    [ Upstream commit 96aa1b22bd6bb9fccf62f6261f390ed6f3e7967f ]

    Tun in IFF_NAPI_FRAGS mode calls napi_gro_frags. Unlike netif_rx and
    netif_gro_receive, this expects skb->data to point to the mac layer.

    But skb_probe_transport_header, __skb_get_hash_symmetric, and
    xdp_do_generic in tun_get_user need skb->data to point to the network
    header. Flow dissection also needs skb->protocol set, so
    eth_type_trans has to be called.

    Ensure the link layer header lies in linear as eth_type_trans pulls
    ETH_HLEN. Then take the same code paths for frags as for not frags.
    Push the link layer header back just before calling napi_gro_frags.

    By pulling up to ETH_HLEN from frag0 into linear, this disables the
    frag0 optimization in the special case when IFF_NAPI_FRAGS is used
    with zero length iov[0] (and thus empty skb->linear).

    Fixes: 90e33d459407 ("tun: enable napi_gro_frags() for TUN/TAP driver")
    Signed-off-by: Willem de Bruijn <willemb@google.com>
    Acked-by: Petar Penkov <ppenkov@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bc0a3f58dfce1a14268e8c6e7b13b4871896c280
```

```
Author: Vasily Averin <vvs@virtuozzo.com>
Date:   Tue Jun 2 15:55:26 2020 +0300

    net_failover: fixed rollback in net_failover_open()

    [ Upstream commit e8224bfe77293494626f6eec1884fee7b87d0ced ]

    found by smatch:
    drivers/net/net_failover.c:65 net_failover_open() error:
     we previously assumed 'primary_dev' could be null (see line 43)

    Fixes: cfc80d9a1163 ("net: Introduce net_failover driver")
    Signed-off-by: Vasily Averin <vvs@virtuozzo.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 47f546cfb4b23b5a8dd94b41c422fd6ef6fde9ab
Author: Vadim Pasternak <vadimp@mellanox.com>
Date:   Sun Jun 7 11:10:27 2020 +0300

    mlxsw: core: Use different get_trend() callbacks for different thermal zones

    [ Upstream commit 2dc2f760052da4925482ecdcdc5c94d4a599153c ]

    The driver registers three different types of thermal zones: For the
    ASIC itself, for port modules and for gearboxes.

    Currently, all three types use the same get_trend() callback which does
    not work correctly for the ASIC thermal zone. The callback assumes that
    the device data is of type 'struct mlxsw_thermal_module', whereas for
    the ASIC thermal zone 'struct mlxsw_thermal' is passed as device data.

    Fix this by using one get_trend() callback for the ASIC thermal zone and
    another for the other two types.

    Fixes: 6f73862fabd9 ("mlxsw: core: Add the hottest thermal zone detection")
    Signed-off-by: Vadim Pasternak <vadimp@mellanox.com>
    Reviewed-by: Jiri Pirko <jiri@mellanox.com>
    Signed-off-by: Ido Schimmel <idosch@mellanox.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b24e451cfb8c33ef5b8b4a80e232706b089914fb
Author: Hangbin Liu <liuhangbin@gmail.com>
Date:   Mon Jun 1 11:55:03 2020 +0800

    ipv6: fix IPV6_ADDRFORM operation logic

    [ Upstream commit 79a1f0ccdbb4ad700590f61b00525b390cb53905 ]

    Socket option IPV6_ADDRFORM supports UDP/UDPLITE and TCP at present.
    Previously the checking logic looks like:
    if (sk->sk_protocol == IPPROTO_UDP || sk->sk_protocol == IPPROTO_UDPLITE)
            do_some_check;
    else if (sk->sk_protocol != IPPROTO_TCP)
            break;

    After commit b6f6118901d1 ("ipv6: restrict IPV6_ADDRFORM operation"), TCP
    was blocked as the logic changed to:
    if (sk->sk_protocol == IPPROTO_UDP || sk->sk_protocol == IPPROTO_UDPLITE)
            do_some_check;
    else if (sk->sk_protocol == IPPROTO_TCP)
            do_some_check;
            break;
    else
```

```
                break;

        Then after commit 82c9ae440857 ("ipv6: fix restrict IPV6_ADDRFORM operation")
        UDP/UDPLITE were blocked as the logic changed to:
        if (sk->sk_protocol == IPPROTO_UDP || sk->sk_protocol == IPPROTO_UDPLITE)
                do_some_check;
        if (sk->sk_protocol == IPPROTO_TCP)
                do_some_check;

        if (sk->sk_protocol != IPPROTO_TCP)
                break;

        Fix it by using Eric's code and simply remove the break in TCP check, which
        looks like:
        if (sk->sk_protocol == IPPROTO_UDP || sk->sk_protocol == IPPROTO_UDPLITE)
                do_some_check;
        else if (sk->sk_protocol == IPPROTO_TCP)
                do_some_check;
        else
                break;

        Fixes: 82c9ae440857 ("ipv6: fix restrict IPV6_ADDRFORM operation")
        Signed-off-by: Hangbin Liu <liuhangbin@gmail.com>
        Signed-off-by: David S. Miller <davem@davemloft.net>
        Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```