```
commit 98e4b991db5ac04be8c23cb2dec277ee31fd22f3
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri May 31 06:43:59 2019 -0700

    Linux 5.1.6

commit 28ed91e0375f1c64384be5e3b275b08edd47489d
Author: Benjamin Coddington <bcodding@redhat.com>
Date:   Thu May 9 07:25:21 2019 -0400

    NFS: Fix a double unlock from nfs_match,get_client

    [ Upstream commit c260121a97a3e4df6536edbc2f26e166eff370ce ]

    Now that nfs_match_client drops the nfs_client_lock, we should be
    careful
    to always return it in the same condition: locked.

    Fixes: 950a578c6128 ("NFS: make nfs_match_client killable")
    Reported-by: syzbot+228a82b263b5da91883d@syzkaller.appspotmail.com
    Signed-off-by: Benjamin Coddington <bcodding@redhat.com>
    Signed-off-by: Anna Schumaker <Anna.Schumaker@Netapp.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c1ad36e9b2a16f322f396b418089f6ebeb3d5938
Author: Maxime Ripard <maxime.ripard@bootlin.com>
Date:   Mon Feb 11 15:41:24 2019 +0100

    drm/sun4i: dsi: Enforce boundaries on the start delay

    [ Upstream commit efa31801203ac2f5c6a82a28cb991c7163ee0f1d ]

    The Allwinner BSP makes sure that we don't end up with a null start delay
    or with a delay larger than vtotal.

    The former condition is likely to happen now with the reworked start delay,
    so make sure we enforce the same boundaries.

    Signed-off-by: Maxime Ripard <maxime.ripard@bootlin.com>
    Reviewed-by: Paul Kocialkowski <paul.kocialkowski@bootlin.com>
    Link:
https://patchwork.freedesktop.org/patch/msgid/c9889cf5f7a3d101ef380905900b45a182596f56.15498
96081.git-series.maxime.ripard@bootlin.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2cfe5e42e054fcd5c779441774b51b87c0d366ce
Author: Brett Creeley <brett.creeley@intel.com>
Date:   Wed Feb 13 10:51:14 2019 -0800

    ice: Put __ICE_PREPARED_FOR_RESET check in ice_prepare_for_reset

    [ Upstream commit 5abac9d7e1bb9a373673811154774d4c89a7f85e ]

    Currently we check if the __ICE_PREPARED_FOR_RESET bit is set prior to
    calling ice_prepare_for_reset in ice_reset_subtask(), but we aren't
    checking that bit in ice_do_reset() before calling
    ice_prepare_for_reset(). This is not consistent and can cause issues if
    ice_prepare_for_reset() is called prior to ice_do_reset(). Fix this by
    checking if the __ICE_PREPARED_FOR_RESET bit is set internal to
    ice_prepare_for_reset().

    Signed-off-by: Brett Creeley <brett.creeley@intel.com>
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
```

```
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 974315320634d255f13b14c17493f1bd8ff6dad3
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Thu Apr 4 22:17:48 2019 +0800

    extcon: axp288: Add a depends on ACPI to the Kconfig entry

    [ Upstream commit fa3c098c2d52a268f6372fa053932e11f50cecb1 ]

    As Hans de Goede pointed, using this driver without ACPI
    makes little sense, so add ACPI dependency to Kconfig entry
    to fix a build error while CONFIG_ACPI is not set.

    drivers/extcon/extcon-axp288.c: In function 'axp288_extcon_probe':
    drivers/extcon/extcon-axp288.c:363:20: error: dereferencing pointer to incomplete type
        put_device(&adev->dev);

    Fixes: 0cf064db948a ("extcon: axp288: Convert to use acpi_dev_get_first_match_dev()")
    Reported-by: Hulk Robot <hulkci@huawei.com>
    Suggested-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Reviewed-by: Mukesh Ojha <mojha@codeaurora.org>
    Signed-off-by: Chanwoo Choi <cw00.choi@samsung.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit db08b2bf67a5510f79f2bbdafe7e7e791343b47b
Author: Farhan Ali <alifm@linux.ibm.com>
Date:   Tue Apr 16 17:23:14 2019 -0400

    vfio-ccw: Prevent quiesce function going into an infinite loop

    [ Upstream commit d1ffa760d22aa1d8190478e5ef555c59a771db27 ]

    The quiesce function calls cio_cancel_halt_clear() and if we
    get an -EBUSY we go into a loop where we:
            - wait for any interrupts
            - flush all I/O in the workqueue
            - retry cio_cancel_halt_clear

    During the period where we are waiting for interrupts or
    flushing all I/O, the channel subsystem could have completed
    a halt/clear action and turned off the corresponding activity
    control bits in the subchannel status word. This means the next
    time we call cio_cancel_halt_clear(), we will again start by
    calling cancel subchannel and so we can be stuck between calling
    cancel and halt forever.

    Rather than calling cio_cancel_halt_clear() immediately after
    waiting, let's try to disable the subchannel. If we succeed in
    disabling the subchannel then we know nothing else can happen
    with the device.

    Suggested-by: Eric Farman <farman@linux.ibm.com>
    Signed-off-by: Farhan Ali <alifm@linux.ibm.com>
    Message-Id:
<4d5a4b98ab1b41ac6131b5c36de18b76c5d66898.1555449329.git.alifm@linux.ibm.com>
    Reviewed-by: Eric Farman <farman@linux.ibm.com>
    Acked-by: Halil Pasic <pasic@linux.ibm.com>
    Signed-off-by: Cornelia Huck <cohuck@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 62431eed2f2514028ba600cb7142da519098d0b8
```

```
Author: Maxime Ripard <maxime.ripard@bootlin.com>
Date:   Mon Feb 11 15:41:22 2019 +0100

    drm/sun4i: dsi: Restrict DSI tcon clock divider

    [ Upstream commit 85fb352666732a9e5caf6027b9c253b3d7881d8f ]

    The current code allows the TCON clock divider to have a range between 4
    and 127 when feeding the DSI controller.

    The only display supported so far had a display clock rate that ended up
    using a divider of 4, but testing with other displays show that only 4
    seems to be functional.

    This also aligns with what Allwinner is doing in their BSP, so let's just
    hardcode that we want a divider of 4 when using the DSI output.

    Reviewed-by: Paul Kocialkowski <paul.kocialkowski@bootlin.com>
    Signed-off-by: Maxime Ripard <maxime.ripard@bootlin.com>
    Link:
https://patchwork.freedesktop.org/patch/msgid/074e88ae472f5e0492e26939c74b44fb4125ffbd.15498
96081.git-series.maxime.ripard@bootlin.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 178fec03ed71f769b75ee90e64ce2f01707780f1
Author: Maxime Ripard <maxime.ripard@bootlin.com>
Date:   Mon Feb 11 15:41:23 2019 +0100

    drm/sun4i: dsi: Change the start delay calculation

    [ Upstream commit da676c6aa6413d59ab0a80c97bbc273025e640b2 ]

    The current calculation for the video start delay in the current DSI driver
    is that it is the total vertical size, minus the front porch and sync length,
    plus 1. This equals to the active vertical size plus the back porch plus 1.

    That 1 is coming in the Allwinner BSP from an variable that is set to 1.
    However, if we look at the Allwinner BSP more closely, and especially in
    the "legacy" code for the display (in drivers/video/sunxi/legacy/), we can
    see that this variable is actually computed from the porches and the sync
    minus 10, clamped between 8 and 100.

    This fixes the start delay symptom we've seen on some panels (vblank
    timeouts with vertical white stripes at the bottom of the panel).

    Reviewed-by: Paul Kocialkowski <paul.kocialkowski@bootlin.com>
    Signed-off-by: Maxime Ripard <maxime.ripard@bootlin.com>
    Link:
https://patchwork.freedesktop.org/patch/msgid/6e5f72e68f47ca0223877464bf12f0c3f3978de8.15498
96081.git-series.maxime.ripard@bootlin.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1ea5230de62b845f0af46a2939294bf85e49a7b
Author: Chris Wilson <chris@chris-wilson.co.uk>
Date:   Fri Aug 4 09:23:28 2017 +0100

    drm: Wake up next in drm_read() chain if we are forced to putback the event

    [ Upstream commit 60b801999c48b6c1dd04e653a38e2e613664264e ]

    After an event is sent, we try to copy it into the user buffer of the
    first waiter in drm_read() and if the user buffer doesn't have enough
    room we put it back onto the list. However, we didn't wake up any
    subsequent waiter, so that event may sit on the list until either a new
    vblank event is sent or a new waiter appears. Rare, but in the worst
```

```
    case may lead to a stuck process.

    Testcase: igt/drm_read/short-buffer-wakeup
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: Daniel Vetter <daniel.vetter@ffwll.ch>
    Reviewed-by: Ville Syrjälä <ville.syrjala@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20170804082328.17173-1-chris@chris-
wilson.co.uk
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa5a0c0b0bb2e9563da7915b4cae2546c176b318
Author: Noralf Trønnes <noralf@tronnes.org>
Date:   Mon Feb 25 15:42:26 2019 +0100

    drm/drv: Hold ref on parent device during drm_device lifetime

    [ Upstream commit 56be6503aab2bc3a30beae408071b9be5e1bae51 ]

    This makes it safe to access drm_device->dev after the parent device has
    been removed/unplugged.

    Signed-off-by: Noralf Trønnes <noralf@tronnes.org>
    Reviewed-by: Gerd Hoffmann <kraxel@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190225144232.20761-2-
noralf@tronnes.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 234b70f0f8f83a34bcb2e1eb3218d82ef449b7e2
Author: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
Date:   Wed Feb 6 10:18:17 2019 -0500

    drm/amd/display: Reset planes that were disabled in init_pipes

    [ Upstream commit 4bc46da4a3aeeb4d55e83dd276cf72756e908286 ]

    [Why]
    Seamless boot tries to reuse planes that were enabled for the first
    commit applied.

    In the case where Raven is booting with two monitors connected and the
    first commit contains two streams the screen corruption would occur
    because the second stream was trying to re-use a tg and plane that
    weren't previously enabled.

    The state on the first commit looks something like the following:

    TG0: enabled=1
    TG1: enabled=0
    TG2: enabled=0
    TG3: enabled=0

    New state: pipe=0, stream=0,    plane=0,       new_tg=0
    New state: pipe=1, stream=1,    plane=1,       new_tg=1
    New state: pipe=2, stream=NULL, plane=NULL,    new_tg=NULL
    New state: pipe=3, stream=NULL, plane=NULL,    new_tg=NULL

    Only one plane/tg is setup before we enter accelerated mode so
    we really want to disabling everything but that first plane.

    [How]

    Check if the stream is not NULL and if the tg is enabled before
    deciding whether to skip the plane disable.

    Also ensure we're also disabling on the current state's pipe_ctx so
```

```
    we don't overwrite the fields in the new pending state.

    Signed-off-by: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
    Reviewed-by: Anthony Koo <Anthony.Koo@amd.com>
    Acked-by: Harry Wentland <Harry.Wentland@amd.com>
    Acked-by: Leo Li <sunpeng.li@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0f0a54bd22977544aa5224d36870d282f5a8ee5a
Author: Anthony Koo <Anthony.Koo@amd.com>
Date:   Wed Feb 6 11:45:42 2019 -0500

    drm/amd/display: Fix exception from AUX acquire failure

    [ Upstream commit dcf1a988678e2e39ce2b4115b8ce14d208c8c481 ]

    [Why]
    AUX arbitration occurs between SW and FW components.
    When AUX acquire fails, it causes engine->ddc to be NULL,
    which leads to an exception when we try to release the AUX
    engine.

    [How]
    When AUX engine acquire fails, it should return from the
    function without trying to continue the operation.
    The upper level will determine if it wants to retry.
    i.e. dce_aux_transfer_with_retries will be used and retry.

    Signed-off-by: Anthony Koo <Anthony.Koo@amd.com>
    Reviewed-by: Aric Cyr <Aric.Cyr@amd.com>
    Acked-by: Leo Li <sunpeng.li@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 95be38a3ee7b14baeca5b7cfa539ba70bb8351ed
Author: Eric Anholt <eric@anholt.net>
Date:   Fri Mar 8 09:43:35 2019 -0800

    drm/v3d: Handle errors from IRQ setup.

    [ Upstream commit fc22771547e7e8a63679f0218e943d72b107de65 ]

    Noted in review by Dave Emett for V3D 4.2 support.

    Signed-off-by: Eric Anholt <eric@anholt.net>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190308174336.7866-1-
eric@anholt.net
    Reviewed-by: Dave Emett <david.emett@broadcom.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8cc1030b18718e5cae6c162f5759d051566459c
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 4 21:30:50 2019 +0100

    ASoC: ti: fix davinci_mcasp_probe dependencies

    [ Upstream commit 7d7b25d05ef1c5a1a9320190e1eeb55534847558 ]

    The SND_SOC_DAVINCI_MCASP driver can use either edma or sdma as
    a back-end, and it takes the presence of the respective dma engine
    drivers in the configuration as an indication to which ones should be
    built. However, this is flawed in multiple ways:

    - With CONFIG_TI_EDMA=m and CONFIG_SND_SOC_DAVINCI_MCASP=y,
```

```
      is enabled as =m, and we get a link error:
      sound/soc/ti/davinci-mcasp.o: In function `davinci_mcasp_probe':
      davinci-mcasp.c:(.text+0x930): undefined reference to `edma_pcm_platform_register'

    - When CONFIG_SND_SOC_DAVINCI_MCASP=m has already been selected by
      another driver, the same link error appears even if CONFIG_TI_EDMA
      is disabled

    There are possibly other issues here, but it seems that the only reasonable
    solution is to always build both SND_SOC_TI_EDMA_PCM and
    SND_SOC_TI_SDMA_PCM as a dependency here. Both are fairly small and
    do not have any other compile-time dependencies, so the cost is
    very small, and makes the configuration stage much more consistent.

    Fixes: f2055e145f29 ("ASoC: ti: Merge davinci and omap directories")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Peter Ujfalusi <peter.ujfalusi@ti.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 41d0c3bd6fc3dd7b802a236ac1853fa70a4466bb
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Thu Mar 7 11:11:30 2019 +0100

    ASoC: davinci-mcasp: Fix clang warning without CONFIG_PM

    [ Upstream commit 8ca5104715cfd14254ea5aecc390ae583b707607 ]

    Building with clang shows a variable that is only used by the
    suspend/resume functions but defined outside of their #ifdef block:

    sound/soc/ti/davinci-mcasp.c:48:12: error: variable 'context_regs' is not needed and
will not be emitted

    We commonly fix these by marking the PM functions as __maybe_unused,
    but here that would grow the davinci_mcasp structure, so instead
    add another #ifdef here.

    Fixes: 1cc0c054f380 ("ASoC: davinci-mcasp: Convert the context save/restore to use
array")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Peter Ujfalusi <peter.ujfalusi@ti.com>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7406043618a52bb5b92c80966b05c6a4fca4e0aa
Author: Chris Lesiak <chris.lesiak@licor.com>
Date:   Thu Mar 7 20:39:00 2019 +0000

    spi: Fix zero length xfer bug

    [ Upstream commit 5442dcaa0d90fc376bdfc179a018931a8f43dea4 ]

    This fixes a bug for messages containing both zero length and
    unidirectional xfers.

    The function spi_map_msg will allocate dummy tx and/or rx buffers
    for use with unidirectional transfers when the hardware can only do
    a bidirectional transfer.  That dummy buffer will be used in place
    of a NULL buffer even when the xfer length is 0.

    Then in the function __spi_map_msg, if he hardware can dma,
    the zero length xfer will have spi_map_buf called on the dummy
    buffer.
```

```
    Eventually, __sg_alloc_table is called and returns -EINVAL
    because nents == 0.

    This fix prevents the error by not using the dummy buffer when
    the xfer length is zero.

    Signed-off-by: Chris Lesiak <chris.lesiak@licor.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2ce9855716ac5b6d5b63a3f6397bb4d49bf73af2
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 14:59:59 2019 +0000

    regulator: da9055: Fix notifier mutex lock warning

    [ Upstream commit 5e6afb3832bedf420dd8e4c5b32ed85117c5087d ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: f6130be652d0 ("regulator: DA9055 regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c8da0508bbcbaef9e45a000546b2e1d9509713e7
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:19:34 2019 +0000

    regulator: da9062: Fix notifier mutex lock warning

    [ Upstream commit 978995def0f6030aa6b3b494682f673aca13881b ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 4068e5182ada ("regulator: da9062: DA9062 regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0cec666abc3a0cd2c6333080dad3e1730caa3bbf
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:46:55 2019 +0000

    regulator: pv88090: Fix notifier mutex lock warning

    [ Upstream commit 275513b7695a61b75b2546406ecd0f8e3d9fc8be ]
```

```
    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: c90456e36d9c ("regulator: pv88090: new regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0ee7dd93e3cde7033a3d2b73bf143883f95a130c
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:48:46 2019 +0000

    regulator: wm831x: Fix notifier mutex lock warning

    [ Upstream commit 119c4f5085c45b60cb23c5595e45d06135b89518 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: e4ee831f949a ("regulator: Add WM831x DC-DC buck convertor support")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Acked-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4e133564312fea05f3f7060c4ec6cc64ebd5515e
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:44:54 2019 +0000

    regulator: pv88080: Fix notifier mutex lock warning

    [ Upstream commit 1867af94cfdf37fc70fe67b3d522e78352800196 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 99cf3af5e2d5 ("regulator: pv88080: new regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a694126eeae02c47b306b0f17b670b9bd97ec2b4
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:23:02 2019 +0000
```

```
    regulator: da9063: Fix notifier mutex lock warning

    [ Upstream commit 29d40b4a5776ec4727c9f0e00a884423dd5e3366 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 69ca3e58d178 ("regulator: da9063: Add Dialog DA9063 voltage regulators support.")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bcffa6db5a0fd5f1ce79aa0ab888fd27d3f87939
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:27:57 2019 +0000

    regulator: da9211: Fix notifier mutex lock warning

    [ Upstream commit 65378de3359d30ebce44762d8b8027f372b5b1c4 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 1028a37daa14 ("regulator: da9211: new regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 28f59b9d1f00cbac7d451588f708b908bd70a8dc
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:32:18 2019 +0000

    regulator: lp8755: Fix notifier mutex lock warning

    [ Upstream commit 89b2758c192c35068b07766a6830433bfbdc1f44 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: b59320cc5a5e ("regulator: lp8755: new driver for LP8755")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 427b0ace3e68fc3750fed2912c4101725c43be6c
Author: Trent Piepho <tpiepho@impinj.com>
Date:   Mon Mar 4 20:18:49 2019 +0000

    spi: imx: stop buffer overflow in RX FIFO flush

    [ Upstream commit c842749ea1d32513f9e603c074d60d7aa07cb2ef ]

    Commit 71abd29057cb ("spi: imx: Add support for SPI Slave mode") added
    an RX FIFO flush before start of a transfer.  In slave mode, the master
    may have sent more data than expected and this data will still be in the
    RX FIFO at the start of the next transfer, and so needs to be flushed.

    However, the code to do the flush was accidentally saving this data into
    the previous transfer's RX buffer, clobbering the contents of whatever
    followed that buffer.

    Change it to empty the FIFO and throw away the data.  Every one of the
    RX functions for the different eCSPI versions and modes reads the RX
    FIFO data using the same readl() call, so just use that, rather than
    using the spi_imx->rx function pointer and making sure all the different
    rx functions have a working "throw away" mode.

    There is another issue, which affects master mode when switching from
    DMA to PIO.  There can be extra data in the RX FIFO which triggers this
    flush code, causing memory corruption in the same manner.  I don't know
    why this data is unexpectedly in the FIFO.  It's likely there is a
    different bug or erratum responsible for that.  But regardless of that,
    I think this is proper fix the for bug at hand here.

    Fixes: 71abd29057cb ("spi: imx: Add support for SPI Slave mode")
    Cc: Jiada Wang <jiada_wang@mentor.com>
    Cc: Fabio Estevam <festevam@gmail.com>
    Cc: Stefan Agner <stefan@agner.ch>
    Cc: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Trent Piepho <tpiepho@impinj.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc7faebca876740aaacc706249dd73a41d1bb6fc
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:42:50 2019 +0000

    regulator: pv88060: Fix notifier mutex lock warning

    [ Upstream commit f58213637206e190453e3bd91f98f535566290a3 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: f307a7e9b7af ("regulator: pv88060: new regulator driver")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 88b8cf5c24d50dca9e95631d9a843b5a0cffff58
Author: Steve Twiss <stwiss.opensource@diasemi.com>
```

```
Date:   Tue Feb 26 15:35:35 2019 +0000

    regulator: ltc3589: Fix notifier mutex lock warning

    [ Upstream commit f132da2534ec6599c78c4adcef15340cff2e9dd9 ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 3eb2c7ecb7ea ("regulator: Add LTC3589 support")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0e278b9d50c9f016d1557928d0d5b2cf5fb047f6
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:38:56 2019 +0000

    regulator: ltc3676: Fix notifier mutex lock warning

    [ Upstream commit 769fc8d4182c1d1875db7859852afeb436714c5c ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: 37b918a034fe ("regulator: Add LTC3676 support")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6c7f758513d5457b6cafac17eeaa6c08ffc78329
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:51:28 2019 +0000

    regulator: wm831x isink: Fix notifier mutex lock warning

    [ Upstream commit f7a621728a6a23bfd2c6ac4d3e42e1303aefde0f ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: d4d6b722e780 ("regulator: Add WM831x ISINK support")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Acked-by: Charles Keepax <ckeepax@opensource.cirrus.com>
```

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

commit fa49bf543095f6f8a2adb0424abb1b0d5348382e
Author: Steve Twiss <stwiss.opensource@diasemi.com>
Date:   Tue Feb 26 15:54:01 2019 +0000

    regulator: wm831x ldo: Fix notifier mutex lock warning

    [ Upstream commit 8be64b6d87bd47d81753b60ddafe70102ebfd76b ]

    The mutex for the regulator_dev must be controlled by the caller of
    the regulator_notifier_call_chain(), as described in the comment
    for that function.

    Failure to mutex lock and unlock surrounding the notifier call results
    in a kernel WARN_ON_ONCE() which will dump a backtrace for the
    regulator_notifier_call_chain() when that function call is first made.
    The mutex can be controlled using the regulator_lock/unlock() API.

    Fixes: d1c6b4fe668b ("regulator: Add WM831x LDO support")
    Suggested-by: Adam Thomson <Adam.Thomson.Opensource@diasemi.com>
    Signed-off-by: Steve Twiss <stwiss.opensource@diasemi.com>
    Acked-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a161fa8b7e03e48bfeec6dd3423b4e254031154
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Tue Mar 12 19:45:13 2019 +0100

    spi: rspi: Fix sequencer reset during initialization

    [ Upstream commit 26843bb128590edd7eba1ad7ce22e4b9f1066ce3 ]

    While the sequencer is reset after each SPI message since commit
    880c6d114fd79a69 ("spi: rspi: Add support for Quad and Dual SPI
    Transfers on QSPI"), it was never reset for the first message, thus
    relying on reset state or bootloader settings.

    Fix this by initializing it explicitly during configuration.

    Fixes: 0b2182ddac4b8837 ("spi: add support for Renesas RSPI")
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4f8a65de1560fc9d253ab128bc8415bbb5bb5900
Author: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
Date:   Thu Sep 13 03:45:06 2018 +0300

    drm/omap: Notify all devices in the pipeline of output disconnection

    [ Upstream commit 27a7e3e18419869cdcc414a404f3fe66f1b4e644 ]

    For HDMI pipelines, when the output gets disconnected the device
    handling CEC needs to be notified. Instead of guessing which device that
    would be (and sometimes getting it wrong), notify all devices in the
    pipeline.

    Signed-off-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Reviewed-by: Sebastian Reichel <sebastian.reichel@collabora.com>
    Tested-by: Sebastian Reichel <sebastian.reichel@collabora.com>
    Signed-off-by: Tomi Valkeinen <tomi.valkeinen@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit c3bb5b0f84f4f6485ce3f89a337860b8b34a388d
Author: Tony Lindgren <tony@atomide.com>
Date:   Thu Feb 7 07:45:16 2019 -0800

    drm/omap: dsi: Fix PM for display blank with paired dss_pll calls

    [ Upstream commit fe4ed1b457943113ee1138c939fbdeede4af6cf3 ]

    Currently dsi_display_init_dsi() calls dss_pll_enable() but it is not
    paired with dss_pll_disable() in dsi_display_uninit_dsi(). This leaves
    the DSS clocks enabled when the display is blanked wasting about extra
    5mW of power while idle.

    The clock that is left on by not calling dss_pll_disable() is
    DSS_CLKCTRL bit 10 OPTFCLKEN_SYS_CLK that is the source clock for
    DSI PLL.

    We can fix this issue by by making the current dsi_pll_uninit() into
    dsi_pll_disable(). This way we can just call dss_pll_disable() from
    dsi_display_uninit_dsi() and the code becomes a bit easier to follow.

    However, we need to also consider that DSI PLL can be muxed for DVI too
    as pointed out by Tomi Valkeinen <tomi.valkeinen@ti.com>. In the DVI
    case, we want to unconditionally disable the clocks. To get around this
    issue, we separate out the DSI lane handling from dsi_pll_enable() and
    dsi_pll_disable() as suggested by Tomi in an earlier experimental patch.

    So we must only toggle the DSI regulator based on the vdds_dsi_enabled
    flag from dsi_display_init_dsi() and dsi_display_uninit_dsi().

    We need to make these two changes together to avoid breaking things
    for DVI when fixing the DSI clock handling. And this all causes a
    slight renumbering of the error path for dsi_display_init_dsi().

    Suggested-by: Tomi Valkeinen <tomi.valkeinen@ti.com>
    Signed-off-by: Tony Lindgren <tony@atomide.com>
    Signed-off-by: Tomi Valkeinen <tomi.valkeinen@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7d188953f269b3cf708ff5b4e8c99682ccd255d1
Author: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
Date:   Thu Feb 21 02:51:37 2019 +0200

    drm: writeback: Fix leak of writeback job

    [ Upstream commit e482ae9b5fdc01a343f22f52930e85a6cfdf85eb ]

    Writeback jobs are allocated when the WRITEBACK_FB_ID is set, and
    deleted when the jobs complete. This results in both a memory leak of
    the job and a leak of the framebuffer if the atomic commit returns
    before the job is queued for processing, for instance if the atomic
    check fails or if the commit runs in test-only mode.

    Fix this by implementing the drm_writeback_cleanup_job() function and
    calling it from __drm_atomic_helper_connector_destroy_state(). As
    writeback jobs are removed from the state when they're queued for
    processing, any job left in the state when the state gets destroyed
    needs to be cleaned up.

    The existing declaration of the drm_writeback_cleanup_job() function
    without an implementation hints that this problem was considered, but
    never addressed.

    Signed-off-by: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
```

```
    Reviewed-by: Brian Starkey <brian.starkey@arm.com>
    Acked-by: Liviu Dudau <liviu.dudau@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ed5025fcfa0becb46a33fa47e0cc6f6349ac6522
Author: Aditya Pakki <pakki001@umn.edu>
Date:   Wed Mar 13 11:55:41 2019 -0500

    spi : spi-topcliff-pch: Fix to handle empty DMA buffers

    [ Upstream commit f37d8e67f39e6d3eaf4cc5471e8a3d21209843c6 ]

    pch_alloc_dma_buf allocated tx, rx DMA buffers which can fail. Further,
    these buffers are used without a check. The patch checks for these
    failures and sends the error upstream.

    Signed-off-by: Aditya Pakki <pakki001@umn.edu>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 51d1fd946444dfa718ba70dbaf788c161f27fded
Author: Li RongQing <lirongqing@baidu.com>
Date:   Thu Mar 7 09:16:24 2019 +0800

    audit: fix a memleak caused by auditing load module

    [ Upstream commit 95e0b46fcebd7dbf6850dee96046e4c4ddc7f69c ]

    module.name will be allocated unconditionally when auditing load
    module, and audit_log_start() can fail with other reasons, or
    audit_log_exit maybe not called, caused module.name is not freed

    so free module.name in audit_free_context and __audit_syscall_exit

    unreferenced object 0xffff88af90837d20 (size 8):
      comm "modprobe", pid 1036, jiffies 4294704867 (age 3069.138s)
      hex dump (first 8 bytes):
        69 78 67 62 65 00 ff ff                          ixgbe...
      backtrace:
        [<0000000008da28fe>] __audit_log_kern_module+0x33/0x80
        [<00000000c1491e61>] load_module+0x64f/0x3850
        [<000000007fc9ae3f>] __do_sys_init_module+0x218/0x250
        [<0000000000d4a478>] do_syscall_64+0x117/0x400
        [<000000004924ded8>] entry_SYSCALL_64_after_hwframe+0x49/0xbe
        [<000000007dc331dd>] 0xffffffffffffffff

    Fixes: ca86cad7380e3 ("audit: log module name on init_module")
    Signed-off-by: Zhang Yu <zhangyu31@baidu.com>
    Signed-off-by: Li RongQing <lirongqing@baidu.com>
    [PM: manual merge fixup in __audit_syscall_exit()]
    Signed-off-by: Paul Moore <paul@paul-moore.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dc974c86b8b3ac9560f1e5dfba79ff5ecbcdc3ea
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:06 2019 -0700

    scsi: lpfc: Fix use-after-free mailbox cmd completion

    [ Upstream commit 9b1640686470fbbd1c6efb35ada6fe1427ea8d0f ]

    When unloading the driver, mailbox commands may be sent without holding a
    reference on the ndlp. By the time the mailbox command completes, the ndlp
    may have reduced its ref counts and been freed.  The problem was reported
    by KASAN.
```

```
    While unregistering due to driver unload, have the completion noop'd by
    setting the ndlp context NULL'd. Due to the unload, no further action was
    necessary.  Also, while reviewing this path, the generic nulling of the
    context after handling should be slightly moved.

    Reported by: Bart Van Assche <bvanassche@acm.org>
    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: Sasha Levin <sashal@kernel.org>

commit 0c80098e7da110ce66f0e8d1ae4d2f3134b32586
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:05 2019 -0700

    scsi: lpfc: Resolve irq-unsafe lockdep heirarchy warning in lpfc_io_free

    [ Upstream commit 50e3f871fb20a9bb644743e2986e8f50f98a25bc ]

    A patch in the 12.2.0.0 set caused a new lockdep warning:

      WARNING: SOFTIRQ-safe -> SOFTIRQ-unsafe lock order detected
      5.0.0-rc8-next-20190301-dbg+ #1 Not tainted

      Possible interrupt unsafe locking scenario:
           CPU0                    CPU1
           ----                    ----
      lock(&(&qp->io_buf_list_put_lock)->rlock);
                                   local_irq_disable();
                                   lock(&(&phba->hbalock)->rlock);
                                   lock(&(&qp->io_buf_list_put_lock)->rlock);
      <Interrupt>
        lock(&(&phba->hbalock)->rlock);

    see: https://www.spinics.net/lists/linux-scsi/msg128389.html

    In summary, the new patch added taking the io_buf_list_put_lock while under
    an irq-disabled hbalock. This created a lock heirarchy dependent upon irq
    being disabled, and there are paths that take the io_buf_list_put_lock
    without disabling irq.

    Looking at the lpfc_io_free routine, which is where the new heirarchy was
    introduced, there is no reason to be taking out the hbalock and raising
    irq, as the functionality is replaced by the io_buf_list_xxx locks.

    Resolve by removing the hbalock/irq calls in lpfc_io_free.

    Fixes: 5e5b511d8bfa ("scsi: lpfc: Partition XRI buffer list across Hardware Queues")
    Reported-by: Bart Van Assche <bvanassche@acm.org>
    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: Sasha Levin <sashal@kernel.org>

commit 950d022e1f528198cfaebab866af1e5683b6e221
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:04 2019 -0700

    scsi: lpfc: Resolve inconsistent check of hdwq in lpfc_scsi_cmd_iocb_cmpl

    [ Upstream commit ff6bf89717b0dc7b8dd0934d1c065f29069831e7 ]

    A prior patch which added support for non-uniform allocation of MSIX
    vectors now causes a smatch complaint:
```

```
      drivers/scsi/lpfc/lpfc_scsi.c:3674 lpfc_scsi_cmd_iocb_cmpl()
        error: we previously assumed 'phba->sli4_hba.hdwq' could be
               null (see line 3667)

    Resolve by removing the unnecessary check for a NULL hdwq table.

    Fixes 6a828b0f6192: ("scsi: lpfc: Support non-uniform allocation of MSIX vectors to
hardware queues")
    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    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: Sasha Levin <sashal@kernel.org>

commit 13989452ec311891f8275ff3c41fed9e97395f41
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:18 2019 -0700

    scsi: lpfc: Fix mailbox hang on adapter init

    [ Upstream commit e8869f5b0a7273fcf20ef99066fd8129e58ba5b7 ]

    The adapter initialization sequence enables interrupts, initializes the
    adapter link_state to LINK_DOWN, then issues commands to initialize the
    adapter. The interrupt handler on the adapter validates the link_state (has
    to be at least LINK_DOWN) and if invalid, will discard the interrupting
    event.

    In most cases, there is not a command completion, thus an interrupt until
    the initialization commands have been sent which is post the setting of
    state to LINK_DOWN.  However, in cases of firmware reset, the reset will
    modify the link_state to an invalid value (indicating a reset of the
    adapter) and there occasionally are cases where the adapter will generate
    an asynchronous event which shares the eq/cq used for mailbox commands. In
    the failure case, an interrupt is generated immediately after enabling them
    due to the async event.  As link_state is invalid, the eq is list and the
    CQ not serviced.  At this point link_state is initialized and the mailbox
    command sent.  As the CQ has not been serviced, it is not armed, so no
    interrupt event is generated when the mailbox command completes.

    Modify the initialization sequence so that interrupts are enabled after
    link_state is properly initialized, which avoids the race condition with
    the async event.

    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: Sasha Levin <sashal@kernel.org>

commit 4b9e930ed2a1a828994f98cbe5e24beed17efa09
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:07 2019 -0700

    scsi: lpfc: Fix SLI3 commands being issued on SLI4 devices

    [ Upstream commit c95a3b4b0fb8d351e2329a96f87c4fc96a149505 ]

    During debug, it was seen that the driver is issuing commands specific to
    SLI3 on SLI4 devices. Although the adapter correctly rejected the command,
    this should not be done.

    Revise the code to stop sending these commands on a SLI4 adapter.

    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: Sasha Levin <sashal@kernel.org>

commit 8839f46b034f90f7f86ed2324178cbdc91206792
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Feb 19 12:01:56 2019 -0500

    media: saa7146: avoid high stack usage with clang

    [ Upstream commit 03aa4f191a36f33fce015387f84efa0eee94408e ]

    Two saa7146/hexium files contain a construct that causes a warning
    when built with clang:

    drivers/media/pci/saa7146/hexium_orion.c:210:12: error: stack frame size of 2272 bytes
in function 'hexium_probe'
          [-Werror,-Wframe-larger-than=]
    static int hexium_probe(struct saa7146_dev *dev)
               ^
    drivers/media/pci/saa7146/hexium_gemini.c:257:12: error: stack frame size of 2304 bytes
in function 'hexium_attach'
          [-Werror,-Wframe-larger-than=]
    static int hexium_attach(struct saa7146_dev *dev, struct saa7146_pci_extension_data
*info)
               ^

    This one happens regardless of KASAN, and the problem is that a
    constructor to initialize a dynamically allocated structure leads
    to a copy of that structure on the stack, whereas gcc initializes
    it in place.

    Link: https://bugs.llvm.org/show_bug.cgi?id=40776

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    [hverkuil-cisco@xs4all.nl: fix checkpatch warnings]
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a6cfca977b060af120090191ca48ea17d7d2518
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:19 2019 -0700

    scsi: lpfc: Fix io lost on host resets

    [ Upstream commit c66a91974634bfdf9d8e8736219d3b27621fa704 ]

    If the driver undergoes repeated host resets it starts losing exchange
    structures and eventually returns SCSI_MLQUEUE_HOST_BUSY and does not
    recover. The offline path is not reclaiming the outstanding ios on the fcp
    pring txcmplq before calling lpfc_destroy_multixripool, which causes the
    txmcplq to be reinit and the resources lost.

    Flush the fcp rings before destroying the multixripools.

    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: Sasha Levin <sashal@kernel.org>

commit 19ac2cd727d2cb936a16ef18885677a53b963685
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:21 2019 -0700
```

```
    scsi: lpfc: Fix fc4type information for FDMI

    [ Upstream commit 32a80c093b524a0682f1c6166c910387b116ffce ]

    The driver is reporting support for NVME even when not configured for NVME
    operation.

    Fix (and make more readable) when NVME protocol support is indicated.

    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: Sasha Levin <sashal@kernel.org>

commit 790395a256d67fe8505c504d2b0c274d99ad5a17
Author: James Smart <jsmart2021@gmail.com>
Date:   Tue Mar 12 16:30:20 2019 -0700

    scsi: lpfc: Fix FDMI manufacturer attribute value

    [ Upstream commit d67f935b79a76ac9d86dde1a27bdd413feb5d987 ]

    The FDMI manufacturer value being reported on Linux is inconsistent with
    other OS's.

    Set the value to "Emulex Corporation" for consistency.

    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: Sasha Levin <sashal@kernel.org>

commit 8075996f87f3637a57a756d38acd4f4b745c15e0
Author: Jernej Skrabec <jernej.skrabec@siol.net>
Date:   Mon Jan 28 15:55:00 2019 -0500

    media: cedrus: Add a quirk for not setting DMA offset

    [ Upstream commit 70a4f5cda82f7197c350099b66fd23506620810e ]

    H6 VPU doesn't work if DMA offset is set.

    Add a quirk for it.

    Signed-off-by: Jernej Skrabec <jernej.skrabec@siol.net>
    Acked-by: Maxime Ripard <maxime.ripard@bootlin.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9fcd2e9fe4521ad4d3c21074b603cae8f0a61203
Author: Hans Verkuil <hverkuil-cisco@xs4all.nl>
Date:   Thu Feb 21 08:35:15 2019 -0500

    media: vim2m: replace devm_kzalloc by kzalloc

    [ Upstream commit ea6c7e34f3b28e165988aa7391310752969842e8 ]

    It is not possible to use devm_kzalloc since that memory is
    freed immediately when the device instance is unbound.

    Various objects like the video device may still be in use
    since someone has the device node open, and when that is closed
    it expects the memory to be around.
```

```
    So use kzalloc and release it at the appropriate time.

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 47517a9f52a12f24d328677a2e3828accad887c5
Author: Hans Verkuil <hverkuil-cisco@xs4all.nl>
Date:   Fri Mar 8 08:02:26 2019 -0500

    media: vimc: zero the media_device on probe

    [ Upstream commit f74267b51cb36321f777807b2e04ca02167ecc08 ]

    The media_device is part of a static global vimc_device struct.
    The media framework expects this to be zeroed before it is
    used, however, since this is a global this is not the case if
    vimc is unbound and then bound again.

    So call memset to ensure any left-over values are cleared.

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 86a474e018e5ff0f9aa7fe482cf33850f5317eec
Author: Steve Longerbeam <slongerbeam@gmail.com>
Date:   Tue Feb 19 20:09:38 2019 -0500

    media: imx: vdic: Restore default case to prepare_vdi_in_buffers()

    [ Upstream commit ce3c2433b074eb9d569a0f63a15d6fd5dbc87f02 ]

    Restore a default case to prepare_vdi_in_buffers() to fix the following
    smatch errors:

    drivers/staging/media/imx/imx-media-vdic.c:236 prepare_vdi_in_buffers() error:
uninitialized symbol 'prev_phys'.
    drivers/staging/media/imx/imx-media-vdic.c:237 prepare_vdi_in_buffers() error:
uninitialized symbol 'curr_phys'.
    drivers/staging/media/imx/imx-media-vdic.c:238 prepare_vdi_in_buffers() error:
uninitialized symbol 'next_phys'.

    Fixes: 6e537b58de772 ("media: imx: vdic: rely on VDIC for correct field order")

    Reported-by: Hans Verkuil <hverkuil@xs4all.nl>
    Signed-off-by: Steve Longerbeam <slongerbeam@gmail.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 70bf8e42a59b76eedea2e129ce92be54434c530e
Author: Alexandre Courbot <acourbot@chromium.org>
Date:   Wed Mar 6 01:15:02 2019 -0500

    media: mtk-vcodec: fix access to vb2_v4l2_buffer struct

    [ Upstream commit 3235d3946429f64b19addfd89fc926a36eaec06a ]

    Commit 0650a91499e0 ("media: mtk-vcodec: Correct return type for mem2mem
    buffer helpers") fixed the return types for mem2mem buffer helper
    functions, but omitted two occurrences that are accessed in the
```

```
    mtk_v4l2_debug() macro. These only trigger compiler errors when DEBUG is
    defined.

    Fixes: 0650a91499e0 ("media: mtk-vcodec: Correct return type for mem2mem buffer
helpers")

    Signed-off-by: Alexandre Courbot <acourbot@chromium.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1fef8fe9d8ccc35341385efaf99c6b4e9be31fd1
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Feb 19 12:01:58 2019 -0500

    media: go7007: avoid clang frame overflow warning with KASAN

    [ Upstream commit ed713a4a1367aca5c0f2f329579465db00c17995 ]

    clang-8 warns about one function here when KASAN is enabled, even
    without the 'asan-stack' option:

    drivers/media/usb/go7007/go7007-fw.c:1551:5: warning: stack frame size of 2656 bytes in
function

    I have reported this issue in the llvm bugzilla, but to make
    it work with the clang-8 release, a small annotation is still
    needed.

    Link: https://bugs.llvm.org/show_bug.cgi?id=38809

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    [hverkuil-cisco@xs4all.nl: fix checkpatch warning]
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f5cbd513bb8b64e8a8bb350454fe7318021dfd68
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Fri Feb 22 09:50:03 2019 -0500

    media: vicodec: avoid clang frame size warning

    [ Upstream commit e855165f3dae6f71da859a5f00b85d5368641d61 ]

    Clang-9 makes some different inlining decisions compared to gcc, which
    leads to a warning about a possible stack overflow problem when building
    with CONFIG_KASAN, including when setting asan-stack=0, which avoids
    most other frame overflow warnings:

    drivers/media/platform/vicodec/codec-fwht.c:673:12: error: stack frame size of 2224
bytes in function 'encode_plane'

    Manually adding noinline_for_stack annotations in those functions
    called by encode_plane() or decode_plane() that require a significant
    amount of kernel stack makes this impossible to happen with any
    compiler.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6fb2ea5513053553956fd677b4a4265cba25f173
Author: Hans Verkuil <hverkuil-cisco@xs4all.nl>
```

```
Date:   Tue Feb 26 07:54:22 2019 -0500

    media: gspca: do not resubmit URBs when streaming has stopped

    [ Upstream commit e6f8bd59c28f758feea403a70d6c3ef28c50959f ]

    When streaming is stopped all URBs are killed, but in fill_frame and in
    bulk_irq this results in an attempt to resubmit the killed URB. That is
    not what you want and causes spurious kernel messages.

    So check if streaming has stopped before resubmitting.

    Also check against gspca_dev->streaming rather than vb2_start_streaming_called()
    since vb2_start_streaming_called() will return true when in stop_streaming,
    but gspca_dev->streaming is set to false when stop_streaming is called.

    Fixes: 6992effe5344 ("gspca: Kill all URBs before releasing any of them")

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fc816cea8c5b24d8953578c6d83e44aee320d0b6
Author: Helen Fornazier <helen.koike@collabora.com>
Date:   Wed Mar 6 17:42:38 2019 -0500

    media: vimc: stream: fix thread state before sleep

    [ Upstream commit 2978a505aaa981b279ef359f74ba93d25098e0a0 ]

    The state TASK_UNINTERRUPTIBLE should be set just before
    schedule_timeout() call, so it knows the sleep mode it should enter.
    There is no point in setting TASK_UNINTERRUPTIBLE at the initialization
    of the thread as schedule_timeout() will set the state back to
    TASK_RUNNING.

    This fixes a warning in __might_sleep() call, as it's expecting the
    task to be in TASK_RUNNING state just before changing the state to
    a sleeping state.

    Reported-by: Hans Verkuil <hverkuil@xs4all.nl>
    Signed-off-by: Helen Koike <helen.koike@collabora.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0a1d3b8e0aaf476f404b89fc92060be4eaa42f60
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Fri Mar 15 02:11:56 2019 -0500

    scsi: ufs: fix a missing check of devm_reset_control_get

    [ Upstream commit 63a06181d7ce169d09843645c50fea1901bc9f0a ]

    devm_reset_control_get could fail, so the fix checks its return value and
    passes the error code upstream in case it fails.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Acked-by: Avri Altman <avri.altman@wdc.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 97b73399f2f2daec358fd5eef7301ac4e3599703
Author: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
Date:   Fri Feb 8 13:21:05 2019 -0500
```

```
    drm/amd/display: Set stream->mode_changed when connectors change

    [ Upstream commit b9952f93cd2cf5fca82b06a8179c0f5f7b769e83 ]

    [Why]
    The kms_plane@plane-position-covered-pipe-*-planes subtests can produce
    a sequence of atomic commits such that neither active_changed nor
    mode_changed but connectors_changed.

    When this happens we remove the old stream from the context and add
    a new stream but the new stream doesn't have mode_changed=true set.

    This incorrect programming sequence causes CRC mismatches to occur in
    the test.

    The stream->mode_changed value should be set whenever a new stream
    is created.

    [How]
    A new stream is created whenever drm_atomic_crtc_needs_modeset is true.
    We previously covered the active_changed and mode_changed conditions
    for the CRTC but connectors_changed is also checked within
    drm_atomic_crtc_needs_modeset.

    So just use drm_atomic_crtc_needs_modeset directly to determine the
    mode_changed flag.

    Signed-off-by: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
    Reviewed-by: Sun peng Li <Sunpeng.Li@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2dc5c6d186b22d5528cff6b01daa72e184d66b4
Author: Martin Leung <martin.leung@amd.com>
Date:   Wed Feb 13 17:06:31 2019 -0500

    drm/amd/display: half bandwidth for YCbCr420 during validation

    [ Upstream commit 162f807858d15bde60cf373a3ad46e03200ad9d8 ]

    [Why]
    used to be unable to run 4:2:0 if using a dongle because 4k60 bandwidth
    exceeded dongle caps

    [How]
    half pixel clock during comparison to dongle cap. *Could get stuck on black
    screen on monitor that don't support 420 but will be selecting 420 as
    preferred mode*

    Signed-off-by: Martin Leung <martin.leung@amd.com>
    Reviewed-by: Wenjing Liu <Wenjing.Liu@amd.com>
    Acked-by: Aidan Wood <Aidan.Wood@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7dfc00de75c4d56aca8e8ffd67c3bf9b9682af20
Author: David Francis <David.Francis@amd.com>
Date:   Mon Feb 11 14:15:19 2019 -0500

    drm/amd/display: Re-add custom degamma support

    [ Upstream commit f91813992c343272813e707343b50f8d06383659 ]
```

```
    [Why]
    The dc_gamma_type CUSTOM_GAMMA is used to represent degamma
    mappings passed in by drm. This type of gamma must be interpolated
    into a transfer function by apply_1d_lut.  The line in
    mod_color_calculate_degamma_params that handled this case
    was erroneously removed.

    [How]
    For CUSTOM_GAMMA degamma, calculate the lut as before.

    Signed-off-by: David Francis <David.Francis@amd.com>
    Reviewed-by: Krunoslav Kovac <Krunoslav.Kovac@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e56c67edecf1ab2d9b024c123e84783f895e60fd
Author: Murton Liu <murton.liu@amd.com>
Date:   Fri Feb 15 15:05:43 2019 -0500

    drm/amd/display: Fix Divide by 0 in memory calculations

    [ Upstream commit 59979bf8be1784ebfc44215031c6c88ca22ae65d ]

    Check if we get any values equal to 0, and set to 1 if so.

    Signed-off-by: Murton Liu <murton.liu@amd.com>
    Reviewed-by: Aric Cyr <Aric.Cyr@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Acked-by: Sivapiriyan Kumarasamy <Sivapiriyan.Kumarasamy@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8f9123dc0d149753b3ef04ecbb6bbfb272c01d4e
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Mar 5 08:29:48 2019 -0500

    media: staging: davinci_vpfe: disallow building with COMPILE_TEST

    [ Upstream commit 49dc762cffd8305a861ca649e82dc5533b3e3344 ]

    The driver should really call dm365_isif_setup_pinmux() through a callback,
    but uses a hack to include a davinci specific machine header file when
    compile testing instead. This works almost everywhere, but not on the
    ARM omap1 platform, which has another header named mach/mux.h. This
    causes a build failure:

    drivers/staging/media/davinci_vpfe/dm365_isif.c:2028:2: error: implicit declaration of
function 'davinci_cfg_reg' [-Werror,-Wimplicit-function-declaration]
            davinci_cfg_reg(DM365_VIN_CAM_WEN);
            ^
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2028:2: error: this function declaration
is not a prototype [-Werror,-Wstrict-prototypes]
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2028:18: error: use of undeclared
identifier 'DM365_VIN_CAM_WEN'
            davinci_cfg_reg(DM365_VIN_CAM_WEN);
                            ^
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2029:18: error: use of undeclared
identifier 'DM365_VIN_CAM_VD'
            davinci_cfg_reg(DM365_VIN_CAM_VD);
                            ^
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2030:18: error: use of undeclared
identifier 'DM365_VIN_CAM_HD'
            davinci_cfg_reg(DM365_VIN_CAM_HD);
```

```
                            ^
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2031:18: error: use of undeclared
identifier 'DM365_VIN_YIN4_7_EN'
            davinci_cfg_reg(DM365_VIN_YIN4_7_EN);
                            ^
    drivers/staging/media/davinci_vpfe/dm365_isif.c:2032:18: error: use of undeclared
identifier 'DM365_VIN_YIN0_3_EN'
            davinci_cfg_reg(DM365_VIN_YIN0_3_EN);
                            ^
    7 errors generated.

    Exclude omap1 from compile-testing, under the assumption that all others
    still work.

    Fixes: 4907c73deefe ("media: staging: davinci_vpfe: allow building with COMPILE_TEST")

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 52faaa96bbc70917352f9205e695cb87a8dae1cb
Author: Wenjing Liu <Wenjing.Liu@amd.com>
Date:   Wed Feb 20 14:00:55 2019 -0500

    drm/amd/display: add pipe lock during stream update

    [ Upstream commit e6bddf6c67f9a3abf6f1ef75e52bc1cd228dfe4d ]

    [why]
    Stream update will adjust both info packets and stream params,
    need to make sure all things are applied togather.

    [how]
    add pipe lock during stream update

    Signed-off-by: Wenjing Liu <Wenjing.Liu@amd.com>
    Reviewed-by: Jun Lei <Jun.Lei@amd.com>
    Acked-by: Leo Li <sunpeng.li@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 706b5abbf18250fb5df2d034b7a41e95f581b6fc
Author: James Hutchinson <jahutchinson99@googlemail.com>
Date:   Sun Jan 13 16:13:47 2019 -0500

    media: m88ds3103: serialize reset messages in m88ds3103_set_frontend

    [ Upstream commit 981fbe3da20a6f35f17977453bce7dfc1664d74f ]

    Ref: https://bugzilla.kernel.org/show_bug.cgi?id=199323

    Users are experiencing problems with the DVBSky S960/S960C USB devices
    since the following commit:

    9d659ae: ("locking/mutex: Add lock handoff to avoid starvation")

    The device malfunctions after running for an indeterminable period of
    time, and the problem can only be cleared by rebooting the machine.

    It is possible to encourage the problem to surface by blocking the
    signal to the LNB.

    Further debugging revealed the cause of the problem.
```

```
    In the following capture:
    - thread #1325 is running m88ds3103_set_frontend
    - thread #42 is running ts2020_stat_work

    a> [1325] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 07 80
       [1325] usb 1-1: dvb_usb_v2_generic_io: <<< 08
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 09 01 01 68 3f
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 08 ff
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 03 11
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 09 01 01 60 3d
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07 ff
    b> [1325] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 07 00
       [1325] usb 1-1: dvb_usb_v2_generic_io: <<< 07
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 03 11
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 09 01 01 60 21
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07 ff
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 03 11
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07
       [42] usb 1-1: dvb_usb_v2_generic_io: >>> 09 01 01 60 66
       [42] usb 1-1: dvb_usb_v2_generic_io: <<< 07 ff
       [1325] usb 1-1: dvb_usb_v2_generic_io: >>> 08 68 02 03 11
       [1325] usb 1-1: dvb_usb_v2_generic_io: <<< 07
       [1325] usb 1-1: dvb_usb_v2_generic_io: >>> 08 60 02 10 0b
       [1325] usb 1-1: dvb_usb_v2_generic_io: <<< 07

    Two i2c messages are sent to perform a reset in m88ds3103_set_frontend:

      a. 0x07, 0x80
      b. 0x07, 0x00

    However, as shown in the capture, the regmap mutex is being handed over
    to another thread (ts2020_stat_work) in between these two messages.

    >From here, the device responds to every i2c message with an 07 message,
    and will only return to normal operation following a power cycle.

    Use regmap_multi_reg_write to group the two reset messages, ensuring
    both are processed before the regmap mutex is unlocked.

    Signed-off-by: James Hutchinson <jahutchinson99@googlemail.com>
    Reviewed-by: Antti Palosaari <crope@iki.fi>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9bfc9c3a8296e632e3d689cfd6a93ed7a62f22ed
Author: Stefan Brüns <stefan.bruens@rwth-aachen.de>
Date:   Sat Jan 19 20:30:04 2019 -0500

    media: dvbsky: Avoid leaking dvb frontend

    [ Upstream commit fdfa59cd63b184e1e96d51ff170fcac739bc6f6f ]

    Commit 14f4eaeddabc ("media: dvbsky: fix driver unregister logic") fixed
    a use-after-free by removing the reference to the frontend after deleting
    the backing i2c device.

    This has the unfortunate side effect the frontend device is never freed
    in the dvb core leaving a dangling device, leading to errors when the
    dvb core tries to register the frontend after e.g. a replug as reported
    here: https://www.spinics.net/lists/linux-media/msg138181.html

    media: dvbsky: issues with DVBSky T680CI
```

```
    ===
    [  561.119145] sp2 8-0040: CIMaX SP2 successfully attached
    [  561.119161] usb 2-3: DVB: registering adapter 0 frontend 0 (Silicon Labs
    Si2168)...
    [  561.119174] sysfs: cannot create duplicate filename '/class/dvb/
    dvb0.frontend0'
    ===

    The use after free happened as dvb_usbv2_disconnect calls in this order:
    - dvb_usb_device::props->exit(...)
    - dvb_usbv2_adapter_frontend_exit(...)
      + if (fe) dvb_unregister_frontend(fe)
      + dvb_usb_device::props->frontend_detach(...)

    Moving the release of the i2c device from exit() to frontend_detach()
    avoids the dangling pointer access and allows the core to unregister
    the frontend.

    This was originally reported for a DVBSky T680CI, but it also affects
    the MyGica T230C. As all supported devices structure the registration/
    unregistration identically, apply the change for all device types.

    Signed-off-by: Stefan Brüns <stefan.bruens@rwth-aachen.de>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c4d4895bb5e53883696ab98dd0b041b89eaa6a1b
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Thu Dec 20 23:54:03 2018 -0500

    media: si2165: fix a missing check of return value

    [ Upstream commit 0ab34a08812a3334350dbaf69a018ee0ab3d2ddd ]

    si2165_readreg8() may fail. Looking into si2165_readreg8(), we will find
    that "val_tmp" will be an uninitialized value when regmap_read() fails.
    "val_tmp" is then assigned to "val". So if si2165_readreg8() fails,
    "val" will be a random value. Further use will lead to undefined
    behaviors. The fix checks if si2165_readreg8() fails, and if so, returns
    its error code upstream.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Reviewed-by: Matthias Schwarzott <zzam@gentoo.org>
    Tested-by: Matthias Schwarzott <zzam@gentoo.org>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8725cb0b2841fe3bb92a92a9b676587301db44f0
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Tue Dec 11 15:59:38 2018 +0800

    igb: Exclude device from suspend direct complete optimization

    [ Upstream commit 5b6e13216be29ced7350d9c354a1af8fe0ad9a3e ]

    igb sets different WoL settings in system suspend callback and runtime
    suspend callback.

    The suspend direct complete optimization leaves igb in runtime suspended
    state with wrong WoL setting during system suspend.

    To fix this, we need to disable suspend direct complete optimization to
```

```
    let igb always use suspend callback to set correct WoL during system
    suspend.

    Signed-off-by: Kai-Heng Feng <kai.heng.feng@canonical.com>
    Tested-by: Aaron Brown <aaron.f.brown@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 14edc56d71ca33cf9effadb490de5bc5a35be1b8
Author: Noralf Trønnes <noralf@tronnes.org>
Date:   Fri Feb 22 13:43:29 2019 +0100

    tinydrm/mipi-dbi: Use dma-safe buffers for all SPI transfers

    [ Upstream commit a89bfc5d9a0732d84b4de311e27133daa0586316 ]

    Buffers passed to spi_sync() must be dma-safe even for tiny buffers since
    some SPI controllers use DMA for all transfers.

    Example splat with CONFIG_DMA_API_DEBUG enabled:

    [   23.750467] DMA-API: dw_dmac_pci 0000:00:15.0: device driver maps memory from stack
[probable addr=000000001e49185d]
    [   23.750529] WARNING: CPU: 1 PID: 1296 at kernel/dma/debug.c:1161
check_for_stack+0xb7/0x190
    [   23.750533] Modules linked in: mmc_block(+) spi_pxa2xx_platform(+) pwm_lpss_pci
pwm_lpss spi_pxa2xx_pci sdhci_pci cqhci intel_mrfld_pwrbtn extcon_intel_mrfld sdhci
intel_mrfld_adc led_class mmc_core ili9341 mipi_dbi tinydrm backlight ti_ads7950
industrialio_triggered_buffer kfifo_buf intel_soc_pmic_mrfld hci_uart btbcm
    [   23.750599] CPU: 1 PID: 1296 Comm: modprobe Not tainted 5.0.0-rc7+ #236
    [   23.750605] Hardware name: Intel Corporation Merrifield/BODEGA BAY, BIOS 542
2015.01.21:18.19.48
    [   23.750620] RIP: 0010:check_for_stack+0xb7/0x190
    [   23.750630] Code: 8b 6d 50 4d 85 ed 75 04 4c 8b 6d 10 48 89 ef e8 2f 8b 44 00 48 89
c6 4a 8d 0c 23 4c 89 ea 48 c7 c7 88 d0 82 b4 e8 40 7c f9 ff <0f> 0b 8b 05 79 00 4b 01 85 c0
74 07 5b 5d 41 5c 41 5d c3 8b 05 54
    [   23.750637] RSP: 0000:ffff97bbc0292fa0 EFLAGS: 00010286
    [   23.750646] RAX: 0000000000000000 RBX: ffff97bbc0290000 RCX: 0000000000000006
    [   23.750652] RDX: 0000000000000007 RSI: 0000000000000002 RDI: ffff94b33e115450
    [   23.750658] RBP: ffff94b33c8578b0 R08: 0000000000000002 R09: 00000000000201c0
    [   23.750664] R10: 00000006ecb0ccc6 R11: 0000000000034f38 R12: 000000000000316c
    [   23.750670] R13: ffff94b33c84b250 R14: ffff94b33dedd5a0 R15: 0000000000000001
    [   23.750679] FS:  0000000000000000(0000) GS:ffff94b33e100000(0063)
knlGS:00000000f7faf690
    [   23.750686] CS:  0010 DS: 002b ES: 002b CR0: 0000000080050033
    [   23.750691] CR2: 00000000f7f54faf CR3: 000000000722c000 CR4: 00000000001006e0
    [   23.750696] Call Trace:
    [   23.750713]  debug_dma_map_sg+0x100/0x340
    [   23.750727]  ? dma_direct_map_sg+0x3b/0xb0
    [   23.750739]  spi_map_buf+0x25a/0x300
    [   23.750751]  __spi_pump_messages+0x2a4/0x680
    [   23.750762]  __spi_sync+0x1dd/0x1f0
    [   23.750773]  spi_sync+0x26/0x40
    [   23.750790]  mipi_dbi_typec3_command_read+0x14d/0x240 [mipi_dbi]
    [   23.750802]  ? spi_finalize_current_transfer+0x10/0x10
    [   23.750821]  mipi_dbi_typec3_command+0x1bc/0x1d0 [mipi_dbi]

    Reported-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Noralf Trønnes <noralf@tronnes.org>
    Tested-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Acked-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20190222124329.23046-1-
noralf@tronnes.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 007130f9c4071c7153c16ea6ba68ff25f37554ed
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Sun Feb 3 01:40:16 2019 +0800

    e1000e: Disable runtime PM on CNP+

    [ Upstream commit 459d69c407f9ba122f12216555c3012284dc9fd7 ]

    There are some new e1000e devices can only be woken up from D3 one time,
    by plugging Ethernet cable. Subsequent cable plugging does set PME bit
    correctly, but it still doesn't get woken up.

    Since e1000e connects to the root complex directly, we rely on ACPI to
    wake it up. In this case, the GPE from _PRW only works once and stops
    working after that. Though it appears to be a platform bug, e1000e
    maintainers confirmed that I219 does not support D3.

    So disable runtime PM on CNP+ chips. We may need to disable earlier
    generations if this bug also hit older platforms.

    Bugzilla: https://bugzilla.kernel.org/attachment.cgi?id=280819
    Signed-off-by: Kai-Heng Feng <kai.heng.feng@canonical.com>
    Tested-by: Aaron Brown <aaron.f.brown@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0f0b754d6220cae80c223b79f026ae22cb9354b2
Author: Akeem G Abodunrin <akeem.g.abodunrin@intel.com>
Date:   Fri Feb 8 12:50:49 2019 -0800

    ice: Fix issue with VF reset and multiple VFs support on PFs

    [ Upstream commit 42b2cc83afb4d1afcf7794148dd4e8e45ba32943 ]

    This patch fixes issues with VF queues being disabled, and VF netdev
    network carrier being lost after reset. Basically, we need to check if VF
    is enabled, and queue configured in reset_all_vfs flow, and disable/enable
    those queues appropriately whenever the function is called after
    Global/CORER/PFR reset/rebuild/replay.

    Signed-off-by: Akeem G Abodunrin <akeem.g.abodunrin@intel.com>
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 37e27e4fd8f493981977776a9bc4052ce4322b1b
Author: Hans Verkuil <hverkuil@xs4all.nl>
Date:   Wed Mar 13 10:47:13 2019 -0400

    media: vicodec: reset last_src/dst_buf based on the IS_OUTPUT

    [ Upstream commit 76eb24fc233b8c94b2156ead5811e08d2046ad58 ]

    When start_streaming was called both last_src_buf and last_dst_buf
    pointers were set to NULL, but this depends on whether the capture
    or output queue starts streaming.

    When decoding with resolution changes in between the capture queue
    has to restart streaming whenever a resolution change occurs. And
    that would reset last_src_buf as well, which causes a problem if
    the decoder was stopped by the application. Since last_src_buf
    is now NULL, the LAST flag is never set for the last capture
    buffer.
```

```
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7a669c40424152c064a430fce2e9c1ecd838d6a2
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 4 15:29:10 2019 -0500

    media: staging/intel-ipu3: mark PM function as __maybe_unused

    [ Upstream commit 948dff7cfa1d7653e7828e7b905863bd24ca5c02 ]

    The imgu_rpm_dummy_cb() looks like an API misuse that is explained
    in the comment above it. Aside from that, it also causes a warning
    when power management support is disabled:

    drivers/staging/media/ipu3/ipu3.c:794:12: error: 'imgu_rpm_dummy_cb' defined but not
used [-Werror=unused-function]

    The warning is at least easy to fix by marking the function as
    __maybe_unused.

    Fixes: 7fc7af649ca7 ("media: staging/intel-ipu3: Add imgu top level pci device driver")

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Sakari Ailus <sakari.ailus@linux.intel.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 979478f208464977fa749a91bf967cc63ce70921
Author: Akinobu Mita <akinobu.mita@gmail.com>
Date:   Mon Mar 11 11:36:02 2019 -0400

    media: ov7670: restore default settings after power-up

    [ Upstream commit 32ab688b280301f7cee9e547564cb74e33e06322 ]

    Since commit 3d6a8fe25605 ("media: ov7670: hook s_power onto v4l2 core"),
    the device is actually powered off while the video stream is stopped.

    The frame format and framerate are restored right after power-up, but
    restoring the default register settings is forgotten.

    Fixes: 3d6a8fe25605 ("media: ov7670: hook s_power onto v4l2 core")

    Cc: Jonathan Corbet <corbet@lwn.net>
    Signed-off-by: Akinobu Mita <akinobu.mita@gmail.com>
    Reviewed-by: Lubomir Rintel <lkundrak@v3.sk>
    Tested-by: Lubomir Rintel <lkundrak@v3.sk>
    Signed-off-by: Sakari Ailus <sakari.ailus@linux.intel.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4f006f631ccc125c87e9b5e96781b4135d9753b4
Author: Sakari Ailus <sakari.ailus@linux.intel.com>
Date:   Sat Mar 2 10:23:12 2019 -0500

    media: v4l2-fwnode: The first default data lane is 0 on C-PHY

    [ Upstream commit fff35d45e16fae125c6000cb87e254cb634ac7fb ]

    C-PHY has no clock lanes. Therefore the first data lane is 0 by default.

    Fixes: edc6d56c2e7e ("media: v4l: fwnode: Support parsing of CSI-2 C-PHY endpoints")
```

```
    Signed-off-by: Sakari Ailus <sakari.ailus@linux.intel.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1413e373fd0fd16519b87969b7c0c2ece070d933
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Tue Mar 12 03:33:28 2019 -0500

    thunderbolt: property: Fix a NULL pointer dereference

    [ Upstream commit 106204b56f60abf1bead7dceb88f2be3e34433da ]

    In case kzalloc fails, the fix releases resources and returns
    -ENOMEM to avoid the NULL pointer dereference.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fad8a67e474468ec9b99a4bebea8cc6165eba897
Author: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
Date:   Thu Feb 28 12:57:59 2019 -0500

    drm/amd/display: Reset alpha state for planes to the correct values

    [ Upstream commit eec3d5efd16d13984a88396b685ae17462fb6d87 ]

    [Why]
    The plane_reset callback is subclassed but hasn't been updated since
    the drm helper got updated to include resetting alpha related state
    (state->alpha and state->pixel_blend_mode). The overlay planes
    exposed by amdgpu_dm were therefore being rendered as invisible by
    default ever since supported was exposed for alpha blending properties
    on overlays.

    This caused regressions in igt@kms_plane_multiple@atomic-tiling-none
    and igt@kms_plane@plane-position-covered-pipe tests.

    [How]
    Reset the plane state values to their correct values as defined in
    the drm helper.

    This fixes the IGT test regression.

    Signed-off-by: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
    Reviewed-by: Harry Wentland <Harry.Wentland@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 60b6c466a4284cf97de891e1fcf0ee444a34ac80
Author: David Francis <David.Francis@amd.com>
Date:   Mon Mar 4 10:31:31 2019 -0500

    drm/amd/display: Update ABM crtc state on non-modeset

    [ Upstream commit b05e2c5e81f9a0be4a145e0926b1dfe62f6347d4 ]

    [Why]
    Somewhere in the atomic check reshuffle ABM got lost.
    ABM is a crtc property (copied from a connector property).
    It can change without a modeset, just like underscan.

    [How]
    In the skip_modeset branch of atomic check crtc updates,
```

```
    copy over the abm property.

    Signed-off-by: David Francis <David.Francis@amd.com>
    Reviewed-by: Nicholas Kazlauskas <Nicholas.Kazlauskas@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5ab2c21538a29a288af46744bb6135a0620ed5bd
Author: Samson Tam <Samson.Tam@amd.com>
Date:   Mon Mar 4 16:21:06 2019 -0500

    drm/amd/display: Link train only when link is DP and backend is enabled

    [ Upstream commit 66acd4418d7de131ef3831e52a8af3d2480e5b15 ]

    [Why]
    In certain cases we do link training when we don't have a backend.

    [How]
    In dc_link_set_preferred_link_settings(), store preferred link settings
    first and then verify that the link is DP and the link stream's backend is
    enabled.  If either is false, then we will not do any link retraining.

    Signed-off-by: Samson Tam <Samson.Tam@amd.com>
    Reviewed-by: Aric Cyr <Aric.Cyr@amd.com>
    Acked-by: Anthony Koo <Anthony.Koo@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 12fd94b117aae1527f61536fd7618e1f6f8b1617
Author: Dmytro Laktyushkin <Dmytro.Laktyushkin@amd.com>
Date:   Thu Mar 7 13:26:13 2019 -0500

    drm/amd/display: fix releasing planes when exiting odm

    [ Upstream commit bc2193992b00488f5734613ac95b78ef2d2803ab ]

    Releasing planes should not release the 2nd odm pipe right away,
    this change leaves us with 2 pipes with null planes and same stream
    when planes are released during odm.

    Signed-off-by: Dmytro Laktyushkin <Dmytro.Laktyushkin@amd.com>
    Reviewed-by: Tony Cheng <Tony.Cheng@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 13b91576a59adaeacf348cddfa0108888e65d2fa
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 4 20:38:29 2019 +0100

    regulator: add regulator_get_linear_step() stub helper

    [ Upstream commit 7287275b4301e230be9e4569431c7dacb67ebc13 ]

    The regulator header has empty inline functions for most interfaces,
    but not regulator_get_linear_step(), which has just grown a user
    that does not depend on regulators otherwise:

    drivers/clk/tegra/clk-tegra124-dfll-fcpu.c: In function 'get_alignment_from_regulator':
    drivers/clk/tegra/clk-tegra124-dfll-fcpu.c:555:19: error: implicit declaration of
function 'regulator_get_linear_step'; did you mean 'regulator_get_drvdata'? [-
Werror=implicit-function-declaration]
```

```
       align->step_uv = regulator_get_linear_step(reg);
                        ^~~~~~~~~~~~~~~~~~~~~~~~
                        regulator_get_drvdata
    cc1: all warnings being treated as errors
    scripts/Makefile.build:278: recipe for target 'drivers/clk/tegra/clk-tegra124-dfll-
fcpu.o' failed

    Add the missing stub along the others.

    Fixes: b3cf8d069505 ("clk: tegra: dfll: CVB calculation alignment with the regulator")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ed49bb0dd557d28aa4a05ac3a1e23799469ecf5a
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Thu Mar 21 13:42:25 2019 +0100

    spi: export tracepoint symbols to modules

    [ Upstream commit ca1438dcb34c7fcad63b6ce14ea63a870b92a69b ]

    The newly added tracepoints in the spi-mxs driver cause a link
    error when the driver is a loadable module:

    ERROR: "__tracepoint_spi_transfer_stop" [drivers/spi/spi-mxs.ko] undefined!
    ERROR: "__tracepoint_spi_transfer_start" [drivers/spi/spi-mxs.ko] undefined!

    I'm not quite sure where to put the export statements, but
    directly after the inclusion of the header seems as good as
    any other place.

    Fixes: f3fdea3af405 ("spi: mxs: add tracing to custom .transfer_one_message callback")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d82c47c58b7994c2802770c9351eb36c338c4857
Author: Aditya Pakki <pakki001@umn.edu>
Date:   Wed Mar 20 10:57:54 2019 -0500

    thunderbolt: Fix to check for kmemdup failure

    [ Upstream commit 2cc12751cf464a722ff57b54d17d30c84553f9c0 ]

    Memory allocated via kmemdup might fail and return a NULL pointer.
    This patch adds a check on the return value of kmemdup and passes the
    error upstream.

    Signed-off-by: Aditya Pakki <pakki001@umn.edu>
    Reviewed-by: Mukesh Ojha <mojha@codeaurora.org>
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7f1a6aba600ae6a18b977ac7a3e20efc9b9cc882
Author: Aditya Pakki <pakki001@umn.edu>
Date:   Wed Mar 20 11:34:09 2019 -0500

    thunderbolt: Fix to check return value of ida_simple_get

    [ Upstream commit 9aabb68568b473bf2f0b179d053b403961e42e4d ]

    In enumerate_services, ida_simple_get on failure can return an error and
    leaks memory. The patch ensures that the dev_set_name is set on non
    failure cases, and releases memory during failure.
```

```
    Signed-off-by: Aditya Pakki <pakki001@umn.edu>
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 86c326d77b2d213ead1c734f2750331a15d8df8e
Author: Rouven Czerwinski <r.czerwinski@pengutronix.de>
Date:   Mon Mar 11 11:58:57 2019 +0100

    hwrng: omap - Set default quality

    [ Upstream commit 62f95ae805fa9e1e84d47d3219adddd97b2654b7 ]

    Newer combinations of the glibc, kernel and openssh can result in long initial
    startup times on OMAP devices:

    [    6.671425] systemd-rc-once[102]: Creating ED25519 key; this may take some time ...
    [  142.652491] systemd-rc-once[102]: Creating ED25519 key; done.

    due to the blocking getrandom(2) system call:

    [  142.610335] random: crng init done

    Set the quality level for the omap hwrng driver allowing the kernel to use the
    hwrng as an entropy source at boot.

    Signed-off-by: Rouven Czerwinski <r.czerwinski@pengutronix.de>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94fbe6c934195a9429ba216f95401bd60d22f1b2
Author: George Hilliard <thirtythreeforty@gmail.com>
Date:   Wed Mar 20 16:42:05 2019 -0600

    staging: mt7621-mmc: Check for nonzero number of scatterlist entries

    [ Upstream commit d4223e06b6aed581625f574ad8faa71b6c0fc903 ]

    The buffer descriptor setup loop is correct only if it is setting up at
    least one bd struct.  Besides, there is an error if dma_map_sg() returns
    0, which is possible and must be handled.

    Additionally, remove the BUG_ON() checking sglen, which is unnecessary
    because we configure DMA with that constraint during init.

    Signed-off-by: George Hilliard <thirtythreeforty@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5a27284c5ba69475fe53aee6b0f36bcb16baac97
Author: Saeed Mahameed <saeedm@mellanox.com>
Date:   Thu Mar 21 15:51:33 2019 -0700

    net/mlx5e: Fix compilation warning in en_tc.c

    [ Upstream commit ee576ec1c1c66ec1cd0c4735bb12bc08f675f530 ]

    Amazingly a mlx5e_tc function is being called from the eswitch layer,
    which is by itself very terrible! The function was declared locally in
    eswitch_offloads.c so it could be used there, which caused the following
    compilation warning, fix that.

    drivers/.../mlx5/core/en_tc.c:3242:6: [-Werror=missing-prototypes]
    error: no previous prototype for ‘mlx5e_tc_clean_fdb_peer_flows’
```

```
    Fixes: 04de7dda7394 ("net/mlx5e: Infrastructure for duplicated offloading of TC flows")
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0729d591559f0de2eb12bde6214e2f38f26aa06d
Author: Pu Wen <puwen@hygon.cn>
Date:   Sat Mar 23 23:42:20 2019 +0800

    x86/CPU/hygon: Fix phys_proc_id calculation logic for multi-die processors

    [ Upstream commit e0ceeae708cebf22c990c3d703a4ca187dc837f5 ]

    The Hygon family 18h multi-die processor platform supports 1, 2 or
    4-Dies per socket. The topology looks like this:

      System View (with 1-Die 2-Socket):
                |------------|
               ------       -----
        SOCKET0 | D0 |       | D1 |  SOCKET1
               ------       -----

      System View (with 2-Die 2-socket):
                 --------------------
                 |     -------------|------
                 |     |            |     |
               ------------       ------------
        SOCKET0 | D1 -- D0 |       | D3 -- D2 | SOCKET1
               ------------       ------------

      System View (with 4-Die 2-Socket) :
                 --------------------
                 |     -------------|------
                 |     |            |     |
               ------------       ------------
               | D1 -- D0 |       | D7 -- D6 |
               | |  \/ |  |       | |  \/ |  |
        SOCKET0 | |  /\ |  |       | |  /\ |  | SOCKET1
               | D2 -- D3 |       | D4 -- D5 |
               ------------       ------------
                 |     |            |     |
                 ------|------------|     |
                       --------------------

    Currently

      phys_proc_id = initial_apicid >> bits

    calculates the physical processor ID from the initial_apicid by shifting
    *bits*.

    However, this does not work for 1-Die and 2-Die 2-socket systems.

    According to document [1] section 2.1.11.1, the bits is the value of
    CPUID_Fn80000008_ECX[12:15]. The possible values are 4, 5 or 6 which
    mean:

      4 - 1 die
      5 - 2 dies
      6 - 3/4 dies.

    Hygon programs the initial ApicId the same way as AMD. The ApicId is
    read from CPUID_Fn00000001_EBX (see section 2.1.11.1 of referrence [1])
    and the definition is as below (see section 2.1.10.2.1.3 of [1]):
```

```
          -------------------------------------------------
    Bit |     6     |   5  4  |    3   |    2   1   0   |
        |-----------|---------|--------|-----------------|
    IDs | Socket ID | Node ID | CCX ID | Core/Thread ID |
        -------------------------------------------------
```

So for 3/4-Die configurations, the bits variable is 6, which is the same as the ApicID definition field.

For 1-Die and 2-Die configurations, bits is 4 or 5, which will cause the right shifted result to not be exactly the value of socket ID.

However, the socket ID should be obtained from ApicId[6]. To fix the problem and match the ApicID field definition, set the shift bits to 6 for all Hygon family 18h multi-die CPUs.

Because AMD doesn't have 2-Socket systems with 1-Die/2-Die processors (see reference [2]), this doesn't need to be changed on the AMD side but only for Hygon.

References:
[1] https://www.amd.com/system/files/TechDocs/54945_PPR_Family_17h_Models_00h-0Fh.pdf
[2] https://www.amd.com/en/products/specifications/processors

[bp: heavily massage commit message.]

Signed-off-by: Pu Wen <puwen@hygon.cn>
Signed-off-by: Borislav Petkov <bp@suse.de>
Cc: H. Peter Anvin <hpa@zytor.com>
Cc: Ingo Molnar <mingo@redhat.com>
Cc: Thomas Gleixner <tglx@linutronix.de>
Cc: Thomas Lendacky <Thomas.Lendacky@amd.com>
Cc: Yazen Ghannam <yazen.ghannam@amd.com>
Cc: x86-ml <x86@kernel.org>
Link: https://lkml.kernel.org/r/1553355740-19999-1-git-send-email-puwen@hygon.cn
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 55d1773ed823dcb1bdde757d1e98123b49a47f8f
Author: Sameer Pujar <spujar@nvidia.com>
Date: Wed Mar 13 17:02:36 2019 +0530

dmaengine: tegra210-adma: use devm_clk_*() helpers

[Upstream commit f6ed6491d565c336a360471e0c29228e34f4380e]

adma driver is using pm_clk_*() interface for managing clock resources. With this it is observed that clocks remain ON always. This happens on Tegra devices which use BPMP co-processor to manage clock resources, where clocks are enabled during prepare phase. This is necessary because clocks to BPMP are always blocking. When pm_clk_*() interface is used on such Tegra devices, clock prepare count is not balanced till remove call happens for the driver and hence clocks are seen ON always. Thus this patch replaces pm_clk_*() with devm_clk_*() framework.

Suggested-by: Mohan Kumar D <mkumard@nvidia.com>
Reviewed-by: Jonathan Hunter <jonathanh@nvidia.com>
Signed-off-by: Sameer Pujar <spujar@nvidia.com>
Signed-off-by: Vinod Koul <vkoul@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d08c911b0a9da6d5c6913c1f9a676a15c78ab119
Author: Linus Lüssing <linus.luessing@c0d3.blue>
Date: Thu Feb 14 16:52:43 2019 +0100

batman-adv: allow updating DAT entry timeouts on incoming ARP Replies

```
    [ Upstream commit 099e6cc1582dc2903fecb898bbeae8f7cf4262c7 ]

    Currently incoming ARP Replies, for example via a DHT-PUT message, do
    not update the timeout for an already existing DAT entry. These ARP
    Replies are dropped instead.

    This however defeats the purpose of the DHCPACK snooping, for instance.
    Right now, a DAT entry in the DHT will be purged every five minutes,
    likely leading to a mesh-wide ARP Request broadcast after this timeout.
    Which then recreates the entry. The idea of the DHCPACK snooping is to
    be able to update an entry before a timeout happens, to avoid ARP Request
    flooding.

    This patch fixes this issue by updating a DAT entry on incoming
    ARP Replies even if a matching DAT entry already exists. While still
    filtering the ARP Reply towards the soft-interface, to avoid duplicate
    messages on the client device side.

    Signed-off-by: Linus Lüssing <linus.luessing@c0d3.blue>
    Acked-by: Antonio Quartulli <a@unstable.cc>
    Signed-off-by: Sven Eckelmann <sven@narfation.org>
    Signed-off-by: Simon Wunderlich <sw@simonwunderlich.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dbc55f8f3ae9b942c6eb319e25a9c125982351ac
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 25 15:23:11 2019 +0100

    selinux: avoid uninitialized variable warning

    [ Upstream commit 98bbbb76f2edcfb8fb2b8f4b3ccc7b6e99d64bd8 ]

    clang correctly points out a code path that would lead
    to an uninitialized variable use:

    security/selinux/netlabel.c:310:6: error: variable 'addr' is used uninitialized whenever
'if' condition is false
          [-Werror,-Wsometimes-uninitialized]
            if (ip_hdr(skb)->version == 4) {
                ^~~~~~~~~~~~~~~~~~~~~~~~~
    security/selinux/netlabel.c:322:40: note: uninitialized use occurs here
            rc = netlbl_conn_setattr(ep->base.sk, addr, &secattr);
                                                  ^~~~
    security/selinux/netlabel.c:310:2: note: remove the 'if' if its condition is always true
            if (ip_hdr(skb)->version == 4) {
            ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    security/selinux/netlabel.c:291:23: note: initialize the variable 'addr' to silence this
warning
            struct sockaddr *addr;
                             ^
                              = NULL

    This is probably harmless since we should not see ipv6 packets
    of CONFIG_IPV6 is disabled, but it's better to rearrange the code
    so this cannot happen.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    [PM: removed old patchwork link, fixed checkpatch.pl style errors]
    Signed-off-by: Paul Moore <paul@paul-moore.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d060bb09ed80be73fa3999bf968e461c3530689a
Author: Dave Ertman <david.m.ertman@intel.com>
Date:   Wed Feb 13 10:51:08 2019 -0800
```

```
    ice: Prevent unintended multiple chain resets

    [ Upstream commit 2ebd4428d93a2f6ce0c813b10a1a43b6a8241fe5 ]

    In the current implementation of ice_reset_subtask, if multiple reset
    types are set in the pf->state, the most intrusive one is meant to be
    performed only, but the bits requesting the other types are not being
    cleared. This would lead to another reset being performed the next time
    the service task is scheduled.

    Change the flow of ice_reset_subtask so that all reset request bits in
    pf->state are cleared, and we still perform the most intrusive of the
    resets requested.

    Signed-off-by: Dave Ertman <david.m.ertman@intel.com>
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3f0fe090c40333a2f9fa42fb220dbbff9afb8683
Author: Dafna Hirschfeld <dafna3@gmail.com>
Date:   Wed Mar 6 16:13:28 2019 -0500

    media: vicodec: bugfix - call v4l2_m2m_buf_copy_metadata also if decoding fails

    [ Upstream commit 8eead25cbdf911e17cff321903bd3397bc6ea22c ]

    The function 'v4l2_m2m_buf_copy_metadata' should
    be called even if decoding/encoding ends with
    status VB2_BUF_STATE_ERROR, so that the metadata
    is copied from the source buffer to the dest buffer.

    Signed-off-by: Dafna Hirschfeld <dafna3@gmail.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 614ed19a761ab39dd43725502192307f24712701
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Fri Mar 22 15:25:49 2019 +0100

    scsi: lpfc: avoid uninitialized variable warning

    [ Upstream commit faf5a744f4f8d76e7c03912b5cd381ac8045f6ec ]

    clang -Wuninitialized incorrectly sees a variable being used without
    initialization:

    drivers/scsi/lpfc/lpfc_nvme.c:2102:37: error: variable 'localport' is uninitialized when
used here
          [-Werror,-Wuninitialized]
                    lport = (struct lpfc_nvme_lport *)localport->private;
                                                      ^~~~~~~~~
    drivers/scsi/lpfc/lpfc_nvme.c:2059:38: note: initialize the variable 'localport' to
silence this warning
            struct nvme_fc_local_port *localport;
                                       ^
                                        = NULL
    1 error generated.

    This is clearly in dead code, as the condition leading up to it is always
    false when CONFIG_NVME_FC is disabled, and the variable is always
    initialized when nvme_fc_register_localport() got called successfully.
```

```
    Change the preprocessor conditional to the equivalent C construct, which
    makes the code more readable and gets rid of the warning.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: James Smart <james.smart@broadcom.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 890ea842f0855eaa0f1724af2f1c1de36e1a1da6
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Fri Mar 22 15:25:03 2019 +0100

    scsi: qla4xxx: avoid freeing unallocated dma memory

    [ Upstream commit 608f729c31d4caf52216ea00d20092a80959256d ]

    Clang -Wuninitialized notices that on is_qla40XX we never allocate any DMA
    memory in get_fw_boot_info() but attempt to free it anyway:

    drivers/scsi/qla4xxx/ql4_os.c:5915:7: error: variable 'buf_dma' is used uninitialized
whenever 'if' condition is false
          [-Werror,-Wsometimes-uninitialized]
                    if (!(val & 0x07)) {
                        ^~~~~~~~~~~~
    drivers/scsi/qla4xxx/ql4_os.c:5985:47: note: uninitialized use occurs here
            dma_free_coherent(&ha->pdev->dev, size, buf, buf_dma);
                                                         ^~~~~~
    drivers/scsi/qla4xxx/ql4_os.c:5915:3: note: remove the 'if' if its condition is always
true
                    if (!(val & 0x07)) {
                    ^~~~~~~~~~~~~~~~~~~
    drivers/scsi/qla4xxx/ql4_os.c:5885:20: note: initialize the variable 'buf_dma' to
silence this warning
            dma_addr_t buf_dma;
                       ^
                        = 0

    Skip the call to dma_free_coherent() here.

    Fixes: 2a991c215978 ("[SCSI] qla4xxx: Boot from SAN support for open-iscsi")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 08cce74e676acfe66273d397a20ab8cb03b578b6
Author: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
Date:   Tue Feb 19 15:04:01 2019 -0800

    ice: Fix for adaptive interrupt moderation

    [ Upstream commit 64a59d05a4b3ddb37eb5ad3a3be0f17148f449f5 ]

    commit 63f545ed1285 ("ice: Add support for adaptive interrupt moderation")
    was meant to add support for adaptive interrupt moderation but there was
    an error on my part while formatting the patch, and thus only part of the
    patch ended up being submitted.

    This patch rectifies the error by adding the rest of the code.

    Fixes: 63f545ed1285 ("ice: Add support for adaptive interrupt moderation")
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
```

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

commit d43589f91b5d5ad7e0d00e4ca3a824b28583ae82
Author: Tony Lindgren <tony@atomide.com>
Date:   Fri Mar 22 14:54:05 2019 -0700

    usb: core: Add PM runtime calls to usb_hcd_platform_shutdown

    [ Upstream commit 8ead7e817224d7832fe51a19783cb8fcadc79467 ]

    If ohci-platform is runtime suspended, we can currently get an "imprecise
    external abort" on reboot with ohci-platform loaded when PM runtime
    is implemented for the SoC.

    Let's fix this by adding PM runtime support to usb_hcd_platform_shutdown.

    Signed-off-by: Tony Lindgren <tony@atomide.com>
    Acked-by: Alan Stern <stern@rowland.harvard.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 30a120c163d856e65a39a0f56f623cee6470ec39
Author: Ludovic Barre <ludovic.barre@st.com>
Date:   Mon Mar 25 18:01:39 2019 +0100

    spi: stm32-qspi: add spi_master_put in release function

    [ Upstream commit a88eceb17ac7e8dc4ad9995681af61c8371668f4 ]

    This patch adds spi_master_put in release function
    to drop the controller's refcount.

    Signed-off-by: Ludovic Barre <ludovic.barre@st.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8063559dfb6df74063a3d55d68b38b1a08196005
Author: Leon Romanovsky <leonro@mellanox.com>
Date:   Tue Mar 12 10:15:44 2019 +0200

    RDMA/rxe: Fix slab-out-bounds access which lead to kernel crash later

    [ Upstream commit a4b7013db23e93824ac53083eeb3e4efdef4b5b0 ]

    BUG: KASAN: slab-out-of-bounds in rxe_mem_init_user+0x6c1/0x740 [rdma_rxe]
    Read of size 8 at addr ffff88805c01a608 by task ib_send_bw/573

    CPU: 24 PID: 573 Comm: ib_send_bw Not tainted 5.0.0-rc5+ #189
    Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.11.0-0-g63451fca13-
prebuilt.qemu-project.org 04/01/2014
    Call Trace:
     rxe_mem_init_user+0x6c1/0x740 [rdma_rxe]
     rxe_reg_user_mr+0x9b/0x110 [rdma_rxe]
     ib_uverbs_reg_mr+0x428/0x9c0 [ib_uverbs]
     ib_uverbs_handler_UVERBS_METHOD_INVOKE_WRITE+0x2b0/0x410 [ib_uverbs]
     ib_uverbs_run_method+0x79c/0x1da0 [ib_uverbs]
     rxe_mem_init_user+0x6c1/0x740 [rdma_rxe]
     rxe_reg_user_mr+0x9b/0x110 [rdma_rxe]
     ib_uverbs_reg_mr+0x428/0x9c0 [ib_uverbs]
     ib_uverbs_handler_UVERBS_METHOD_INVOKE_WRITE+0x2b0/0x410 [ib_uverbs]
     ib_uverbs_run_method+0x79c/0x1da0 [ib_uverbs]
     ib_uverbs_cmd_verbs+0x5f2/0xf20 [ib_uverbs]
     ib_uverbs_ioctl+0x202/0x310 [ib_uverbs]
     do_vfs_ioctl+0x193/0x1440
     ksys_ioctl+0x3a/0x70
```

```
       __x64_sys_ioctl+0x6f/0xb0
       do_syscall_64+0x13f/0x570
       entry_SYSCALL_64_after_hwframe+0x49/0xbe

      Allocated by task 573:
       __kasan_kmalloc.constprop.5+0xc1/0xd0
       __kmalloc+0x161/0x310
       rxe_mem_alloc+0x52/0x470 [rdma_rxe]
       rxe_mem_init_user+0x113/0x740 [rdma_rxe]
       rxe_reg_user_mr+0x9b/0x110 [rdma_rxe]
       ib_uverbs_reg_mr+0x428/0x9c0 [ib_uverbs]
       ib_uverbs_handler_UVERBS_METHOD_INVOKE_WRITE+0x2b0/0x410 [ib_uverbs]
       ib_uverbs_run_method+0x79c/0x1da0 [ib_uverbs]
       ib_uverbs_cmd_verbs+0x5f2/0xf20 [ib_uverbs]
       ib_uverbs_ioctl+0x202/0x310 [ib_uverbs]
       do_vfs_ioctl+0x193/0x1440
       ksys_ioctl+0x3a/0x70
       __x64_sys_ioctl+0x6f/0xb0
       do_syscall_64+0x13f/0x570
       entry_SYSCALL_64_after_hwframe+0x49/0xbe

      Freed by task 0:
       __kasan_slab_free+0x12e/0x180
       kfree+0x10a/0x2c0
       rcu_process_callbacks+0xa77/0x1260
       __do_softirq+0x2ad/0xacb

      Test scenario:
       ib_send_bw -x 1 -d rxe0 -a &
       ib_send_bw -x 1 -d rxe0 -a localhost

      Fixes: 8700e3e7c485 ("Soft RoCE driver")
      Reported-by: Parav Pandit <parav@mellanox.com>
      Reviewed-by: Zhu Yanjun <yanjun.zhu@oracle.com>
      Tested-by: Zhu Yanjun <yanjun.zhu@oracle.com>
      Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
      Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 6937a0525c18d87d06ed2af01900c2fb457905a5
  Author: Kangjie Lu <kjlu@umn.edu>
  Date:   Mon Mar 25 15:29:22 2019 -0500

      x86/platform/uv: Fix missing checks of kcalloc() return values

      [ Upstream commit 766460852cfaeca4042e5f3aeb9616b3689147bc ]

      Handle potential errors returned from kcalloc().

       [ bp: rewrite commit message. ]

      Signed-off-by: Kangjie Lu <kjlu@umn.edu>
      Signed-off-by: Borislav Petkov <bp@suse.de>
      Cc: Andrew Banman <abanman@hpe.com>
      Cc: Andy Shevchenko <andy@infradead.org>
      Cc: Colin Ian King <colin.king@canonical.com>
      Cc: Darren Hart <dvhart@infradead.org>
      Cc: "Gustavo A. R. Silva" <gustavo@embeddedor.com>
      Cc: "H. Peter Anvin" <hpa@zytor.com>
      Cc: Ingo Molnar <mingo@redhat.com>
      Cc: Kees Cook <keescook@chromium.org>
      Cc: Mike Travis <mike.travis@hpe.com>
      Cc: Nicolai Stange <nstange@suse.de>
      Cc: pakki001@umn.edu
      Cc: platform-driver-x86@vger.kernel.org
```

```
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Varsha Rao <rvarsha016@gmail.com>
    Cc: x86-ml <x86@kernel.org>
    Link: https://lkml.kernel.org/r/20190325202924.4624-1-kjlu@umn.edu
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e8cfc3265cdce02a8c26b734b1b2d6b1368f04ba
Author: Neeraj Upadhyay <neeraju@codeaurora.org>
Date:   Mon Mar 11 17:28:03 2019 +0530

    rcu: Do a single rhp->func read in rcu_head_after_call_rcu()

    [ Upstream commit b699cce1604e828f19c39845252626eb78cdf38a ]

    The rcu_head_after_call_rcu() function reads the rhp->func pointer twice,
    which can result in a false-positive WARN_ON_ONCE() if the callback
    were passed to call_rcu() between the two reads.  Although racing
    rcu_head_after_call_rcu() with call_rcu() is to be a dubious use case
    (the return value is not reliable in that case), intermittent and
    irreproducible warnings are also quite dubious.  This commit therefore
    uses a single READ_ONCE() to pick up the value of rhp->func once, then
    tests that value twice, thus guaranteeing consistent processing within
    rcu_head_after_call_rcu()().

    Neverthless, racing rcu_head_after_call_rcu() with call_rcu() is still
    a dubious use case.

    Signed-off-by: Neeraj Upadhyay <neeraju@codeaurora.org>
    [ paulmck: Add blank line after declaration per checkpatch.pl. ]
    Signed-off-by: Paul E. McKenney <paulmck@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f618b46f9d282e5d00b197df60e98dbe8c1d6673
Author: Paul E. McKenney <paulmck@linux.ibm.com>
Date:   Thu Mar 21 10:26:41 2019 -0700

    rcuperf: Fix cleanup path for invalid perf_type strings

    [ Upstream commit ad092c027713a68a34168942a5ef422e42e039f4 ]

    If the specified rcuperf.perf_type is not in the rcu_perf_init()
    function's perf_ops[] array, rcuperf prints some console messages and
    then invokes rcu_perf_cleanup() to set state so that a future torture
    test can run.  However, rcu_perf_cleanup() also attempts to end the
    test that didn't actually start, and in doing so relies on the value
    of cur_ops, a value that is not particularly relevant in this case.
    This can result in confusing output or even follow-on failures due to
    attempts to use facilities that have not been properly initialized.

    This commit therefore sets the value of cur_ops to NULL in this case and
    inserts a check near the beginning of rcu_perf_cleanup(), thus avoiding
    relying on an irrelevant cur_ops value.

    Signed-off-by: Paul E. McKenney <paulmck@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 776613c51fdb69db2cddf58c397aab8a31059782
Author: Yazen Ghannam <yazen.ghannam@amd.com>
Date:   Fri Jul 27 16:40:09 2018 -0500

    x86/mce: Handle varying MCA bank counts

    [ Upstream commit 006c077041dc73b9490fffc4c6af5befe0687110 ]

    Linux reads MCG_CAP[Count] to find the number of MCA banks visible to a
```

```
    CPU. Currently, this number is the same for all CPUs and a warning is
    shown if there is a difference. The number of banks is overwritten with
    the MCG_CAP[Count] value of each following CPU that boots.

    According to the Intel SDM and AMD APM, the MCG_CAP[Count] value gives
    the number of banks that are available to a "processor implementation".
    The AMD BKDGs/PPRs further clarify that this value is per core. This
    value has historically been the same for every core in the system, but
    that is not an architectural requirement.

    Future AMD systems may have different MCG_CAP[Count] values per core,
    so the assumption that all CPUs will have the same MCG_CAP[Count] value
    will no longer be valid.

    Also, the first CPU to boot will allocate the struct mce_banks[] array
    using the number of banks based on its MCG_CAP[Count] value. The machine
    check handler and other functions use the global number of banks to
    iterate and index into the mce_banks[] array. So it's possible to use an
    out-of-bounds index on an asymmetric system where a following CPU sees a
    MCG_CAP[Count] value greater than its predecessors.

    Thus, allocate the mce_banks[] array to the maximum number of banks.
    This will avoid the potential out-of-bounds index since the value of
    mca_cfg.banks is capped to MAX_NR_BANKS.

    Set the value of mca_cfg.banks equal to the max of the previous value
    and the value for the current CPU. This way mca_cfg.banks will always
    represent the max number of banks detected on any CPU in the system.

    This will ensure that all CPUs will access all the banks that are
    visible to them. A CPU that can access fewer than the max number of
    banks will find the registers of the extra banks to be read-as-zero.

    Furthermore, print the resulting number of MCA banks in use. Do this in
    mcheck_late_init() so that the final value is printed after all CPUs
    have been initialized.

    Finally, get bank count from target CPU when doing injection with mce-inject
    module.

     [ bp: Remove out-of-bounds example, passify and cleanup commit message. ]

    Signed-off-by: Yazen Ghannam <yazen.ghannam@amd.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: linux-edac <linux-edac@vger.kernel.org>
    Cc: Pu Wen <puwen@hygon.cn>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Tony Luck <tony.luck@intel.com>
    Cc: Vishal Verma <vishal.l.verma@intel.com>
    Cc: x86-ml <x86@kernel.org>
    Link: https://lkml.kernel.org/r/20180727214009.78289-1-Yazen.Ghannam@amd.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f55e548fc751e5d7822f073c248938596326e68b
Author: Paul E. McKenney <paulmck@linux.ibm.com>
Date:   Thu Mar 21 09:27:28 2019 -0700

    rcutorture: Fix cleanup path for invalid torture_type strings

    [ Upstream commit b813afae7ab6a5e91b4e16cc567331d9c2ae1f04 ]

    If the specified rcutorture.torture_type is not in the rcu_torture_init()
    function's torture_ops[] array, rcutorture prints some console messages
```

```
    and then invokes rcu_torture_cleanup() to set state so that a future
    torture test can run.  However, rcu_torture_cleanup() also attempts to
    end the test that didn't actually start, and in doing so relies on the
    value of cur_ops, a value that is not particularly relevant in this case.
    This can result in confusing output or even follow-on failures due to
    attempts to use facilities that have not been properly initialized.

    This commit therefore sets the value of cur_ops to NULL in this case
    and inserts a check near the beginning of rcu_torture_cleanup(),
    thus avoiding relying on an irrelevant cur_ops value.

    Reported-by: kernel test robot <rong.a.chen@intel.com>
    Signed-off-by: Paul E. McKenney <paulmck@linux.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 34315916ec973a81d7ab26e9be45bfcbe9979ae6
Author: Tony Luck <tony.luck@intel.com>
Date:   Tue Mar 12 10:09:38 2019 -0700

    x86/mce: Fix machine_check_poll() tests for error types

    [ Upstream commit f19501aa07f18268ab14f458b51c1c6b7f72a134 ]

    There has been a lurking "TBD" in the machine check poll routine ever
    since it was first split out from the machine check handler. The
    potential issue is that the poll routine may have just begun a read from
    the STATUS register in a machine check bank when the hardware logs an
    error in that bank and signals a machine check.

    That race used to be pretty small back when machine checks were
    broadcast, but the addition of local machine check means that the poll
    code could continue running and clear the error from the bank before the
    local machine check handler on another CPU gets around to reading it.

    Fix the code to be sure to only process errors that need to be processed
    in the poll code, leaving other logged errors alone for the machine
    check handler to find and process.

     [ bp: Massage a bit and flip the "== 0" check to the usual !(..) test. ]

    Fixes: b79109c3bbcf ("x86, mce: separate correct machine check poller and fatal
exception handler")
    Fixes: ed7290d0ee8f ("x86, mce: implement new status bits")
    Reported-by: Ashok Raj <ashok.raj@intel.com>
    Signed-off-by: Tony Luck <tony.luck@intel.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: Ashok Raj <ashok.raj@intel.com>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: linux-edac <linux-edac@vger.kernel.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: x86-ml <x86@kernel.org>
    Cc: Yazen Ghannam <Yazen.Ghannam@amd.com>
    Link: https://lkml.kernel.org/r/20190312170938.GA23035@agluck-desk
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3d4afffaac1087bd6562e6cbe511ce6923e2f7a
Author: Leon Romanovsky <leonro@mellanox.com>
Date:   Sun Mar 17 12:11:14 2019 +0200

    overflow: Fix -Wtype-limits compilation warnings

    [ Upstream commit dc7fe518b0493faa0af0568d6d8c2a33c00f58d0 ]

    Attempt to use check_shl_overflow() with inputs of unsigned type
```

```
    produces the following compilation warnings.

    drivers/infiniband/hw/mlx5/qp.c: In function _set_user_rq_size_:
    ./include/linux/overflow.h:230:6: warning: comparison of unsigned
    expression >= 0 is always true [-Wtype-limits]
       _s >= 0 && _s < 8 * sizeof(*d) ? _s : 0;  \
          ^~
    drivers/infiniband/hw/mlx5/qp.c:5820:6: note: in expansion of macro _check_shl_overflow_
      if (check_shl_overflow(rwq->wqe_count, rwq->wqe_shift,
    &rwq->buf_size))
          ^~~~~~~~~~~~~~~~~
    ./include/linux/overflow.h:232:26: warning: comparison of unsigned expression < 0 is
 always false [-Wtype-limits]
      (_to_shift != _s || *_d < 0 || _a < 0 ||   \
                          ^
    drivers/infiniband/hw/mlx5/qp.c:5820:6: note: in expansion of macro _check_shl_overflow_
      if (check_shl_overflow(rwq->wqe_count, rwq->wqe_shift, &rwq->buf_size))
          ^~~~~~~~~~~~~~~~~
    ./include/linux/overflow.h:232:36: warning: comparison of unsigned expression < 0 is
 always false [-Wtype-limits]
      (_to_shift != _s || *_d < 0 || _a < 0 ||   \
                                     ^
    drivers/infiniband/hw/mlx5/qp.c:5820:6: note: in expansion of macro _check_shl_overflow_
      if (check_shl_overflow(rwq->wqe_count, rwq->wqe_shift,&rwq->buf_size))
          ^~~~~~~~~~~~~~~~~

    Fixes: 0c66847793d1 ("overflow.h: Add arithmetic shift helper")
    Reviewed-by: Bart Van Assche <bvanassche@acm.org>
    Acked-by: Kees Cook <keescook@chromium.org>
    Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit a36c3c66af62ead7f65debdd3b0d043403290ec8
 Author: George Hilliard <thirtythreeforty@gmail.com>
 Date:   Tue Mar 26 19:50:57 2019 -0600

    staging: mt7621-mmc: Initialize completions a single time during probe

    [ Upstream commit 7ca8c2c8bbeda2a2a2a9898cd35066bc1dc83836 ]

    The module was initializing completions whenever it was going to wait on
    them, and not when the completion was allocated.  This is incorrect
    according to the completion docs:

        Calling init_completion() on the same completion object twice is
        most likely a bug [...]

    Re-initialization is also unnecessary because the module never uses
    complete_all().  Fix this by only ever initializing the completion a
    single time, and log if the completions are not consumed as intended
    (this is not a fatal problem, but should not go unnoticed).

    Signed-off-by: George Hilliard <thirtythreeforty@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit a04b2936a060fb94f2059a333129e512184399c0
 Author: Kangjie Lu <kjlu@umn.edu>
 Date:   Fri Mar 15 02:07:12 2019 -0500

    tty: ipwireless: fix missing checks for ioremap

    [ Upstream commit 1bbb1c318cd8a3a39e8c3e2e83d5e90542d6c3e3 ]
```

```
    ipw->attr_memory and ipw->common_memory are assigned with the
    return value of ioremap. ioremap may fail, but no checks
    are enforced. The fix inserts the checks to avoid potential
    NULL pointer dereferences.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ddbc7bfaa7cffc43045127966c3828d78a8859df
Author: Pankaj Gupta <pagupta@redhat.com>
Date:   Tue Mar 19 11:34:06 2019 +0530

    virtio_console: initialize vtermno value for ports

    [ Upstream commit 4b0a2c5ff7215206ea6135a405f17c5f6fca7d00 ]

    For regular serial ports we do not initialize value of vtermno
    variable. A garbage value is assigned for non console ports.
    The value can be observed as a random integer with [1].

    [1] vim /sys/kernel/debug/virtio-ports/vport*p*

    This patch initialize the value of vtermno for console serial
    ports to '1' and regular serial ports are initiaized to '0'.

    Reported-by: siliu@redhat.com
    Signed-off-by: Pankaj Gupta <pagupta@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aaf5aa44934ad069cac805923c49f6968b9a0d49
Author: Thierry Escande <thierry.escande@linaro.org>
Date:   Thu Mar 7 10:12:23 2019 +0000

    misc: fastrpc: Fix a possible double free

    [ Upstream commit b49f6d83e290f17e20f4e5cf31288d3bb4955ea6 ]

    This patch fixes the error exit path of fastrpc_init_create_process().
    If the DMA allocation or the DSP invoke fails the fastrpc_map was freed
    but not removed from the mapping list leading to a double free once the
    mapping list is emptied in fastrpc_device_release().

    [srinivas kandagatla]: Cleaned up error path labels and reset init mem
    to NULL after free
    Fixes: d73f71c7c6ee("misc: fastrpc: Add support for create remote init process")
    Signed-off-by: Thierry Escande <thierry.escande@linaro.org>
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8b29b2bf0a8d5768d3e419634731511ea819f2b5
Author: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
Date:   Thu Mar 7 10:12:24 2019 +0000

    misc: fastrpc: make sure memory read and writes are visible

    [ Upstream commit 415a0729bd1225f0ffbc0ba82888dd65772554f7 ]

    dma_alloc_coherent buffers could have writes queued in store buffers so
    commit them before sending buffer to DSP using correct dma barriers.
    Same with vice-versa.
```

```
    Fixes: c68cfb718c8f ("misc: fastrpc: Add support for context Invoke method")
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 954edc466128479872731d06f026d0e71840d153
Author: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
Date:   Thu Mar 7 10:12:26 2019 +0000

    misc: fastrpc: consider address offset before sending to DSP

    [ Upstream commit 80f3afd72bd4149c57daf852905476b43bb47647 ]

    While passing address phy address to DSP, take care of the offset
    calculated from virtual address vma.

    Fixes: c68cfb718c8f ("misc: fastrpc: Add support for context Invoke method")
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9b1ce019c1f7552ec4333433e878dab11d97276c
Author: Chad Dupuis <cdupuis@marvell.com>
Date:   Tue Mar 26 00:38:35 2019 -0700

    scsi: qedf: Add missing return in qedf_post_io_req() in the fcport offload check

    [ Upstream commit c5e06ba2f76809ad1492fdad312e81335df46bc5 ]

    Fixes the following crash as the return was missing from the check if an
    fcport is offloaded. If we hit this code we continue to try to post an
    invalid task which can lead to the crash:

    [30259.616411] [0000:61:00.3]:[qedf_post_io_req:989]:3: Session not offloaded yet.
    [30259.616413] [0000:61:00.3]:[qedf_upload_connection:1340]:3: Uploading connection
port_id=490020.
    [30259.623769] BUG: unable to handle kernel NULL pointer dereference at 0000000000000198
    [30259.631645] IP: [<ffffffffc035b1ed>] qedf_init_task.isra.16+0x3d/0x450 [qedf]
    [30259.638816] PGD 0
    [30259.640841] Oops: 0000 [#1] SMP
    [30259.644098] Modules linked in: fuse xt_CHECKSUM iptable_mangle ipt_MASQUERADE
nf_nat_masquerade_ipv4 iptable_nat nf_nat_ipv4 nf_nat nf_conntrack_ipv4 nf_defrag_ipv4
xt_conntrack nf_conntrack ipt_REJECT nf_reject_ipv4 tun bridge stp llc ebtable_filter
ebtables devlink ip6table_filter ip6_tables iptable_filter vfat fat ib_isert
iscsi_target_mod ib_srpt target_core_mod ib_srp scsi_transport_srp ib_ipoib ib_ucm ib_umad
dm_service_time skx_edac intel_powerclamp coretemp intel_rapl iosf_mbi kvm_intel kvm
irqbypass crc32_pclmul ghash_clmulni_intel aesni_intel rpcrdma sunrpc rdma_ucm ib_uverbs lrw
gf128mul ib_iser rdma_cm iw_cm ib_cm libiscsi scsi_transport_iscsi qedr(OE) glue_helper
ablk_helper cryptd ib_core dm_round_robin joydev pcspkr ipmi_ssif ses enclosure ipmi_si
ipmi_devintf ipmi_msghandler mei_me
    [30259.715529]  mei sg hpilo hpwdt shpchp wmi lpc_ich acpi_power_meter dm_multipath
ip_tables xfs libcrc32c sd_mod crc_t10dif crct10dif_generic uas usb_storage mgag200 qedf(OE)
i2c_algo_bit libfcoe drm_kms_helper libfc syscopyarea sysfillrect scsi_transport_fc qede(OE)
sysimgblt fb_sys_fops ptp ttm pps_core drm qed(OE) smartpqi crct10dif_pclmul
crct10dif_common crc32c_intel i2c_core scsi_transport_sas scsi_tgt dm_mirror dm_region_hash
dm_log dm_mod
    [30259.754237] CPU: 9 PID: 977 Comm: kdmwork-253:7 Kdump: loaded Tainted: G        W  OE
------------   3.10.0-862.el7.x86_64 #1
    [30259.765664] Hardware name: HPE Synergy 480 Gen10/Synergy 480 Gen10 Compute Module,
BIOS I42 04/04/2018
    [30259.775000] task: ffff8c801efd0000 ti: ffff8c801efd8000 task.ti: ffff8c801efd8000
    [30259.782505] RIP: 0010:[<ffffffffc035b1ed>]  [<ffffffffc035b1ed>]
qedf_init_task.isra.16+0x3d/0x450 [qedf]
    [30259.792116] RSP: 0018:ffff8c801efdbbb0  EFLAGS: 00010046
    [30259.797444] RAX: 0000000000000000 RBX: ffffa7f1450948d8 RCX: ffff8c7fe5bc40c8
```

```
    [30259.804600] RDX: ffff8c800715b300 RSI: ffffa7f1450948d8 RDI: ffff8c80169c2480
    [30259.811755] RBP: ffff8c801efdbc30 R08: 00000000000000ae R09: ffff8c800a314540
    [30259.818911] R10: ffff8c7fe5bc40c8 R11: ffff8c801efdb8ae R12: 0000000000000000
    [30259.826068] R13: ffff8c800715b300 R14: ffff8c80169c2480 R15: ffff8c8005da28e0
    [30259.833223] FS:  0000000000000000(0000) GS:ffff8c803f840000(0000)
knlGS:0000000000000000
    [30259.841338] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [30259.847100] CR2: 0000000000000198 CR3: 000000081242e000 CR4: 00000000007607e0
    [30259.854256] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [30259.861412] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [30259.868568] PKRU: 00000000
    [30259.871278] Call Trace:
    [30259.873737]  [<ffffffffc035c948>] qedf_post_io_req+0x148/0x680 [qedf]
    [30259.880201]  [<ffffffffc035d070>] qedf_queuecommand+0x1f0/0x240 [qedf]
    [30259.886749]  [<ffffffffa329b050>] scsi_dispatch_cmd+0xb0/0x240
    [30259.892600]  [<ffffffffa32a45bc>] scsi_request_fn+0x4cc/0x680
    [30259.898364]  [<ffffffffa3118ad9>] __blk_run_queue+0x39/0x50
    [30259.903954]  [<ffffffffa3114393>] __elv_add_request+0xd3/0x260
    [30259.909805]  [<ffffffffa311baf0>] blk_insert_cloned_request+0xf0/0x1b0
    [30259.916358]  [<ffffffffc010b622>] map_request+0x142/0x220 [dm_mod]
    [30259.922560]  [<ffffffffc010b716>] map_tio_request+0x16/0x40 [dm_mod]
    [30259.928932]  [<ffffffffa2ebb1f5>] kthread_worker_fn+0x85/0x180
    [30259.934782]  [<ffffffffa2ebb170>] ? kthread_stop+0xf0/0xf0
    [30259.940284]  [<ffffffffa2ebae31>] kthread+0xd1/0xe0
    [30259.945176]  [<ffffffffa2ebad60>] ? insert_kthread_work+0x40/0x40
    [30259.951290]  [<ffffffffa351f61d>] ret_from_fork_nospec_begin+0x7/0x21
    [30259.957750]  [<ffffffffa2ebad60>] ? insert_kthread_work+0x40/0x40
    [30259.963860] Code: fe 41 55 49 89 d5 41 54 53 48 89 f3 48 83 ec 58 4c 8b 67 28 4c 8b
4e 18 65 48 8b 04 25 28 00 00 00 48 89 45 d0 31 c0 4c 8b 7e 58 <49> 8b 84 24 98 01 00 00 48
8b 00 f6 80 31 01 00 00 10 0f 85 0b
    [30259.983372] RIP  [<ffffffffc035b1ed>] qedf_init_task.isra.16+0x3d/0x450 [qedf]
    [30259.990630]  RSP <ffff8c801efdbbb0>
    [30259.994127] CR2: 0000000000000198

    Signed-off-by: Chad Dupuis <cdupuis@marvell.com>
    Signed-off-by: Saurav Kashyap <skashyap@marvell.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1a2fdf1c536a81a39006506dc8567dc7d5ed0d7d
Author: Artemy Kovalyov <artemyko@mellanox.com>
Date:   Tue Mar 19 11:24:39 2019 +0200

    IB/mlx5: Compare only index part of a memory window rkey

    [ Upstream commit d623dfd2836114507d647c9793a80d213d8bffe8 ]

    The InfiniBand Architecture Specification section 10.6.7.2.4 TYPE 2 MEMORY
    WINDOWS says that if the CI supports the Base Memory Management Extensions
    defined in this specification, the R_Key format for a Type 2 Memory Window
    must consist of:

    * 24 bit index in the most significant bits of the R_Key, which is owned
      by the CI, and
    * 8 bit key in the least significant bits of the R_Key, which is owned by
      the Consumer.

    This means that the kernel should compare only the index part of a R_Key
    to determine equality with another R_Key.

    Fixes: db570d7deafb ("IB/mlx5: Add ODP support to MW")
    Signed-off-by: Artemy Kovalyov <artemyko@mellanox.com>
    Signed-off-by: Moni Shoua <monis@mellanox.com>
    Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
```

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

commit 5e04fcbab4b417ac1110ec917f653d7eb38fd056
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Sat Mar 23 11:36:19 2019 +0100

    timekeeping: Force upper bound for setting CLOCK_REALTIME

    [ Upstream commit 7a8e61f8478639072d402a26789055a4a4de8f77 ]

    Several people reported testing failures after setting CLOCK_REALTIME close
    to the limits of the kernel internal representation in nanoseconds,
    i.e. year 2262.

    The failures are exposed in subsequent operations, i.e. when arming timers
    or when the advancing CLOCK_MONOTONIC makes the calculation of
    CLOCK_REALTIME overflow into negative space.

    Now people start to paper over the underlying problem by clamping
    calculations to the valid range, but that's just wrong because such
    workarounds will prevent detection of real issues as well.

    It is reasonable to force an upper bound for the various methods of setting
    CLOCK_REALTIME. Year 2262 is the absolute upper bound. Assume a maximum
    uptime of 30 years which is plenty enough even for esoteric embedded
    systems. That results in an upper bound of year 2232 for setting the time.

    Once that limit is reached in reality this limit is only a small part of
    the problem space. But until then this stops people from trying to paper
    over the problem at the wrong places.

    Reported-by: Xiongfeng Wang <wangxiongfeng2@huawei.com>
    Reported-by: Hongbo Yao <yaohongbo@huawei.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: John Stultz <john.stultz@linaro.org>
    Cc: Stephen Boyd <sboyd@kernel.org>
    Cc: Miroslav Lichvar <mlichvar@redhat.com>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Cc: Richard Cochran <richardcochran@gmail.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link:
https://lkml.kernel.org/r/alpine.DEB.2.21.1903231125480.2157@nanos.tec.linutronix.de
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6eb883c480055fe2d9c0d6d898ecbc92ff824a52
Author: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
Date:   Tue Mar 12 18:18:17 2019 +0200

    drm: rcar-du: lvds: Fix post-DLL divider calculation

    [ Upstream commit 167e535438ecc73d299340bb1269616432020dfb ]

    The PLL parameters are computed by looping over the range of acceptable
    M, N and E values, and selecting the combination that produces the
    output frequency closest to the target. The internal frequency
    constraints are taken into account by restricting the tested values for
    the PLL parameters, reducing the search space. The target frequency,
    however, is only taken into account when computing the post-PLL divider,
    which can result in a 0 value for the divider when the PLL output
    frequency being tested is lower than half of the target frequency.
    Subsequent loops will produce a better set of PLL parameters, but for
    some of the iterations this can result in a division by 0.

    Fix it by clamping the divider value. We could instead restrict the E
    values being tested in the inner loop, but that would require additional
```

```
      calculation that would likely be less efficient as the E parameter can
      only take three different values.

      Fixes: c25c01361199 ("drm: rcar-du: lvds: D3/E3 support")
      Signed-off-by: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
      Reviewed-by: Kieran Bingham <kieran.bingham+renesas@ideasonboard.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 7beeeb717378f37cb8c61be2cc785bb37c817cbc
  Author: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
  Date:   Wed Mar 6 22:48:35 2019 +0200

      drm: rcar-du: lvds: Set LVEN and LVRES bits together on D3

      [ Upstream commit 00d082cc4ea6e42ec4fed832a1020231bb1ca150 ]

      On the D3 SoC the LVDS PHY must be enabled in the same register write
      that enables the LVDS output. Skip writing the LVEN bit independently
      on that platform, it will be set by the write that sets LVRES.

      Signed-off-by: Laurent Pinchart <laurent.pinchart+renesas@ideasonboard.com>
      Reviewed-by: Jacopo Mondi <jacopo+renesas@jmondi.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit fffcfb227ecf3fec958442965492a4f38384f00b
  Author: Aditya Pakki <pakki001@umn.edu>
  Date:   Mon Mar 25 16:25:22 2019 -0500

      thunderbolt: Fix to check the return value of kmemdup

      [ Upstream commit fd21b79e541e4666c938a344f3ad2df74b4f5120 ]

      uuid in add_switch is allocted via kmemdup which can fail. The patch
      logs the error and cleans up the allocated memory for switch.

      Signed-off-by: Aditya Pakki <pakki001@umn.edu>
      Reviewed-by: Mukesh Ojha <mojha@codeaurora.org>
      Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit f0cc2ffb37b407b79875d1ee9412f3639dc81f37
  Author: Kangjie Lu <kjlu@umn.edu>
  Date:   Mon Mar 25 15:23:08 2019 -0500

      thunderbolt: property: Fix a missing check of kzalloc

      [ Upstream commit 6183d5a51866f3acdeeb66b75e87d44025b01a55 ]

      No check is enforced for the return value of kzalloc,
      which may lead to NULL-pointer dereference.

      The patch fixes this issue.

      Signed-off-by: Kangjie Lu <kjlu@umn.edu>
      Reviewed-by: Mukesh Ojha <mojha@codeaurora.org>
      Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit d1d090c475348da13e3507e49a837c26632a1736
  Author: Alexandre Courbot <acourbot@chromium.org>
  Date:   Tue Mar 26 03:44:23 2019 -0400

      media: mtk-vcodec: fix access to incorrect planes member

      [ Upstream commit 52fafc58c3535c9f4f53864686dbaee3bcbadcb4 ]
```

```
    Commit 0650a91499e0 ("media: mtk-vcodec: Correct return type for mem2mem
    buffer helpers") fixed the return types for mem2mem buffer helper
    functions by changing a few local variables from vb2_buffer to
    vb2_v4l2_buffer. However, it left a few accesses to vb2_buffer::planes
    as-is, accidentally turning them into accesses to
    vb2_v4l2_buffer::planes and resulting in values being read from/written
    to the wrong place.

    Fix this by inserting vb2_buf into these accesses so they mimic their
    original behavior.

    Fixes: 0650a91499e0 ("media: mtk-vcodec: Correct return type for mem2mem buffer
helpers")

    Signed-off-by: Alexandre Courbot <acourbot@chromium.org>
    Reviewed-by: Ezequiel Garcia <ezequiel@collabora.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 44f212eb2aafbd2e62fad86d76200fdafe72cafd
Author: Ard Biesheuvel <ard.biesheuvel@linaro.org>
Date:   Thu Mar 28 20:34:26 2019 +0100

    efifb: Omit memory map check on legacy boot

    [ Upstream commit c2999c281ea2d2ebbdfce96cecc7b52e2ae7c406 ]

    Since the following commit:

      38ac0287b7f4 ("fbdev/efifb: Honour UEFI memory map attributes when mapping the FB")

    efifb_probe() checks its memory range via efi_mem_desc_lookup(),
    and this leads to a spurious error message:

       EFI_MEMMAP is not enabled

    at every boot on KVM.  This is quite annoying since the error message
    appears even if you set "quiet" boot option.

    Since this happens on legacy boot, which strangely enough exposes
    a EFI framebuffer via screen_info, let's double check that we are
    doing an EFI boot before attempting to access the EFI memory map.

    Reported-by: Takashi Iwai <tiwai@suse.de>
    Tested-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Ard Biesheuvel <ard.biesheuvel@linaro.org>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Matt Fleming <matt@codeblueprint.co.uk>
    Cc: Peter Jones <pjones@redhat.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: linux-efi@vger.kernel.org
    Link: http://lkml.kernel.org/r/20190328193429.21373-3-ard.biesheuvel@linaro.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b1c4294fe94e571fc24e41b17032521b4c0bc922
Author: Ezequiel Garcia <ezequiel@collabora.com>
Date:   Thu Feb 28 10:28:34 2019 -0500

    media: gspca: Kill URBs on USB device disconnect

    [ Upstream commit 9b9ea7c2b57a0c9c3341fc6db039d1f7971a432e ]
```

```
    In order to prevent ISOC URBs from being infinitely resubmitted,
    the driver's USB disconnect handler must kill all the in-flight URBs.

    While here, change the URB packet status message to a debug level,
    to avoid spamming the console too much.

    This commit fixes a lockup caused by an interrupt storm coming
    from the URB completion handler.

    Signed-off-by: Ezequiel Garcia <ezequiel@collabora.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7e8750d0c2b2e22b1fe9de69b698dfa682174fe9
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Tue Mar 26 01:12:07 2019 -0400

    media: wl128x: prevent two potential buffer overflows

    [ Upstream commit 9c2ccc324b3a6cbc865ab8b3e1a09e93d3c8ade9 ]

    Smatch marks skb->data as untrusted so it warns that "evt_hdr->dlen"
    can copy up to 255 bytes and we only have room for two bytes.  Even
    if this comes from the firmware and we trust it, the new policy
    generally is just to fix it as kernel hardenning.

    I can't test this code so I tried to be very conservative.  I considered
    not allowing "evt_hdr->dlen == 1" because it doesn't initialize the
    whole variable but in the end I decided to allow it and manually
    initialized "asic_id" and "asic_ver" to zero.

    Fixes: e8454ff7b9a4 ("[media] drivers:media:radio: wl128x: FM Driver Common sources")

    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ff560d4bba9cd32b3c02d0a816f1833227560a0f
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Sat Mar 9 02:20:56 2019 -0500

    media: video-mux: fix null pointer dereferences

    [ Upstream commit aeb0d0f581e2079868e64a2e5ee346d340376eae ]

    devm_kcalloc may fail and return a null pointer. The fix returns
    -ENOMEM upon failures to avoid null pointer dereferences.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Reviewed-by: Philipp Zabel <p.zabel@pengutronix.de>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2472b70889d535d56ae284345a1c51a725943504
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Sun Mar 17 14:02:31 2019 +0900

    kobject: Don't trigger kobject_uevent(KOBJ_REMOVE) twice.

    [ Upstream commit c03a0fd0b609e2f5c669c2b7f27c8e1928e9196e ]
```

```
    syzbot is hitting use-after-free bug in uinput module [1]. This is because
    kobject_uevent(KOBJ_REMOVE) is called again due to commit 0f4dafc0563c6c49
    ("Kobject: auto-cleanup on final unref") after memory allocation fault
    injection made kobject_uevent(KOBJ_REMOVE) from device_del() from
    input_unregister_device() fail, while uinput_destroy_device() is expecting
    that kobject_uevent(KOBJ_REMOVE) is not called after device_del() from
    input_unregister_device() completed.

    That commit intended to catch cases where nobody even attempted to send
    "remove" uevents. But there is no guarantee that an event will ultimately
    be sent. We are at the point of no return as far as the rest of the kernel
    is concerned; there are no repeats or do-overs.

    Also, it is not clear whether some subsystem depends on that commit.
    If no subsystem depends on that commit, it will be better to remove
    the state_{add,remove}_uevent_sent logic. But we don't want to risk
    a regression (in a patch which will be backported) by trying to remove
    that logic. Therefore, as a first step, let's avoid the use-after-free bug
    by making sure that kobject_uevent(KOBJ_REMOVE) won't be triggered twice.

    [1] https://syzkaller.appspot.com/bug?id=8b17c134fe938bbddd75a45afaa9e68af43a362d

    Reported-by: syzbot <syzbot+f648cfb7e0b52bf7ae32@syzkaller.appspotmail.com>
    Analyzed-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
    Fixes: 0f4dafc0563c6c49 ("Kobject: auto-cleanup on final unref")
    Cc: Kay Sievers <kay@vrfy.org>
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9965948a3bbcaccc1e9cf3d41e7c6675b2e1c032
Author: Oded Gabbay <oded.gabbay@gmail.com>
Date:   Sun Mar 31 21:37:42 2019 +0300

    habanalabs: prevent CPU soft lockup on Palladium

    [ Upstream commit e850b89f50d2c1439f58d547b888ee6e43312dea ]

    Unmapping ptes in the device MMU on Palladium can take a long time, which
    can cause a kernel BUG of CPU soft lockup.

    This patch minimize the chances for this bug by sleeping a little between
    unmapping ptes.

    Signed-off-by: Oded Gabbay <oded.gabbay@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1955cfc8164245cea716dbff43b3a641e01c6692
Author: Sowjanya Komatineni <skomatineni@nvidia.com>
Date:   Tue Mar 26 22:56:32 2019 -0700

    spi: tegra114: reset controller on probe

    [ Upstream commit 019194933339b3e9b486639c8cb3692020844d65 ]

    Fixes: SPI driver can be built as module so perform SPI controller reset
    on probe to make sure it is in valid state before initiating transfer.

    Signed-off-by: Sowjanya Komatineni <skomatineni@nvidia.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ddd85a2fe831f8e5b881d43dd9d2b784380950e9
Author: Hans de Goede <hdegoede@redhat.com>
Date:   Fri Mar 22 08:41:40 2019 +0100
```

```
    HID: logitech-hidpp: change low battery level threshold from 31 to 30 percent

    [ Upstream commit 1f87b0cd32b3456d7efdfb017fcf74d0bfe3ec29 ]

    According to hidpp20_batterylevel_get_battery_info my Logitech K270
    keyboard reports only 2 battery levels. This matches with what I've seen
    after testing with batteries at varying level of fullness, it always
    reports either 5% or 30%.

    Windows reports "battery good" for the 30% level. I've captured an USB
    trace of Windows reading the battery and it is getting the same info
    as the Linux hidpp code gets.

    Now that Linux handles these devices as hidpp devices, it reports the
    battery as being low as it treats anything under 31% as low, this leads
    to the user constantly getting a "Keyboard battery is low" warning from
    GNOME3, which is very annoying.

    This commit fixes this by changing the low threshold to anything under
    30%, which I assume is what Windows does.

    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Jiri Kosina <jkosina@suse.cz>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f67fd9a686a09a8f99e9260db7b4c32715cceaf2
Author: Takeshi Kihara <takeshi.kihara.df@renesas.com>
Date:   Fri Sep 28 16:33:06 2018 +0900

    clk: renesas: rcar-gen3: Correct parent clock of Audio-DMAC

    [ Upstream commit b9df2ea2b8d09ad850afe4d4a0403cb23d9e0c02 ]

    The clock sources of the AXI-bus clock (266.66 MHz) used for Audio-DMAC
    DMA transfers are:

        Channel        R-Car H3    R-Car M3-W    R-Car M3-N    R-Car E3
        ------------------------------------------------------------------
        Audio-DMAC0    S1D2        S1D2          S1D2          S1D2
        Audio-DMAC1    S1D2        S1D2          S1D2          -

    As a result, change the parent clocks of the Audio-DMAC{0,1} module
    clocks on R-Car H3, R-Car M3-W, and R-Car M3-N to S1D2, and change the
    parent clock of the Audio-DMAC0 module on R-Car E3 to S1D2.

    NOTE: This information will be reflected in a future revision of the
          R-Car Gen3 Hardware Manual.

    Signed-off-by: Takeshi Kihara <takeshi.kihara.df@renesas.com>
    [geert: Update R-Car D3, RZ/G2M, and RZ/G2E]
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Simon Horman <horms+renesas@verge.net.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 36478bcfcf86cd91bd12572fa9740fd4fd1a4444
Author: Ming Lei <ming.lei@redhat.com>
Date:   Fri Mar 29 15:07:54 2019 +0800

    block: pass page to xen_biovec_phys_mergeable

    [ Upstream commit 0383ad4374f7ad7edd925a2ee4753035c3f5508a ]

    xen_biovec_phys_mergeable() only needs .bv_page of the 2nd bio bvec
    for checking if the two bvecs can be merged, so pass page to
```

```
    xen_biovec_phys_mergeable() directly.

    No function change.

    Cc: ris Ostrovsky <boris.ostrovsky@oracle.com>
    Cc: Juergen Gross <jgross@suse.com>
    Cc: xen-devel@lists.xenproject.org
    Cc: Omar Sandoval <osandov@fb.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Boris Ostrovsky <boris.ostrovsky@oracle.com>
    Signed-off-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0aa9a8d4983dea88e181c519cd50544f62238ad8
Author: Ming Lei <ming.lei@redhat.com>
Date:   Sun Mar 17 18:01:04 2019 +0800

    block: avoid to break XEN by multi-page bvec

    [ Upstream commit db5ebd6edd2627d7e81a031643cf43587f63e66c ]

    XEN has special page merge requirement, see xen_biovec_phys_mergeable().
    We can't merge pages into one bvec simply for XEN.

    So move XEN's specific check on page merge into __bio_try_merge_page(),
    then abvoid to break XEN by multi-page bvec.

    Cc: ris Ostrovsky <boris.ostrovsky@oracle.com>
    Cc: xen-devel@lists.xenproject.org
    Cc: Omar Sandoval <osandov@fb.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Juergen Gross <jgross@suse.com>
    Signed-off-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 46a9cbe92c46d67c0369312845d17381abf20f0f
Author: Takeshi Kihara <takeshi.kihara.df@renesas.com>
Date:   Fri Sep 28 16:18:00 2018 +0900

    clk: renesas: rcar-gen3: Correct parent clock of SYS-DMAC

    [ Upstream commit 3c772f71a552d343a96868ed9a809f9047be94f5 ]

    The clock sources of the AXI BUS clock (266.66 MHz) used for SYS-DMAC
    DMA transfers are:

        Channel      R-Car H3    R-Car M3-W    R-Car M3-N
        ------------------------------------------------
        SYS-DMAC0    S0D3        S0D3          S0D3
        SYS-DMAC1    S3D1        S3D1          S3D1
        SYS-DMAC2    S3D1        S3D1          S3D1

    As a result, change the parent clocks of the SYS-DMAC{1,2} module clocks
    on R-Car H3, R-Car M3-W, and R-Car M3-N to S3D1.

    NOTE: This information will be reflected in a future revision of the
          R-Car Gen3 Hardware Manual.

    Signed-off-by: Takeshi Kihara <takeshi.kihara.df@renesas.com>
    [geert: Update RZ/G2M]
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Simon Horman <horms+renesas@verge.net.au>
```

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

commit c9e691c25d6392708a08bfdb21b90dfa0ef716c1
Author: Gustavo A. R. Silva <gustavo@embeddedor.com>
Date:   Fri Mar 29 10:27:26 2019 -0500

    cxgb3/l2t: Fix undefined behaviour

    [ Upstream commit 76497732932f15e7323dc805e8ea8dc11bb587cf ]

    The use of zero-sized array causes undefined behaviour when it is not
    the last member in a structure. As it happens to be in this case.

    Also, the current code makes use of a language extension to the C90
    standard, but the preferred mechanism to declare variable-length
    types such as this one is a flexible array member, introduced in
    C99:

    struct foo {
            int stuff;
            struct boo array[];
    };

    By making use of the mechanism above, we will get a compiler warning
    in case the flexible array does not occur last. Which is beneficial
    to cultivate a high-quality code.

    Fixes: e48f129c2f20 ("[SCSI] cxgb3i: convert cdev->l2opt to use rcu to prevent NULL
dereference")
    Signed-off-by: Gustavo A. R. Silva <gustavo@embeddedor.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 58492aad455755e0b4b29251600b8c304f6452e8
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Tue Feb 26 16:17:49 2019 +0800

    ASoC: wcd9335: fix a leaked reference by adding missing of_node_put

    [ Upstream commit 64b92de9603f22b5455da925ee57268ef7fb4e80 ]

    The call to of_parse_phandle returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./sound/soc/codecs/wcd9335.c:5193:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 5183, but without a correspon    ding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Liam Girdwood <lgirdwood@gmail.com>
    Cc: Mark Brown <broonie@kernel.org>
    Cc: Jaroslav Kysela <perex@perex.cz>
    Cc: Takashi Iwai <tiwai@suse.com>
    Cc: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Cc: Vinod Koul <vkoul@kernel.org>
    Cc: Dan Carpenter <dan.carpenter@oracle.com> (commit_signer:1/11=9%,authored:1/11=9%)
    Cc: alsa-devel@alsa-project.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f7d84b846b3716a8102e2ccfd8f4cea7c9c50e77
Author: Wen Yang <wen.yang99@zte.com.cn>
```

```
Date:   Tue Feb 26 16:17:50 2019 +0800

    ASoC: fsl_utils: fix a leaked reference by adding missing of_node_put

    [ Upstream commit c705247136a523488eac806bd357c3e5d79a7acd ]

    The call to of_parse_phandle returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./sound/soc/fsl/fsl_utils.c:74:2-8: ERROR: missing of_node_put; acquired a node pointer
with refcount incremented on line 38, but without a corresponding     object release within
this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Timur Tabi <timur@kernel.org>
    Cc: Nicolin Chen <nicoleotsuka@gmail.com>
    Cc: Xiubo Li <Xiubo.Lee@gmail.com>
    Cc: Fabio Estevam <festevam@gmail.com>
    Cc: Liam Girdwood <lgirdwood@gmail.com>
    Cc: Mark Brown <broonie@kernel.org>
    Cc: Jaroslav Kysela <perex@perex.cz>
    Cc: Takashi Iwai <tiwai@suse.com>
    Cc: alsa-devel@alsa-project.org
    Cc: linuxppc-dev@lists.ozlabs.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 28779829000bd062fb6260bf2c255e5e736d991e
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Tue Feb 26 16:17:51 2019 +0800

    ASoC: eukrea-tlv320: fix a leaked reference by adding missing of_node_put

    [ Upstream commit b820d52e7eed7b30b2dfef5f4213a2bc3cbea6f3 ]

    The call to of_parse_phandle returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./sound/soc/fsl/eukrea-tlv320.c:121:3-9: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 102, but without a correspo    nding object
release within this function.
    ./sound/soc/fsl/eukrea-tlv320.c:127:3-9: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 102, but without a correspo    nding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Liam Girdwood <lgirdwood@gmail.com>
    Cc: Mark Brown <broonie@kernel.org>
    Cc: Jaroslav Kysela <perex@perex.cz>
    Cc: Takashi Iwai <tiwai@suse.com>
    Cc: alsa-devel@alsa-project.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a30cdaf1eda802f6f7ef66fe7f33c21071d23b7a
Author: Nicolas Saenz Julienne <nsaenzjulienne@suse.de>
Date:   Wed Mar 27 11:18:48 2019 +0100

    HID: core: move Usage Page concatenation to Main item
```

```
    [ Upstream commit 58e75155009cc800005629955d3482f36a1e0eec ]

    As seen on some USB wireless keyboards manufactured by Primax, the HID
    parser was using some assumptions that are not always true. In this case
    it's s the fact that, inside the scope of a main item, an Usage Page
    will always precede an Usage.

    The spec is not pretty clear as 6.2.2.7 states "Any usage that follows
    is interpreted as a Usage ID and concatenated with the Usage Page".
    While 6.2.2.8 states "When the parser encounters a main item it
    concatenates the last declared Usage Page with a Usage to form a
    complete usage value." Being somewhat contradictory it was decided to
    match Window's implementation, which follows 6.2.2.8.

    In summary, the patch moves the Usage Page concatenation from the local
    item parsing function to the main item parsing function.

    Signed-off-by: Nicolas Saenz Julienne <nsaenzjulienne@suse.de>
    Reviewed-by: Terry Junge <terry.junge@poly.com>
    Signed-off-by: Benjamin Tissoires <benjamin.tissoires@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 653117ea9acbfbc80aed3a919c8f18679f2dbe42
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Mon Feb 11 13:58:43 2019 +0100

    sh: sh7786: Add explicit I/O cast to sh7786_mm_sel()

    [ Upstream commit 8440bb9b944c02222c7a840d406141ed42e945cd ]

    When compile-testing on arm:

        arch/sh/include/cpu-sh4/cpu/sh7786.h: In function ‘sh7786_mm_sel’:
        arch/sh/include/cpu-sh4/cpu/sh7786.h:135:21: warning: passing argument 1 of
‘__raw_readl’ makes pointer from integer without a cast [-Wint-conversion]
          return __raw_readl(0xFC400020) & 0x7;
                             ^~~~~~~~~~
        In file included from include/linux/io.h:25:0,
                         from arch/sh/include/cpu-sh4/cpu/sh7786.h:14,
                         from drivers/pinctrl/sh-pfc/pfc-sh7786.c:15:
        arch/arm/include/asm/io.h:113:21: note: expected ‘const volatile void *’ but
argument is of type ‘unsigned int’
         #define __raw_readl __raw_readl
                             ^
        arch/arm/include/asm/io.h:114:19: note: in expansion of macro ‘__raw_readl’
         static inline u32 __raw_readl(const volatile void __iomem *addr)
                           ^~~~~~~~~~

    __raw_readl() on SuperH is a macro that casts the passed I/O address to
    the correct type, while the implementations on most other architectures
    expect to be passed the correct pointer type.

    Add an explicit cast to fix this.

    Note that this also gets rid of a sparse warning on SuperH:

        arch/sh/include/cpu-sh4/cpu/sh7786.h:135:16: warning: incorrect type in argument 1
(different base types)
        arch/sh/include/cpu-sh4/cpu/sh7786.h:135:16:    expected void const volatile
[noderef] <asn:2>*<noident>
        arch/sh/include/cpu-sh4/cpu/sh7786.h:135:16:    got unsigned int

    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Reviewed-by: Simon Horman <horms+renesas@verge.net.au>
```

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

commit 75b841b1520dcae80e3d2bc4b55701d022f30757
Author: Leon Romanovsky <leonro@mellanox.com>
Date:   Tue Mar 19 11:10:08 2019 +0200

    RDMA/hns: Fix bad endianess of port_pd variable

    [ Upstream commit 6734b2973565e36659e97e12ab0d0faf1d9f3fbe ]

    port_pd is treated as le32 in declaration and read, fix assignment to be
    in le32 too. This change fixes the following compilation warnings.

    drivers/infiniband/hw/hns/hns_roce_ah.c:67:24: warning: incorrect type
    in assignment (different base types)
    drivers/infiniband/hw/hns/hns_roce_ah.c:67:24: expected restricted __le32 [usertype]
port_pd
    drivers/infiniband/hw/hns/hns_roce_ah.c:67:24: got restricted __be32 [usertype]

    Fixes: 9a4435375cd1 ("IB/hns: Add driver files for hns RoCE driver")
    Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
    Reviewed-by: Gal Pressman <galpress@amazon.com>
    Reviewed-by: Lijun Ou <ouliun@huawei.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c106ddc7f3c4d412aa3829ac12676e6d56f379d1
Author: Chengguang Xu <cgxu519@gmx.com>
Date:   Fri Feb 15 20:27:11 2019 +0800

    chardev: add additional check for minor range overlap

    [ Upstream commit de36e16d1557a0b6eb328bc3516359a12ba5c25c ]

    Current overlap checking cannot correctly handle
    a case which is baseminor < existing baseminor &&
    baseminor + minorct > existing baseminor + minorct.

    Signed-off-by: Chengguang Xu <cgxu519@gmx.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b086be4c5e57194f3ddb252ebcf851a22d9e379f
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Wed Apr 3 09:39:45 2019 +0200

    x86/uaccess: Fix up the fixup

    [ Upstream commit b69656fa7ea2f75e47d7bd5b9430359fa46488af ]

    New tooling got confused about this:

      arch/x86/lib/memcpy_64.o: warning: objtool: .fixup+0x7: return with UACCESS enabled

    While the code isn't wrong, it is tedious (if at all possible) to
    figure out what function a particular chunk of .fixup belongs to.

    This then confuses the objtool uaccess validation. Instead of
    returning directly from the .fixup, jump back into the right function.

    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
```

```
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 025c323c155cd9400a1c002aca17531ee547c586
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Mon Feb 25 12:56:35 2019 +0100

    x86/ia32: Fix ia32_restore_sigcontext() AC leak

    [ Upstream commit 67a0514afdbb8b2fc70b771b8c77661a9cb9d3a9 ]

    Objtool spotted that we call native_load_gs_index() with AC set.
    Re-arrange the code to avoid that.

    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1bd3284b25041869d57a9b005731e8d72d9f4a62
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Wed Apr 3 09:39:48 2019 +0200

    x86/uaccess, signal: Fix AC=1 bloat

    [ Upstream commit 88e4718275c1bddca6f61f300688b4553dc8584b ]

    Occasionally GCC is less agressive with inlining and the following is
    observed:

      arch/x86/kernel/signal.o: warning: objtool: restore_sigcontext()+0x3cc: call to
force_valid_ss.isra.5() with UACCESS enabled
      arch/x86/kernel/signal.o: warning: objtool: do_signal()+0x384: call to
frame_uc_flags.isra.0() with UACCESS enabled

    Cure this by moving this code out of the AC=1 region, since it really
    isn't needed for the user access.

    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Reviewed-by: Andy Lutomirski <luto@kernel.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3ff4740d2664fcf13900ec198a587b7c89a6e25c
Author: Johannes Berg <johannes.berg@intel.com>
Date:   Tue Feb 19 14:22:11 2019 +0100

    iwlwifi: mvm: IBSS: use BE FIFO for multicast

    [ Upstream commit 192a7e1f731fd9a64216cce35287eb23360437f6 ]

    Back in commit 4d339989acd7 ("iwlwifi: mvm: support ibss in dqa mode")
    we changed queue selection for IBSS to be:

        if (ieee80211_is_probe_resp(fc) || ieee80211_is_auth(fc) ||
```

```
            ieee80211_is_deauth(fc))
                return IWL_MVM_DQA_AP_PROBE_RESP_QUEUE;
        if (info->hw_queue == info->control.vif->cab_queue)
                return info->hw_queue;
        return IWL_MVM_DQA_AP_PROBE_RESP_QUEUE;
```

    Clearly, the thought at the time must've been that mac80211 will
    select the hw_queue as the cab_queue, so that we'll return and use
    that, where we store the multicast queue for IBSS. This, however,
    isn't true because mac80211 doesn't implement powersave for IBSS
    and thus selects the normal IBSS interface AC queue (best effort).

    This therefore always used the probe response queue, which maps to
    the BE FIFO.

    In commit cfbc6c4c5b91 ("iwlwifi: mvm: support mac80211 TXQs model")
    we rethought this code, and as a consequence now started mapping the
    multicast traffic to the multicast hardware queue since we no longer
    relied on mac80211 selecting the queue, doing it ourselves instead.
    This queue is mapped to the MCAST FIFO. however, this isn't actually
    enabled/controlled by the firmware in IBSS mode because we don't
    implement powersave, and frames from this queue can never go out in
    this case.

    Therefore, we got queue hang reports such as
    https://bugzilla.kernel.org/show_bug.cgi?id=201707

    Fix this by mapping the multicast queue to the BE FIFO in IBSS so
    that all the frames can go out.

    Fixes: cfbc6c4c5b91 ("iwlwifi: mvm: support mac80211 TXQs model")
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Luca Coelho <luciano.coelho@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b65b70ba068b7cdbfeb65eee87cce84a74618603
Author: Peter Zijlstra <peterz@infradead.org>
Date: Thu Mar 7 11:09:13 2019 +0100

    x86/uaccess, ftrace: Fix ftrace_likely_update() vs. SMAP

    [Upstream commit 4a6c91fbdef846ec7250b82f2eeeb87ac5f18cf9]

    For CONFIG_TRACE_BRANCH_PROFILING=y the likely/unlikely things get
    overloaded and generate callouts to this code, and thus also when
    AC=1.

    Make it safe.

    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 38e068cd4847dc0eaa0d73d8a56857c3d68cda4d
Author: Lior David <liord@codeaurora.org>
Date: Thu Feb 28 11:35:01 2019 +0200

    wil6210: fix return code of wmi_mgmt_tx and wmi_mgmt_tx_ext

```
    [ Upstream commit 49122ec42634f73babb1dc96f170023e5228d080 ]

    The functions that send management TX frame have 3 possible
    results: success and other side acknowledged receive (ACK=1),
    success and other side did not acknowledge receive(ACK=0) and
    failure to send the frame. The current implementation
    incorrectly reports the ACK=0 case as failure.

    Signed-off-by: Lior David <liord@codeaurora.org>
    Signed-off-by: Maya Erez <merez@codeaurora.org>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 68cccec09c25cde0915482370b3b385008573148
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Tue Mar 19 13:18:56 2019 +0100

    locking/static_key: Fix false positive warnings on concurrent dec/inc

    [ Upstream commit a1247d06d01045d7ab2882a9c074fbf21137c690 ]

    Even though the atomic_dec_and_mutex_lock() in
    __static_key_slow_dec_cpuslocked() can never see a negative value in
    key->enabled the subsequent sanity check is re-reading key->enabled, which may
    have been set to -1 in the meantime by static_key_slow_inc_cpuslocked().

                    CPU  A                                  CPU B

     __static_key_slow_dec_cpuslocked():          static_key_slow_inc_cpuslocked():
                                  # enabled = 1
       atomic_dec_and_mutex_lock()
                                  # enabled = 0
                                                  atomic_read() == 0
                                                  atomic_set(-1)
                                  # enabled = -1
       val = atomic_read()
       # Oops - val == -1!

    The test case is TCP's clean_acked_data_enable() / clean_acked_data_disable()
    as tickled by KTLS (net/ktls).

    Suggested-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Reported-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Tested-by: Jakub Kicinski <jakub.kicinski@netronome.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Andrew Morton <akpm@linux-foundation.org>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Paul E. McKenney <paulmck@linux.vnet.ibm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: ard.biesheuvel@linaro.org
    Cc: oss-drivers@netronome.com
    Cc: pbonzini@redhat.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2f5decc2aa13c7d1670a35705d9aebdc5fb7c9e9
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Tue Mar 5 19:34:05 2019 +0800

    arm64: cpu_ops: fix a leaked reference by adding missing of_node_put

    [ Upstream commit 92606ec9285fb84cd9b5943df23f07d741384bfc ]
```

```
    The call to of_get_next_child returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
      ./arch/arm64/kernel/cpu_ops.c:102:1-7: ERROR: missing of_node_put;
      acquired a node pointer with refcount incremented on line 69, but
      without a corresponding object release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: linux-arm-kernel@lists.infradead.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fcff7bdf15232f2329323446180763c7e213d976
Author: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
Date:   Mon Mar 11 10:33:35 2019 -0400

    drm/amd/display: Prevent cursor hotspot overflow for RV overlay planes

    [ Upstream commit 6752bea8b03e77c98be7d8d25b0a9d86a00b3cf7 ]

    [Why]
    The actual position for the cursor on the screen is essentially:

    x_out = x - x_plane - x_hotspot
    y_out = y - y_plane - y_hotspot

    The register values for cursor position and cursor hotspot need to be
    greater than zero when programmed, but we also need to subtract off
    the plane position to display the cursor at the correct position.

    Since we don't want x or y to be less than zero, we add the plane
    position as a positive value to x_hotspot or y_hotspot. However, what
    this doesn't take into account is that the hotspot registers are limited
    by the maximum cursor size.

    On DCN10 the cursor hotspot regitsers are masked to 0xFF, so they have
    a maximum value of 0-255. Values greater this will wrap, causing the
    cursor to display in the wrong position.

    In practice this means that for sufficiently large plane positions, the
    cursor will be drawn twice on the screen, and can cause screen flashes
    or p-state WARNS depending on what the wrapped value is.

    So we need a way to remove the value from x_plane and y_plane without
    exceeding the maximum cursor size.

    [How]
    Subtract as much as x_plane/y_plane as possible from x and y and place
    the remainder in the cursor hotspot register.

    The value for x_hotspot and y_hotspot can still wrap around but it
    won't happen in a case where the cursor is actually enabled.

    The cursor plane needs to intersect at least one pixel of the plane's
    rectangle to be enabled, so the cursor position + hotspot provided by
    userspace must always be strictly less than the maximum cursor size for
    the cursor to actually be enabled.

    Signed-off-by: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
```

```
    Reviewed-by: Sun peng Li <Sunpeng.Li@amd.com>
    Acked-by: Bhawanpreet Lakha <Bhawanpreet.Lakha@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a9b2666bcaf39c47e6d1f18b65ab2c26aaa15c63
Author: Yannick Fertré <yannick.fertre@st.com>
Date:   Thu Mar 21 09:04:05 2019 +0100

    drm/panel: otm8009a: Add delay at the end of initialization

    [ Upstream commit 0084c3c71126fc878c6dab8a6ab8ecc484c2be02 ]

    At the end of initialization, a delay is required by the panel. Without
    this delay, the panel could received a frame early & generate a crash of
    panel (black screen).

    Signed-off-by: Yannick Fertré <yannick.fertre@st.com>
    Reviewed-by: Philippe Cornu <philippe.cornu@st.com>
    Tested-by: Philippe Cornu <philippe.cornu@st.com>
    Signed-off-by: Thierry Reding <treding@nvidia.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/1553155445-13407-1-git-send-email-
yannick.fertre@st.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 81ba6b9ddcc3ea0921cf85c4c88fec46f560c2ad
Author: Stanley Chu <stanley.chu@mediatek.com>
Date:   Thu Mar 28 17:16:24 2019 +0800

    scsi: ufs: Avoid configuring regulator with undefined voltage range

    [ Upstream commit 3b141e8cfd54ba3e5c610717295b2a02aab26a05 ]

    For regulators used by UFS, vcc, vccq and vccq2 will have voltage range
    initialized by ufshcd_populate_vreg(), however other regulators may have
    undefined voltage range if dt-bindings have no such definition.

    In above undefined case, both "min_uV" and "max_uV" fields in ufs_vreg
    struct will be zero values and these values will be configured on
    regulators in different power modes.

    Currently this may have no harm if both "min_uV" and "max_uV" always keep
    "zero values" because regulator_set_voltage() will always bypass such
    invalid values and return "good" results.

    However improper values shall be fixed to avoid potential bugs.  Simply
    bypass voltage configuration if voltage range is not defined.

    Signed-off-by: Stanley Chu <stanley.chu@mediatek.com>
    Reviewed-by: Avri Altman <avri.altman@wdc.com>
    Acked-by: Alim Akhtar <alim.akhtar@samsung.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d59bc35df0ffdd56c44294af192b15ba2936bf6e
Author: Stanley Chu <stanley.chu@mediatek.com>
Date:   Thu Mar 28 17:16:25 2019 +0800

    scsi: ufs: Fix regulator load and icc-level configuration

    [ Upstream commit 0487fff76632ec023d394a05b82e87a971db8c03 ]

    Currently if a regulator has "<name>-fixed-regulator" property in device
    tree, it will skip current limit initialization.  This lead to a zero
    "max_uA" value in struct ufs_vreg.
```

```
    However, "regulator_set_load" operation shall be required on regulators
    which have valid current limits, otherwise a zero "max_uA" set by
    "regulator_set_load" may cause unexpected behavior when this regulator is
    enabled or set as high power mode.

    Similarly, in device's icc_level configuration flow, the target icc_level
    shall be updated if regulator also has valid current limit, otherwise a
    wrong icc_level will be calculated by zero "max_uA" and thus causes
    unexpected results after it is written to device.

    Signed-off-by: Stanley Chu <stanley.chu@mediatek.com>
    Reviewed-by: Avri Altman <avri.altman@wdc.com>
    Acked-by: Alim Akhtar <alim.akhtar@samsung.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a7704ab60c30f2de92e3cb183895f8d41b0b4d88
Author: Ping-Ke Shih <pkshih@realtek.com>
Date:   Tue Mar 12 17:06:48 2019 +0800

    rtlwifi: fix potential NULL pointer dereference

    [ Upstream commit 60209d482b97743915883d293c8b85226d230c19 ]

    In case dev_alloc_skb fails, the fix safely returns to avoid
    potential NULL pointer dereference.

    Signed-off-by: Ping-Ke Shih <pkshih@realtek.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 91b6a56451fdf7e881df7ab6512be0287c0948b7
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Wed Apr 3 16:46:56 2019 +0200

    spi: Add missing error handling for CS GPIOs

    [ Upstream commit 1723fdec5fcbc4de3d26bbb23a9e1704ee258955 ]

    While devm_gpiod_get_index_optional() returns NULL if the GPIO is not
    present (i.e. -ENOENT), it may still return other error codes, like
    -EPROBE_DEFER.  Currently these are not handled, leading to
    unrecoverable failures later in case of probe deferral:

        gpiod_set_consumer_name: invalid GPIO (errorpointer)
        gpiod_direction_output: invalid GPIO (errorpointer)
        gpiod_set_value_cansleep: invalid GPIO (errorpointer)
        gpiod_set_value_cansleep: invalid GPIO (errorpointer)
        gpiod_set_value_cansleep: invalid GPIO (errorpointer)

    Detect and propagate errors to fix this.

    Fixes: f3186dd876697e69 ("spi: Optionally use GPIO descriptors for CS GPIOs")
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3c328af802c679f2fe556b1869a7d63988bda826
Author: Alexandre Belloni <alexandre.belloni@bootlin.com>
Date:   Wed Mar 20 13:32:27 2019 +0100

    rtc: xgene: fix possible race condition

    [ Upstream commit a652e00ee1233e251a337c28e18a1da59224e5ce ]
```

```
    The IRQ is requested before the struct rtc is allocated and registered, but
    this struct is used in the IRQ handler. This may lead to a NULL pointer
    dereference.

    Switch to devm_rtc_allocate_device/rtc_register_device to allocate the rtc
    struct before requesting the IRQ.

    Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 335e7c032edeaa65475572cfe80151bcadce902e
Author: Piotr Figiel <p.figiel@camlintechnologies.com>
Date:   Wed Mar 13 09:52:01 2019 +0000

    brcmfmac: fix Oops when bringing up interface during USB disconnect

    [ Upstream commit 24d413a31afaee9bbbf79226052c386b01780ce2 ]

    Fix a race which leads to an Oops with NULL pointer dereference.  The
    dereference is in brcmf_config_dongle() when cfg_to_ndev() attempts to get
    net_device structure of interface with index 0 via if2bss mapping. This
    shouldn't fail because of check for bus being ready in brcmf_netdev_open(),
    but it's not synchronised with USB disconnect and there is a race: after
    the check the bus can be marked down and the mapping for interface 0 may be
    gone.

    Solve this by modifying disconnect handling so that the removal of mapping
    of ifidx to brcmf_if structure happens after netdev removal (which is
    synchronous with brcmf_netdev_open() thanks to rtln being locked in
    devinet_ioctl()). This assures brcmf_netdev_open() returns before the
    mapping is removed during disconnect.

    Unable to handle kernel NULL pointer dereference at virtual address 00000008
    pgd = bcae2612
    [00000008] *pgd=8be73831
    Internal error: Oops: 17 [#1] PREEMPT SMP ARM
    Modules linked in: brcmfmac brcmutil nf_log_ipv4 nf_log_common xt_LOG xt_limit
    iptable_mangle xt_connmark xt_tcpudp xt_conntrack nf_conntrack nf_defrag_ipv6
    nf_defrag_ipv4 iptable_filter ip_tables x_tables usb_f_mass_storage usb_f_rndis
    u_ether usb_serial_simple usbserial cdc_acm smsc95xx usbnet ci_hdrc_imx ci_hdrc
    usbmisc_imx ulpi 8250_exar 8250_pci 8250 8250_base libcomposite configfs
    udc_core [last unloaded: brcmutil]
    CPU: 2 PID: 24478 Comm: ifconfig Not tainted 4.19.23-00078-ga62866d-dirty #115
    Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
    PC is at brcmf_cfg80211_up+0x94/0x29c [brcmfmac]
    LR is at brcmf_cfg80211_up+0x8c/0x29c [brcmfmac]
    pc : [<7f26a91c>]    lr : [<7f26a914>]    psr: a0070013
    sp : eca99d28  ip : 00000000  fp : ee9c6c00
    r10: 00000036  r9 : 00000000  r8 : ece4002c
    r7 : edb5b800  r6 : 00000000  r5 : 80f08448  r4 : edb5b968
    r3 : ffffffff  r2 : 00000000  r1 : 00000002  r0 : 00000000
    Flags: NzCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
    Control: 10c5387d  Table: 7ca0c04a  DAC: 00000051
    Process ifconfig (pid: 24478, stack limit = 0xd9e85a0e)
    Stack: (0xeca99d28 to 0xeca9a000)
    9d20:                   00000000 80f873b0 0000000d 80f08448 eca99d68 50d45f32
    9d40: 7f27de94 ece40000 80f08448 80f08448 7f27de94 ece4002c 00000000 00000036
    9d60: ee9c6c00 7f27262c 00001002 50d45f32 ece40000 00000000 80f08448 80772008
    9d80: 00000001 00001043 00001002 ece40000 00000000 50d45f32 ece40000 00000001
    9da0: 80f08448 00001043 00001002 807723d0 00000000 50d45f32 80f08448 eca99e58
    9dc0: 80f87113 50d45f32 80f08448 ece40000 ece40138 00001002 80f08448 00000000
    9de0: 00000000 80772434 edbd5380 eca99e58 edbd5380 80f08448 ee9c6c0c 80805f70
    9e00: 00000000 ede08e00 00008914 ece40000 00000014 ee9c6c0c 600c0013 00001043
    9e20: 0208a8c0 ffffffff 00000000 50d45f32 eca98000 80f08448 7ee9fc38 00008914
```

```
    9e40: 80f68e40 00000051 eca98000 00000036 00000003 80808b9c 6e616c77 00000030
    9e60: 00000000 00000000 00001043 0208a8c0 ffffffff 00000000 80f08448 00000000
    9e80: 00000000 816d8b20 600c0013 00000001 ede09320 801763d4 00000000 50d45f32
    9ea0: eca98000 80f08448 7ee9fc38 50d45f32 00008914 80f08448 7ee9fc38 80f68e40
    9ec0: ed531540 8074721c 00000800 00000001 00000000 6e616c77 00000030 00000000
    9ee0: 00000000 00001002 0208a8c0 ffffffff 00000000 50d45f32 80f08448 7ee9fc38
    9f00: ed531560 ec8fc900 80285a6c 80285138 edb910c0 00000000 ecd91008 ede08e00
    9f20: 80f08448 00000000 00000000 816d8b20 600c0013 00000001 ede09320 801763d4
    9f40: 00000000 50d45f32 00021000 edb91118 edb910c0 80f08448 01b29000 edb91118
    9f60: eca99f7c 50d45f32 00021000 ec8fc900 00000003 ec8fc900 00008914 7ee9fc38
    9f80: eca98000 00000036 00000003 80285a6c 00086364 7ee9fe1c 000000c3 00000036
    9fa0: 801011c4 80101000 00086364 7ee9fe1c 00000003 00008914 7ee9fc38 00086364
    9fc0: 00086364 7ee9fe1c 000000c3 00000036 0008630c 7ee9fe1c 7ee9fc38 00000003
    9fe0: 000a42b8 7ee9fbd4 00019914 76e09acc 600c0010 00000003 00000000 00000000
    [<7f26a91c>] (brcmf_cfg80211_up [brcmfmac]) from [<7f27262c>]
(brcmf_netdev_open+0x74/0xe8 [brcmfmac])
    [<7f27262c>] (brcmf_netdev_open [brcmfmac]) from [<80772008>] (__dev_open+0xcc/0x150)
    [<80772008>] (__dev_open) from [<807723d0>] (__dev_change_flags+0x168/0x1b4)
    [<807723d0>] (__dev_change_flags) from [<80772434>] (dev_change_flags+0x18/0x48)
    [<80772434>] (dev_change_flags) from [<80805f70>] (devinet_ioctl+0x67c/0x79c)
    [<80805f70>] (devinet_ioctl) from [<80808b9c>] (inet_ioctl+0x210/0x3d4)
    [<80808b9c>] (inet_ioctl) from [<8074721c>] (sock_ioctl+0x350/0x524)
    [<8074721c>] (sock_ioctl) from [<80285138>] (do_vfs_ioctl+0xb0/0x9b0)
    [<80285138>] (do_vfs_ioctl) from [<80285a6c>] (ksys_ioctl+0x34/0x5c)
    [<80285a6c>] (ksys_ioctl) from [<80101000>] (ret_fast_syscall+0x0/0x28)
    Exception stack(0xeca99fa8 to 0xeca99ff0)
    9fa0:                   00086364 7ee9fe1c 00000003 00008914 7ee9fc38 00086364
    9fc0: 00086364 7ee9fe1c 000000c3 00000036 0008630c 7ee9fe1c 7ee9fc38 00000003
    9fe0: 000a42b8 7ee9fbd4 00019914 76e09acc
    Code: e5970328 eb002021 e1a02006 e3a01002 (e5909008)
    ---[ end trace 5cbac2333f3ac5df ]---

    Signed-off-by: Piotr Figiel <p.figiel@camlintechnologies.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9429cd8ca96574a161316b45bcf618ca92ad4f50
Author: Piotr Figiel <p.figiel@camlintechnologies.com>
Date:   Fri Mar 8 15:25:04 2019 +0000

    brcmfmac: fix race during disconnect when USB completion is in progress

    [ Upstream commit db3b9e2e1d58080d0754bdf9293dabf8c6491b67 ]

    It was observed that rarely during USB disconnect happening shortly after
    connect (before full initialization completes) usb_hub_wq would wait
    forever for the dev_init_lock to be unlocked. dev_init_lock would remain
    locked though because of infinite wait during usb_kill_urb:

    [ 2730.656472] kworker/0:2     D    0   260      2 0x00000000
    [ 2730.660700] Workqueue: events request_firmware_work_func
    [ 2730.664807] [<809dca20>] (__schedule) from [<809dd164>] (schedule+0x4c/0xac)
    [ 2730.670587] [<809dd164>] (schedule) from [<8069af44>] (usb_kill_urb+0xdc/0x114)
    [ 2730.676815] [<8069af44>] (usb_kill_urb) from [<7f258b50>] (brcmf_usb_free_q+0x34/0xa8
[brcmfmac])
    [ 2730.684833] [<7f258b50>] (brcmf_usb_free_q [brcmfmac]) from [<7f2517d4>]
(brcmf_detach+0xa0/0xb8 [brcmfmac])
    [ 2730.693557] [<7f2517d4>] (brcmf_detach [brcmfmac]) from [<7f251a34>]
(brcmf_attach+0xac/0x3d8 [brcmfmac])
    [ 2730.702094] [<7f251a34>] (brcmf_attach [brcmfmac]) from [<7f2587ac>]
(brcmf_usb_probe_phase2+0x468/0x4a0 [brcmfmac])
    [ 2730.711601] [<7f2587ac>] (brcmf_usb_probe_phase2 [brcmfmac]) from [<7f252888>]
(brcmf_fw_request_done+0x194/0x220 [brcmfmac])
    [ 2730.721795] [<7f252888>] (brcmf_fw_request_done [brcmfmac]) from [<805748e4>]
(request_firmware_work_func+0x4c/0x88)
```

```
      [ 2730.731125] [<805748e4>] (request_firmware_work_func) from [<80141474>]
  (process_one_work+0x228/0x808)
      [ 2730.739223] [<80141474>] (process_one_work) from [<80141a80>]
  (worker_thread+0x2c/0x564)
      [ 2730.746105] [<80141a80>] (worker_thread) from [<80147bcc>] (kthread+0x13c/0x16c)
      [ 2730.752227] [<80147bcc>] (kthread) from [<801010b4>] (ret_from_fork+0x14/0x20)

      [ 2733.099695] kworker/0:3     D    0  1065      2 0x00000000
      [ 2733.103926] Workqueue: usb_hub_wq hub_event
      [ 2733.106914] [<809dca20>] (__schedule) from [<809dd164>] (schedule+0x4c/0xac)
      [ 2733.112693] [<809dd164>] (schedule) from [<809e2a8c>] (schedule_timeout+0x214/0x3e4)
      [ 2733.119621] [<809e2a8c>] (schedule_timeout) from [<809dde2c>]
  (wait_for_common+0xc4/0x1c0)
      [ 2733.126810] [<809dde2c>] (wait_for_common) from [<7f258d00>]
  (brcmf_usb_disconnect+0x1c/0x4c [brcmfmac])
      [ 2733.135206] [<7f258d00>] (brcmf_usb_disconnect [brcmfmac]) from [<8069e0c8>]
  (usb_unbind_interface+0x5c/0x1e4)
      [ 2733.143943] [<8069e0c8>] (usb_unbind_interface) from [<8056d3e8>]
  (device_release_driver_internal+0x164/0x1fc)
      [ 2733.152769] [<8056d3e8>] (device_release_driver_internal) from [<8056c078>]
  (bus_remove_device+0xd0/0xfc)
      [ 2733.161138] [<8056c078>] (bus_remove_device) from [<8056977c>]
  (device_del+0x11c/0x310)
      [ 2733.167939] [<8056977c>] (device_del) from [<8069cba8>]
  (usb_disable_device+0xa0/0x1cc)
      [ 2733.174743] [<8069cba8>] (usb_disable_device) from [<8069507c>]
  (usb_disconnect+0x74/0x1dc)
      [ 2733.181823] [<8069507c>] (usb_disconnect) from [<80695e88>] (hub_event+0x478/0xf88)
      [ 2733.188278] [<80695e88>] (hub_event) from [<80141474>] (process_one_work+0x228/0x808)
      [ 2733.194905] [<80141474>] (process_one_work) from [<80141a80>]
  (worker_thread+0x2c/0x564)
      [ 2733.201724] [<80141a80>] (worker_thread) from [<80147bcc>] (kthread+0x13c/0x16c)
      [ 2733.207913] [<80147bcc>] (kthread) from [<801010b4>] (ret_from_fork+0x14/0x20)

      It was traced down to a case where usb_kill_urb would be called on an URB
      structure containing more or less random data, including large number in
      its use_count. During the debugging it appeared that in brcmf_usb_free_q()
      the traversal over URBs' lists is not synchronized with operations on those
      lists in brcmf_usb_rx_complete() leading to handling
      brcmf_usbdev_info structure (holding lists' head) as lists' element and in
      result causing above problem.

      Fix it by walking through all URBs during brcmf_cancel_all_urbs using the
      arrays of requests instead of linked lists.

      Signed-off-by: Piotr Figiel <p.figiel@camlintechnologies.com>
      Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit d3e2c7b59816fe55472e47949641a2e2f837b084
  Author: Piotr Figiel <p.figiel@camlintechnologies.com>
  Date:   Mon Mar 4 15:42:49 2019 +0000

      brcmfmac: fix WARNING during USB disconnect in case of unempty psq

      [ Upstream commit c80d26e81ef1802f30364b4ad1955c1443a592b9 ]

      brcmu_pkt_buf_free_skb emits WARNING when attempting to free a sk_buff
      which is part of any queue. After USB disconnect this may have happened
      when brcmf_fws_hanger_cleanup() is called as per-interface psq was never
      cleaned when removing the interface.
      Change brcmf_fws_macdesc_cleanup() in a way that it removes the
      corresponding packets from hanger table (to avoid double-free when
      brcmf_fws_hanger_cleanup() is called) and add a call to clean-up the
      interface specific packet queue.
```

```
    Below is a WARNING during USB disconnect with Raspberry Pi WiFi dongle
    running in AP mode. This was reproducible when the interface was
    transmitting during the disconnect and is fixed with this commit.

    ------------[ cut here ]------------
    WARNING: CPU: 0 PID: 1171 at drivers/net/wireless/broadcom/brcm80211/brcmutil/utils.c:49
brcmu_pkt_buf_free_skb+0x3c/0x40
    Modules linked in: nf_log_ipv4 nf_log_common xt_LOG xt_limit iptable_mangle xt_connmark
xt_tcpudp xt_conntrack nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 iptable_filter ip_tables
x_tables usb_f_mass_storage usb_f_rndis u_ether cdc_acm smsc95xx usbnet ci_hdrc_imx ci_hdrc
ulpi usbmisc_imx 8250_exar 8250_pci 8250 8250_base libcomposite configfs udc_core
    CPU: 0 PID: 1171 Comm: kworker/0:0 Not tainted 4.19.23-00075-gde33ed8 #99
    Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
    Workqueue: usb_hub_wq hub_event
    [<8010ff84>] (unwind_backtrace) from [<8010bb64>] (show_stack+0x10/0x14)
    [<8010bb64>] (show_stack) from [<80840278>] (dump_stack+0x88/0x9c)
    [<80840278>] (dump_stack) from [<8011f5ec>] (__warn+0xfc/0x114)
    [<8011f5ec>] (__warn) from [<8011f71c>] (warn_slowpath_null+0x40/0x48)
    [<8011f71c>] (warn_slowpath_null) from [<805a476c>] (brcmu_pkt_buf_free_skb+0x3c/0x40)
    [<805a476c>] (brcmu_pkt_buf_free_skb) from [<805bb6c4>] (brcmf_fws_cleanup+0x1e4/0x22c)
    [<805bb6c4>] (brcmf_fws_cleanup) from [<805bc854>] (brcmf_fws_del_interface+0x58/0x68)
    [<805bc854>] (brcmf_fws_del_interface) from [<805b66ac>]
(brcmf_remove_interface+0x40/0x150)
    [<805b66ac>] (brcmf_remove_interface) from [<805b6870>] (brcmf_detach+0x6c/0xb0)
    [<805b6870>] (brcmf_detach) from [<805bdbb8>] (brcmf_usb_disconnect+0x30/0x4c)
    [<805bdbb8>] (brcmf_usb_disconnect) from [<805e5d64>] (usb_unbind_interface+0x5c/0x1e0)
    [<805e5d64>] (usb_unbind_interface) from [<804aab10>]
(device_release_driver_internal+0x154/0x1ec)
    [<804aab10>] (device_release_driver_internal) from [<804a97f4>]
(bus_remove_device+0xcc/0xf8)
    [<804a97f4>] (bus_remove_device) from [<804a6fc0>] (device_del+0x118/0x308)
    [<804a6fc0>] (device_del) from [<805e488c>] (usb_disable_device+0xa0/0x1c8)
    [<805e488c>] (usb_disable_device) from [<805dcf98>] (usb_disconnect+0x70/0x1d8)
    [<805dcf98>] (usb_disconnect) from [<805ddd84>] (hub_event+0x464/0xf50)
    [<805ddd84>] (hub_event) from [<80135a70>] (process_one_work+0x138/0x3f8)
    [<80135a70>] (process_one_work) from [<80135d5c>] (worker_thread+0x2c/0x554)
    [<80135d5c>] (worker_thread) from [<8013b1a0>] (kthread+0x124/0x154)
    [<8013b1a0>] (kthread) from [<801010e8>] (ret_from_fork+0x14/0x2c)
    Exception stack(0xecf8dfb0 to 0xecf8dff8)
    dfa0:                                     00000000 00000000 00000000 00000000
    dfc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    dfe0: 00000000 00000000 00000000 00000000 00000013 00000000
    ---[ end trace 38d234018e9e2a90 ]---
    ------------[ cut here ]------------

    Signed-off-by: Piotr Figiel <p.figiel@camlintechnologies.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 100ba77f36899245dcddffc9b9bb05885ab94e8d
Author: Piotr Figiel <p.figiel@camlintechnologies.com>
Date:   Wed Mar 13 09:52:42 2019 +0000

    brcmfmac: convert dev_init_lock mutex to completion

    [ Upstream commit a9fd0953fa4a62887306be28641b4b0809f3b2fd ]

    Leaving dev_init_lock mutex locked in probe causes BUG and a WARNING when
    kernel is compiled with CONFIG_PROVE_LOCKING. Convert mutex to completion
    which silences those warnings and improves code readability.

    Fix below errors when connecting the USB WiFi dongle:

    brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac43143 for chip BCM43143/2
```

```
BUG: workqueue leaked lock or atomic: kworker/0:2/0x00000000/434
     last function: hub_event
1 lock held by kworker/0:2/434:
 #0: 18d5dcdf (&devinfo->dev_init_lock){+.+.}, at: brcmf_usb_probe+0x78/0x550 [brcmfmac]
CPU: 0 PID: 434 Comm: kworker/0:2 Not tainted 4.19.23-00084-g454a789-dirty #123
Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
Workqueue: usb_hub_wq hub_event
[<8011237c>] (unwind_backtrace) from [<8010d74c>] (show_stack+0x10/0x14)
[<8010d74c>] (show_stack) from [<809c4324>] (dump_stack+0xa8/0xd4)
[<809c4324>] (dump_stack) from [<8014195c>] (process_one_work+0x710/0x808)
[<8014195c>] (process_one_work) from [<80141a80>] (worker_thread+0x2c/0x564)
[<80141a80>] (worker_thread) from [<80147bcc>] (kthread+0x13c/0x16c)
[<80147bcc>] (kthread) from [<801010b4>] (ret_from_fork+0x14/0x20)
Exception stack(0xed1d9fb0 to 0xed1d9ff8)
9fa0:                                     00000000 00000000 00000000 00000000
9fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
9fe0: 00000000 00000000 00000000 00000000 00000013 00000000

======================================================
WARNING: possible circular locking dependency detected
4.19.23-00084-g454a789-dirty #123 Not tainted
------------------------------------------------------
kworker/0:2/434 is trying to acquire lock:
e29cf799 ((wq_completion)"events"){+.+.}, at: process_one_work+0x174/0x808

but task is already holding lock:
18d5dcdf (&devinfo->dev_init_lock){+.+.}, at: brcmf_usb_probe+0x78/0x550 [brcmfmac]

which lock already depends on the new lock.

the existing dependency chain (in reverse order) is:

-> #2 (&devinfo->dev_init_lock){+.+.}:
       mutex_lock_nested+0x1c/0x24
       brcmf_usb_probe+0x78/0x550 [brcmfmac]
       usb_probe_interface+0xc0/0x1bc
       really_probe+0x228/0x2c0
       __driver_attach+0xe4/0xe8
       bus_for_each_dev+0x68/0xb4
       bus_add_driver+0x19c/0x214
       driver_register+0x78/0x110
       usb_register_driver+0x84/0x148
       process_one_work+0x228/0x808
       worker_thread+0x2c/0x564
       kthread+0x13c/0x16c
       ret_from_fork+0x14/0x20
         (null)

-> #1 (brcmf_driver_work){+.+.}:
       worker_thread+0x2c/0x564
       kthread+0x13c/0x16c
       ret_from_fork+0x14/0x20
         (null)

-> #0 ((wq_completion)"events"){+.+.}:
       process_one_work+0x1b8/0x808
       worker_thread+0x2c/0x564
       kthread+0x13c/0x16c
       ret_from_fork+0x14/0x20
         (null)

other info that might help us debug this:

Chain exists of:
  (wq_completion)"events" --> brcmf_driver_work --> &devinfo->dev_init_lock
```

```
      Possible unsafe locking scenario:

            CPU0                    CPU1
            ----                    ----
       lock(&devinfo->dev_init_lock);
                                    lock(brcmf_driver_work);
                                    lock(&devinfo->dev_init_lock);
       lock((wq_completion)"events");

      *** DEADLOCK ***

     1 lock held by kworker/0:2/434:
      #0: 18d5dcdf (&devinfo->dev_init_lock){+.+.}, at: brcmf_usb_probe+0x78/0x550 [brcmfmac]

     stack backtrace:
     CPU: 0 PID: 434 Comm: kworker/0:2 Not tainted 4.19.23-00084-g454a789-dirty #123
     Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
     Workqueue: events request_firmware_work_func
     [<8011237c>] (unwind_backtrace) from [<8010d74c>] (show_stack+0x10/0x14)
     [<8010d74c>] (show_stack) from [<809c4324>] (dump_stack+0xa8/0xd4)
     [<809c4324>] (dump_stack) from [<80172838>] (print_circular_bug+0x210/0x330)
     [<80172838>] (print_circular_bug) from [<80175940>] (__lock_acquire+0x160c/0x1a30)
     [<80175940>] (__lock_acquire) from [<8017671c>] (lock_acquire+0xe0/0x268)
     [<8017671c>] (lock_acquire) from [<80141404>] (process_one_work+0x1b8/0x808)
     [<80141404>] (process_one_work) from [<80141a80>] (worker_thread+0x2c/0x564)
     [<80141a80>] (worker_thread) from [<80147bcc>] (kthread+0x13c/0x16c)
     [<80147bcc>] (kthread) from [<801010b4>] (ret_from_fork+0x14/0x20)
     Exception stack(0xed1d9fb0 to 0xed1d9ff8)
     9fa0:                                     00000000 00000000 00000000 00000000
     9fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     9fe0: 00000000 00000000 00000000 00000000 00000013 00000000

     Signed-off-by: Piotr Figiel <p.figiel@camlintechnologies.com>
     Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e96af6faf9f2e444e1b0972955eaed29327e3195
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Fri Mar 22 15:37:02 2019 +0100

     b43: shut up clang -Wuninitialized variable warning

     [ Upstream commit d825db346270dbceef83b7b750dbc29f1d7dcc0e ]

     Clang warns about what is clearly a case of passing an uninitalized
     variable into a static function:

     drivers/net/wireless/broadcom/b43/phy_lp.c:1852:23: error: variable 'gains' is
uninitialized when used here
           [-Werror,-Wuninitialized]
                     lpphy_papd_cal(dev, gains, 0, 1, 30);
                                         ^~~~~
     drivers/net/wireless/broadcom/b43/phy_lp.c:1838:2: note: variable 'gains' is declared
here
             struct lpphy_tx_gains gains, oldgains;
             ^
     1 error generated.

     However, this function is empty, and its arguments are never evaluated,
     so gcc in contrast does not warn here. Both compilers behave in a
     reasonable way as far as I can tell, so we should change the code
     to avoid the warning everywhere.

     We could just eliminate the lpphy_papd_cal() function entirely,
```

```
    given that it has had the TODO comment in it for 10 years now
    and is rather unlikely to ever get done. I'm doing a simpler
    change here, and just pass the 'oldgains' variable in that has
    been initialized, based on the guess that this is what was
    originally meant.

    Fixes: 2c0d6100da3e ("b43: LP-PHY: Begin implementing calibration & software RFKILL
support")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Larry Finger <Larry.Finger@lwfinger.net>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 135870bd383e4b17a9faf308025c076cb6a9e61f
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Fri Mar 15 12:04:32 2019 -0500

    brcmfmac: fix missing checks for kmemdup

    [ Upstream commit 46953f97224d56a12ccbe9c6acaa84ca0dab2780 ]

    In case kmemdup fails, the fix sets conn_info->req_ie_len and
    conn_info->resp_ie_len to zero to avoid buffer overflows.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Acked-by: Arend van Spriel <arend.vanspriel@broadcom.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1d8e898aa48d050f390dc6da2d9a6466318a4bba
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Tue Mar 12 15:03:58 2019 +0800

    mwifiex: Fix mem leak in mwifiex_tm_cmd

    [ Upstream commit 003b686ace820ce2d635a83f10f2d7f9c147dabc ]

    'hostcmd' is alloced by kzalloc, should be freed before
    leaving from the error handling cases, otherwise it will
    cause mem leak.

    Fixes: 3935ccc14d2c ("mwifiex: add cfg80211 testmode support")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8dc032a293d4f5454dabf02b46f7ee0783fc8af4
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Tue Mar 12 02:56:33 2019 -0500

    rtlwifi: fix a potential NULL pointer dereference

    [ Upstream commit 765976285a8c8db3f0eb7f033829a899d0c2786e ]

    In case alloc_workqueue fails, the fix reports the error and
    returns to avoid NULL pointer dereference.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f382ab1afe260e60dc555619e5d196aa52afaa10
Author: Daniel T. Lee <danieltimlee@gmail.com>
Date:   Thu Apr 4 07:17:55 2019 +0900
```

```
    selftests/bpf: ksym_search won't check symbols exists

    [ Upstream commit 0979ff7992fb6f4eb837995b12f4071dcafebd2d ]

    Currently, ksym_search located at trace_helpers won't check symbols are
    existing or not.

    In ksym_search, when symbol is not found, it will return &syms[0](_stext).
    But when the kernel symbols are not loaded, it will return NULL, which is
    not a desired action.

    This commit will add verification logic whether symbols are loaded prior
    to the symbol search.

    Signed-off-by: Daniel T. Lee <danieltimlee@gmail.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3f52cbfe83967fbef59e8e174764409d0539ea5e
Author: Jian Shen <shenjian15@huawei.com>
Date:   Thu Apr 4 16:17:55 2019 +0800

    net: hns3: add protect when handling mac addr list

    [ Upstream commit 389775a6605e040dddea21a778a88eaaa57c068d ]

    It used netdev->uc and netdev->mc list in function
    hns3_recover_hw_addr() and hns3_remove_hw_addr().
    We should add protect for them.

    Fixes: f05e21097121 ("net: hns3: Clear mac vlan table entries when unload driver or
function reset")
    Signed-off-by: Jian Shen <shenjian15@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9cc0b2b5877fd9292be2f74b7265fa36fabaf7a3
Author: Huazhong Tan <tanhuazhong@huawei.com>
Date:   Thu Apr 4 16:17:56 2019 +0800

    net: hns3: check resetting status in hns3_get_stats()

    [ Upstream commit c4e401e5a934bb0798ebbba98e08dab129695eff ]

    hns3_get_stats() should check the resetting status firstly,
    since the device will be reinitialized when resetting. If the
    reset has not completed, the hns3_get_stats() may access
    invalid memory.

    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8f52f33185ff13e76da8b1617ab0a82c495fcba8
Author: Justin Chen <justinpopo6@gmail.com>
Date:   Thu Feb 28 14:16:48 2019 -0800

    iio: adc: ti-ads7950: Fix improper use of mlock

    [ Upstream commit abbde2792999c9ad3514dd25d7f8d9a96034fe16 ]
```

```
    Indio->mlock is used for protecting the different iio device modes.
    It is currently not being used in this way. Replace the lock with
    an internal lock specifically used for protecting the SPI transfer
    buffer.

    Signed-off-by: Justin Chen <justinpopo6@gmail.com>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1b98d51db5865b569b5857e8fdc6f4cde4d6f334
Author: Nathan Chancellor <natechancellor@gmail.com>
Date:   Thu Mar 7 14:45:46 2019 -0700

    iio: common: ssp_sensors: Initialize calculated_time in ssp_common_process_data

    [ Upstream commit 6f9ca1d3eb74b81f811a87002de2d51640d135b1 ]

    When building with -Wsometimes-uninitialized, Clang warns:

    drivers/iio/common/ssp_sensors/ssp_iio.c:95:6: warning: variable
    'calculated_time' is used uninitialized whenever 'if' condition is false
    [-Wsometimes-uninitialized]

    While it isn't wrong, this will never be a problem because
    iio_push_to_buffers_with_timestamp only uses calculated_time
    on the same condition that it is assigned (when scan_timestamp
    is not zero). While iio_push_to_buffers_with_timestamp is marked
    as inline, Clang does inlining in the optimization stage, which
    happens after the semantic analysis phase (plus inline is merely
    a hint to the compiler).

    Fix this by just zero initializing calculated_time.

    Link: https://github.com/ClangBuiltLinux/linux/issues/394
    Signed-off-by: Nathan Chancellor <natechancellor@gmail.com>
    Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f899898e265ae5de7b709e1aea1c28542d2a1300
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Sat Mar 16 17:08:33 2019 -0500

    iio: hmc5843: fix potential NULL pointer dereferences

    [ Upstream commit 536cc27deade8f1ec3c1beefa60d5fbe0f6fcb28 ]

    devm_regmap_init_i2c may fail and return NULL. The fix returns
    the error when it fails.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5e526a75033d02e7cf0bf964d21b58a6105a66d3
Author: Lars-Peter Clausen <lars@metafoo.de>
Date:   Tue Mar 19 13:37:55 2019 +0200

    iio: ad_sigma_delta: Properly handle SPI bus locking vs CS assertion

    [ Upstream commit df1d80aee963480c5c2938c64ec0ac3e4a0df2e0 ]

    For devices from the SigmaDelta family we need to keep CS low when doing a
    conversion, since the device will use the MISO line as a interrupt to
    indicate that the conversion is complete.
```

```
    This is why the driver locks the SPI bus and when the SPI bus is locked
    keeps as long as a conversion is going on. The current implementation gets
    one small detail wrong though. CS is only de-asserted after the SPI bus is
    unlocked. This means it is possible for a different SPI device on the same
    bus to send a message which would be wrongfully be addressed to the
    SigmaDelta device as well. Make sure that the last SPI transfer that is
    done while holding the SPI bus lock de-asserts the CS signal.

    Signed-off-by: Lars-Peter Clausen <lars@metafoo.de>
    Signed-off-by: Alexandru Ardelean <Alexandru.Ardelean@analog.com>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eaf6e69d164fb2ea1007b942c43822fcc412b191
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Thu Apr 4 00:04:13 2019 +0800

    drm/pl111: fix possible object reference leak

    [ Upstream commit bc29d3a69d4c1bd1a103e8b3c1ed81b807c1870b ]

    The call to of_find_matching_node_and_match returns a node pointer with
    refcount incremented thus it must be explicitly decremented after the
    last usage.

    Detected by coccinelle with the following warnings:
    drivers/gpu/drm/pl111/pl111_versatile.c:333:3-9: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.
    drivers/gpu/drm/pl111/pl111_versatile.c:340:3-9: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.
    drivers/gpu/drm/pl111/pl111_versatile.c:346:3-9: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.
    drivers/gpu/drm/pl111/pl111_versatile.c:354:2-8: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.
    drivers/gpu/drm/pl111/pl111_versatile.c:395:3-9: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.
    drivers/gpu/drm/pl111/pl111_versatile.c:402:1-7: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 317, but without a corresponding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Eric Anholt <eric@anholt.net> (supporter:DRM DRIVER FOR ARM PL111 CLCD)
    Cc: David Airlie <airlied@linux.ie> (maintainer:DRM DRIVERS)
    Cc: Daniel Vetter <daniel@ffwll.ch> (maintainer:DRM DRIVERS)
    Cc: dri-devel@lists.freedesktop.org (open list:DRM DRIVERS)
    Cc: linux-kernel@vger.kernel.org (open list)
    Signed-off-by: Eric Anholt <eric@anholt.net>
    Link: https://patchwork.freedesktop.org/patch/msgid/1554307455-40361-6-git-send-email-
wen.yang99@zte.com.cn
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 34e3da15025a12ee1081d355dd4869b19c7ed253
Author: Ranjani Sridharan <ranjani.sridharan@linux.intel.com>
Date:   Thu Apr 4 17:30:40 2019 -0700

    ASoC: core: remove link components before cleaning up card resources

    [ Upstream commit f96fb7d198ca624fe33c4145a004eb5a3d0eddec ]
```

```
    When the card is registered by the machine driver,
    dai link components are probed after the snd_card is
    created. This is done in snd_soc_bind_card() which calls
    snd_soc_instantiate_card() to first create the snd_card
    and then probes the link components by calling
    soc_probe_link_components(). The snd_card is used by the
    component driver to add the kcontrols associated
    with dapm widgets to the card.

    When the machine driver is unregistered, the snd_card
    is freed when the card resources are cleaned up.
    But the snd_card needs to be valid while unloading the
    topology dapm widgets in order to remove the kcontrols
    from the card.

    Since, unloading topology is done when the component
    driver is removed, the link components should be removed
    in snd_soc_unbind_card(). This will ensure that the kcontrols
    are removed before the card resources are cleaned up and
    the snd_card itself is freed.

    Signed-off-by: Ranjani Sridharan <ranjani.sridharan@linux.intel.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 18127d11a49815897d541a096b3ddbad4740905c
Author: Charles Keepax <ckeepax@opensource.cirrus.com>
Date:   Thu Apr 4 16:32:18 2019 +0100

    regulator: core: Avoid potential deadlock on regulator_unregister

    [ Upstream commit 063773011d33bb36588a90385aa9eb75d13c6d80 ]

    Lockdep reports the following issue on my setup:

    Possible unsafe locking scenario:

    CPU0                    CPU1
    ----                    ----
    lock((work_completion)(&(&rdev->disable_work)->work));
                            lock(regulator_list_mutex);
                            lock((work_completion)(&(&rdev->disable_work)->work));
    lock(regulator_list_mutex);

    The problem is that regulator_unregister takes the
    regulator_list_mutex and then calls flush_work on disable_work. But
    regulator_disable_work calls regulator_lock_dependent which will
    also take the regulator_list_mutex. Resulting in a deadlock if the
    flush_work call actually needs to flush the work.

    Fix this issue by moving the flush_work outside of the
    regulator_list_mutex. The list mutex is not used to guard the point at
    which the delayed work is queued, so its use adds no additional safety.

    Fixes: f8702f9e4aa7 ("regulator: core: Use ww_mutex for regulators locking")
    Signed-off-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Reviewed-by: Dmitry Osipenko <digetx@gmail.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cf0e0ec168173c1a58d7805751e3c310e5f6789d
Author: Andrey Smirnov <andrew.smirnov@gmail.com>
Date:   Tue Apr 2 21:01:28 2019 -0700

    spi: Don't call spi_get_gpio_descs() before device name is set
```

```
    [ Upstream commit 0a919ae49223d32ac0e8be3494547fcd1e4aa0aa ]

    Move code calling spi_get_gpio_descs() to happen after ctlr->dev's
    name is set in order to have proper GPIO consumer names.

    Before:

    cat /sys/kernel/debug/gpio
    gpiochip0: GPIOs 0-31, parent: platform/40049000.gpio, vf610-gpio:
     gpio-6   (                    |regulator-usb0-vbus ) out lo

    gpiochip1: GPIOs 32-63, parent: platform/4004a000.gpio, vf610-gpio:
     gpio-36  (                    |scl                 ) in  hi
     gpio-37  (                    |sda                 ) in  hi
     gpio-40  (                    |(null) CS1          ) out lo
     gpio-41  (                    |(null) CS0          ) out lo ACTIVE LOW
     gpio-42  (                    |miso                ) in  hi
     gpio-43  (                    |mosi                ) in  lo
     gpio-44  (                    |sck                 ) out lo

    After:

    cat /sys/kernel/debug/gpio
    gpiochip0: GPIOs 0-31, parent: platform/40049000.gpio, vf610-gpio:
     gpio-6   (                    |regulator-usb0-vbus ) out lo

    gpiochip1: GPIOs 32-63, parent: platform/4004a000.gpio, vf610-gpio:
     gpio-36  (                    |scl                 ) in  hi
     gpio-37  (                    |sda                 ) in  hi
     gpio-40  (                    |spi0 CS1            ) out lo
     gpio-41  (                    |spi0 CS0            ) out lo ACTIVE LOW
     gpio-42  (                    |miso                ) in  hi
     gpio-43  (                    |mosi                ) in  lo
     gpio-44  (                    |sck                 ) out lo

    Signed-off-by: Andrey Smirnov <andrew.smirnov@gmail.com>
    Cc: Mark Brown <broonie@kernel.org>
    Cc: Chris Healy <cphealy@gmail.com>
    Cc: linux-spi@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5fa810fc6ab57408390041ba9c1e968f11d5a603
Author: Kees Cook <keescook@chromium.org>
Date:   Thu Apr 4 14:40:27 2019 -0700

    x86/build: Keep local relocations with ld.lld

    [ Upstream commit 7c21383f3429dd70da39c0c7f1efa12377a47ab6 ]

    The LLVM linker (ld.lld) defaults to removing local relocations, which
    causes KASLR boot failures. ld.bfd and ld.gold already handle this
    correctly. This adds the explicit instruction "--discard-none" during
    the link phase. There is no change in output for ld.bfd and ld.gold,
    but ld.lld now produces an image with all the needed relocations.

    Signed-off-by: Kees Cook <keescook@chromium.org>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Nick Desaulniers <ndesaulniers@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: clang-built-linux@googlegroups.com
```

```
    Cc: x86-ml <x86@kernel.org>
    Link: https://lkml.kernel.org/r/20190404214027.GA7324@beast
    Link: https://github.com/ClangBuiltLinux/linux/issues/404
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 10a8c316aaeac45bb5e0e1678b50dd4f4e4aa3ce
Author: Alexei Starovoitov <ast@kernel.org>
Date:   Thu Apr 4 14:37:14 2019 -0700

    samples/bpf: fix build with new clang

    [ Upstream commit 636e78b1cdb40b77a79b143dbd9d94847b360efa ]

    clang started to error on invalid asm clobber usage in x86 headers
    and many bpf program samples failed to build with the message:

      CLANG-bpf  /data/users/ast/bpf-next/samples/bpf/xdp_redirect_kern.o
    In file included from /data/users/ast/bpf-next/samples/bpf/xdp_redirect_kern.c:14:
    In file included from ../include/linux/in.h:23:
    In file included from ../include/uapi/linux/in.h:24:
    In file included from ../include/linux/socket.h:8:
    In file included from ../include/linux/uio.h:14:
    In file included from ../include/crypto/hash.h:16:
    In file included from ../include/linux/crypto.h:26:
    In file included from ../include/linux/uaccess.h:5:
    In file included from ../include/linux/sched.h:15:
    In file included from ../include/linux/sem.h:5:
    In file included from ../include/uapi/linux/sem.h:5:
    In file included from ../include/linux/ipc.h:9:
    In file included from ../include/linux/refcount.h:72:
    ../arch/x86/include/asm/refcount.h:72:36: error: asm-specifier for input or output
variable conflicts with asm clobber list
                                            r->refs.counter, e, "er", i, "cx");
                                                                         ^
    ../arch/x86/include/asm/refcount.h:86:27: error: asm-specifier for input or output
variable conflicts with asm clobber list
                                            r->refs.counter, e, "cx");
                                                                ^
    2 errors generated.

    Override volatile() to workaround the problem.

    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f1d84fe4bdd8871c77a8f009480ebde21e54b02f
Author: Oded Gabbay <oded.gabbay@gmail.com>
Date:   Sat Apr 6 13:23:54 2019 +0300

    habanalabs: all FD must be closed before removing device

    [ Upstream commit caa3c8e52582fc4d2ed82afd5e7ea164c18ef4fe ]

    This patch fixes a bug in the implementation of the function that removes
    the device.

    The bug can happen when the device is removed but not the driver itself
    (e.g. remove by the OS due to PCI freeze in Power architecture).

    In that case, there maybe open users that are calling IOCTLs while the
    device is removed. This is a possible race condition that the driver must
    handle. Otherwise, a kernel panic may occur.

    This race is prevented in the hard-reset flow, because the driver makes
```

```
    sure the users are closed before continuing with the hard-reset. This
    race can not occur when the driver itself is removed because the OS makes
    sure all the file descriptors are closed.

    The fix is to make sure the open users close their file descriptors and if
    they don't (after a certain amount of time), the driver sends them a
    SIGKILL, because the remove of the device can't be stopped.

    The patch re-uses the same code that is called from the hard-reset flow.

    Signed-off-by: Oded Gabbay <oded.gabbay@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7de03fc00debb96c758817781d867de60656c96f
Author: Oded Gabbay <oded.gabbay@gmail.com>
Date:   Sat Apr 6 15:33:38 2019 +0300

    habanalabs: prevent device PTE read/write during hard-reset

    [ Upstream commit 9f201aba56b92c3daa4b76efae056ddbb80d91e6 ]

    During hard-reset, contexts are closed as part of the tear-down process.
    After a context is closed, the driver cleans up the page tables of that
    context in the device's DRAM. This action is both dangerous and
    unnecessary.

    It is unnecessary, because the device is going through a hard-reset, which
    means the device's DRAM contents are no longer valid and the device's MMU
    is being reset.

    It is dangerous, because if the hard-reset came as a result of a PCI
    freeze, this action may cause the entire host machine to hang.

    Therefore, prevent all device PTE updates when a hard-reset operation is
    pending.

    Signed-off-by: Oded Gabbay <oded.gabbay@gmail.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7c1c65c50558d1cd7986acd4e8947d1064ea15e1
Author: David Kozub <zub@linux.fjfi.cvut.cz>
Date:   Thu Feb 14 01:15:53 2019 +0100

    block: sed-opal: fix IOC_OPAL_ENABLE_DISABLE_MBR

    [ Upstream commit 78bf47353b0041865564deeed257a54f047c2fdc ]

    The implementation of IOC_OPAL_ENABLE_DISABLE_MBR handled the value
    opal_mbr_data.enable_disable incorrectly: enable_disable is expected
    to be one of OPAL_MBR_ENABLE(0) or OPAL_MBR_DISABLE(1). enable_disable
    was passed directly to set_mbr_done and set_mbr_enable_disable where
    is was interpreted as either OPAL_TRUE(1) or OPAL_FALSE(0). The end
    result was that calling IOC_OPAL_ENABLE_DISABLE_MBR with OPAL_MBR_ENABLE
    actually disabled the shadow MBR and vice versa.

    This patch adds correct conversion from OPAL_MBR_DISABLE/ENABLE to
    OPAL_FALSE/TRUE. The change affects existing programs using
    IOC_OPAL_ENABLE_DISABLE_MBR but this is typically used only once when
    setting up an Opal drive.

    Acked-by: Jon Derrick <jonathan.derrick@intel.com>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Reviewed-by: Scott Bauer <sbauer@plzdonthack.me>
    Signed-off-by: David Kozub <zub@linux.fjfi.cvut.cz>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
```

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

commit 91efbd9dabb03fe332e221621f818a1677cbf38b
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:48 2019 +0800

    cpufreq: ap806: fix possible object reference leak

    [ Upstream commit b623fa320f8360f049a6f3c3ccc487cb85af4c5b ]

    The call to of_find_compatible_node returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/armada-8k-cpufreq.c:187:1-7: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 130, but without a corresponding object
release within this function.
    ./drivers/cpufreq/armada-8k-cpufreq.c:191:1-7: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 130, but without a corresponding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Jason Cooper <jason@lakedaemon.net>
    Cc: Andrew Lunn <andrew@lunn.ch>
    Cc: Gregory Clement <gregory.clement@bootlin.com>
    Cc: Sebastian Hesselbarth <sebastian.hesselbarth@gmail.com>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: linux-arm-kernel@lists.infradead.org
    Cc: linux-pm@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e2a8a9472840c2cd02ecef2fbfb7dfb5d064d4a
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:49 2019 +0800

    cpufreq: imx6q: fix possible object reference leak

    [ Upstream commit ddb64c5db3cc8fb9c1242214d5798b2c2865681c ]

    The call to of_node_get returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/imx6q-cpufreq.c:391:4-10: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 348, but without a corresponding object release
within this function.
    ./drivers/cpufreq/imx6q-cpufreq.c:395:3-9: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 348, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: Shawn Guo <shawnguo@kernel.org>
    Cc: Sascha Hauer <s.hauer@pengutronix.de>
    Cc: Pengutronix Kernel Team <kernel@pengutronix.de>
    Cc: Fabio Estevam <festevam@gmail.com>
    Cc: NXP Linux Team <linux-imx@nxp.com>
    Cc: linux-pm@vger.kernel.org
    Cc: linux-arm-kernel@lists.infradead.org
```

```
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6446fdec59cd07ec24ccc7fa6720015969b4d46d
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:50 2019 +0800

    cpufreq: kirkwood: fix possible object reference leak

    [ Upstream commit 7c468966f05ac9c17bb5948275283d34e6fe0660 ]

    The call to of_get_child_by_name returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/kirkwood-cpufreq.c:127:2-8: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 118, but without a corresponding object
release within this function.
    ./drivers/cpufreq/kirkwood-cpufreq.c:133:2-8: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 118, but without a corresponding object
release within this function.

    and also do some cleanup:
    - of_node_put(np);
    - np = NULL;
    ...
    of_node_put(np);

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: linux-pm@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e84c252b995c9d3e0db52084f0a037cf34a0bf06
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:53 2019 +0800

    cpufreq: pmac32: fix possible object reference leak

    [ Upstream commit 8d10dc28a9ea6e8c02e825dab28699f3c72b02d9 ]

    The call to of_find_node_by_name returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/pmac32-cpufreq.c:557:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 552, but without a corresponding object release
within this function.
    ./drivers/cpufreq/pmac32-cpufreq.c:569:1-7: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 552, but without a corresponding object release
within this function.
    ./drivers/cpufreq/pmac32-cpufreq.c:598:1-7: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 587, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: Benjamin Herrenschmidt <benh@kernel.crashing.org>
```

```
    Cc: Paul Mackerras <paulus@samba.org>
    Cc: Michael Ellerman <mpe@ellerman.id.au>
    Cc: linux-pm@vger.kernel.org
    Cc: linuxppc-dev@lists.ozlabs.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 37792cedb809ce0aabf657280235276da74c57f3
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:52 2019 +0800

    cpufreq/pasemi: fix possible object reference leak

    [ Upstream commit a9acc26b75f652f697e02a9febe2ab0da648a571 ]

    The call to of_get_cpu_node returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/pasemi-cpufreq.c:212:1-7: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 147, but without a corresponding object release
within this function.
    ./drivers/cpufreq/pasemi-cpufreq.c:220:1-7: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 147, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: linuxppc-dev@lists.ozlabs.org
    Cc: linux-pm@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1f82d9100de2dfab482f5a1d3f16b49b3f60ce4
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Mon Apr 1 09:37:54 2019 +0800

    cpufreq: ppc_cbe: fix possible object reference leak

    [ Upstream commit 233298032803f2802fe99892d0de4ab653bfece4 ]

    The call to of_get_cpu_node returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/cpufreq/ppc_cbe_cpufreq.c:89:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 76, but without a corresponding object release
within this function.
    ./drivers/cpufreq/ppc_cbe_cpufreq.c:89:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 76, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: "Rafael J. Wysocki" <rjw@rjwysocki.net>
    Cc: Viresh Kumar <viresh.kumar@linaro.org>
    Cc: linux-pm@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit bbc8d4d4138b12a25938b7211f878eba08cd7e17
Author: Huazhong Tan <tanhuazhong@huawei.com>
Date:   Sat Apr 6 15:43:34 2019 +0800

    net: hns3: add error handler for initializing command queue

    [ Upstream commit 4339ef396ab65a61f7f22f36d7ba94b6e9e0939b ]

    This patch adds error handler for the failure of command queue
    initialization both PF and VF.

    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3063f0f57fef1f399b1236c1ae83dd6e694f4c5e
Author: Kristian Evensen <kristian.evensen@gmail.com>
Date:   Sun Apr 7 15:39:09 2019 +0200

    qmi_wwan: Add quirk for Quectel dynamic config

    [ Upstream commit e4bf63482c309287ca84d91770ffa7dcc18e37eb ]

    Most, if not all, Quectel devices use dynamic interface numbers, and
    users are able to change the USB configuration at will. Matching on for
    example interface number is therefore not possible.

    Instead, the QMI device can be identified by looking at the interface
    class, subclass and protocol (all 0xff), as well as the number of
    endpoints. The reason we need to look at the number of endpoints, is
    that the diagnostic port interface has the same class, subclass and
    protocol as QMI. However, the diagnostic port only has two endpoints,
    while QMI has three.

    Until now, we have identified the QMI device by combining a match on
    class, subclass and protocol, with a call to the function
    quectel_diag_detect(). In quectel_diag_detect(), we check if the number
    of endpoints matches for known Quectel vendor/product ids.

    Adding new vendor/product ids to quectel_diag_detect() is not a good
    long-term solution. This commit replaces the function with a quirk, and
    applies the quirk to affected Quectel devices that I have been able to
    test the change with (EP06, EM12 and EC25). If the quirk is set and the
    number of endpoints equal two, we return from qmi_wwan_probe() with
    -ENODEV.

    Signed-off-by: Kristian Evensen <kristian.evensen@gmail.com>
    Acked-by: Bjørn Mork <bjorn@mork.no>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e88bd7e3df92ce0e6fb62ee716c9f4c845c7c763
Author: Huazhong Tan <tanhuazhong@huawei.com>
Date:   Sat Apr 6 15:43:36 2019 +0800

    net: hns3: fix keep_alive_timer not stop problem

    [ Upstream commit e233516e6a92baeec20aa40fa5b63be6b94f1627 ]

    When hclgevf_client_start() fails or VF driver unloaded, there is
    nobody to disable keep_alive_timer.

    So this patch fixes them.
```

```
    Fixes: a6d818e31d08 ("net: hns3: Add vport alive state checking support")
    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 930d69e957979cb4dd52baf2be667f07e25d9762
Author: Roman Gushchin <guro@fb.com>
Date:   Mon Apr 8 15:12:30 2019 -0700

    selftests: cgroup: fix cleanup path in test_memcg_subtree_control()

    [ Upstream commit e14d314c7a489f060d6d691866fef5f131281718 ]

    Dan reported, that cleanup path in test_memcg_subtree_control()
    triggers a static checker warning:
      ./tools/testing/selftests/cgroup/test_memcontrol.c:76 \
      test_memcg_subtree_control()
      error: uninitialized symbol 'child2'.

    Fix this by initializing child2 and parent2 variables and
    split the cleanup path into few stages.

    Signed-off-by: Roman Gushchin <guro@fb.com>
    Fixes: 84092dbcf901 ("selftests: cgroup: add memory controller self-tests")
    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Cc: Dan Carpenter <dan.carpenter@oracle.com>
    Cc: Shuah Khan (Samsung OSG) <shuah@kernel.org>
    Cc: Mike Rapoport <rppt@linux.vnet.ibm.com>
    Signed-off-by: Shuah Khan <shuah@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ccbab981dee6089f0e2eff8516b0619fb2e5614c
Author: Wenjing Liu <Wenjing.Liu@amd.com>
Date:   Thu Mar 21 13:05:36 2019 -0400

    drm/amd/display: use proper formula to calculate bandwidth from timing

    [ Upstream commit e49f69363adf8920883fff7e8ffecb802d897c6b ]

    [why]
    The existing calculation uses a wrong formula to
    calculate bandwidth from timing.

    [how]
    Expose the existing proper function that calculates the bandwidth,
    so dc_link can use it to calculate timing bandwidth correctly.

    Signed-off-by: Wenjing Liu <Wenjing.Liu@amd.com>
    Reviewed-by: Jun Lei <Jun.Lei@amd.com>
    Acked-by: Leo Li <sunpeng.li@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb7401e75582b2028541b6ba97895c2549ba849a
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Apr 8 23:26:20 2019 +0200

    s390: cio: fix cio_irb declaration

    [ Upstream commit e91012ee855ad9f5ef2ab106a3de51db93fe4d0c ]

    clang points out that the declaration of cio_irb does not match the
    definition exactly, it is missing the alignment attribute:
```

```
    ../drivers/s390/cio/cio.c:50:1: warning: section does not match previous declaration [-
Wsection]
    DEFINE_PER_CPU_ALIGNED(struct irb, cio_irb);
    ^
    ../include/linux/percpu-defs.h:150:2: note: expanded from macro 'DEFINE_PER_CPU_ALIGNED'
            DEFINE_PER_CPU_SECTION(type, name, PER_CPU_ALIGNED_SECTION)     \
            ^
    ../include/linux/percpu-defs.h:93:9: note: expanded from macro 'DEFINE_PER_CPU_SECTION'
            extern __PCPU_ATTRS(sec) __typeof__(type) name;                 \
                   ^
    ../include/linux/percpu-defs.h:49:26: note: expanded from macro '__PCPU_ATTRS'
            __percpu __attribute__((section(PER_CPU_BASE_SECTION sec)))     \
                                    ^
    ../drivers/s390/cio/cio.h:118:1: note: previous attribute is here
    DECLARE_PER_CPU(struct irb, cio_irb);
    ^
    ../include/linux/percpu-defs.h:111:2: note: expanded from macro 'DECLARE_PER_CPU'
            DECLARE_PER_CPU_SECTION(type, name, "")
            ^
    ../include/linux/percpu-defs.h:87:9: note: expanded from macro 'DECLARE_PER_CPU_SECTION'
            extern __PCPU_ATTRS(sec) __typeof__(type) name
                   ^
    ../include/linux/percpu-defs.h:49:26: note: expanded from macro '__PCPU_ATTRS'
            __percpu __attribute__((section(PER_CPU_BASE_SECTION sec)))     \
                                    ^
    Use DECLARE_PER_CPU_ALIGNED() here, to make the two match.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Sebastian Ott <sebott@linux.ibm.com>
    Signed-off-by: Martin Schwidefsky <schwidefsky@de.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a5615d324afc61cc6a82cf61952126205c956d09
Author: Thomas Huth <thuth@redhat.com>
Date:   Sun Apr 7 14:55:09 2019 +0200

    s390/mm: silence compiler warning when compiling without CONFIG_PGSTE

    [ Upstream commit 81a8f2beb32a5951ecf04385301f50879abc092b ]

    If CONFIG_PGSTE is not set (e.g. when compiling without KVM), GCC complains:

      CC      arch/s390/mm/pgtable.o
    arch/s390/mm/pgtable.c:413:15: warning: ‘pmd_alloc_map’ defined but not
     used [-Wunused-function]
     static pmd_t *pmd_alloc_map(struct mm_struct *mm, unsigned long addr)
                   ^~~~~~~~~~~~~

    Wrap the function with "#ifdef CONFIG_PGSTE" to silence the warning.

    Signed-off-by: Thomas Huth <thuth@redhat.com>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Signed-off-by: Heiko Carstens <heiko.carstens@de.ibm.com>
    Signed-off-by: Martin Schwidefsky <schwidefsky@de.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 28968ec7fda32db4af34c4987bd9ab4cebb5bbba
Author: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
Date:   Fri Mar 22 09:59:32 2019 -0400

    drm/amd/display: Initialize stream_update with memset

    [ Upstream commit 2aa632c5ffbedb2ee0e68857683466ea788f17eb ]
```

```
    The brace initialization used here generates warnings on some
    compilers. For example, on GCC 4.9:

    [...] In function ‘dm_determine_update_type_for_commit’:
    [...] error: missing braces around initializer [-Werror=missing-braces]
       struct dc_stream_update stream_update = { 0 };
              ^


    Use memset to make this more portable.

    v2: Specify the compiler / diagnostic in the commit message (Paul)

    Cc: Sun peng Li <Sunpeng.Li@amd.com>
    Cc: Harry Wentland <Harry.Wentland@amd.com>
    Signed-off-by: Nicholas Kazlauskas <nicholas.kazlauskas@amd.com>
    Reviewed-by: Leo Li <sunpeng.li@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 82222b9679a75d74f10e72c17b9eca692e7182a7
Author: Borislav Petkov <bp@suse.de>
Date:   Thu Apr 4 22:14:07 2019 +0200

    x86/microcode: Fix the ancient deprecated microcode loading method

    [ Upstream commit 24613a04ad1c0588c10f4b5403ca60a73d164051 ]

    Commit

      2613f36ed965 ("x86/microcode: Attempt late loading only when new microcode is
present")

    added the new define UCODE_NEW to denote that an update should happen
    only when newer microcode (than installed on the system) has been found.

    But it missed adjusting that for the old /dev/cpu/microcode loading
    interface. Fix it.

    Fixes: 2613f36ed965 ("x86/microcode: Attempt late loading only when new microcode is
present")
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Reviewed-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: Jann Horn <jannh@google.com>
    Link: https://lkml.kernel.org/r/20190405133010.24249-3-bp@alien8.de
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 73205be1e0b1c2e0cc91073d7380debada8dab83
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Apr 8 23:26:18 2019 +0200

    s390: zcrypt: initialize variables before_use

    [ Upstream commit 913140e221567b3ecd21b4242257a7e3fa279026 ]

    The 'func_code' variable gets printed in debug statements without
    a prior initialization in multiple functions, as reported when building
    with clang:

    drivers/s390/crypto/zcrypt_api.c:659:6: warning: variable 'func_code' is used
uninitialized whenever 'if' condition is true
          [-Wsometimes-uninitialized]
            if (mex->outputdatalength < mex->inputdatalength) {
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    drivers/s390/crypto/zcrypt_api.c:725:29: note: uninitialized use occurs here
            trace_s390_zcrypt_rep(mex, func_code, rc,
```

```
                                        ^~~~~~~~~
    drivers/s390/crypto/zcrypt_api.c:659:2: note: remove the 'if' if its condition is always
 false
            if (mex->outputdatalength < mex->inputdatalength) {
            ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    drivers/s390/crypto/zcrypt_api.c:654:24: note: initialize the variable 'func_code' to
 silence this warning
            unsigned int func_code;
                                ^
```

    Add initializations to all affected code paths to shut up the warning
    and make the warning output consistent.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Martin Schwidefsky <schwidefsky@de.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d1058b175fee37b5358e898a38336d3ca0f76dcb
Author: Michael Tretter <m.tretter@pengutronix.de>
Date: Tue Mar 19 11:01:46 2019 +0100

    clk: zynqmp: fix check for fractional clock

    [Upstream commit c06e64407e031e71c67f45f07981510ca4c880a1]

    The firmware sets BIT(13) in clkflag to mark a divider as fractional
    divider. The clock driver copies the clkflag straight to the flags of
    the common clock framework. In the common clk framework flags, BIT(13)
    is defined as CLK_DUTY_CYCLE_PARENT.

    Add a new field to the zynqmp_clk_divider to specify if a divider is a
    fractional devider. Set this field based on the clkflag when registering
    a divider.

    At the same time, unset BIT(13) from clkflag when copying the flags to
    the common clk framework flags.

    Signed-off-by: Michael Tretter <m.tretter@pengutronix.de>
    Signed-off-by: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4296aca938f7c7adc6174b9190f5b8eca4ad5865
Author: Douglas Anderson <dianders@chromium.org>
Date: Tue Apr 9 13:47:06 2019 -0700

    clk: rockchip: Make rkpwm a critical clock on rk3288

    [Upstream commit dfe7fb21cd9e730230d55a79bc72cf2ece67cdd5]

    Most rk3288-based boards are derived from the EVB and thus use a PWM
    regulator for the logic rail. However, most rk3288-based boards don't
    specify the PWM regulator in their device tree. We'll deal with that
    by making it critical.

    NOTE: it's important to make it critical and not just IGNORE_UNUSED
    because all PWMs in the system share the same clock. We don't want
    another PWM user to turn the clock on and off and kill the logic rail.

    This change is in preparation for actually having the PWMs in the
    rk3288 device tree actually point to the proper PWM clock. Up until
    now they've all pointed to the clock for the old IP block and they've
    all worked due to the fact that rkpwm was IGNORE_UNUSED and that the
    clock rates for both clocks were the same.

    Signed-off-by: Douglas Anderson <dianders@chromium.org>

```
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 95c559cb4194cea35169a7c9a5570c7449322ed0
Author: Charles Keepax <ckeepax@opensource.cirrus.com>
Date:   Thu Apr 4 17:33:56 2019 +0100

    extcon: arizona: Disable mic detect if running when driver is removed

    [ Upstream commit 00053de52231117ddc154042549f2256183ffb86 ]

    Microphone detection provides the button detection features on the
    Arizona CODECs as such it will be running if the jack is currently
    inserted. If the driver is unbound whilst the jack is still inserted
    this will cause warnings from the regulator framework as the MICVDD
    regulator is put but was never disabled.

    Correct this by disabling microphone detection on driver removal and if
    the microphone detection was running disable the regulator and put the
    runtime reference that was currently held.

    Signed-off-by: Charles Keepax <ckeepax@opensource.cirrus.com>
    Signed-off-by: Chanwoo Choi <cw00.choi@samsung.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 038c427d6e730ccb9c6c77eef1911c4845d5aa19
Author: Douglas Anderson <dianders@chromium.org>
Date:   Thu Apr 11 06:55:55 2019 -0700

    clk: rockchip: Fix video codec clocks on rk3288

    [ Upstream commit 00c0cd9e59d265b393553e9afa54fee8b10e8158 ]

    It appears that there is a typo in the rk3288 TRM.  For
    GRF_SOC_CON0[7] it says that 0 means "vepu" and 1 means "vdpu".  It's
    the other way around.

    How do I know?  Here's my evidence:

    1. Prior to commit 4d3e84f99628 ("clk: rockchip: describe aclk_vcodec
       using the new muxgrf type on rk3288") we always pretended that we
       were using "aclk_vdpu" and the comment in the code said that this
       matched the default setting in the system.  In fact the default
       setting is 0 according to the TRM and according to reading memory
       at bootup.  In addition rk3288-based Chromebooks ran like this and
       the video codecs worked.
    2. With the existing clock code if you boot up and try to enable the
       new VIDEO_ROCKCHIP_VPU as a module (and without "clk_ignore_unused"
       on the command line), you get errors like "failed to get ack on
       domain 'pd_video', val=0x80208".  After flipping vepu/vdpu things
       init OK.
    3. If I export and add both the vepu and vdpu to the list of clocks
       for RK3288_PD_VIDEO I can get past the power domain errors, but now
       I freeze when the vpu_mmu gets initted.
    4. If I just mark the "vdpu" as IGNORE_UNUSED then everything boots up
       and probes OK showing that somehow the "vdpu" was important to keep
       enabled.  This is because we were actually using it as a parent.
    5. After this change I can hack "aclk_vcodec_pre" to parent from
       "aclk_vepu" using assigned-clocks and the video codec still probes
       OK.
    6. Rockchip has said so on the mailing list [1].

    ...so let's fix it.

    Let's also add CLK_SET_RATE_PARENT to "aclk_vcodec_pre" as suggested
```

```
    by Jonas Karlman.  Prior to the same commit you could do
    clk_set_rate() on "aclk_vcodec" and it would change "aclk_vdpu".
    That's because "aclk_vcodec" was a simple gate clock (always gets
    CLK_SET_RATE_PARENT) and its direct parent was "aclk_vdpu".  After
    that commit "aclk_vcodec_pre" gets in the way so we need to add
    CLK_SET_RATE_PARENT to it too.

    [1] https://lkml.kernel.org/r/1d17b015-9e17-34b9-baf8-c285dc1957aa@rock-chips.com

    Fixes: 4d3e84f99628 ("clk: rockchip: describe aclk_vcodec using the new muxgrf type on
rk3288")
    Suggested-by: Jonas Karlman <jonas@kwiboo.se>
    Suggested-by: Randy Li <ayaka@soulik.info>
    Signed-off-by: Douglas Anderson <dianders@chromium.org>
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fdb3ecd0d65f9ac00a7b60da30e0031690e2821f
Author: Ulf Hansson <ulf.hansson@linaro.org>
Date:   Wed Apr 10 11:55:16 2019 +0200

    PM / core: Propagate dev->power.wakeup_path when no callbacks

    [ Upstream commit dc351d4c5f4fe4d0f274d6d660227be0c3a03317 ]

    The dev->power.direct_complete flag may become set in device_prepare() in
    case the device don't have any PM callbacks (dev->power.no_pm_callbacks is
    set). This leads to a broken behaviour, when there is child having wakeup
    enabled and relies on its parent to be used in the wakeup path.

    More precisely, when the direct complete path becomes selected for the
    child in __device_suspend(), the propagation of the dev->power.wakeup_path
    becomes skipped as well.

    Let's address this problem, by checking if the device is a part the wakeup
    path or has wakeup enabled, then prevent the direct complete path from
    being used.

    Reported-by: Loic Pallardy <loic.pallardy@st.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    [ rjw: Comment cleanup ]
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5344c9ef9afe03dbaf6753ee22dfdbca1fc32df5
Author: Christian König <christian.koenig@amd.com>
Date:   Fri Mar 29 19:30:23 2019 +0100

    drm/amdgpu: fix old fence check in amdgpu_fence_emit

    [ Upstream commit 3d2aca8c8620346abdba96c6300d2c0b90a1d0cc ]

    We don't hold a reference to the old fence, so it can go away
    any time we are waiting for it to signal.

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

commit 8b4ad8fbfb95ad3167a13d26bdfa6cdd2211f4f6
Author: Peng Li <lipeng321@huawei.com>
Date:   Sun Apr 14 09:47:45 2019 +0800

    net: hns3: free the pending skb when clean RX ring
```

```
    [ Upstream commit cc5ff6e90f808f9a4c8229bf2f1de0dfe5d7931c ]

    If there is pending skb in RX flow when close the port, and the
    pending buffer is not cleaned, the new packet will be added to
    the pending skb when the port opens again, and the first new
    packet has error data.

    This patch cleans the pending skb when clean RX ring.

    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 71fb483c4bb876ce36a3b40614c3c14004238ecb
Author: Yinbo Zhu <yinbo.zhu@nxp.com>
Date:   Mon Mar 11 02:16:40 2019 +0000

    mmc: sdhci-of-esdhc: add erratum eSDHC-A001 and A-008358 support

    [ Upstream commit 05cb6b2a66fa7837211a060878e91be5eb10cb07 ]

    eSDHC-A001: The data timeout counter (SYSCTL[DTOCV]) is not
    reliable for DTOCV values 0x4(2^17 SD clock), 0x8(2^21 SD clock),
    and 0xC(2^25 SD clock). The data timeout counter can count from
    2^13–2^27, but for values 2^17, 2^21, and 2^25, the timeout
    counter counts for only 2^13 SD clocks.
    A-008358: The data timeout counter value loaded into the timeout
    counter is less than expected and can result into early timeout
    error in case of eSDHC data transactions. The table below shows
    the expected vs actual timeout period for different values of
    SYSCTL[DTOCV]:
    these two erratum has the same quirk to control it, and set
    SDHCI_QUIRK_RESET_AFTER_REQUEST to fix above issue.

    Signed-off-by: Yinbo Zhu <yinbo.zhu@nxp.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 20b0a12e383579375eab157ac0d2c4e25cfc80f3
Author: Yinbo Zhu <yinbo.zhu@nxp.com>
Date:   Mon Mar 11 02:16:44 2019 +0000

    mmc: sdhci-of-esdhc: add erratum A-009204 support

    [ Upstream commit 5dd195522562542bc6ebe6e7bd47890d8b7ca93c ]

    In the event of that any data error (like, IRQSTAT[DCE]) occurs
    during an eSDHC data transaction where DMA is used for data
    transfer to/from the system memory, setting the SYSCTL[RSTD]
    register may cause a system hang. If software sets the register
    SYSCTL[RSTD] to 1 for error recovery while DMA transferring is
    not complete, eSDHC may hang the system bus. This happens because
    the software register SYSCTL[RSTD] resets the DMA engine without
    waiting for the completion of pending system transactions. This
    erratum is to fix this issue.

    Signed-off-by: Yinbo Zhu <yinbo.zhu@nxp.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ebe0205812b51c3f503d04a87c1b1b80bac38362
```

```
Author: Yinbo Zhu <yinbo.zhu@nxp.com>
Date:   Mon Mar 11 02:16:36 2019 +0000

    mmc: sdhci-of-esdhc: add erratum eSDHC5 support

    [ Upstream commit a46e42712596b51874f04c73f1cdf1017f88df52 ]

    Software writing to the Transfer Type configuration register
    (system clock domain) can cause a setup/hold violation in the
    CRC flops (card clock domain), which can cause write accesses
    to be sent with corrupt CRC values. This issue occurs only for
    write preceded by read. this erratum is to fix this issue.

    Signed-off-by: Yinbo Zhu <yinbo.zhu@nxp.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dec5d5480a20cba674cd824970357e7ce0b2e939
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Mon Mar 11 00:53:33 2019 -0500

    mmc_spi: add a status check for spi_sync_locked

    [ Upstream commit 611025983b7976df0183390a63a2166411d177f1 ]

    In case spi_sync_locked fails, the fix reports the error and
    returns the error code upstream.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Reviewed-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 022d0ac1a2b8f0611e9a179f3dffa32d9057e3db
Author: Andrea Merello <andrea.merello@gmail.com>
Date:   Fri Apr 5 10:34:58 2019 +0200

    mmc: core: make pwrseq_emmc (partially) support sleepy GPIO controllers

    [ Upstream commit 002ee28e8b322d4d4b7b83234b5d0f4ebd428eda ]

    pwrseq_emmc.c implements a HW reset procedure for eMMC chip by driving a
    GPIO line.

    It registers the .reset() cb on mmc_pwrseq_ops and it registers a system
    restart notification handler; both of them perform reset by unconditionally
    calling gpiod_set_value().

    If the eMMC reset line is tied to a GPIO controller whose driver can sleep
    (i.e. I2C GPIO controller), then the kernel would spit warnings when trying
    to reset the eMMC chip by means of .reset() mmc_pwrseq_ops cb (that is
    exactly what I'm seeing during boot).

    Furthermore, on system reset we would gets to the system restart
    notification handler with disabled interrupts - local_irq_disable() is
    called in machine_restart() at least on ARM/ARM64 - and we would be in
    trouble when the GPIO driver tries to sleep (which indeed doesn't happen
    here, likely because in my case the machine specific code doesn't call
    do_kernel_restart(), I guess..).

    This patch fixes the .reset() cb to make use of gpiod_set_value_cansleep(),
    so that the eMMC gets reset on boot without complaints, while, since there
    isn't that much we can do, we avoid register the restart handler if the
    GPIO controller has a sleepy driver (and we spit a dev_notice() message to
```

```
    let people know)..

    This had been tested on a downstream 4.9 kernel with backported
    commit 83f37ee7ba33 ("mmc: pwrseq: Add reset callback to the struct
    mmc_pwrseq_ops") and commit ae60fb031cf2 ("mmc: core: Don't do eMMC HW
    reset when resuming the eMMC card"), because I couldn't boot my board
    otherwise. Maybe worth to RFT.

    Signed-off-by: Andrea Merello <andrea.merello@gmail.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 43a25a4a716e519f55ebbf89b349f5d82f918c2e
Author: John Garry <john.garry@huawei.com>
Date:   Fri Apr 12 16:57:56 2019 +0800

    scsi: libsas: Do discovery on empty PHY to update PHY info

    [ Upstream commit d8649fc1c5e40e691d589ed825998c36a947491c ]

    When we discover the PHY is empty in sas_rediscover_dev(), the PHY
    information (like negotiated linkrate) is not updated.

    As such, for a user examining sysfs for that PHY, they would see
    incorrect values:

    root@(none)$ cd /sys/class/sas_phy/phy-0:0:20
    root@(none)$ more negotiated_linkrate
    3.0 Gbit
    root@(none)$ echo 0 > enable
    root@(none)$ more negotiated_linkrate
    3.0 Gbit

    So fix this, simply discover the PHY again, even though we know it's empty;
    in the above example, this gives us:

    root@(none)$ more negotiated_linkrate
    Phy disabled

    We must do this after unregistering the device associated with the PHY
    (in sas_unregister_devs_sas_addr()).

    Signed-off-by: John Garry <john.garry@huawei.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e363683dd6e7428648c63b3074c4ca228ac0d4e8
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Thu Apr 4 10:52:43 2019 -0700

    hwmon: (f71805f) Use request_muxed_region for Super-IO accesses

    [ Upstream commit 73e6ff71a7ea924fb7121d576a2d41e3be3fc6b5 ]

    Super-IO accesses may fail on a system with no or unmapped LPC bus.

    Unable to handle kernel paging request at virtual address ffffffbffee0002e
    pgd = ffffffc1d68d4000
    [ffffffbffee0002e] *pgd=0000000000000000, *pud=0000000000000000
    Internal error: Oops: 94000046 [#1] PREEMPT SMP
    Modules linked in: f71805f(+) hwmon
    CPU: 3 PID: 1659 Comm: insmod Not tainted 4.5.0+ #88
    Hardware name: linux,dummy-virt (DT)
    task: ffffffc1f6665400 ti: ffffffc1d6418000 task.ti: ffffffc1d6418000
    PC is at f71805f_find+0x6c/0x358 [f71805f]
```

```
    Also, other drivers may attempt to access the LPC bus at the same time,
    resulting in undefined behavior.

    Use request_muxed_region() to ensure that IO access on the requested
    address space is supported, and to ensure that access by multiple
    drivers is synchronized.

    Fixes: e53004e20a58e ("hwmon: New f71805f driver")
    Reported-by: Kefeng Wang <wangkefeng.wang@huawei.com>
    Reported-by: John Garry <john.garry@huawei.com>
    Cc: John Garry <john.garry@huawei.com>
    Acked-by: John Garry <john.garry@huawei.com>
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 64e5c7c82d52dc2dfbf4b775507a50b7e9fe2149
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Thu Apr 4 11:16:20 2019 -0700

    hwmon: (pc87427) Use request_muxed_region for Super-IO accesses

    [ Upstream commit 755a9b0f8aaa5639ba5671ca50080852babb89ce ]

    Super-IO accesses may fail on a system with no or unmapped LPC bus.

    Also, other drivers may attempt to access the LPC bus at the same time,
    resulting in undefined behavior.

    Use request_muxed_region() to ensure that IO access on the requested
    address space is supported, and to ensure that access by multiple drivers
    is synchronized.

    Fixes: ba224e2c4f0a7 ("hwmon: New PC87427 hardware monitoring driver")
    Reported-by: Kefeng Wang <wangkefeng.wang@huawei.com>
    Reported-by: John Garry <john.garry@huawei.com>
    Cc: John Garry <john.garry@huawei.com>
    Acked-by: John Garry <john.garry@huawei.com>
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e231d73fb6bde8a15fd359433fea831d57df6748
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Thu Apr 4 11:22:42 2019 -0700

    hwmon: (smsc47b397) Use request_muxed_region for Super-IO accesses

    [ Upstream commit 8c0826756744c0ac1df600a5e4cca1a341b13101 ]

    Super-IO accesses may fail on a system with no or unmapped LPC bus.

    Also, other drivers may attempt to access the LPC bus at the same time,
    resulting in undefined behavior.

    Use request_muxed_region() to ensure that IO access on the requested
    address space is supported, and to ensure that access by multiple drivers
    is synchronized.

    Fixes: 8d5d45fb1468 ("I2C: Move hwmon drivers (2/3)")
    Reported-by: Kefeng Wang <wangkefeng.wang@huawei.com>
    Reported-by: John Garry <john.garry@huawei.com>
    Cc: John Garry <john.garry@huawei.com>
    Acked-by: John Garry <john.garry@huawei.com>
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 02d5ae1cc84d9afde59ba1875c9f26e28f6a6276
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Thu Apr 4 11:28:37 2019 -0700

    hwmon: (smsc47m1) Use request_muxed_region for Super-IO accesses

    [ Upstream commit d6410408ad2a798c4cc685252c1baa713be0ad69 ]

    Super-IO accesses may fail on a system with no or unmapped LPC bus.

    Also, other drivers may attempt to access the LPC bus at the same time,
    resulting in undefined behavior.

    Use request_muxed_region() to ensure that IO access on the requested
    address space is supported, and to ensure that access by multiple drivers
    is synchronized.

    Fixes: 8d5d45fb1468 ("I2C: Move hwmon drivers (2/3)")
    Reported-by: Kefeng Wang <wangkefeng.wang@huawei.com>
    Reported-by: John Garry <john.garry@huawei.com>
    Cc: John Garry <john.garry@huawei.com>
    Acked-by: John Garry <john.garry@huawei.com>
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e2cab96556f2666758642daac95563258f4c3cc8
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Fri Apr 5 08:53:08 2019 -0700

    hwmon: (vt1211) Use request_muxed_region for Super-IO accesses

    [ Upstream commit 14b97ba5c20056102b3dd22696bf17b057e60976 ]

    Super-IO accesses may fail on a system with no or unmapped LPC bus.

    Also, other drivers may attempt to access the LPC bus at the same time,
    resulting in undefined behavior.

    Use request_muxed_region() to ensure that IO access on the requested
    address space is supported, and to ensure that access by multiple drivers
    is synchronized.

    Fixes: 2219cd81a6cd ("hwmon/vt1211: Add probing of alternate config index port")
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6b110830cd98e4a4dbf19f24c073e29c8d4902de
Author: Enric Balletbo i Serra <enric.balletbo@collabora.com>
Date:   Wed Mar 13 13:22:53 2019 +0100

    PM / devfreq: Fix static checker warning in try_then_request_governor

    [ Upstream commit b53b0128052ffd687797d5f4deeb76327e7b5711 ]

    The patch 23c7b54ca1cd: "PM / devfreq: Fix devfreq_add_device() when
    drivers are built as modules." leads to the following static checker
    warning:

        drivers/devfreq/devfreq.c:1043 governor_store()
        warn: 'governor' can also be NULL

    The reason is that the try_then_request_governor() function returns both
    error pointers and NULL. It should just return error pointers, so fix
    this by returning a ERR_PTR to the error intead of returning NULL.
```

```
    Fixes: 23c7b54ca1cd ("PM / devfreq: Fix devfreq_add_device() when drivers are built as
modules.")
    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Enric Balletbo i Serra <enric.balletbo@collabora.com>
    Reviewed-by: Chanwoo Choi <cw00.choi@samsung.com>
    Signed-off-by: MyungJoo Ham <myungjoo.ham@samsung.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d15467459100861bec784f464365d61471dc3edb
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Tue Apr 2 12:45:06 2019 -0700

    perf/x86/intel/cstate: Add Icelake support

    [ Upstream commit f08c47d1f86c6dc666c7e659d94bf6d4492aa9d7 ]

    Icelake uses the same C-state residency events as Sandy Bridge.

    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: acme@kernel.org
    Cc: jolsa@kernel.org
    Link: https://lkml.kernel.org/r/20190402194509.2832-10-kan.liang@linux.intel.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d2b4a1c782a042d31a04a681843095fc05113d5e
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Tue Apr 2 12:45:07 2019 -0700

    perf/x86/intel/rapl: Add Icelake support

    [ Upstream commit b3377c3acb9e54cf86efcfe25f2e792bca599ed4 ]

    Icelake support the same RAPL counters as Skylake.

    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: acme@kernel.org
    Cc: jolsa@kernel.org
    Link: https://lkml.kernel.org/r/20190402194509.2832-11-kan.liang@linux.intel.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dcec07cd2bc687f94ea76a3452cc0c811aec5a0a
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Tue Apr 2 12:45:08 2019 -0700
```

```
    perf/x86/msr: Add Icelake support

    [ Upstream commit cf50d79a8cfe5adae37fec026220b009559bbeed ]

    Icelake is the same as the existing Skylake parts.

    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Arnaldo Carvalho de Melo <acme@redhat.com>
    Cc: Jiri Olsa <jolsa@redhat.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Vince Weaver <vincent.weaver@maine.edu>
    Cc: acme@kernel.org
    Cc: jolsa@kernel.org
    Link: https://lkml.kernel.org/r/20190402194509.2832-12-kan.liang@linux.intel.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8bcc53fb09779ea011623e7766f56e2ea24bdcf0
Author: Eric Biggers <ebiggers@google.com>
Date:   Thu Apr 11 14:32:15 2019 -0700

    fscrypt: use READ_ONCE() to access ->i_crypt_info

    [ Upstream commit e37a784d8b6a1e726de5ddc7b4809c086a08db09 ]

    ->i_crypt_info starts out NULL and may later be locklessly set to a
    non-NULL value by the cmpxchg() in fscrypt_get_encryption_info().

    But ->i_crypt_info is used directly, which technically is incorrect.
    It's a data race, and it doesn't include the data dependency barrier
    needed to safely dereference the pointer on at least one architecture.

    Fix this by using READ_ONCE() instead.  Note: we don't need to use
    smp_load_acquire(), since dereferencing the pointer only requires a data
    dependency barrier, which is already included in READ_ONCE().  We also
    don't need READ_ONCE() in places where ->i_crypt_info is unconditionally
    dereferenced, since it must have already been checked.

    Also downgrade the cmpxchg() to cmpxchg_release(), since RELEASE
    semantics are sufficient on the write side.

    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d66a2891b79962cfa742967844da7fc5d9bdab87
Author: Colin Ian King <colin.king@canonical.com>
Date:   Sat Apr 13 17:00:26 2019 +0100

    RDMA/cxgb4: Fix null pointer dereference on alloc_skb failure

    [ Upstream commit a6d2a5a92e67d151c98886babdc86d530d27111c ]

    Currently if alloc_skb fails to allocate the skb a null skb is passed to
    t4_set_arp_err_handler and this ends up dereferencing the null skb.  Avoid
    the NULL pointer dereference by checking for a NULL skb and returning
    early.

    Addresses-Coverity: ("Dereference null return")
    Fixes: b38a0ad8ec11 ("RDMA/cxgb4: Set arp error handler for PASS_ACCEPT_RPL messages")
```

```
    Signed-off-by: Colin Ian King <colin.king@canonical.com>
    Acked-by: Potnuri Bharat Teja <bharat@chelsio.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cb9552b932fa036281c3ab2fd6456abd2a708fa2
Author: Vincenzo Frascino <vincenzo.frascino@arm.com>
Date:   Tue Apr 16 17:14:30 2019 +0100

    arm64: vdso: Fix clock_getres() for CLOCK_REALTIME

    [ Upstream commit 81fb8736dd81da3fe94f28968dac60f392ec6746 ]

    clock_getres() in the vDSO library has to preserve the same behaviour
    of posix_get_hrtimer_res().

    In particular, posix_get_hrtimer_res() does:

        sec = 0;
        ns = hrtimer_resolution;

    where 'hrtimer_resolution' depends on whether or not high resolution
    timers are enabled, which is a runtime decision.

    The vDSO incorrectly returns the constant CLOCK_REALTIME_RES. Fix this
    by exposing 'hrtimer_resolution' in the vDSO datapage and returning that
    instead.

    Reviewed-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Vincenzo Frascino <vincenzo.frascino@arm.com>
    [will: Use WRITE_ONCE(), move adr off COARSE path, renumber labels, use 'w' reg]
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6388d9ec21c6caeb398a84b4fa85675961f7fcbc
Author: Kefeng Wang <wangkefeng.wang@huawei.com>
Date:   Mon Apr 8 23:21:12 2019 +0800

    ACPI/IORT: Reject platform device creation on NUMA node mapping failure

    [ Upstream commit 36a2ba07757df790b4a874efb1a105b9330a9ae7 ]

    In a system where, through IORT firmware mappings, the SMMU device is
    mapped to a NUMA node that is not online, the kernel bootstrap results
    in the following crash:

      Unable to handle kernel paging request at virtual address 0000000000001388
      Mem abort info:
        ESR = 0x96000004
        Exception class = DABT (current EL), IL = 32 bits
        SET = 0, FnV = 0
        EA = 0, S1PTW = 0
      Data abort info:
        ISV = 0, ISS = 0x00000004
        CM = 0, WnR = 0
      [0000000000001388] user address but active_mm is swapper
      Internal error: Oops: 96000004 [#1] SMP
      Modules linked in:
      CPU: 5 PID: 1 Comm: swapper/0 Not tainted 5.0.0 #15
      pstate: 80c00009 (Nzcv daif +PAN +UAO)
      pc : __alloc_pages_nodemask+0x13c/0x1068
      lr : __alloc_pages_nodemask+0xdc/0x1068
      ...
      Process swapper/0 (pid: 1, stack limit = 0x(____ptrval____))
      Call trace:
```

```
        __alloc_pages_nodemask+0x13c/0x1068
        new_slab+0xec/0x570
        ___slab_alloc+0x3e0/0x4f8
        __slab_alloc+0x60/0x80
        __kmalloc_node_track_caller+0x10c/0x478
        devm_kmalloc+0x44/0xb0
        pinctrl_bind_pins+0x4c/0x188
        really_probe+0x78/0x2b8
        driver_probe_device+0x64/0x110
        device_driver_attach+0x74/0x98
        __driver_attach+0x9c/0xe8
        bus_for_each_dev+0x84/0xd8
        driver_attach+0x30/0x40
        bus_add_driver+0x170/0x218
        driver_register+0x64/0x118
        __platform_driver_register+0x54/0x60
        arm_smmu_driver_init+0x24/0x2c
        do_one_initcall+0xbc/0x328
        kernel_init_freeable+0x304/0x3ac
        kernel_init+0x18/0x110
        ret_from_fork+0x10/0x1c
      Code: f90013b5 b9410fa1 1a9f0694 b50014c2 (b9400804)
      ---[ end trace dfeaed4c373a32da ]--

    Change the dev_set_proximity() hook prototype so that it returns a
    value and make it return failure if the PXM->NUMA-node mapping
    corresponds to an offline node, fixing the crash.

    Acked-by: Lorenzo Pieralisi <lorenzo.pieralisi@arm.com>
    Signed-off-by: Kefeng Wang <wangkefeng.wang@huawei.com>
    Link: https://lore.kernel.org/linux-arm-kernel/20190315021940.86905-1-
wangkefeng.wang@huawei.com/
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f7b0c61dc60ee602bd5f547cdf50c316a3e8e1a4
Author: Nicholas Nunley <nicholas.d.nunley@intel.com>
Date:   Wed Feb 6 15:08:17 2019 -0800

    i40e: don't allow changes to HW VLAN stripping on active port VLANs

    [ Upstream commit bfb0ebed53857cfc57f11c63fa3689940d71c1c8 ]

    Modifying the VLAN stripping options when a port VLAN is configured
    will break traffic for the VSI, and conceptually doesn't make sense,
    so don't allow this.

    Signed-off-by: Nicholas Nunley <nicholas.d.nunley@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit decaf30165b8dadc210657c30008329160a0d136
Author: Adam Ludkiewicz <adam.ludkiewicz@intel.com>
Date:   Wed Feb 6 15:08:25 2019 -0800

    i40e: Able to add up to 16 MAC filters on an untrusted VF

    [ Upstream commit 06b6e2a2333eb3581567a7ac43ca465ef45f4daa ]

    This patch fixes the problem with the driver being able to add only 7
    multicast MAC address filters instead of 16. The problem is fixed by
    changing the maximum number of MAC address filters to 16+1+1 (two extra
    are needed because the driver uses 1 for unicast MAC address and 1 for
    broadcast).
```

```
    Signed-off-by: Adam Ludkiewicz <adam.ludkiewicz@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9ea0adc47c554d84cec06bbec60e78ee111c17a4
Author: Ioana Ciocoi Radulescu <ruxandra.radulescu@nxp.com>
Date:   Tue Apr 16 17:13:28 2019 +0000

    dpaa2-eth: Fix Rx classification status

    [ Upstream commit df8e249be866e2f762be11b14a9e7a94752614d4 ]

    Set the Rx flow classification enable flag only if key config
    operation is successful.

    Fixes 3f9b5c9 ("dpaa2-eth: Configure Rx flow classification key")

    Signed-off-by: Ioana Radulescu <ruxandra.radulescu@nxp.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a416e2058aa389ba629e10917c3412ce9d28c4a
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 4 21:23:57 2019 +0100

    phy: ti: usb2: fix OMAP_CONTROL_PHY dependency

    [ Upstream commit d41ce98a122c13ea77938af04ef06fb12ae0c69e ]

    With randconfig build testing on arm64, we can run into a configuration
    that has CONFIG_OMAP_CONTROL_PHY=m and CONFIG_OMAP_USB2=y, which in turn
    causes a link failure:

    drivers/phy/ti/phy-omap-usb2.o: In function `omap_usb_phy_power':
    phy-omap-usb2.c:(.text+0x17c): undefined reference to `omap_control_phy_power'

    I could not come up with a good way to correctly describe the relation
    of the two symbols, but if we just select CONFIG_OMAP_CONTROL_PHY
    during compile testing, we can no longer run into the broken configuration.

    Fixes: 6777cee3a872 ("phy: ti: usb2: Add support for AM654 USB2 PHY")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Kishon Vijay Abraham I <kishon@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 03b812eade3d431166c6e7674e02f74f0a8614fc
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Mon Mar 4 21:23:58 2019 +0100

    phy: mapphone-mdm6600: add gpiolib dependency

    [ Upstream commit 208d3423ee463ab257908456f6bbca4024ab63f7 ]

    gcc points out that when CONFIG_GPIOLIB is disabled,
    gpiod_get_array_value_cansleep() returns 0 but fails to set its output:

    drivers/phy/motorola/phy-mapphone-mdm6600.c: In function 'phy_mdm6600_status':
    drivers/phy/motorola/phy-mapphone-mdm6600.c:220:24: error: 'values[0]' is used
uninitialized in this function [-Werror=uninitialized]

    This could be fixed more generally in gpiolib by returning a failure
    code, but for this specific case, the easier workaround is to add a
    gpiolib dependency.
```

```
    Fixes: 5d1ebbda0318 ("phy: mapphone-mdm6600: Add USB PHY driver for MDM6600 on Droid 4")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Tony Lindgren <tony@atomide.com>
    Signed-off-by: Kishon Vijay Abraham I <kishon@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb4b60cda1179de867897ed96d1dad4c48236df8
Author: Paul Kocialkowski <paul.kocialkowski@bootlin.com>
Date:   Thu Mar 14 14:05:18 2019 +0100

    phy: sun4i-usb: Make sure to disable PHY0 passby for peripheral mode

    [ Upstream commit e6f32efb1b128344a2c7df9875bc1a1abaa1d395 ]

    On platforms where the MUSB and HCI controllers share PHY0, PHY passby
    is required when using the HCI controller with the PHY, but it must be
    disabled when the MUSB controller is used instead.

    Without this, PHY0 passby is always enabled, which results in broken
    peripheral mode on such platforms (e.g. H3/H5).

    Fixes: ba4bdc9e1dc0 ("PHY: sunxi: Add driver for sunxi usb phy")

    Signed-off-by: Paul Kocialkowski <paul.kocialkowski@bootlin.com>
    Signed-off-by: Kishon Vijay Abraham I <kishon@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bf0280e58717b585fd63ec03a512aa2bd730099a
Author: Evan Green <evgreen@chromium.org>
Date:   Thu Mar 21 10:17:54 2019 -0700

    dt-bindings: phy-qcom-qmp: Add UFS PHY reset

    [ Upstream commit 95cee0b4e30a09a411a17e9a3bc6b72ed92063da ]

    Add a required reset to the SDM845 UFS phy to express the PHY reset
    bit inside the UFS controller register space. Before this change, this
    reset was not expressed in the DT, and the driver utilized two different
    callbacks (phy_init and phy_poweron) to implement a two-phase
    initialization procedure that involved deasserting this reset between
    init and poweron. This abused the two callbacks and diluted their
    purpose.

    That scheme does not work as regulators cannot be turned off in
    phy_poweroff because they were turned on in init, rather than poweron.
    The net result is that regulators are left on in suspend that shouldn't
    be.

    This new scheme gives the UFS reset to the PHY, so that it can fully
    initialize itself in a single callback. We can then turn regulators on
    during poweron and off during poweroff.

    Signed-off-by: Evan Green <evgreen@chromium.org>
    Reviewed-by: Rob Herring <robh@kernel.org>
    Reviewed-by: Stephen Boyd <swboyd@chromium.org>
    Signed-off-by: Kishon Vijay Abraham I <kishon@ti.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bf18396dbe081161f2bc484598819e5ba962f1f0
Author: Russell King <rmk+kernel@armlinux.org.uk>
Date:   Mon Feb 25 10:51:30 2019 +0000

    drm: etnaviv: avoid DMA API warning when importing buffers

    [ Upstream commit 1262cc8893ecb0eb2c21e042d0d268cc180edb61 ]
```

```
    During boot, I get this kernel warning:

    WARNING: CPU: 0 PID: 19001 at kernel/dma/debug.c:1301 debug_dma_map_sg+0x284/0x3dc
    etnaviv etnaviv: DMA-API: mapping sg segment longer than device claims to support
[len=3145728] [max=65536]
    Modules linked in: ip6t_REJECT nf_reject_ipv6 ip6t_rpfilter xt_tcpudp ipt_REJECT
nf_reject_ipv4 xt_conntrack ip_set nfnetlink ebtable_broute ebtable_nat ip6table_raw
ip6table_nat nf_nat_ipv6 ip6table_mangle iptable_raw iptable_nat nf_nat_ipv4 nf_nat
nf_conntrack nf_defrag_ipv4 nf_defrag_ipv6 libcrc32c iptable_mangle ebtable_filter ebtables
ip6table_filter ip6_tables iptable_filter caam_jr error snd_soc_imx_spdif imx_thermal
snd_soc_imx_audmux nvmem_imx_ocotp snd_soc_sgtl5000
    caam imx_sdma virt_dma coda rc_cec v4l2_mem2mem snd_soc_fsl_ssi snd_soc_fsl_spdif
imx_vdoa imx_pcm_dma videobuf2_dma_contig etnaviv dw_hdmi_cec gpu_sched dw_hdmi_ahb_audio
imx6q_cpufreq nfsd sch_fq_codel ip_tables x_tables
    CPU: 0 PID: 19001 Comm: Xorg Not tainted 4.20.0+ #307
    Hardware name: Freescale i.MX6 Quad/DualLite (Device Tree)
    [<c0019658>] (unwind_backtrace) from [<c001489c>] (show_stack+0x10/0x14)
    [<c001489c>] (show_stack) from [<c07fb420>] (dump_stack+0x9c/0xd4)
    [<c07fb420>] (dump_stack) from [<c00312dc>] (__warn+0xf8/0x124)
    [<c00312dc>] (__warn) from [<c00313d0>] (warn_slowpath_fmt+0x38/0x48)
    [<c00313d0>] (warn_slowpath_fmt) from [<c00b14e8>] (debug_dma_map_sg+0x284/0x3dc)
    [<c00b14e8>] (debug_dma_map_sg) from [<c046eb40>] (drm_gem_map_dma_buf+0xc4/0x13c)
    [<c046eb40>] (drm_gem_map_dma_buf) from [<c04c3314>] (dma_buf_map_attachment+0x38/0x5c)
    [<c04c3314>] (dma_buf_map_attachment) from [<c046e728>]
(drm_gem_prime_import_dev+0x74/0x104)
    [<c046e728>] (drm_gem_prime_import_dev) from [<c046e5bc>]
(drm_gem_prime_fd_to_handle+0x84/0x17c)
    [<c046e5bc>] (drm_gem_prime_fd_to_handle) from [<c046edd0>]
(drm_prime_fd_to_handle_ioctl+0x38/0x4c)
    [<c046edd0>] (drm_prime_fd_to_handle_ioctl) from [<c0460efc>]
(drm_ioctl_kernel+0x90/0xc8)
    [<c0460efc>] (drm_ioctl_kernel) from [<c0461114>] (drm_ioctl+0x1e0/0x3b0)
    [<c0461114>] (drm_ioctl) from [<c01cae20>] (do_vfs_ioctl+0x90/0xa48)
    [<c01cae20>] (do_vfs_ioctl) from [<c01cb80c>] (ksys_ioctl+0x34/0x60)
    [<c01cb80c>] (ksys_ioctl) from [<c0009000>] (ret_fast_syscall+0x0/0x28)
    Exception stack(0xd81a9fa8 to 0xd81a9ff0)
    9fa0:                   b6c69c88 bec613f8 00000009 c00c642e bec613f8 b86c4600
    9fc0: b6c69c88 bec613f8 c00c642e 00000036 012762e0 01276348 00000300 012d91f8
    9fe0: b6989f18 bec613dc b697185c b667be5c
    irq event stamp: 47905
    hardirqs last  enabled at (47913): [<c0098824>] console_unlock+0x46c/0x680
    hardirqs last disabled at (47922): [<c0098470>] console_unlock+0xb8/0x680
    softirqs last  enabled at (47754): [<c000a484>] __do_softirq+0x344/0x540
    softirqs last disabled at (47701): [<c0038700>] irq_exit+0x124/0x144
    ---[ end trace af477747acbcc642 ]---

    The reason is the contiguous buffer exceeds the default maximum segment
    size of 64K as specified by dma_get_max_seg_size() in
    linux/dma-mapping.h.  Fix this by providing our own segment size, which
    is set to 2GiB to cover the window found in MMUv1 GPUs.

    Signed-off-by: Russell King <rmk+kernel@armlinux.org.uk>
    Signed-off-by: Lucas Stach <l.stach@pengutronix.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5c907df83abe178590c9a7675a1b96a60f8e7a15
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Sun Apr 14 17:59:38 2019 +0200

    x86/irq/64: Limit IST stack overflow check to #DB stack

    [ Upstream commit 7dbcf2b0b770eeb803a416ee8dcbef78e6389d40 ]

    Commit
```

```
       37fe6a42b343 ("x86: Check stack overflow in detail")

    added a broad check for the full exception stack area, i.e. it considers
    the full exception stack area as valid.

    That's wrong in two aspects:

     1) It does not check the individual areas one by one

     2) #DF, NMI and #MCE are not enabling interrupts which means that a
        regular device interrupt cannot happen in their context. In fact if a
        device interrupt hits one of those IST stacks that's a bug because some
        code path enabled interrupts while handling the exception.

    Limit the check to the #DB stack and consider all other IST stacks as
    'overflow' or invalid.

    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Mitsuo Hayasaka <mitsuo.hayasaka.hu@hitachi.com>
    Cc: Nicolai Stange <nstange@suse.de>
    Cc: Sean Christopherson <sean.j.christopherson@intel.com>
    Cc: x86-ml <x86@kernel.org>
    Link: https://lkml.kernel.org/r/20190414160143.682135110@linutronix.de
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit abda52890034331e2dd12048b81843af0c510b16
Author: Alan Stern <stern@rowland.harvard.edu>
Date:   Tue Apr 16 10:50:01 2019 -0400

    USB: core: Don't unbind interfaces following device reset failure

    [ Upstream commit 381419fa720060ba48b7bbc483be787d5b1dca6f ]

    The SCSI core does not like to have devices or hosts unregistered
    while error recovery is in progress.  Trying to do so can lead to
    self-deadlock: Part of the removal code tries to obtain a lock already
    held by the error handler.

    This can cause problems for the usb-storage and uas drivers, because
    their error handler routines perform a USB reset, and if the reset
    fails then the USB core automatically goes on to unbind all drivers
    from the device's interfaces -- all while still in the context of the
    SCSI error handler.

    As it turns out, practically all the scenarios leading to a USB reset
    failure end up causing a device disconnect (the main error pathway in
    usb_reset_and_verify_device(), at the end of the routine, calls
    hub_port_logical_disconnect() before returning).  As a result, the
    hub_wq thread will soon become aware of the problem and will unbind
    all the device's drivers in its own context, not in the
    error-handler's context.

    This means that usb_reset_device() does not need to call
    usb_unbind_and_rebind_marked_interfaces() in cases where
    usb_reset_and_verify_device() has returned an error, because hub_wq
    will take care of everything anyway.

    This particular problem was observed in somewhat artificial
    circumstances, by using usbfs to tell a hub to power-down a port
```

```
    connected to a USB-3 mass storage device using the UAS protocol.  With
    the port turned off, the currently executing command timed out and the
    error handler started running.  The USB reset naturally failed,
    because the hub port was off, and the error handler deadlocked as
    described above.  Not carrying out the call to
    usb_unbind_and_rebind_marked_interfaces() fixes this issue.

    Signed-off-by: Alan Stern <stern@rowland.harvard.edu>
    Reported-by: Kento Kobayashi <Kento.A.Kobayashi@sony.com>
    Tested-by: Kento Kobayashi <Kento.A.Kobayashi@sony.com>
    CC: Bart Van Assche <bvanassche@acm.org>
    CC: Martin K. Petersen <martin.petersen@oracle.com>
    CC: Jacky Cao <Jacky.Cao@sony.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dfb23004c1191e01d9e88c06533db58fe451ddc6
Author: Julian Wiedmann <jwi@linux.ibm.com>
Date:   Wed Apr 17 18:17:29 2019 +0200

    s390/qeth: handle error from qeth_update_from_chp_desc()

    [ Upstream commit a4cdc9baee0740748f16e50cd70c2607510df492 ]

    Subsequent code relies on the values that qeth_update_from_chp_desc()
    reads from the CHP descriptor. Rather than dealing with weird errors
    later on, just handle it properly here.

    Signed-off-by: Julian Wiedmann <jwi@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a3d844b924ebd3553edc0f3ce4eba4daa7df842
Author: Mika Westerberg <mika.westerberg@linux.intel.com>
Date:   Tue Mar 19 16:48:41 2019 +0200

    thunderbolt: Take domain lock in switch sysfs attribute callbacks

    [ Upstream commit 09f11b6c99feaf86a26444bca85dc693b3f58f8b ]

    switch_lock was introduced because it allowed serialization of device
    authorization requests from userspace without need to take the big
    domain lock (tb->lock). This was fine because device authorization with
    ICM is just one command that is sent to the firmware. Now that we start
    to handle all tunneling in the driver switch_lock is not enough because
    we need to walk over the topology to establish paths.

    For this reason drop switch_lock from the driver completely in favour of
    big domain lock.

    There is one complication, though. If userspace is waiting for the lock
    in tb_switch_set_authorized(), it keeps the device_del() from removing
    the sysfs attribute because it waits for active users to release the
    attribute first which leads into following splat:

        INFO: task kworker/u8:3:73 blocked for more than 61 seconds.
              Tainted: G        W         5.1.0-rc1+ #244
        "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message.
        kworker/u8:3    D12976    73      2 0x80000000
        Workqueue: thunderbolt0 tb_handle_hotplug [thunderbolt]
        Call Trace:
         ? __schedule+0x2e5/0x740
         ? _raw_spin_lock_irqsave+0x12/0x40
         ? prepare_to_wait_event+0xc5/0x160
         schedule+0x2d/0x80
```

```
          __kernfs_remove.part.17+0x183/0x1f0
          ? finish_wait+0x80/0x80
          kernfs_remove_by_name_ns+0x4a/0x90
          remove_files.isra.1+0x2b/0x60
          sysfs_remove_group+0x38/0x80
          sysfs_remove_groups+0x24/0x40
          device_remove_attrs+0x3d/0x70
          device_del+0x14c/0x360
          device_unregister+0x15/0x50
          tb_switch_remove+0x9e/0x1d0 [thunderbolt]
          tb_handle_hotplug+0x119/0x5a0 [thunderbolt]
          ? process_one_work+0x1b7/0x420
          process_one_work+0x1b7/0x420
          worker_thread+0x37/0x380
          ? _raw_spin_unlock_irqrestore+0xf/0x30
          ? process_one_work+0x420/0x420
          kthread+0x118/0x130
          ? kthread_create_on_node+0x60/0x60
          ret_from_fork+0x35/0x40

    We deal this by following what network stack did for some of their
    attributes and use mutex_trylock() with restart_syscall(). This makes
    userspace release the attribute allowing sysfs attribute removal to
    progress before the write is restarted and eventually fail when the
    attribute is removed.

    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5f335f7c418236760322940d035559d3f1a816c1
Author: Nicholas Piggin <npiggin@gmail.com>
Date:   Tue Apr 9 19:34:03 2019 +1000

    irq_work: Do not raise an IPI when queueing work on the local CPU

    [ Upstream commit 471ba0e686cb13752bc1ff3216c54b69a2d250ea ]

    The QEMU PowerPC/PSeries machine model was not expecting a self-IPI,
    and it may be a bit surprising thing to do, so have irq_work_queue_on
    do local queueing when target is the current CPU.

    Suggested-by: Steven Rostedt <rostedt@goodmis.org>
    Reported-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Tested-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Signed-off-by: Nicholas Piggin <npiggin@gmail.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Reviewed-by: Frederic Weisbecker <frederic@kernel.org>
    Acked-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: =?UTF-8?q?C=C3=A9dric=20Le=20Goater?= <clg@kaod.org>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Paul Mackerras <paulus@samba.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Suraj Jitindar Singh <sjitindarsingh@gmail.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lkml.kernel.org/r/20190409093403.20994-1-npiggin@gmail.com
    [ Simplified the preprocessor comments.
      Fixed unbalanced curly brackets pointed out by Thomas. ]
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a02a8367e8599785f7dbaca3d460784028f42457
Author: Luca Weiss <luca@z3ntu.xyz>
Date:   Sat Mar 2 13:35:29 2019 +0100

    drm/msm: Fix NULL pointer dereference
```

```
    [ Upstream commit 7603df38cc8c1e5d540b18ec9eb9d62d823197d0 ]

    [    3.707412] Unable to handle kernel NULL pointer dereference at virtual address
 0000009c
    [    3.714511] pgd = (ptrval)
    [    3.722742] [0000009c] *pgd=00000000
    [    3.725238] Internal error: Oops: 5 [#1] PREEMPT SMP ARM
    [    3.728968] Modules linked in:
    [    3.734265] CPU: 3 PID: 112 Comm: kworker/3:2 Tainted: G        W         5.0.0-rc7-
 00183-g06a1c31df9eb #4
    [    3.737142] Hardware name: Generic DT based system
    [    3.746778] Workqueue: events deferred_probe_work_func
    [    3.751542] PC is at msm_gem_map_vma+0x3c/0xac
    [    3.756669] LR is at msm_gem_get_and_pin_iova+0xd8/0x134
    [    3.761086] pc : [<c07d3b7c>]    lr : [<c07d14f8>]    psr: 60000013
    [    3.766560] sp : ee297be8  ip : ed9ab1c0  fp : ed93b800
    [    3.772546] r10: ee35e180  r9 : 00000000  r8 : ee297c80
    [    3.777752] r7 : 00000000  r6 : 7c100000  r5 : 00000000  r4 : ee35e180
    [    3.782968] r3 : 00000001  r2 : 00000003  r1 : ee35e180  r0 : 00000000
    [    3.789562] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
    [    3.796079] Control: 10c5787d  Table: 2e3a806a  DAC: 00000051
    [    3.803282] Process kworker/3:2 (pid: 112, stack limit = 0x(ptrval))
    [    3.809006] Stack: (0xee297be8 to 0xee298000)
    [    3.815445] 7be0:                   00000000 c1108c48 eda8c000 00000003 eda8c0fc
 c1108c48
    [    3.819715] 7c00: eda8c000 00000003 eda8c0fc c07d14f8 00000001 c07d1100 7c100000
 00000000
    [    3.827873] 7c20: eda8c000 bb7ffb78 00000000 eda8c000 00000000 00000000 c0c8b1d4
 ee3bfa00
    [    3.836037] 7c40: ee3b9800 c07d1684 00000000 c1108c48 ee0d7810 ee3b9800 c0c8b1d4
 c07d222c
    [    3.844193] 7c60: ee3bfd84 ee297c80 00000000 c0b1d5b0 ee3bfc40 c07dcfd8 ee3bfd84
 ee297c80
    [    3.852357] 7c80: 0000006d ee3bfc40 ee0d7810 bb7ffb78 c0c8b1d4 00000000 ee3bfc40
 c07ddb48
    [    3.860516] 7ca0: 00002004 c0eba384 ee3bfc40 c079eba0 ee3bd040 ee3b9800 00000001
 ed93b800
    [    3.868673] 7cc0: ed9aa100 c07db7e8 ee3bf240 ed9a6500 00000001 ee3b9800 ee3bf2d4
 c07a0a30
    [    3.876834] 7ce0: ed93b800 7d100000 c1108c48 ee0d7610 ee3b9800 ed93b800 c1108c48
 00000000
    [    3.884991] 7d00: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 00000000
    [    3.893151] 7d20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 bb7ffb78
    [    3.901310] 7d40: c12113c4 ed93b800 ee3b9800 c1108c48 ee9eec10 00000000 ed93b800
 7d100000
    [    3.909472] 7d60: eff7b000 c07cf748 7d100000 00000000 c0e9a350 c0b1d5b0 c12113c4
 c0961e40
    [    3.917633] 7d80: c12113c4 40000113 eeff4bec c0ebe004 00000019 c0b1d230 ee9eeda8
 60000113
    [    3.925791] 7da0: ee35d300 ee9eeda8 c07ce260 bb7ffb78 c07ce260 ee35d2c0 00000028
 00000002
    [    3.933950] 7dc0: eeb76280 c118f884 ee0be640 c11c6128 c07ce260 c07ea4ac 00000000
 c0962b48
    [    3.942108] 7de0: c118f868 00000001 c0ebbc98 ee35d2c0 00000000 eeb76280 00000000
 c118f87c
    [    3.950270] 7e00: ee35d2c0 00000000 c11c63e0 c118f694 00000019 c07ea5d0 ee0d7810
 00000000
    [    3.958430] 7e20: c118f694 00000000 00000000 c07f2b0c c120f55c ee0d7810 c120f560
 00000000
    [    3.966590] 7e40: 00000000 c07f08c4 c07f0e8c ee0d7810 c11ba3d0 ee0d7810 c118f694
 c07f0e8c
    [    3.974748] 7e60: c1108c48 00000001 c0ebc3cc c11c63f8 c11ba3d0 c07f0c08 00000001
```

```
 c07f2f8c
     [    3.982908] 7e80: c118f694 00000000 ee297ed4 c07f0e8c c1108c48 00000001 c0ebc3cc
 c11c63f8
     [    3.991068] 7ea0: c11ba3d0 c07ee8a0 c11ba3d0 ee82686c ee0baf38 bb7ffb78 ee0d7810
 ee0d7810
     [    3.999227] 7ec0: c1108c48 ee0d7844 c118faac c07f05b0 ee0d7810 ee0d7810 00000001
 bb7ffb78
     [    4.007389] 7ee0: ee0d7810 ee0d7810 c118fd18 c118faac c11c63e0 c07ef7d0 ee0d7810
 c118fa90
     [    4.015548] 7f00: c118fa90 c07efd68 c118fac8 ee27fe00 eefd9c80 eefdcd00 00000000
 c118facc
     [    4.023708] 7f20: 00000000 c033c038 eefd9c80 eefd9c80 00000008 ee27fe00 ee27fe14
 eefd9c80
     [    4.031866] 7f40: 00000008 c1103d00 eefd9c98 ee296000 eefd9c80 c033ce54 ee907eac
 c0b1d230
     [    4.040026] 7f60: ee907eac eea24440 ee285000 00000000 ee296000 ee27fe00 c033ce24
 eea2445c
     [    4.048188] 7f80: ee907eac c0341db0 00000000 ee285000 c0341c8c 00000000 00000000
 00000000
     [    4.056346] 7fa0: 00000000 00000000 00000000 c03010e8 00000000 00000000 00000000
 00000000
     [    4.064505] 7fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 00000000
     [    4.072665] 7fe0: 00000000 00000000 00000000 00000000 00000013 00000000 00000000
 00000000
     [    4.080828] [<c07d3b7c>] (msm_gem_map_vma) from [<c07d14f8>]
 (msm_gem_get_and_pin_iova+0xd8/0x134)
     [    4.088983] [<c07d14f8>] (msm_gem_get_and_pin_iova) from [<c07d1684>]
 (_msm_gem_kernel_new+0x38/0xac)
     [    4.097839] [<c07d1684>] (_msm_gem_kernel_new) from [<c07d222c>]
 (msm_gem_kernel_new+0x24/0x2c)
     [    4.107130] [<c07d222c>] (msm_gem_kernel_new) from [<c07dcfd8>]
 (dsi_tx_buf_alloc_6g+0x44/0x90)
     [    4.115631] [<c07dcfd8>] (dsi_tx_buf_alloc_6g) from [<c07ddb48>]
 (msm_dsi_host_modeset_init+0x80/0x104)
     [    4.124313] [<c07ddb48>] (msm_dsi_host_modeset_init) from [<c07db7e8>]
 (msm_dsi_modeset_init+0x34/0x1c0)
     [    4.133691] [<c07db7e8>] (msm_dsi_modeset_init) from [<c07a0a30>]
 (mdp5_kms_init+0x764/0x7e0)
     [    4.143409] [<c07a0a30>] (mdp5_kms_init) from [<c07cf748>] (msm_drm_bind+0x56c/0x740)
     [    4.151824] [<c07cf748>] (msm_drm_bind) from [<c07ea4ac>]
 (try_to_bring_up_master+0x238/0x2b4)
     [    4.159636] [<c07ea4ac>] (try_to_bring_up_master) from [<c07ea5d0>]
 (component_add+0xa8/0x170)
     [    4.168146] [<c07ea5d0>] (component_add) from [<c07f2b0c>]
 (platform_drv_probe+0x48/0x9c)
     [    4.176737] [<c07f2b0c>] (platform_drv_probe) from [<c07f08c4>]
 (really_probe+0x278/0x404)
     [    4.184981] [<c07f08c4>] (really_probe) from [<c07f0c08>]
 (driver_probe_device+0x78/0x1c0)
     [    4.193147] [<c07f0c08>] (driver_probe_device) from [<c07ee8a0>]
 (bus_for_each_drv+0x74/0xb8)
     [    4.201389] [<c07ee8a0>] (bus_for_each_drv) from [<c07f05b0>]
 (__device_attach+0xd0/0x164)
     [    4.209984] [<c07f05b0>] (__device_attach) from [<c07ef7d0>]
 (bus_probe_device+0x84/0x8c)
     [    4.218143] [<c07ef7d0>] (bus_probe_device) from [<c07efd68>]
 (deferred_probe_work_func+0x48/0xc4)
     [    4.226398] [<c07efd68>] (deferred_probe_work_func) from [<c033c038>]
 (process_one_work+0x204/0x574)
     [    4.235254] [<c033c038>] (process_one_work) from [<c033ce54>]
 (worker_thread+0x30/0x560)
     [    4.244534] [<c033ce54>] (worker_thread) from [<c0341db0>] (kthread+0x124/0x154)
     [    4.252606] [<c0341db0>] (kthread) from [<c03010e8>] (ret_from_fork+0x14/0x2c)
     [    4.259966] Exception stack(0xee297fb0 to 0xee297ff8)
```

```
     [    4.266998] 7fa0:                                     00000000 00000000 00000000
 00000000
     [    4.272143] 7fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 00000000
     [    4.280297] 7fe0: 00000000 00000000 00000000 00000000 00000013 00000000
     [    4.288451] Code: e5813080 1a000013 e3a03001 e5c4307c (e590009c)
     [    4.294933] ---[ end trace 18729cc2bca2b4b3 ]---

     Signed-off-by: Luca Weiss <luca@z3ntu.xyz>
     Signed-off-by: Rob Clark <robdclark@gmail.com>
     Signed-off-by: Rob Clark <robdclark@chromium.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 48f41424127ea4812a86c1a2819852f92605f3e9
 Author: Sean Paul <seanpaul@chromium.org>
 Date:   Wed Jan 30 11:32:12 2019 -0500

     drm/msm: dpu: Don't set frame_busy_mask for async updates

     [ Upstream commit f98baa3109cea46083d2361ab14a0207d1b1bd16 ]

     The frame_busy mask is used in frame_done event handling, which is not
     invoked for async commits. So an async commit will leave the
     frame_busy mask populated after it completes and future commits will start
     with the busy mask incorrect.

     This showed up on disable after cursor move. I was hitting the "this should
     not happen" comment in the frame event worker since frame_busy was set,
     we queued the event, but there were no frames pending (since async
     also doesn't set that).

     Reviewed-by: Fritz Koenig <frkoenig@google.com>
     Signed-off-by: Sean Paul <seanpaul@chromium.org>
     Link: https://patchwork.freedesktop.org/patch/msgid/20190130163220.138637-1-
 sean@poorly.run
     Signed-off-by: Rob Clark <robdclark@chromium.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 15180fd9558244e7d181a88ee4035eafa2743760
 Author: Wen Yang <wen.yang99@zte.com.cn>
 Date:   Thu Apr 4 00:04:11 2019 +0800

     drm/msm: a5xx: fix possible object reference leak

     [ Upstream commit 6cd5235c3135ea84b32469ea51b2aae384eda8af ]

     The call to of_get_child_by_name returns a node pointer with refcount
     incremented thus it must be explicitly decremented after the last
     usage.

     Detected by coccinelle with the following warnings:
     drivers/gpu/drm/msm/adreno/a5xx_gpu.c:57:2-8: ERROR: missing of_node_put; acquired a
 node pointer with refcount incremented on line 47, but without a corresponding object
 release within this function.
     drivers/gpu/drm/msm/adreno/a5xx_gpu.c:66:2-8: ERROR: missing of_node_put; acquired a
 node pointer with refcount incremented on line 47, but without a corresponding object
 release within this function.
     drivers/gpu/drm/msm/adreno/a5xx_gpu.c:118:1-7: ERROR: missing of_node_put; acquired a
 node pointer with refcount incremented on line 47, but without a corresponding object
 release within this function.
     drivers/gpu/drm/msm/adreno/a5xx_gpu.c:57:2-8: ERROR: missing of_node_put; acquired a
 node pointer with refcount incremented on line 51, but without a corresponding object
 release within this function.
     drivers/gpu/drm/msm/adreno/a5xx_gpu.c:66:2-8: ERROR: missing of_node_put; acquired a
 node pointer with refcount incremented on line 51, but without a corresponding object
```

```
release within this function.
    drivers/gpu/drm/msm/adreno/a5xx_gpu.c:118:1-7: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 51, but without a corresponding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Rob Clark <robdclark@gmail.com>
    Cc: Sean Paul <sean@poorly.run>
    Cc: David Airlie <airlied@linux.ie>
    Cc: Daniel Vetter <daniel@ffwll.ch>
    Cc: Jordan Crouse <jcrouse@codeaurora.org>
    Cc: Mamta Shukla <mamtashukla555@gmail.com>
    Cc: Thomas Zimmermann <tzimmermann@suse.de>
    Cc: Sharat Masetty <smasetty@codeaurora.org>
    Cc: linux-arm-msm@vger.kernel.org
    Cc: dri-devel@lists.freedesktop.org
    Cc: freedreno@lists.freedesktop.org
    Cc: linux-kernel@vger.kernel.org (open list)
    Reviewed-by: Jordan Crouse <jcrouse@codeaurora.org>
    Signed-off-by: Rob Clark <robdclark@gmail.com>
    Signed-off-by: Rob Clark <robdclark@chromium.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aa6b53c2bcf9bb88b825af07e49df7fad5cf6324
Author: Jeykumar Sankaran <jsanka@codeaurora.org>
Date:   Wed Feb 13 17:19:12 2019 -0800

    drm/msm/dpu: release resources on modeset failure

    [ Upstream commit a7fcc3237f31a4e206953bb73cf41bd429442f09 ]

    release resources allocated in mode_set if any of
    the hw check fails. Most of these checks are not
    necessary and they will be removed in the follow up
    patches with state based resource allocations.

    Signed-off-by: Jeykumar Sankaran <jsanka@codeaurora.org>
    Signed-off-by: Sean Paul <seanpaul@chromium.org>
    Link: https://patchwork.freedesktop.org/patch/msgid/1550107156-17625-4-git-send-email-
jsanka@codeaurora.org
    Signed-off-by: Rob Clark <robdclark@chromium.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 216fb3cc905789e2a5216bfa11d9f229d151935a
Author: Nicholas Mc Guire <hofrat@osadl.org>
Date:   Fri Apr 19 01:31:08 2019 +0200

    staging: vc04_services: handle kzalloc failure

    [ Upstream commit a5112277872a56017b777770e2fd4324d4a6c866 ]

    The kzalloc here was being used without checking the return - if the
    kzalloc fails return VCHIQ_ERROR. The call-site of
    vchiq_platform_init_state() vchiq_init_state() was not responding
    to an allocation failure so checks for != VCHIQ_SUCCESS
    and pass VCHIQ_ERROR up to vchiq_platform_init() which then
    will fail with -EINVAL.

    Signed-off-by: Nicholas Mc Guire <hofrat@osadl.org>
    Reported-by: kbuild test robot <lkp@intel.com>
    Acked-By: Stefan Wahren <stefan.wahren@i2se.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 16e061eb21f67c4162badc977f6dcb0b5d178e66
```

```
Author: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
Date:   Wed Feb 27 11:10:18 2019 +0300

    sched/core: Handle overflow in cpu_shares_write_u64

    [ Upstream commit 5b61d50ab4ef590f5e1d4df15cd2cea5f5715308 ]

    Bit shift in scale_load() could overflow shares. This patch saturates
    it to MAX_SHARES like following sched_group_set_shares().

    Example:

     # echo 9223372036854776832 > cpu.shares
     # cat cpu.shares

    Before patch: 1024
    After pattch: 262144

    Signed-off-by: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
    Acked-by: Peter Zijlstra <a.p.zijlstra@chello.nl>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: http://lkml.kernel.org/r/155125501891.293431.3345233332801109696.stgit@buzz
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b698e991346ed5dc9713e3797a5a2176d544d2ca
Author: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
Date:   Wed Feb 27 11:10:17 2019 +0300

    sched/rt: Check integer overflow at usec to nsec conversion

    [ Upstream commit 1a010e29cfa00fee2888fd2fd4983f848cbafb58 ]

    Example of unhandled overflows:

     # echo 18446744073709651 > cpu.rt_runtime_us
     # cat cpu.rt_runtime_us
     99

     # echo 18446744073709900 > cpu.rt_period_us
     # cat cpu.rt_period_us
     348

    After this patch they will fail with -EINVAL.

    Signed-off-by: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
    Acked-by: Peter Zijlstra <a.p.zijlstra@chello.nl>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: http://lkml.kernel.org/r/155125501739.293431.5252197504404771496.stgit@buzz
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dcfdbf774d832d29ec9fbb9df7afcd41f1617c04
Author: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
Date:   Wed Feb 27 11:10:20 2019 +0300

    sched/core: Check quota and period overflow at usec to nsec conversion

    [ Upstream commit 1a8b4540db732ca16c9e43ac7c08b1b8f0b252d8 ]

    Large values could overflow u64 and pass following sanity checks.
```

```
      # echo 18446744073750000 > cpu.cfs_period_us
      # cat cpu.cfs_period_us
      40448

      # echo 18446744073750000 > cpu.cfs_quota_us
      # cat cpu.cfs_quota_us
      40448

    After this patch they will fail with -EINVAL.

    Signed-off-by: Konstantin Khlebnikov <khlebnikov@yandex-team.ru>
    Acked-by: Peter Zijlstra <a.p.zijlstra@chello.nl>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: http://lkml.kernel.org/r/155125502079.293431.3947497929372138600.stgit@buzz
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fa31707881df755e5479a9efb2439a0f6db27817
Author: Roman Gushchin <guro@fb.com>
Date:   Fri Apr 19 10:03:03 2019 -0700

    cgroup: protect cgroup->nr_(dying_)descendants by css_set_lock

    [ Upstream commit 4dcabece4c3a9f9522127be12cc12cc120399b2f ]

    The number of descendant cgroups and the number of dying
    descendant cgroups are currently synchronized using the cgroup_mutex.

    The number of descendant cgroups will be required by the cgroup v2
    freezer, which will use it to determine if a cgroup is frozen
    (depending on total number of descendants and number of frozen
    descendants). It's not always acceptable to grab the cgroup_mutex,
    especially from quite hot paths (e.g. exit()).

    To avoid this, let's additionally synchronize these counters using
    the css_set_lock.

    So, it's safe to read these counters with either cgroup_mutex or
    css_set_lock locked, and for changing both locks should be acquired.

    Signed-off-by: Roman Gushchin <guro@fb.com>
    Signed-off-by: Tejun Heo <tj@kernel.org>
    Cc: kernel-team@fb.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 50a41f60e5deb7edb36f3861db1ff45e2d9c8c63
Author: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
Date:   Sat Apr 20 00:09:51 2019 -0400

    random: add a spinlock_t to struct batched_entropy

    [ Upstream commit b7d5dc21072cda7124d13eae2aefb7343ef94197 ]

    The per-CPU variable batched_entropy_uXX is protected by get_cpu_var().
    This is just a preempt_disable() which ensures that the variable is only
    from the local CPU. It does not protect against users on the same CPU
    from another context. It is possible that a preemptible context reads
    slot 0 and then an interrupt occurs and the same value is read again.

    The above scenario is confirmed by lockdep if we add a spinlock:
    | ================================
    | WARNING: inconsistent lock state
```

```
      | 5.1.0-rc3+ #42 Not tainted
      | ------------------------------
      | inconsistent {SOFTIRQ-ON-W} -> {IN-SOFTIRQ-W} usage.
      | ksoftirqd/9/56 [HC0[0]:SC1[1]:HE0:SE0] takes:
      | (____ptrval____) (batched_entropy_u32.lock){+.?.}, at: get_random_u32+0x3e/0xe0
      | {SOFTIRQ-ON-W} state was registered at:
      |   _raw_spin_lock+0x2a/0x40
      |   get_random_u32+0x3e/0xe0
      |   new_slab+0x15c/0x7b0
      |   ___slab_alloc+0x492/0x620
      |   __slab_alloc.isra.73+0x53/0xa0
      |   kmem_cache_alloc_node+0xaf/0x2a0
      |   copy_process.part.41+0x1e1/0x2370
      |   _do_fork+0xdb/0x6d0
      |   kernel_thread+0x20/0x30
      |   kthreadd+0x1ba/0x220
      |   ret_from_fork+0x3a/0x50
      …
      | other info that might help us debug this:
      |  Possible unsafe locking scenario:
      |
      |        CPU0
      |        ----
      |   lock(batched_entropy_u32.lock);
      |   <Interrupt>
      |     lock(batched_entropy_u32.lock);
      |
      |  *** DEADLOCK ***
      |
      | stack backtrace:
      | Call Trace:
      …
      |  kmem_cache_alloc_trace+0x20e/0x270
      |  ipmi_alloc_recv_msg+0x16/0x40
      …
      |  __do_softirq+0xec/0x48d
      |  run_ksoftirqd+0x37/0x60
      |  smpboot_thread_fn+0x191/0x290
      |  kthread+0xfe/0x130
      |  ret_from_fork+0x3a/0x50

    Add a spinlock_t to the batched_entropy data structure and acquire the
    lock while accessing it. Acquire the lock with disabled interrupts
    because this function may be used from interrupt context.

    Remove the batched_entropy_reset_lock lock. Now that we have a lock for
    the data scructure, we can access it from a remote CPU.

    Signed-off-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6f9c63bc3bfa95a82a5c8fa017ce898d47cbfa06
Author: Jon DeVree <nuxi@vault24.org>
Date:   Fri Apr 19 23:35:16 2019 -0400

    random: fix CRNG initialization when random.trust_cpu=1

    [ Upstream commit fe6f1a6a8eedc1aa538fee0baa612b6a59639cf8 ]

    When the system boots with random.trust_cpu=1 it doesn't initialize the
    per-NUMA CRNGs because it skips the rest of the CRNG startup code. This
    means that the code from 1e7f583af67b ("random: make /dev/urandom scalable
    for silly userspace programs") is not used when random.trust_cpu=1.
```

```
    crash> dmesg | grep random:
    [    0.000000] random: get_random_bytes called from start_kernel+0x94/0x530 with
 crng_init=0
    [    0.314029] random: crng done (trusting CPU's manufacturer)
    crash> print crng_node_pool
    $6 = (struct crng_state **) 0x0

    After adding the missing call to numa_crng_init() the per-NUMA CRNGs are
    initialized again:

    crash> dmesg | grep random:
    [    0.000000] random: get_random_bytes called from start_kernel+0x94/0x530 with
 crng_init=0
    [    0.314031] random: crng done (trusting CPU's manufacturer)
    crash> print crng_node_pool
    $1 = (struct crng_state **) 0xffff9a915f4014a0

    The call to invalidate_batched_entropy() was also missing. This is
    important for architectures like PPC and S390 which only have the
    arch_get_random_seed_* functions.

    Fixes: 39a8883a2b98 ("random: add a config option to trust the CPU's hwrng")
    Signed-off-by: Jon DeVree <nuxi@vault24.org>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit ee079354fba85bda54b9c64f662b8cd16260fdf8
 Author: Russell Currey <ruscur@russell.cc>
 Date:   Wed Mar 27 14:35:54 2019 +1100

    powerpc/64: Fix booting large kernels with STRICT_KERNEL_RWX

    [ Upstream commit 56c46bba9bbfe229b4472a5be313c44c5b714a39 ]

    With STRICT_KERNEL_RWX enabled anything marked __init is placed at a 16M
    boundary.  This is necessary so that it can be repurposed later with
    different permissions.  However, in kernels with text larger than 16M,
    this pushes early_setup past 32M, incapable of being reached by the
    branch instruction.

    Fix this by setting the CTR and branching there instead.

    Fixes: 1e0fc9d1eb2b ("powerpc/Kconfig: Enable STRICT_KERNEL_RWX for some configs")
    Signed-off-by: Russell Currey <ruscur@russell.cc>
    [mpe: Fix it to work on BE by using DOTSYM()]
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit c4fb49c727050a488d113ad5acb46a1822c161f7
 Author: Nathan Lynch <nathanl@linux.ibm.com>
 Date:   Thu Apr 18 13:56:57 2019 -0500

    powerpc/numa: improve control of topology updates

    [ Upstream commit 2d4d9b308f8f8dec68f6dbbff18c68ec7c6bd26f ]

    When booted with "topology_updates=no", or when "off" is written to
    /proc/powerpc/topology_updates, NUMA reassignments are inhibited for
    PRRN and VPHN events. However, migration and suspend unconditionally
    re-enable reassignments via start_topology_update(). This is
    incoherent.

    Check the topology_updates_enabled flag in
    start/stop_topology_update() so that callers of those APIs need not be
    aware of whether reassignments are enabled. This allows the
```

```
    administrative decision on reassignments to remain in force across
    migrations and suspensions.

    Signed-off-by: Nathan Lynch <nathanl@linux.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5090dd8f2804fa746e668fe5c047ad1664ef9528
Author: Yufen Yu <yuyufen@huawei.com>
Date:   Tue Apr 2 20:06:34 2019 +0800

    block: fix use-after-free on gendisk

    [ Upstream commit 2c88e3c7ec32d7a40cc7c9b4a487cf90e4671bdd ]

    commit 2da78092dda "block: Fix dev_t minor allocation lifetime"
    specifically moved blk_free_devt(dev->devt) call to part_release()
    to avoid reallocating device number before the device is fully
    shutdown.

    However, it can cause use-after-free on gendisk in get_gendisk().
    We use md device as example to show the race scenes:

    Process1                Worker                  Process2
    md_free
                                                    blkdev_open
    del_gendisk
      add delete_partition_work_fn() to wq
                                                    __blkdev_get
                                                    get_gendisk
    put_disk
      disk_release
        kfree(disk)
                                                    find part from ext_devt_idr
                                                    get_disk_and_module(disk)
                                                    cause use after free

                            delete_partition_work_fn
                            put_device(part)
                            part_release
                            remove part from ext_devt_idr

    Before <devt, hd_struct pointer> is removed from ext_devt_idr by
    delete_partition_work_fn(), we can find the devt and then access
    gendisk by hd_struct pointer. But, if we access the gendisk after
    it have been freed, it can cause in use-after-freeon gendisk in
    get_gendisk().

    We fix this by adding a new helper blk_invalidate_devt() in
    delete_partition() and del_gendisk(). It replaces hd_struct
    pointer in idr with value 'NULL', and deletes the entry from
    idr in part_release() as we do now.

    Thanks to Jan Kara for providing the solution and more clear comments
    for the code.

    Fixes: 2da78092dda1 ("block: Fix dev_t minor allocation lifetime")
    Cc: Al Viro <viro@zeniv.linux.org.uk>
    Reviewed-by: Bart Van Assche <bvanassche@acm.org>
    Reviewed-by: Keith Busch <keith.busch@intel.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Suggested-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Yufen Yu <yuyufen@huawei.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 3404d705d04133f3efff06da6724b93406964bd6
Author: Fabrice Gasnier <fabrice.gasnier@st.com>
Date:   Mon Apr 15 15:00:50 2019 +0200

    iio: adc: stm32-dfsdm: fix unmet direct dependencies detected

    [ Upstream commit ba7ecfe43d6bf12e2aa76705c45f7d187ae3d7c0 ]

    This fixes unmet direct dependencies seen when CONFIG_STM32_DFSDM_ADC
    is selected:

    WARNING: unmet direct dependencies detected for IIO_BUFFER_HW_CONSUMER
      Depends on [n]: IIO [=y] && IIO_BUFFER [=n]
      Selected by [y]:
      - STM32_DFSDM_ADC [=y] && IIO [=y] && (ARCH_STM32 [=y] && OF [=y] ||
        COMPILE_TEST [=n])

    Fixes: e2e6771c6462 ("IIO: ADC: add STM32 DFSDM sigma delta ADC support")

    Signed-off-by: Fabrice Gasnier <fabrice.gasnier@st.com>
    Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5ffb30e6fa840af9dc4e478eb1c31f0ba40e5f26
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Mon Apr 8 05:52:38 2019 -0400

    media: pvrusb2: Prevent a buffer overflow

    [ Upstream commit c1ced46c7b49ad7bc064e68d966e0ad303f917fb ]

    The ctrl_check_input() function is called from pvr2_ctrl_range_check().
    It's supposed to validate user supplied input and return true or false
    depending on whether the input is valid or not.  The problem is that
    negative shifts or shifts greater than 31 are undefined in C.  In
    practice with GCC they result in shift wrapping so this function returns
    true for some inputs which are not valid and this could result in a
    buffer overflow:

        drivers/media/usb/pvrusb2/pvrusb2-ctrl.c:205 pvr2_ctrl_get_valname()
        warn: uncapped user index 'names[val]'

    The cptr->hdw->input_allowed_mask mask is configured in pvr2_hdw_create()
    and the highest valid bit is BIT(4).

    Fixes: 7fb20fa38caa ("V4L/DVB (7299): pvrusb2: Improve logic which handles input choice
availability")

    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 02dce75ac8aa16d0ec5a2632286e97204c9f98f7
Author: Shuah Khan <shuah@kernel.org>
Date:   Mon Apr 1 20:43:17 2019 -0400

    media: au0828: Fix NULL pointer dereference in au0828_analog_stream_enable()

    [ Upstream commit 898bc40bfcc26abb6e06e960d6d4754c36c58b50 ]

    Fix au0828_analog_stream_enable() to check if device is in the right
    state first. When unbind happens while bind is in progress, usbdev
    pointer could be invalid in au0828_analog_stream_enable() and a call
```

```
    to usb_ifnum_to_if() will result in the null pointer dereference.

    This problem is found with the new media_dev_allocator.sh test.

    kernel: [  590.359623] BUG: unable to handle kernel NULL pointer dereference at
 00000000000004e8
    kernel: [  590.359627] #PF error: [normal kernel read fault]
    kernel: [  590.359629] PGD 0 P4D 0
    kernel: [  590.359632] Oops: 0000 [#1] SMP PTI
    kernel: [  590.359634] CPU: 3 PID: 1458 Comm: v4l_id Not tainted 5.1.0-rc2+ #30
    kernel: [  590.359636] Hardware name: Dell Inc. OptiPlex 7 90/0HY9JP, BIOS A18
 09/24/2013
    kernel: [  590.359641] RIP: 0010:usb_ifnum_to_if+0x6/0x60
    kernel: [  590.359643] Code: 5d 41 5e 41 5f 5d c3 48 83 c4
     10 b8 fa ff ff ff 5b 41 5c 41 5d 41 5e 41 5f 5d c3 b8 fa ff ff ff c3 0f 1f 00 6
    6 66 66 66 90 55 <48> 8b 97 e8 04 00 00 48 89 e5 48 85 d2 74 41 0f b6 4a 04 84 c
    9 74
    kernel: [  590.359645] RSP: 0018:ffffad3cc3c1fc00 EFLAGS: 00010246
    kernel: [  590.359646] RAX: 0000000000000000 RBX: ffff8ded b1f3c000 RCX:
 1f377e4500000000
    kernel: [  590.359648] RDX: ffff8dedfa3a6b50 RSI: 00000000 00000000 RDI:
 0000000000000000
    kernel: [  590.359649] RBP: ffffad3cc3c1fc28 R08: 00000000 8574acc2 R09:
 ffff8dedfa3a6b50
    kernel: [  590.359650] R10: 0000000000000001 R11: 00000000 00000000 R12:
 0000000000000000
    kernel: [  590.359652] R13: ffff8dedb1f3f0f0 R14: ffffffff adcf7ec0 R15:
 0000000000000000
    kernel: [  590.359654] FS:  00007f7917198540(0000) GS:ffff 8dee258c0000(0000)
 knlGS:0000000000000000
    kernel: [  590.359655] CS:  0010 DS: 0000 ES: 0000 CR0: 00 00000080050033
    kernel: [  590.359657] CR2: 00000000000004e8 CR3: 00000001 a388e002 CR4:
 00000000000606e0
    kernel: [  590.359658] Call Trace:
    kernel: [  590.359664]  ? au0828_analog_stream_enable+0x2c/0x180
    kernel: [  590.359666]  au0828_v4l2_open+0xa4/0x110
    kernel: [  590.359670]  v4l2_open+0x8b/0x120
    kernel: [  590.359674]  chrdev_open+0xa6/0x1c0
    kernel: [  590.359676]  ? cdev_put.part.3+0x20/0x20
    kernel: [  590.359678]  do_dentry_open+0x1f6/0x360
    kernel: [  590.359681]  vfs_open+0x2f/0x40
    kernel: [  590.359684]  path_openat+0x299/0xc20
    kernel: [  590.359688]  do_filp_open+0x9b/0x110
    kernel: [  590.359695]  ? _raw_spin_unlock+0x27/0x40
    kernel: [  590.359697]  ? __alloc_fd+0xb2/0x160
    kernel: [  590.359700]  do_sys_open+0x1ba/0x260
    kernel: [  590.359702]  ? do_sys_open+0x1ba/0x260
    kernel: [  590.359712]  __x64_sys_openat+0x20/0x30
    kernel: [  590.359715]  do_syscall_64+0x5a/0x120
    kernel: [  590.359718]  entry_SYSCALL_64_after_hwframe+0x44/0xa9

    Signed-off-by: Shuah Khan <shuah@kernel.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 74a8b114a22ed7045d9e901aeefd024a8dfcc0c8
 Author: Hugues Fruchet <hugues.fruchet@st.com>
 Date:   Mon Apr 1 04:56:09 2019 -0400

    media: stm32-dcmi: fix crash when subdev do not expose any formats

    [ Upstream commit 33dfeb62e23c31619d2197850f7e8b50e8cc5466 ]

    Do not access sd_formats[] if num_of_sd_formats is zero, ie
```

```
    subdev sensor didn't expose any formats.

    Signed-off-by: Hugues Fruchet <hugues.fruchet@st.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bfda64256006d38974be3ae396bf254e3f650d1f
Author: Wenwen Wang <wang6495@umn.edu>
Date:   Fri Apr 19 20:49:29 2019 -0500

    audit: fix a memory leak bug

    [ Upstream commit 70c4cf17e445264453bc5323db3e50aa0ac9e81f ]

    In audit_rule_change(), audit_data_to_entry() is firstly invoked to
    translate the payload data to the kernel's rule representation. In
    audit_data_to_entry(), depending on the audit field type, an audit tree may
    be created in audit_make_tree(), which eventually invokes kmalloc() to
    allocate the tree.  Since this tree is a temporary tree, it will be then
    freed in the following execution, e.g., audit_add_rule() if the message
    type is AUDIT_ADD_RULE or audit_del_rule() if the message type is
    AUDIT_DEL_RULE. However, if the message type is neither AUDIT_ADD_RULE nor
    AUDIT_DEL_RULE, i.e., the default case of the switch statement, this
    temporary tree is not freed.

    To fix this issue, only allocate the tree when the type is AUDIT_ADD_RULE
    or AUDIT_DEL_RULE.

    Signed-off-by: Wenwen Wang <wang6495@umn.edu>
    Reviewed-by: Richard Guy Briggs <rgb@redhat.com>
    Signed-off-by: Paul Moore <paul@paul-moore.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e76dde35199c9911e8e34ccea2c9ae93d8c5d53d
Author: Akinobu Mita <akinobu.mita@gmail.com>
Date:   Sat Mar 30 10:01:31 2019 -0400

    media: ov2659: make S_FMT succeed even if requested format doesn't match

    [ Upstream commit bccb89cf9cd07a0690d519696a00c00a973b3fe4 ]

    This driver returns an error if unsupported media bus pixel code is
    requested by VIDIOC_SUBDEV_S_FMT.

    But according to Documentation/media/uapi/v4l/vidioc-subdev-g-fmt.rst,

    Drivers must not return an error solely because the requested format
    doesn't match the device capabilities. They must instead modify the
    format to match what the hardware can provide.

    So select default format code and return success in that case.

    This is detected by v4l2-compliance.

    Cc: "Lad, Prabhakar" <prabhakar.csengg@gmail.com>
    Signed-off-by: Akinobu Mita <akinobu.mita@gmail.com>
    Acked-by: Lad, Prabhakar <prabhakar.csengg@gmail.com>
    Signed-off-by: Sakari Ailus <sakari.ailus@linux.intel.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 84e9e876d003068db327171c4bc23b65766aaacc
Author: Hans Verkuil <hverkuil@xs4all.nl>
Date:   Tue Apr 2 03:24:15 2019 -0400
```

```
    media: au0828: stop video streaming only when last user stops

    [ Upstream commit f604f0f5afb88045944567f604409951b5eb6af8 ]

    If the application was streaming from both videoX and vbiX, and streaming
    from videoX was stopped, then the vbi streaming also stopped.

    The cause being that stop_streaming for video stopped the subdevs as well,
    instead of only doing that if dev->streaming_users reached 0.

    au0828_stop_vbi_streaming was also wrong since it didn't stop the subdevs
    at all when dev->streaming_users reached 0.

    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Tested-by: Shuah Khan <shuah@kernel.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb9d3fab8cc920848baa7bbb1b4890ff58497d41
Author: Janusz Krzysztofik <jmkrzyszt@gmail.com>
Date:   Fri Mar 29 21:06:09 2019 -0400

    media: ov6650: Move v4l2_clk_get() to ov6650_video_probe() helper

    [ Upstream commit ccdd85d518d8b9320ace1d87271f0ba2175f21fa ]

    In preparation for adding asynchronous subdevice support to the driver,
    don't acquire v4l2_clk from the driver .probe() callback as that may
    fail if the clock is provided by a bridge driver which may be not yet
    initialized.  Move the v4l2_clk_get() to ov6650_video_probe() helper
    which is going to be converted to v4l2_subdev_internal_ops.registered()
    callback, executed only when the bridge driver is ready.

    Signed-off-by: Janusz Krzysztofik <jmkrzyszt@gmail.com>
    Signed-off-by: Sakari Ailus <sakari.ailus@linux.intel.com>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d209a6be666d7f22e7d41079fd5084d1fcabad5f
Author: Philipp Zabel <p.zabel@pengutronix.de>
Date:   Mon Apr 8 08:32:49 2019 -0400

    media: coda: clear error return value before picture run

    [ Upstream commit bbeefa7357a648afe70e7183914c87c3878d528d ]

    The error return value is not written by some firmware codecs, such as
    MPEG-2 decode on CodaHx4. Clear the error return value before starting
    the picture run to avoid misinterpreting unrelated values returned by
    sequence initialization as error return value.

    Signed-off-by: Philipp Zabel <p.zabel@pengutronix.de>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8ccd99e2337c99803b82c56bd202d544922174e
Author: Nicolas Ferre <nicolas.ferre@microchip.com>
Date:   Wed Apr 3 12:23:57 2019 +0200

    dmaengine: at_xdmac: remove BUG_ON macro in tasklet

    [ Upstream commit e2c114c06da2d9ffad5b16690abf008d6696f689 ]
```

```
    Even if this case shouldn't happen when controller is properly programmed,
    it's still better to avoid dumping a kernel Oops for this.
    As the sequence may happen only for debugging purposes, log the error and
    just finish the tasklet call.

    Signed-off-by: Nicolas Ferre <nicolas.ferre@microchip.com>
    Acked-by: Ludovic Desroches <ludovic.desroches@microchip.com>
    Signed-off-by: Vinod Koul <vkoul@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8cc056ecca22fb2d3be4492dc98fe97a04e1bd69
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Tue Apr 16 16:24:24 2019 +0100

    perf/arm-cci: Remove broken race mitigation

    [ Upstream commit 0d2e2a82d4de298d006bf8eddc86829e3c7da820 ]

    Uncore PMU drivers face an awkward cyclic dependency wherein:

     - They have to pick a valid online CPU to associate with before
       registering the PMU device, since it will get exposed to userspace
       immediately.
     - The PMU registration has to be be at least partly complete before
       hotplug events can be handled, since trying to migrate an
       uninitialised context would be bad.
     - The hotplug handler has to be ready as soon as a CPU is chosen, lest
       it go offline without the user-visible cpumask value getting updated.

    The arm-cci driver has tried to solve this by using get_cpu() to pick
    the current CPU and prevent it from disappearing while both
    registrations are performed, but that results in taking mutexes with
    preemption disabled, which makes certain configurations very unhappy:

    [ 1.983337] BUG: sleeping function called from invalid context at
kernel/locking/rtmutex.c:2004
    [ 1.983340] in_atomic(): 1, irqs_disabled(): 0, pid: 1, name: swapper/0
    [ 1.983342] Preemption disabled at:
    [ 1.983353] [<ffffff80089801f4>] cci_pmu_probe+0x1dc/0x488
    [ 1.983360] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 4.18.20-rt8-yocto-preempt-rt #1
    [ 1.983362] Hardware name: ZynqMP ZCU102 Rev1.0 (DT)
    [ 1.983364] Call trace:
    [ 1.983369] dump_backtrace+0x0/0x158
    [ 1.983372] show_stack+0x24/0x30
    [ 1.983378] dump_stack+0x80/0xa4
    [ 1.983383] ___might_sleep+0x138/0x160
    [ 1.983386] __might_sleep+0x58/0x90
    [ 1.983391] __rt_mutex_lock_state+0x30/0xc0
    [ 1.983395] _mutex_lock+0x24/0x30
    [ 1.983400] perf_pmu_register+0x2c/0x388
    [ 1.983404] cci_pmu_probe+0x2bc/0x488
    [ 1.983409] platform_drv_probe+0x58/0xa8

    It is not feasible to resolve all the possible races outside of the perf
    core itself, so address the immediate bug by following the example of
    nearly every other PMU driver and not even trying to do so. Registering
    the hotplug notifier first should minimise the window in which things
    can go wrong, so that's about as much as we can reasonably do here. This
    also revealed an additional race in assigning the global pointer too
    late relative to the hotplug notifier, which gets fixed in the process.

    Reported-by: Li, Meng <Meng.Li@windriver.com>
    Tested-by: Corentin Labbe <clabbe.montjoie@gmail.com>
    Reviewed-by: Suzuki K Poulose <suzuki.poulose@arm.com>
    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
```

```
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d7c7530cb5495a59c43b02809ccf1db97f266279
Author: Douglas Anderson <dianders@chromium.org>
Date:   Fri Apr 12 09:17:47 2019 -0700

    clk: rockchip: undo several noc and special clocks as critical on rk3288

    [ Upstream commit f4033db5b84ebe4b32c25ba2ed65ab20b628996a ]

    This is mostly a revert of commit 55bb6a633c33 ("clk: rockchip: mark
    noc and some special clk as critical on rk3288") except that we're
    keeping "pmu_hclk_otg0" as critical still.

    NOTE: turning these clocks off doesn't seem to do a whole lot in terms
    of power savings (checking the power on the logic rail).  It appears
    to save maybe 1-2mW.  ...but still it seems like we should turn the
    clocks off if they aren't needed.

    About "pmu_hclk_otg0" (the one clock from the original commit we're
    still keeping critical) from an email thread:

    > pmu ahb clock
    >
    > Function: Clock to pmu module when hibernation and/or ADP is
    > enabled. Must be greater than or equal to 30 MHz.
    >
    > If the SOC design does not support hibernation/ADP function, only have
    > hclk_otg, this clk can be switched according to the usage of otg.
    > If the SOC design support hibernation/ADP, has two clocks, hclk_otg and
    > pmu_hclk_otg0.
    > Hclk_otg belongs to the closed part of otg logic, which can be switched
    > according to the use of otg.
    >
    > pmu_hclk_otg0 belongs to the always on part.
    >
    > As for whether pmu_hclk_otg0 can be turned off when otg is not in use,
    > we have not tested. IC suggest make pmu_hclk_otg0 always on.

    For the rest of the clocks:

    atclk: No documentation about this clock other than that it goes to
    the CPU.  CPU functions fine without it on.  Maybe needed for JTAG?

    jtag: Presumably this clock is only needed if you're debugging with
    JTAG.  It doesn't seem like it makes sense to waste power for every
    rk3288 user.  In any case to do JTAG you'd need private patches to
    adjust the pinctrl the mux the JTAG out anyway.

    pclk_dbg, pclk_core_niu: On veyron Chromebooks we turn these two
    clocks on only during kernel panics in order to access some coresight
    registers.  Since nothing in the upstream kernel does this we should
    be able to leave them off safely.  Maybe also needed for JTAG?

    hsicphy12m_xin12m: There is no indication of why this clock would need
    to be turned on for boards that don't use HSIC.

    pclk_ddrupctl[0-1], pclk_publ0[0-1]: On veyron Chromebooks we turn
    these 4 clocks on only when doing DDR transitions and they are off
    otherwise.  I see no reason why they'd need to be on in the upstream
    kernel which doesn't support DDRFreq.

    Signed-off-by: Douglas Anderson <dianders@chromium.org>
    Reviewed-by: Elaine Zhang <zhangqing@rock-chips.com>
```

```
    Signed-off-by: Heiko Stuebner <heiko@sntech.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d7f620722baed4d752a54f5b81f92ed63d1ff003
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Fri Apr 12 14:02:22 2019 +0800

    pinctrl: samsung: fix leaked of_node references

    [ Upstream commit 44b9f86cd41db6c522effa5aec251d664a52fbc0 ]

    The call to of_find_compatible_node returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/pinctrl/samsung/pinctrl-exynos-arm.c:76:2-8: ERROR: missing of_node_put;
acquired a node pointer with refcount incremented on line 66, but without a corresponding
object release within this function.
    ./drivers/pinctrl/samsung/pinctrl-exynos-arm.c:82:1-7: ERROR: missing of_node_put;
acquired a node pointer with refcount incremented on line 66, but without a corresponding
object release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Cc: Tomasz Figa <tomasz.figa@gmail.com>
    Cc: Sylwester Nawrocki <s.nawrocki@samsung.com>
    Cc: Kukjin Kim <kgene@kernel.org>
    Cc: linux-samsung-soc@vger.kernel.org
    Cc: linux-gpio@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Reviewed-by: Krzysztof Kozlowski <krzk@kernel.org>
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e2ef3ec3d96d7856b3a4b8d189591d84848d2212
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Fri Apr 12 14:02:21 2019 +0800

    pinctrl: st: fix leaked of_node references

    [ Upstream commit 483d70d73beaecab55882fcd2a357af72674e24c ]

    The call to of_get_child_by_name returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/pinctrl/pinctrl-st.c:1188:3-9: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 1175, but without a corresponding object release
within this function.
    ./drivers/pinctrl/pinctrl-st.c:1188:3-9: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 1175, but without a corresponding object release
within this function.
    ./drivers/pinctrl/pinctrl-st.c:1199:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 1175, but without a corresponding object release
within this function.
    ./drivers/pinctrl/pinctrl-st.c:1199:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 1175, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Patrice Chotard <patrice.chotard@st.com>
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Cc: linux-gpio@vger.kernel.org
```

```
    Cc: linux-kernel@vger.kernel.org (open list)
    Reviewed-by: Patrice Chotard <patrice.chotard@st.com>
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a101de06e263ee97d9205e19a6268f31b86f074a
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Fri Apr 12 14:02:19 2019 +0800

    pinctrl: pistachio: fix leaked of_node references

    [ Upstream commit 44a4455ac2c6b0981eace683a2b6eccf47689022 ]

    The call to of_get_child_by_name returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/pinctrl/pinctrl-pistachio.c:1422:1-7: ERROR: missing of_node_put; acquired a
node pointer with refcount incremented on line 1360, but without a corresponding object
release within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Cc: linux-gpio@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fe340ccadb787545e00363cecce461350430ebb5
Author: Hans de Goede <hdegoede@redhat.com>
Date:   Sat Apr 20 13:22:10 2019 +0200

    HID: logitech-hidpp: use RAP instead of FAP to get the protocol version

    [ Upstream commit 096377525cdb8251e4656085efc988bdf733fb4c ]

    According to the logitech_hidpp_2.0_specification_draft_2012-06-04.pdf doc:
    https://lekensteyn.nl/files/logitech/logitech_hidpp_2.0_specification_draft_2012-06-
04.pdf

    We should use a register-access-protocol request using the short input /
    output report ids. This is necessary because 27MHz HID++ receivers have
    a max-packetsize on their HIP++ endpoint of 8, so they cannot support
    long reports. Using a feature-access-protocol request (which is always
    long or very-long) with these will cause a timeout error, followed by
    the hidpp driver treating the device as not being HID++ capable.

    This commit fixes this by switching to using a rap request to get the
    protocol version.

    Besides being tested with a (046d:c517) 27MHz receiver with various
    27MHz keyboards and mice, this has also been tested to not cause
    regressions on a non-unifying dual-HID++ nano receiver (046d:c534) with
    k270 and m185 HID++-2.0 devices connected and on a unifying/dj receiver
    (046d:c52b) with a HID++-2.0 Logitech Rechargeable Touchpad T650.

    Signed-off-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Benjamin Tissoires <benjamin.tissoires@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a07c802b2486e1a1982f1a1a3c037037e656fc14
Author: Sean Wang <sean.wang@mediatek.com>
Date:   Tue Mar 5 08:14:25 2019 +0800
```

```
    Bluetooth: mediatek: Fixed incorrect type in assignment

    [ Upstream commit cac63f9b163700fb70a609ad220697c61b797d6b ]

    Fixed warning: incorrect type in assignment reported by kbuild test robot.
    The detailed warning is shown as below.

    make ARCH=x86_64 allmodconfig
    make C=1 CF='-fdiagnostic-prefix -D__CHECK_ENDIAN__'

    All warnings (new ones prefixed by >>):

    btmtkuart.c:671:18: sparse:    warning: incorrect type in assignment
                                   (different base types)
    btmtkuart.c:671:18: sparse:    expected unsigned int [usertype] baudrate
    btmtkuart.c:671:18: sparse:    got restricted __le32 [usertype]

    sparse warnings: (new ones prefixed by >>)
    btmtkuart.c:671:18: sparse: warning: incorrect type in assignment
                                   (different base types)
    btmtkuart.c:671:18: sparse:    expected unsigned int [usertype] baudrate
    btmtkuart.c:671:18: sparse:    got restricted __le32 [usertype]

    vim +671 drivers/bluetooth/btmtkuart.c

       659
       660  static int btmtkuart_change_baudrate(struct hci_dev *hdev)
       661  {
       662          struct btmtkuart_dev *bdev = hci_get_drvdata(hdev);
       663          struct btmtk_hci_wmt_params wmt_params;
       664          u32 baudrate;
       665          u8 param;
       666          int err;
       667
       668          /* Indicate the device to enter the probe state the host is
       669           * ready to change a new baudrate.
       670           */
     > 671          baudrate = cpu_to_le32(bdev->desired_speed);
       672          wmt_params.op = MTK_WMT_HIF;

    Fixes: 22eaf6c9946a ("Bluetooth: mediatek: add support for MediaTek MT7663U and MT7668U
UART devices")
    Reported-by: kernel test robot <lkp@intel.com>
    Signed-off-by: Sean Wang <sean.wang@mediatek.com>
    Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0bc530b89431963d121a4df479e7c4d1872cc96e
Author: Ferry Toth <ftoth@exalondelft.nl>
Date:   Tue Apr 9 16:15:50 2019 +0200

    Bluetooth: btbcm: Add default address for BCM43341B

    [ Upstream commit 5035726128cd2e3813ee44deedb9898509edb232 ]

    The BCM43341B has the default MAC address 43:34:1B:00:1F:AC if none
    is given. This address was found when enabling Bluetooth on multiple
    Intel Edison modules. It also contains the sequence 43341B, the name
    the chip identifies itself as. Using the same BD_ADDR is problematic
    when having multiple Intel Edison modules in each others range.
    The default address also has the LAA (locally administered address)
    bit set which prevents a BNEP device from being created, needed for
    BT tethering.

    Add this to the list of black listed default MAC addresses and let
```

```
    the user configure a valid one using f.i.
    `btmgmt -i hci0 public-addr xx:xx:xx:xx:xx:xx`

    Suggested-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Ferry Toth <ftoth@exalondelft.nl>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c5da31c4fda8a3f804145be39ca32720b5046e39
Author: Balakrishna Godavarthi <bgodavar@codeaurora.org>
Date:   Mon Apr 1 15:19:08 2019 +0530

    Bluetooth: hci_qca: Give enough time to ROME controller to bootup.

    [ Upstream commit 7f09d5a6c33be66a5ca19bf9dd1c2d90c5dfcf0d ]

    This patch enables enough time to ROME controller to bootup
    after we bring the enable pin out of reset.

    Fixes: 05ba533c5c11 ("Bluetooth: hci_qca: Add serdev support").
    Signed-off-by: Balakrishna Godavarthi <bgodavar@codeaurora.org>
    Reviewed-by: Rocky Liao <rjliao@codeaurora.org>
    Tested-by: Rocky Liao <rjliao@codeaurora.org>
    Tested-by: Claire Chang <tientzu@chromium.org>
    Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0d4ca252063faea296e98ec5be04e2431d0c1a44
Author: Matthias Kaehlcke <mka@chromium.org>
Date:   Tue Apr 23 11:16:52 2019 -0700

    Bluetooth: hci_qca: Fix crash with non-serdev devices

    [ Upstream commit ecf2b768bd11e2ff09ecbe621b387d0d58e970cf ]

    qca_set_baudrate() calls serdev_device_wait_until_sent() assuming that
    the HCI is always associated with a serdev device. This isn't true for
    ROME controllers instantiated through ldisc, where the call causes a
    crash due to a NULL pointer dereferentiation. Only call the function
    when we have a serdev device. The timeout for ROME devices at the end
    of qca_set_baudrate() is long enough to be reasonably sure that the
    command was sent.

    Fixes: fa9ad876b8e0 ("Bluetooth: hci_qca: Add support for Qualcomm Bluetooth chip
wcn3990")
    Reported-by: Balakrishna Godavarthi <bgodavar@codeaurora.org>
    Reported-by: Rocky Liao <rjliao@codeaurora.org>
    Signed-off-by: Matthias Kaehlcke <mka@chromium.org>
    Reviewed-by: Rocky Liao <rjliao@codeaurora.org>
    Tested-by: Rocky Liao <rjliao@codeaurora.org>
    Reviewed-by: Balakrishna Godavarthi <bgodavar@codeaurora.org>
    Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 095eaec4e9e5ca1bbc02c93151bd383a28a383c8
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Wed Apr 24 09:19:25 2019 +0200

    mm/uaccess: Use 'unsigned long' to placate UBSAN warnings on older GCC versions

    [ Upstream commit 29da93fea3ea39ab9b12270cc6be1b70ef201c9e ]

    Randy reported objtool triggered on his (GCC-7.4) build:
```

```
      lib/strncpy_from_user.o: warning: objtool: strncpy_from_user()+0x315: call to
__ubsan_handle_add_overflow() with UACCESS enabled
      lib/strnlen_user.o: warning: objtool: strnlen_user()+0x337: call to
__ubsan_handle_sub_overflow() with UACCESS enabled

    This is due to UBSAN generating signed-overflow-UB warnings where it
    should not. Prior to GCC-8 UBSAN ignored -fwrapv (which the kernel
    uses through -fno-strict-overflow).

    Make the functions use 'unsigned long' throughout.

    Reported-by: Randy Dunlap <rdunlap@infradead.org>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Acked-by: Randy Dunlap <rdunlap@infradead.org> # build-tested
    Acked-by: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: luto@kernel.org
    Link: http://lkml.kernel.org/r/20190424072208.754094071@infradead.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9ab589a014be40001058e4d2fc5111e03680948a
Author: Jiri Kosina <jkosina@suse.cz>
Date:   Wed Apr 24 09:04:57 2019 +0200

    x86/mm: Remove in_nmi() warning from 64-bit implementation of vmalloc_fault()

    [ Upstream commit a65c88e16f32aa9ef2e8caa68ea5c29bd5eb0ff0 ]

    In-NMI warnings have been added to vmalloc_fault() via:

      ebc8827f75 ("x86: Barf when vmalloc and kmemcheck faults happen in NMI")

    back in the time when our NMI entry code could not cope with nested NMIs.

    These days, it's perfectly fine to take a fault in NMI context and we
    don't have to care about the fact that IRET from the fault handler might
    cause NMI nesting.

    This warning has already been removed from 32-bit implementation of
    vmalloc_fault() in:

      6863ea0cda8 ("x86/mm: Remove in_nmi() warning from vmalloc_fault()")

    but the 64-bit version was omitted.

    Remove the bogus warning also from 64-bit implementation of vmalloc_fault().

    Reported-by: Nicolai Stange <nstange@suse.de>
    Signed-off-by: Jiri Kosina <jkosina@suse.cz>
    Acked-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Dave Hansen <dave.hansen@linux.intel.com>
    Cc: Frederic Weisbecker <fweisbec@gmail.com>
    Cc: Joerg Roedel <jroedel@suse.de>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Fixes: 6863ea0cda8 ("x86/mm: Remove in_nmi() warning from vmalloc_fault()")
    Link: http://lkml.kernel.org/r/nycvar.YFH.7.76.1904240902280.9803@cbobk.fhfr.pm
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit c0b481eb9da90713a8cf488925798e43477d74a5
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Wed Apr 24 09:19:24 2019 +0200

    x86/uaccess: Dont leak the AC flag into __put_user() argument evaluation

    [ Upstream commit 6ae865615fc43d014da2fd1f1bba7e81ee622d1b ]

    The __put_user() macro evaluates it's @ptr argument inside the
    __uaccess_begin() / __uaccess_end() region. While this would normally
    not be expected to be an issue, an UBSAN bug (it ignored -fwrapv,
    fixed in GCC 8+) would transform the @ptr evaluation for:

      drivers/gpu/drm/i915/i915_gem_execbuffer.c: if (unlikely(__put_user(offset,
&urelocs[r-stack].presumed_offset))) {

    into a signed-overflow-UB check and trigger the objtool AC validation.

    Finish this commit:

      2a418cf3f5f1 ("x86/uaccess: Don't leak the AC flag into __put_user() value
evaluation")

    and explicitly evaluate all 3 arguments early.

    Reported-by: Randy Dunlap <rdunlap@infradead.org>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Acked-by: Randy Dunlap <rdunlap@infradead.org> # build-tested
    Acked-by: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: luto@kernel.org
    Fixes: 2a418cf3f5f1 ("x86/uaccess: Don't leak the AC flag into __put_user() value
evaluation")
    Link: http://lkml.kernel.org/r/20190424072208.695962771@infradead.org
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bd323b5fe30f1e0a3b391057d712f63ab8ca6334
Author: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
Date:   Wed Apr 24 10:52:53 2019 +0200

    smpboot: Place the __percpu annotation correctly

    [ Upstream commit d4645d30b50d1691c26ff0f8fa4e718b08f8d3bb ]

    The test robot reported a wrong assignment of a per-CPU variable which
    it detected by using sparse and sent a report. The assignment itself is
    correct. The annotation for sparse was wrong and hence the report.
    The first pointer is a "normal" pointer and points to the per-CPU memory
    area. That means that the __percpu annotation has to be moved.

    Move the __percpu annotation to pointer which points to the per-CPU
    area. This change affects only the sparse tool (and is ignored by the
    compiler).

    Reported-by: kbuild test robot <lkp@intel.com>
    Signed-off-by: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Paul E. McKenney <paulmck@linux.ibm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Fixes: f97f8f06a49fe ("smpboot: Provide infrastructure for percpu hotplug threads")
    Link: http://lkml.kernel.org/r/20190424085253.12178-1-bigeasy@linutronix.de
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
```

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

commit a43bf1035db6f04dd6432e6c7bc968c891399916
Author: Kees Cook <keescook@chromium.org>
Date:   Tue Apr 23 11:38:27 2019 -0700

    x86/build: Move _etext to actual end of .text

    [ Upstream commit 392bef709659abea614abfe53cf228e7a59876a4 ]

    When building x86 with Clang LTO and CFI, CFI jump regions are
    automatically added to the end of the .text section late in linking. As a
    result, the _etext position was being labelled before the appended jump
    regions, causing confusion about where the boundaries of the executable
    region actually are in the running kernel, and broke at least the fault
    injection code. This moves the _etext mark to outside (and immediately
    after) the .text area, as it already the case on other architectures
    (e.g. arm64, arm).

    Reported-and-tested-by: Sami Tolvanen <samitolvanen@google.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Cc: Borislav Petkov <bp@suse.de>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: http://lkml.kernel.org/r/20190423183827.GA4012@beast
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2014ac6aed377fb82e1ac0403a0adc2bc65eb21e
Author: Farhan Ali <alifm@linux.ibm.com>
Date:   Mon Apr 8 17:05:33 2019 -0400

    vfio-ccw: Release any channel program when releasing/removing vfio-ccw mdev

    [ Upstream commit b49bdc8602b7c9c7a977758bee4125683f73e59f ]

    When releasing the vfio-ccw mdev, we currently do not release
    any existing channel program and its pinned pages. This can
    lead to the following warning:

    [1038876.561565] WARNING: CPU: 2 PID: 144727 at drivers/vfio/vfio_iommu_type1.c:1494
vfio_sanity_check_pfn_list+0x40/0x70 [vfio_iommu_type1]

    ....

    1038876.561921] Call Trace:
    [1038876.561935] ([<00000009897fb870>] 0x9897fb870)
    [1038876.561949]  [<000003ff8013bf62>] vfio_iommu_type1_detach_group+0xda/0x2f0
[vfio_iommu_type1]
    [1038876.561965]  [<000003ff8007b634>] __vfio_group_unset_container+0x64/0x190 [vfio]
    [1038876.561978]  [<000003ff8007b87e>] vfio_group_put_external_user+0x26/0x38 [vfio]
    [1038876.562024]  [<000003ff806fc608>] kvm_vfio_group_put_external_user+0x40/0x60 [kvm]
    [1038876.562045]  [<000003ff806fcb9e>] kvm_vfio_destroy+0x5e/0xd0 [kvm]
    [1038876.562065]  [<000003ff806f63fc>] kvm_put_kvm+0x2a4/0x3d0 [kvm]
    [1038876.562083]  [<000003ff806f655e>] kvm_vm_release+0x36/0x48 [kvm]
    [1038876.562098]  [<00000000003c2dc4>] __fput+0x144/0x228
    [1038876.562113]  [<000000000016ee82>] task_work_run+0x8a/0xd8
    [1038876.562125]  [<000000000014c7a8>] do_exit+0x5d8/0xd90
    [1038876.562140]  [<000000000014d084>] do_group_exit+0xc4/0xc8
    [1038876.562155]  [<000000000015c046>] get_signal+0x9ae/0xa68
    [1038876.562169]  [<0000000000108d66>] do_signal+0x66/0x768
    [1038876.562185]  [<0000000000b9e37e>] system_call+0x1ea/0x2d8
    [1038876.562195] 2 locks held by qemu-system-s39/144727:
    [1038876.562205]  #0: 00000000537abaf9 (&container->group_lock){++++}, at:
```

```
  __vfio_group_unset_container+0x3c/0x190 [vfio]
      [1038876.562230]  #1: 00000000670008b5 (&iommu->lock){+.+.}, at:
  vfio_iommu_type1_detach_group+0x36/0x2f0 [vfio_iommu_type1]
      [1038876.562250] Last Breaking-Event-Address:
      [1038876.562262]  [<000003ff8013aa24>] vfio_sanity_check_pfn_list+0x3c/0x70
  [vfio_iommu_type1]
      [1038876.562272] irq event stamp: 4236481
      [1038876.562287] hardirqs last  enabled at (4236489): [<0000000001cee7a>]
  console_unlock+0x6d2/0x740
      [1038876.562299] hardirqs last disabled at (4236496): [<0000000001ce87e>]
  console_unlock+0xd6/0x740
      [1038876.562311] softirqs last  enabled at (4234162): [<0000000000b9fa1e>]
  __do_softirq+0x556/0x598
      [1038876.562325] softirqs last disabled at (4234153): [<000000000014e4cc>]
  irq_exit+0xac/0x108
      [1038876.562337] ---[ end trace 6c96d467b1c3ca06 ]---

      Similarly we do not free the channel program when we are removing
      the vfio-ccw device. Let's fix this by resetting the device and freeing
      the channel program and pinned pages in the release path. For the remove
      path we can just quiesce the device, since in the remove path the mediated
      device is going away for good and so we don't need to do a full reset.

      Signed-off-by: Farhan Ali <alifm@linux.ibm.com>
      Message-Id:
  <ae9f20dc8873f2027f7b3c5d2aaa0bdfe06850b8.1554756534.git.alifm@linux.ibm.com>
      Acked-by: Eric Farman <farman@linux.ibm.com>
      Signed-off-by: Cornelia Huck <cohuck@redhat.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 5824fa74965eb9d0413b8e6bc74523f680183e1b
  Author: Farhan Ali <alifm@linux.ibm.com>
  Date:   Mon Apr 8 17:05:31 2019 -0400

      vfio-ccw: Do not call flush_workqueue while holding the spinlock

      [ Upstream commit cea5dde42a83b5f0a039da672f8686455936b8d8 ]

      Currently we call flush_workqueue while holding the subchannel
      spinlock. But flush_workqueue function can go to sleep, so
      do not call the function while holding the spinlock.

      Fixes the following bug:

      [  285.203430] BUG: scheduling while atomic: bash/14193/0x00000002
      [  285.203434] INFO: lockdep is turned off.
      ....
      [  285.203485] Preemption disabled at:
      [  285.203488] [<000003ff80243e5c>] vfio_ccw_sch_quiesce+0xbc/0x120 [vfio_ccw]
      [  285.203496] CPU: 7 PID: 14193 Comm: bash Tainted: G        W
      ....
      [  285.203504] Call Trace:
      [  285.203510] ([<0000000000113772>] show_stack+0x82/0xd0)
      [  285.203514]  [<0000000000b7a102>] dump_stack+0x92/0xd0
      [  285.203518]  [<000000000017b8be>] __schedule_bug+0xde/0xf8
      [  285.203524]  [<0000000000b95b5a>] __schedule+0x7a/0xc38
      [  285.203528]  [<0000000000b9678a>] schedule+0x72/0xb0
      [  285.203533]  [<0000000000b9bfbc>] schedule_timeout+0x34/0x528
      [  285.203538]  [<0000000000b97608>] wait_for_common+0x118/0x1b0
      [  285.203544]  [<0000000000166d6a>] flush_workqueue+0x182/0x548
      [  285.203550]  [<000003ff80243e6e>] vfio_ccw_sch_quiesce+0xce/0x120 [vfio_ccw]
      [  285.203556]  [<000003ff80245278>] vfio_ccw_mdev_reset+0x38/0x70 [vfio_ccw]
      [  285.203562]  [<000003ff802458b0>] vfio_ccw_mdev_remove+0x40/0x78 [vfio_ccw]
      [  285.203567]  [<000003ff801a499c>] mdev_device_remove_ops+0x3c/0x80 [mdev]
      [  285.203573]  [<000003ff801a4d5c>] mdev_device_remove+0xc4/0x130 [mdev]
```

```
     [  285.203578]  [<000003ff801a5074>] remove_store+0x6c/0xa8 [mdev]
     [  285.203582]  [<000000000046f494>] kernfs_fop_write+0x14c/0x1f8
     [  285.203588]  [<00000000003c1530>] __vfs_write+0x38/0x1a8
     [  285.203593]  [<00000000003c187c>] vfs_write+0xb4/0x198
     [  285.203597]  [<00000000003c1af2>] ksys_write+0x5a/0xb0
     [  285.203601]  [<0000000000b9e270>] system_call+0xdc/0x2d8

     Signed-off-by: Farhan Ali <alifm@linux.ibm.com>
     Reviewed-by: Eric Farman <farman@linux.ibm.com>
     Reviewed-by: Pierre Morel <pmorel@linux.ibm.com>
     Message-Id:
 <626bab8bb2958ae132452e1ddaf1b20882ad5a9d.1554756534.git.alifm@linux.ibm.com>
     Signed-off-by: Cornelia Huck <cohuck@redhat.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 7412bdf50a1f6fb7a688ecd29d8d48a3a401622b
 Author: Parav Pandit <parav@mellanox.com>
 Date:   Wed Apr 10 11:23:04 2019 +0300

     RDMA/cma: Consider scope_id while binding to ipv6 ll address

     [ Upstream commit 5d7ed2f27bbd482fd29e6b2e204b1a1ee8a0b268 ]

     When two netdev have same link local addresses (such as vlan and non
     vlan), two rdma cm listen id should be able to bind to following different
     addresses.

     listener-1: addr=lla, scope_id=A, port=X
     listener-2: addr=lla, scope_id=B, port=X

     However while comparing the addresses only addr and port are considered,
     due to which 2nd listener fails to listen.

     In below example of two listeners, 2nd listener is failing with address in
     use error.

     $ rping -sv -a fe80::268a:7ff:feb3:d113%ens2f1 -p 4545&

     $ rping -sv -a fe80::268a:7ff:feb3:d113%ens2f1.200 -p 4545
     rdma_bind_addr: Address already in use

     To overcome this, consider the scope_ids as well which forms the accurate
     IPv6 link local address.

     Signed-off-by: Parav Pandit <parav@mellanox.com>
     Reviewed-by: Daniel Jurgens <danielj@mellanox.com>
     Signed-off-by: Leon Romanovsky <leonro@mellanox.com>
     Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 9b143f35bb90b1f21c3230f5a819addf056bff65
 Author: Arnd Bergmann <arnd@arndb.de>
 Date:   Thu Apr 25 00:48:28 2019 +0800

     bcache: avoid clang -Wunintialized warning

     [ Upstream commit 78d4eb8ad9e1d413449d1b7a060f50b6efa81ebd ]

     clang has identified a code path in which it thinks a
     variable may be unused:

     drivers/md/bcache/alloc.c:333:4: error: variable 'bucket' is used uninitialized whenever
 'if' condition is false
           [-Werror,-Wsometimes-uninitialized]
                         fifo_pop(&ca->free_inc, bucket);
```

```
                          ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    drivers/md/bcache/util.h:219:27: note: expanded from macro 'fifo_pop'
     #define fifo_pop(fifo, i)       fifo_pop_front(fifo, (i))
                                    ^~~~~~~~~~~~~~~~~~~~~~~~
    drivers/md/bcache/util.h:189:6: note: expanded from macro 'fifo_pop_front'
            if (_r) {                                                       \
                ^~
    drivers/md/bcache/alloc.c:343:46: note: uninitialized use occurs here
                          allocator_wait(ca, bch_allocator_push(ca, bucket));
                                                                  ^~~~~~
    drivers/md/bcache/alloc.c:287:7: note: expanded from macro 'allocator_wait'
                    if (cond)                                               \
                        ^~~~
    drivers/md/bcache/alloc.c:333:4: note: remove the 'if' if its condition is always true
                          fifo_pop(&ca->free_inc, bucket);
                          ^
    drivers/md/bcache/util.h:219:27: note: expanded from macro 'fifo_pop'
     #define fifo_pop(fifo, i)       fifo_pop_front(fifo, (i))
                                    ^
    drivers/md/bcache/util.h:189:2: note: expanded from macro 'fifo_pop_front'
            if (_r) {                                                       \
            ^
    drivers/md/bcache/alloc.c:331:15: note: initialize the variable 'bucket' to silence this
 warning
                          long bucket;
                                      ^


    This cannot happen in practice because we only enter the loop
    if there is at least one element in the list.

    Slightly rearranging the code makes this clearer to both the
    reader and the compiler, which avoids the warning.

    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Coly Li <colyli@suse.de>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit b24e16ebbc326f4eb4cfe95d4735e80e543f4421
 Author: Coly Li <colyli@suse.de>
 Date:   Thu Apr 25 00:48:34 2019 +0800

    bcache: add failure check to run_cache_set() for journal replay

    [ Upstream commit ce3e4cfb59cb382f8e5ce359238aa580d4ae7778 ]

    Currently run_cache_set() has no return value, if there is failure in
    bch_journal_replay(), the caller of run_cache_set() has no idea about
    such failure and just continue to execute following code after
    run_cache_set().  The internal failure is triggered inside
    bch_journal_replay() and being handled in async way. This behavior is
    inefficient, while failure handling inside bch_journal_replay(), cache
    register code is still running to start the cache set. Registering and
    unregistering code running as same time may introduce some rare race
    condition, and make the code to be more hard to be understood.

    This patch adds return value to run_cache_set(), and returns -EIO if
    bch_journal_rreplay() fails. Then caller of run_cache_set() may detect
    such failure and stop registering code flow immedidately inside
    register_cache_set().

    If journal replay fails, run_cache_set() can report error immediately
    to register_cache_set(). This patch makes the failure handling for
    bch_journal_replay() be in synchronized way, easier to understand and
```

```
    debug, and avoid poetential race condition for register-and-unregister
    in same time.

    Signed-off-by: Coly Li <colyli@suse.de>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b299330755656e19a272628f4393a568f7eb7c90
Author: Tang Junhui <tang.junhui.linux@gmail.com>
Date:   Thu Apr 25 00:48:41 2019 +0800

    bcache: fix failure in journal relplay

    [ Upstream commit 631207314d88e9091be02fbdd1fdadb1ae2ed79a ]

    journal replay failed with messages:
    Sep 10 19:10:43 ceph kernel: bcache: error on
    bb379a64-e44e-4812-b91d-a5599871a3b1: bcache: journal entries
    2057493-2057567 missing! (replaying 2057493-2076601), disabling
    caching

    The reason is in journal_reclaim(), when discard is enabled, we send
    discard command and reclaim those journal buckets whose seq is old
    than the last_seq_now, but before we write a journal with last_seq_now,
    the machine is restarted, so the journal with the last_seq_now is not
    written to the journal bucket, and the last_seq_wrote in the newest
    journal is old than last_seq_now which we expect to be, so when we doing
    replay, journals from last_seq_wrote to last_seq_now are missing.

    It's hard to write a journal immediately after journal_reclaim(),
    and it harmless if those missed journal are caused by discarding
    since those journals are already wrote to btree node. So, if miss
    seqs are started from the beginning journal, we treat it as normal,
    and only print a message to show the miss journal, and point out
    it maybe caused by discarding.

    Patch v2 add a judgement condition to ignore the missed journal
    only when discard enabled as Coly suggested.

    (Coly Li: rebase the patch with other changes in bch_journal_replay())

    Signed-off-by: Tang Junhui <tang.junhui.linux@gmail.com>
    Tested-by: Dennis Schridde <devurandom@gmx.net>
    Signed-off-by: Coly Li <colyli@suse.de>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e067f2f0a17973f7573fcbc645320c64b7a6b060
Author: Coly Li <colyli@suse.de>
Date:   Thu Apr 25 00:48:36 2019 +0800

    bcache: return error immediately in bch_journal_replay()

    [ Upstream commit 68d10e6979a3b59e3cd2e90bfcafed79c4cf180a ]

    When failure happens inside bch_journal_replay(), calling
    cache_set_err_on() and handling the failure in async way is not a good
    idea. Because after bch_journal_replay() returns, registering code will
    continue to execute following steps, and unregistering code triggered
    by cache_set_err_on() is running in same time. First it is unnecessary
    to handle failure and unregister cache set in an async way, second there
    might be potential race condition to run register and unregister code
    for same cache set.

    So in this patch, if failure happens in bch_journal_replay(), we don't
```

```
    call cache_set_err_on(), and just print out the same error message to
    kernel message buffer, then return -EIO immediately caller. Then caller
    can detect such failure and handle it in synchrnozied way.

    Signed-off-by: Coly Li <colyli@suse.de>
    Reviewed-by: Hannes Reinecke <hare@suse.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 11425640d7983e67c0c229e0974895f8e20cda68
Author: Shenghui Wang <shhuiw@foxmail.com>
Date:   Thu Apr 25 00:48:43 2019 +0800

    bcache: avoid potential memleak of list of journal_replay(s) in the CACHE_SYNC branch of
run_cache_set

    [ Upstream commit 95f18c9d1310730d075499a75aaf13bcd60405a7 ]

    In the CACHE_SYNC branch of run_cache_set(), LIST_HEAD(journal) is used
    to collect journal_replay(s) and filled by bch_journal_read().

    If all goes well, bch_journal_replay() will release the list of
    jounal_replay(s) at the end of the branch.

    If something goes wrong, code flow will jump to the label "err:" and leave
    the list unreleased.

    This patch will release the list of journal_replay(s) in the case of
    error detected.

    v1 -> v2:
    * Move the release code to the location after label 'err:' to
      simply the change.

    Signed-off-by: Shenghui Wang <shhuiw@foxmail.com>
    Signed-off-by: Coly Li <colyli@suse.de>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8d4819fdff5a636eab44315ee7ba3b2033b65d8c
Author: Corentin Labbe <clabbe.montjoie@gmail.com>
Date:   Thu Apr 18 10:17:34 2019 +0200

    crypto: sun4i-ss - Fix invalid calculation of hash end

    [ Upstream commit f87391558acf816b48f325a493d81d45dec40da0 ]

    When nbytes < 4, end is wronlgy set to a negative value which, due to
    uint, is then interpreted to a large value leading to a deadlock in the
    following code.

    This patch fix this problem.

    Fixes: 6298e948215f ("crypto: sunxi-ss - Add Allwinner Security System crypto
accelerator")
    Signed-off-by: Corentin Labbe <clabbe.montjoie@gmail.com>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f34237009e4de37a9e376ce90400c26e4ef7541f
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Wed Apr 24 11:53:17 2019 -0700

    nvme-tcp: fix a NULL deref when an admin connect times out
```

```
    [ Upstream commit 7a42589654ae79e1177f0d74306a02d6cef7bddf ]

    If we timeout the admin startup sequence we might not yet have
    an I/O tagset allocated which causes the teardown sequence to crash.
    Make nvme_tcp_teardown_io_queues safe by not iterating inflight tags
    if the tagset wasn't allocated.

    Fixes: 39d57757467b ("nvme-tcp: fix timeout handler")
    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b4e256d88eeeececb1dc353240d283fb867b397a
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Wed Apr 24 11:53:18 2019 -0700

    nvme-rdma: fix a NULL deref when an admin connect times out

    [ Upstream commit 1007709d7d06fab09bf2d007657575958676282b ]

    If we timeout the admin startup sequence we might not yet have
    an I/O tagset allocated which causes the teardown sequence to crash.
    Make nvme_tcp_teardown_io_queues safe by not iterating inflight tags
    if the tagset wasn't allocated.

    Fixes: 4c174e636674 ("nvme-rdma: fix timeout handler")
    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 12b83abc3006a9328217d451483c9e4fa99d77af
Author: Sagi Grimberg <sagi@grimberg.me>
Date:   Mon Mar 11 15:02:25 2019 -0700

    nvme: set 0 capacity if namespace block size exceeds PAGE_SIZE

    [ Upstream commit 01fa017484ad98fccdeaab32db0077c574b6bd6f ]

    If our target exposed a namespace with a block size that is greater
    than PAGE_SIZE, set 0 capacity on the namespace as we do not support it.

    This issue encountered when the nvmet namespace was backed by a tempfile.

    Signed-off-by: Sagi Grimberg <sagi@grimberg.me>
    Reviewed-by: Keith Busch <keith.busch@intel.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6fb42f3c379e10d2929787b6cff0ffa4c178febd
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Tue Mar 12 03:05:02 2019 -0500

    net: cw1200: fix a NULL pointer dereference

    [ Upstream commit 0ed2a005347400500a39ea7c7318f1fea57fb3ca ]

    In case create_singlethread_workqueue fails, the fix free the
    hardware and returns NULL to avoid NULL pointer dereference.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 90905286c19176b2806a0b277610eb17b8334263
Author: Aditya Pakki <pakki001@umn.edu>
```

```
Date:   Sat Mar 23 15:49:16 2019 -0500

    rsi: Fix NULL pointer dereference in kmalloc

    [ Upstream commit d5414c2355b20ea8201156d2e874265f1cb0d775 ]

    kmalloc can fail in rsi_register_rates_channels but memcpy still attempts
    to write to channels. The patch replaces these calls with kmemdup and
    passes the error upstream.

    Signed-off-by: Aditya Pakki <pakki001@umn.edu>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae745a8b592d44c31b9b657051863e15a61a6344
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Thu Apr 4 11:44:23 2019 +0300

    mwifiex: prevent an array overflow

    [ Upstream commit b4c35c17227fe437ded17ce683a6927845f8c4a4 ]

    The "rate_index" is only used as an index into the phist_data->rx_rate[]
    array in the mwifiex_hist_data_set() function.  That array has
    MWIFIEX_MAX_AC_RX_RATES (74) elements and it's used to generate some
    debugfs information.  The "rate_index" variable comes from the network
    skb->data[] and it is a u8 so it's in the 0-255 range.  We need to cap
    it to prevent an array overflow.

    Fixes: cbf6e05527a7 ("mwifiex: add rx histogram statistics support")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 50c312ad3ebd0c96e947b726dc06b9642a63aad1
Author: Xiaoli Feng <fengxiaoli0714@gmail.com>
Date:   Sat Mar 16 09:43:30 2019 +0800

    Fix nfs4.2 return -EINVAL when do dedupe operation

    [ Upstream commit ce96e888fe48ecfa868c9a39adc03292c78a80ff ]

    dedupe_file_range operations is combiled into remap_file_range.
    But in nfs42_remap_file_range, it's skiped for dedupe operations.
    Before this patch:
      # dd if=/dev/zero of=nfs/file bs=1M count=1
      # xfs_io -c "dedupe nfs/file 4k 64k 4k" nfs/file
      XFS_IOC_FILE_EXTENT_SAME: Invalid argument
    After this patch:
      # dd if=/dev/zero of=nfs/file bs=1M count=1
      # xfs_io -c "dedupe nfs/file 4k 64k 4k" nfs/file
      deduped 4096/4096 bytes at offset 65536
      4 KiB, 1 ops; 0.0046 sec (865.988 KiB/sec and 216.4971 ops/sec)

    Signed-off-by: Xiaoli Feng <fengxiaoli0714@gmail.com>
    Signed-off-by: Anna Schumaker <Anna.Schumaker@Netapp.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e2130db17d007cf7f216fcb3f9a2d08ffe897620
Author: Daniel Baluta <daniel.baluta@nxp.com>
Date:   Sun Apr 21 19:39:08 2019 +0000

    ASoC: fsl_sai: Update is_slave_mode with correct value

    [ Upstream commit ddb351145a967ee791a0fb0156852ec2fcb746ba ]
```

```
    is_slave_mode defaults to false because sai structure
    that contains it is kzalloc'ed.

    Anyhow, if we decide to set the following configuration
    SAI slave -> SAI master, is_slave_mode will remain set on true
    although SAI being master it should be set to false.

    Fix this by updating is_slave_mode for each call of
    fsl_sai_set_dai_fmt.

    Signed-off-by: Daniel Baluta <daniel.baluta@nxp.com>
    Acked-by: Nicolin Chen <nicoleotsuka@gmail.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a973c17c5cebecc2d1b3098a19ca0c5466b8ae4d
Author: Linus Walleij <linus.walleij@linaro.org>
Date:   Sat Apr 20 13:34:30 2019 +0200

    regulator: core: Actually put the gpiod after use

    [ Upstream commit 78927aa40bc82f32de07323ddc1c9de07ac68180 ]

    I went to great lengths to hand over the management of the GPIO
    descriptors to the regulator core, and some stray rebased
    oneliner in the old patch must have been assuming the devices
    were still doing devres management of it.

    We handed the management over to the regulator core, so of
    course the regulator core shall issue gpiod_put() when done.

    Sorry for the descriptor leak.

    Fixes: 541d052d7215 ("regulator: core: Only support passing enable GPIO descriptors")
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c3bbd47c07d454ed0b7f80f51c013543ffa3e2e3
Author: Kangjie Lu <kjlu@umn.edu>
Date:   Sat Apr 13 11:34:47 2019 +0100

    slimbus: fix a potential NULL pointer dereference in of_qcom_slim_ngd_register

    [ Upstream commit 06d5d6b7f9948a89543e1160ef852d57892c750d ]

    In case platform_device_alloc fails, the fix returns an error
    code to avoid the NULL pointer dereference.

    Signed-off-by: Kangjie Lu <kjlu@umn.edu>
    Signed-off-by: Srinivas Kandagatla <srinivas.kandagatla@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f1df511d34aec569b618e5da080dafb734e8b723
Author: Daniel T. Lee <danieltimlee@gmail.com>
Date:   Wed Apr 24 05:24:56 2019 +0900

    libbpf: fix samples/bpf build failure due to undefined UINT32_MAX

    [ Upstream commit 32e621e55496a0009f44fe4914cd4a23cade4984 ]

    Currently, building bpf samples will cause the following error.
```

```
         ./tools/lib/bpf/bpf.h:132:27: error: 'UINT32_MAX' undeclared here (not in a
 function) ..
          #define BPF_LOG_BUF_SIZE (UINT32_MAX >> 8) /* verifier maximum in kernels <= 5.1 */
                                    ^
         ./samples/bpf/bpf_load.h:31:25: note: in expansion of macro 'BPF_LOG_BUF_SIZE'
          extern char bpf_log_buf[BPF_LOG_BUF_SIZE];
                                  ^~~~~~~~~~~~~~~~
```

    Due to commit 4519efa6f8ea ("libbpf: fix BPF_LOG_BUF_SIZE off-by-one error")
    hard-coded size of BPF_LOG_BUF_SIZE has been replaced with UINT32_MAX which is
    defined in <stdint.h> header.

    Even with this change, bpf selftests are running fine since these are built
    with clang and it includes header(-idirafter) from clang/6.0.0/include.
    (it has <stdint.h>)

```
         clang -I. -I./include/uapi -I../../../include/uapi -idirafter /usr/local/include -
 idirafter /usr/include \
         -idirafter /usr/lib/llvm-6.0/lib/clang/6.0.0/include -idirafter /usr/include/x86_64-
 linux-gnu \
         -Wno-compare-distinct-pointer-types -O2 -target bpf -emit-llvm -c
 progs/test_sysctl_prog.c -o - | \
         llc -march=bpf -mcpu=generic  -filetype=obj -o
 /linux/tools/testing/selftests/bpf/test_sysctl_prog.o
```

    But bpf samples are compiled with GCC, and it only searches and includes
    headers declared at the target file. As '#include <stdint.h>' hasn't been
    declared in tools/lib/bpf/bpf.h, it causes build failure of bpf samples.

```
         gcc -Wp,-MD,./samples/bpf/.sockex3_user.o.d -Wall -Wmissing-prototypes -Wstrict-
 prototypes \
         -O2 -fomit-frame-pointer -std=gnu89 -I./usr/include -I./tools/lib/ -
 I./tools/testing/selftests/bpf/ \
         -I./tools/  lib/ -I./tools/include -I./tools/perf -c -o ./samples/bpf/sockex3_user.o
 ./samples/bpf/sockex3_user.c;
```

    This commit add declaration of '#include <stdint.h>' to tools/lib/bpf/bpf.h
    to fix this problem.

    Signed-off-by: Daniel T. Lee <danieltimlee@gmail.com>
    Acked-by: Yonghong Song <yhs@fb.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

```
commit 59c8fa1f382d63fcf088e2978dcf6f2f74d71764
Author: Masahiro Yamada <yamada.masahiro@socionext.com>
Date:   Fri Mar 29 20:32:41 2019 +0900
```

    drm: prefix header search paths with $(srctree)/

    [Upstream commit 43068cb7ba1f6ceb1523e947c84002b2a61fd6d4]

    Currently, the Kbuild core manipulates header search paths in a crazy
    way [1].

    To fix this mess, I want all Makefiles to add explicit $(srctree)/ to
    the search paths in the srctree. Some Makefiles are already written in
    that way, but not all. The goal of this work is to make the notation
    consistent, and finally get rid of the gross hacks.

    Having whitespaces after -I does not matter since commit 48f6e3cf5bc6
    ("kbuild: do not drop -I without parameter").

    [1]: https://patchwork.kernel.org/patch/9632347/

```
    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Reviewed-by: Sam Ravnborg <sam@ravnborg.org>
    Reviewed-by: James Qian Wang (Arm Technology China) <james.qian.wang@arm.com>
    Acked-by: Liviu Dudau <liviu.dudau@arm.com>
    Signed-off-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Link: https://patchwork.freedesktop.org/patch/msgid/1553859161-2628-1-git-send-email-
yamada.masahiro@socionext.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4659bb96d64b97b210c0274096ee848389e4773d
Author: Sergey Matyukevich <sergey.matyukevich.os@quantenna.com>
Date:   Tue Mar 26 09:27:37 2019 +0000

    mac80211/cfg80211: update bss channel on channel switch

    [ Upstream commit 5dc8cdce1d722c733f8c7af14c5fb595cfedbfa8 ]

    FullMAC STAs have no way to update bss channel after CSA channel switch
    completion. As a result, user-space tools may provide inconsistent
    channel info. For instance, consider the following two commands:
    $ sudo iw dev wlan0 link
    $ sudo iw dev wlan0 info
    The latter command gets channel info from the hardware, so most probably
    its output will be correct. However the former command gets channel info
    from scan cache, so its output will contain outdated channel info.
    In fact, current bss channel info will not be updated until the
    next [re-]connect.

    Note that mac80211 STAs have a workaround for this, but it requires
    access to internal cfg80211 data, see ieee80211_chswitch_work:

            /* XXX: shouldn't really modify cfg80211-owned data! */
            ifmgd->associated->channel = sdata->csa_chandef.chan;

    This patch suggests to convert mac80211 workaround into cfg80211 behavior
    and to update current bss channel in cfg80211_ch_switch_notify.

    Signed-off-by: Sergey Matyukevich <sergey.matyukevich.os@quantenna.com>
    Signed-off-by: Johannes Berg <johannes.berg@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1606cbce895d02607d3e2405b88c0978998ac87e
Author: Sugar Zhang <sugar.zhang@rock-chips.com>
Date:   Wed Apr 3 19:06:22 2019 +0800

    dmaengine: pl330: _stop: clear interrupt status

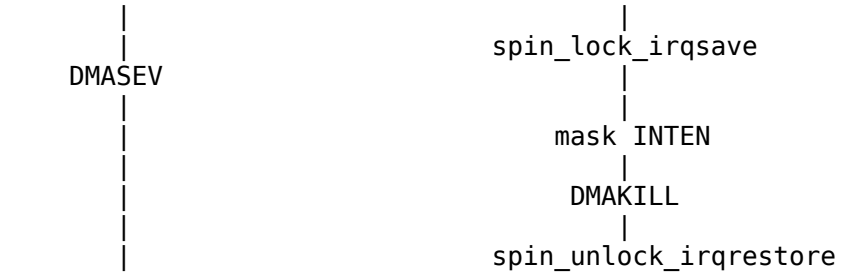
    [ Upstream commit 2da254cc7908105a60a6bb219d18e8dced03dcb9 ]

    This patch kill instructs the DMAC to immediately terminate
    execution of a thread. and then clear the interrupt status,
    at last, stop generating interrupts for DMA_SEV. to guarantee
    the next dma start is clean. otherwise, one interrupt maybe leave
    to next start and make some mistake.

    we can reporduce the problem as follows:

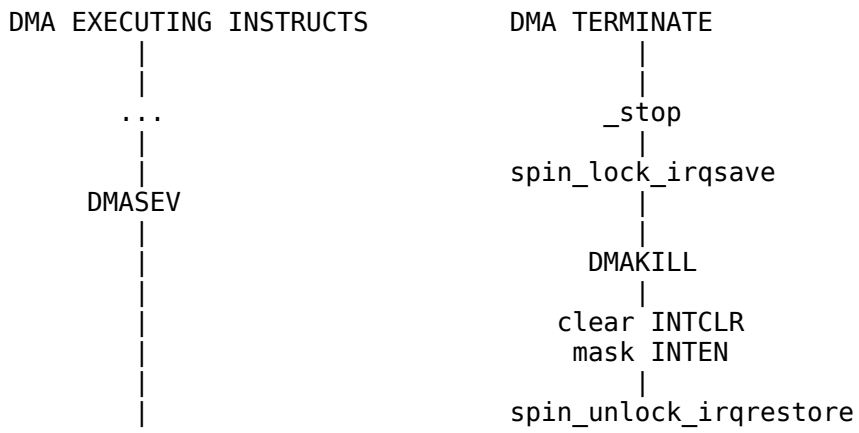
    DMASEV: modify the event-interrupt resource, and if the INTEN sets
    function as interrupt, the DMAC will set irq<event_num> HIGH to
    generate interrupt. write INTCLR to clear interrupt.

            DMA EXECUTING INSTRUCTS         DMA TERMINATE
                    |                               |
                    |                               |
                   ...                            _stop
```



in above case, a interrupt was left, and if we unmask INTEN, the DMAC
will set irq<event_num> HIGH to generate interrupt.

to fix this, do as follows:



Signed-off-by: Sugar Zhang <sugar.zhang@rock-chips.com>
Signed-off-by: Vinod Koul <vkoul@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4918adcf5b6306c598c64a19ff11af17058c619a
Author: Huazhong Tan <tanhuazhong@huawei.com>
Date: Thu Apr 25 20:42:49 2019 +0800

net: hns3: use atomic_t replace u32 for arq's count

[Upstream commit 30780a8b1677e7409b32ae52a9a84f7d41ae6b43]

Since irq handler and mailbox task will both update arq's count,
so arq's count should use atomic_t instead of u32, otherwise
its value may go wrong finally.

Fixes: 07a0556a3a73 ("net: hns3: Changes to support ARQ(Asynchronous Receive Queue)")
Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
Signed-off-by: Peng Li <lipeng321@huawei.com>
Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1c05aa606c47c0d05dc494a7d57493cd2ecb9716
Author: Will Deacon <will.deacon@arm.com>
Date: Mon Apr 8 12:45:09 2019 +0100

arm64: futex: Fix FUTEX_WAKE_OP atomic ops with non-zero result value

[Upstream commit 84ff7a09c371bc7417eabfda19bf7f113ec917b6]

Rather embarrassingly, our futex() FUTEX_WAKE_OP implementation doesn't
explicitly set the return value on the non-faulting path and instead
leaves it holding the result of the underlying atomic operation. This
means that any FUTEX_WAKE_OP atomic operation which computes a non-zero
value will be reported as having failed. Regrettably, I wrote the buggy

```
    code back in 2011 and it was upstreamed as part of the initial arm64
    support in 2012.

    The reasons we appear to get away with this are:

      1. FUTEX_WAKE_OP is rarely used and therefore doesn't appear to get
         exercised by futex() test applications

      2. If the result of the atomic operation is zero, the system call
         behaves correctly

      3. Prior to version 2.25, the only operation used by GLIBC set the
         futex to zero, and therefore worked as expected. From 2.25 onwards,
         FUTEX_WAKE_OP is not used by GLIBC at all.

    Fix the implementation by ensuring that the return value is either 0
    to indicate that the atomic operation completed successfully, or -EFAULT
    if we encountered a fault when accessing the user mapping.

    Cc: <stable@kernel.org>
    Fixes: 6170a97460db ("arm64: Atomic operations")
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8f1f40f7bb485507607ff83177afc890e56e4ecc
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Thu Apr 25 18:25:54 2019 +0200

    s390: qeth: address type mismatch warning

    [ Upstream commit 46b83629dede262315aa82179d105581f11763b6 ]

    clang produces a harmless warning for each use for the qeth_adp_supported
    macro:

    drivers/s390/net/qeth_l2_main.c:559:31: warning: implicit conversion from enumeration
type 'enum qeth_ipa_setadp_cmd' to
          different enumeration type 'enum qeth_ipa_funcs' [-Wenum-conversion]
            if (qeth_adp_supported(card, IPA_SETADP_SET_PROMISC_MODE))
                ~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~
    drivers/s390/net/qeth_core.h:179:41: note: expanded from macro 'qeth_adp_supported'
            qeth_is_ipa_supported(&c->options.adp, f)
            ~~~~~~~~~~~~~~~~~~~~                   ^

    Add a version of this macro that uses the correct types, and
    remove the unused qeth_adp_enabled() macro that has the same
    problem.

    Reviewed-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Julian Wiedmann <jwi@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0dd7e77b7fabc2b6c63af51a74da28e8420263f4
Author: Heiner Kallweit <hkallweit1@gmail.com>
Date:   Wed Apr 24 21:41:06 2019 +0200

    net: phy: improve genphy_soft_reset

    [ Upstream commit 8c90b795e90f7753d23c18e8b95dd71b4a18c5d9 ]

    PHY's behave differently when being reset. Some reset registers to
    defaults, some don't. Some trigger an autoneg restart, some don't.
```

```
    So let's also set the autoneg restart bit when resetting. Then PHY
    behavior should be more consistent. Clearing BMCR_ISOLATE serves the
    same purpose and is borrowed from genphy_restart_aneg.

    BMCR holds the speed / duplex settings in fixed mode. Therefore
    we may have an issue if a soft reset resets BMCR to its default.
    So better call genphy_setup_forced() afterwards in fixed mode.
    We've seen no related complaint in the last >10 yrs, so let's
    treat it as an improvement.

    Signed-off-by: Heiner Kallweit <hkallweit1@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4c1c9dabb6e9f91a8bec334e1110adc3a403cd11
Author: Yunsheng Lin <linyunsheng@huawei.com>
Date:   Thu Apr 25 20:42:46 2019 +0800

    net: hns3: fix for TX clean num when cleaning TX BD

    [ Upstream commit 63380a1ae4ced8aef67659ff9547c69ef8b9613a ]

    hns3_desc_unused() returns how many BD have been cleaned, but new
    buffer has not been attached to them. The register of
    HNS3_RING_RX_RING_FBDNUM_REG returns how many BD need allocating new
    buffer to or need to cleaned. So the remaining BD need to be clean
    is HNS3_RING_RX_RING_FBDNUM_REG - hns3_desc_unused().

    Also, new buffer can not attach to the pending BD when the last BD is
    not handled, because memcpy has not been done on the first pending BD.

    This patch fixes by subtracting the pending BD num from unused_count
    after 'HNS3_RING_RX_RING_FBDNUM_REG - unused_count' is used to calculate
    the BD bum need to be clean.

    Fixes: e55970950556 ("net: hns3: Add handling of GRO Pkts not fully RX'ed in NAPI poll")
    Signed-off-by: Yunsheng Lin <linyunsheng@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f5fc42d952c6ef294c967c51e20b207b479ad23d
Author: Huazhong Tan <tanhuazhong@huawei.com>
Date:   Thu Apr 25 20:42:52 2019 +0800

    net: hns3: fix pause configure fail problem

    [ Upstream commit fba2efdae8b4f998f66a2ff4c9f0575e1c4bbc40 ]

    When configure pause, current implementation returns directly
    after setup PFC without setup BP, which is not sufficient.

    So this patch fixes it, only return while setting PFC failed.

    Fixes: 44e59e375bf7 ("net: hns3: do not return GE PFC setting err when initializing")
    Signed-off-by: Huazhong Tan <tanhuazhong@huawei.com>
    Signed-off-by: Peng Li <lipeng321@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae9505c0959113e153e12fd451e93772eca3afa8
Author: Mariusz Bialonczyk <manio@skyboo.net>
Date:   Thu Mar 21 11:52:55 2019 +0100
```

```
    w1: fix the resume command API

    [ Upstream commit 62909da8aca048ecf9fbd7e484e5100608f40a63 ]

    >From the DS2408 datasheet [1]:
    "Resume Command function checks the status of the RC flag and, if it is set,
     directly transfers control to the control functions, similar to a Skip ROM
     command. The only way to set the RC flag is through successfully executing
     the Match ROM, Search ROM, Conditional Search ROM, or Overdrive-Match ROM
     command"

    The function currently works perfectly fine in a multidrop bus, but when we
    have only a single slave connected, then only a Skip ROM is used and Match
    ROM is not called at all. This is leading to problems e.g. with single one
    DS2408 connected, as the Resume Command is not working properly and the
    device is responding with failing results after the Resume Command.

    This commit is fixing this by using a Skip ROM instead in those cases.
    The bandwidth / performance advantage is exactly the same.

    Refs:
    [1] https://datasheets.maximintegrated.com/en/ds/DS2408.pdf

    Signed-off-by: Mariusz Bialonczyk <manio@skyboo.net>
    Reviewed-by: Jean-Francois Dagenais <jeff.dagenais@gmail.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit edb515810d5bfd70ffa691a12a3bc214bb958f2f
Author: Grygorii Strashko <grygorii.strashko@ti.com>
Date:   Fri Apr 26 20:12:33 2019 +0300

    net: ethernet: ti: cpsw: fix allmulti cfg in dual_mac mode

    [ Upstream commit 06095f34f8a0a2c4c83a19514c272699edd5f80b ]

    Now CPSW ALE will set/clean Host port bit in Unregistered Multicast Flood
    Mask (UNREG_MCAST_FLOOD_MASK) for every VLAN without checking if this port
    belongs to VLAN or not when ALLMULTI mode flag is set for nedev. This is
    working in non dual_mac mode, but in dual_mac - it causes
    enabling/disabling ALLMULTI flag for both ports.

    Hence fix it by adding additional parameter to cpsw_ale_set_allmulti() to
    specify ALE port number for which ALLMULTI has to be enabled and check if
    port belongs to VLAN before modifying UNREG_MCAST_FLOOD_MASK.

    Signed-off-by: Grygorii Strashko <grygorii.strashko@ti.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5cd3fddb97b20fb4471b53095d87ad6db086078f
Author: Nicholas Piggin <npiggin@gmail.com>
Date:   Fri Apr 12 14:26:13 2019 +1000

    sched/nohz: Run NOHZ idle load balancer on HK_FLAG_MISC CPUs

    [ Upstream commit 9b019acb72e4b5741d88e8936d6f200ed44b66b2 ]

    The NOHZ idle balancer runs on the lowest idle CPU. This can
    interfere with isolated CPUs, so confine it to HK_FLAG_MISC
    housekeeping CPUs.

    HK_FLAG_SCHED is not used for this because it is not set anywhere
    at the moment. This could be folded into HK_FLAG_SCHED once that
    option is fixed.
```

```
    The problem was observed with increased jitter on an application
    running on CPU0, caused by NOHZ idle load balancing being run on
    CPU1 (an SMT sibling).

    Signed-off-by: Nicholas Piggin <npiggin@gmail.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Frederic Weisbecker <fweisbec@gmail.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lkml.kernel.org/r/20190412042613.28930-1-npiggin@gmail.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c13f2ccff5572b82c64aa2fc77781da9ea219cff
Author: Bard liao <yung-chuan.liao@linux.intel.com>
Date:   Sun Apr 28 04:53:39 2019 +0800

    ALSA: hda: fix unregister device twice on ASoC driver

    [ Upstream commit 4d95c51776b2edb4d4ebcea00b6e5a1fe538ce66 ]

    snd_hda_codec_device_new() is used by both legacy HDA and ASoC
    driver. However, we will call snd_hdac_device_unregister() in
    snd_hdac_ext_bus_device_remove() for ASoC device. This patch uses
    the type flag in hdac_device struct to determine is it a ASoC device
    or legacy HDA device and call snd_hdac_device_unregister() in
    snd_hda_codec_dev_free() only if it is a legacy HDA device.

    Signed-off-by: Bard liao <yung-chuan.liao@linux.intel.com>
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea9e874b707bc511a41506cdf6ea6864b6e46338
Author: Philipp Rudo <prudo@linux.ibm.com>
Date:   Mon Apr 1 12:48:43 2019 +0200

    s390/kexec_file: Fix detection of text segment in ELF loader

    [ Upstream commit 729829d775c9a5217abc784b2f16087d79c4eec8 ]

    To register data for the next kernel (command line, oldmem_base, etc.) the
    current kernel needs to find the ELF segment that contains head.S. This is
    currently done by checking ifor 'phdr->p_paddr == 0'. This works fine for
    the current kernel build but in theory the first few pages could be
    skipped. Make the detection more robust by checking if the entry point lies
    within the segment.

    Signed-off-by: Philipp Rudo <prudo@linux.ibm.com>
    Signed-off-by: Martin Schwidefsky <schwidefsky@de.ibm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1df245c3020a19e7808bf7dfa6158fed6905c666
Author: Manish Rangankar <mrangankar@marvell.com>
Date:   Fri Apr 26 03:55:45 2019 -0700

    scsi: qedi: Abort ep termination if offload not scheduled

    [ Upstream commit f848bfd8e167210a29374e8a678892bed591684f ]

    Sometimes during connection recovery when there is a failure to resolve
    ARP, and offload connection was not issued, driver tries to flush pending
    offload connection work which was not queued up.
```

```
     kernel: WARNING: CPU: 19 PID: 10110 at kernel/workqueue.c:3030
  __flush_work.isra.34+0x19c/0x1b0
     kernel: CPU: 19 PID: 10110 Comm: iscsid Tainted: G W 5.1.0-rc4 #11
     kernel: Hardware name: Dell Inc. PowerEdge R730/0599V5, BIOS 2.9.1 12/04/2018
     kernel: RIP: 0010:__flush_work.isra.34+0x19c/0x1b0
     kernel: Code: 8b fb 66 0f 1f 44 00 00 31 c0 eb ab 48 89 ef c6 07 00 0f 1f 40 00 fb 66 0f
  1f 44 00 00 31 c0 eb 96 e8 08 16 fe ff 0f 0b eb 8d <0f> 0b 31 c0 eb 87 0f 1f 40 00 66 2e 0f
  1
     f 84 00 00 00 00 0f 1f
     kernel: RSP: 0018:ffffa6b4054dba68 EFLAGS: 00010246
     kernel: RAX: 0000000000000000 RBX: ffff91df21c36fc0 RCX: 0000000000000000
     kernel: RDX: 0000000000000001 RSI: 0000000000000000 RDI: ffff91df21c36fc0
     kernel: RBP: ffff91df21c36ef0 R08: 0000000000000000 R09: 0000000000000000
     kernel: R10: 0000000000000038 R11: ffffa6b4054dbd60 R12: ffffffffc05e72c0
     kernel: R13: ffff91db10280820 R14: 0000000000000048 R15: 0000000000000000
     kernel: FS:  00007f5d83cc1740(0000) GS:ffff91df2f840000(0000) knlGS:0000000000000000
     kernel: CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     kernel: CR2: 0000000001cc5000 CR3: 0000000465450002 CR4: 00000000001606e0
     kernel: Call Trace:
     kernel: ? try_to_del_timer_sync+0x4d/0x80
     kernel: qedi_ep_disconnect+0x3b/0x410 [qedi]
     kernel: ? 0xffffffffc083c000
     kernel: ? klist_iter_exit+0x14/0x20
     kernel: ? class_find_device+0x93/0xf0
     kernel: iscsi_if_ep_disconnect.isra.18+0x58/0x70 [scsi_transport_iscsi]
     kernel: iscsi_if_recv_msg+0x10e2/0x1510 [scsi_transport_iscsi]
     kernel: ? copyout+0x22/0x30
     kernel: ? _copy_to_iter+0xa0/0x430
     kernel: ? _cond_resched+0x15/0x30
     kernel: ? __kmalloc_node_track_caller+0x1f9/0x270
     kernel: iscsi_if_rx+0xa5/0x1e0 [scsi_transport_iscsi]
     kernel: netlink_unicast+0x17f/0x230
     kernel: netlink_sendmsg+0x2d2/0x3d0
     kernel: sock_sendmsg+0x36/0x50
     kernel: ___sys_sendmsg+0x280/0x2a0
     kernel: ? timerqueue_add+0x54/0x80
     kernel: ? enqueue_hrtimer+0x38/0x90
     kernel: ? hrtimer_start_range_ns+0x19f/0x2c0
     kernel: __sys_sendmsg+0x58/0xa0
     kernel: do_syscall_64+0x5b/0x180
     kernel: entry_SYSCALL_64_after_hwframe+0x44/0xa9

     Signed-off-by: Manish Rangankar <mrangankar@marvell.com>
     Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit a4e1b27e3a15e9c1f4cf3cc9b200faa66eb4534d
  Author: Fabien Dessenne <fabien.dessenne@st.com>
  Date:   Wed Apr 24 14:26:48 2019 +0200

     rtc: stm32: manage the get_irq probe defer case

     [ Upstream commit cf612c5949aca2bd81a1e28688957c8149ea2693 ]

     Manage the -EPROBE_DEFER error case for the wake IRQ.

     Signed-off-by: Fabien Dessenne <fabien.dessenne@st.com>
     Acked-by: Amelie Delaunay <amelie.delaunay@st.com>
     Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 88053c93e6c6ad3d3a57fbaaef0862e79edbb062
  Author: Sven Van Asbroeck <thesven73@gmail.com>
  Date:   Fri Apr 26 14:36:35 2019 -0400
```

```
      rtc: 88pm860x: prevent use-after-free on device remove

      [ Upstream commit f22b1ba15ee5785aa028384ebf77dd39e8e47b70 ]

      The device's remove() attempts to shut down the delayed_work scheduled
      on the kernel-global workqueue by calling flush_scheduled_work().

      Unfortunately, flush_scheduled_work() does not prevent the delayed_work
      from re-scheduling itself. The delayed_work might run after the device
      has been removed, and touch the already de-allocated info structure.
      This is a potential use-after-free.

      Fix by calling cancel_delayed_work_sync() during remove(): this ensures
      that the delayed work is properly cancelled, is no longer running, and
      is not able to re-schedule itself.

      This issue was detected with the help of Coccinelle.

      Signed-off-by: Sven Van Asbroeck <TheSven73@gmail.com>
      Signed-off-by: Alexandre Belloni <alexandre.belloni@bootlin.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 902e3a83ccfd4aaf0f5d38293eef3d6370c31e3e
  Author: Johannes Berg <johannes.berg@intel.com>
  Date:   Tue Mar 5 10:31:11 2019 +0100

      iwlwifi: pcie: don't crash on invalid RX interrupt

      [ Upstream commit 30f24eabab8cd801064c5c37589d803cb4341929 ]

      If for some reason the device gives us an RX interrupt before we're
      ready for it, perhaps during device power-on with misconfigured IRQ
      causes mapping or so, we can crash trying to access the queues.

      Prevent that by checking that we actually have RXQs and that they
      were properly allocated.

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

  commit 40b111b4170c3ebfeb8e078848905960f3a27aa5
  Author: Qu Wenruo <wqu@suse.com>
  Date:   Tue Feb 26 16:33:56 2019 +0800

      btrfs: Don't panic when we can't find a root key

      [ Upstream commit 7ac1e464c4d473b517bb784f30d40da1f842482e ]

      When we failed to find a root key in btrfs_update_root(), we just panic.

      That's definitely not cool, fix it by outputting an unique error
      message, aborting current transaction and return -EUCLEAN. This should
      not normally happen as the root has been used by the callers in some
      way.

      Reviewed-by: Filipe Manana <fdmanana@suse.com>
      Reviewed-by: Johannes Thumshirn <jthumshirn@suse.de>
      Signed-off-by: Qu Wenruo <wqu@suse.com>
      Reviewed-by: David Sterba <dsterba@suse.com>
      Signed-off-by: David Sterba <dsterba@suse.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 2d39b32c2f1448a423c5f7cd724bb3c7750dcea0
  Author: Josef Bacik <josef@toxicpanda.com>
```

```
Date:   Mon Feb 25 11:14:45 2019 -0500

    btrfs: fix panic during relocation after ENOSPC before writeback happens

    [ Upstream commit ff612ba7849964b1898fd3ccd1f56941129c6aab ]

    We've been seeing the following sporadically throughout our fleet

    panic: kernel BUG at fs/btrfs/relocation.c:4584!
    netversion: 5.0-0
    Backtrace:
     #0 [ffffc90003adb880] machine_kexec at ffffffff81041da8
     #1 [ffffc90003adb8c8] __crash_kexec at ffffffff8110396c
     #2 [ffffc90003adb988] crash_kexec at ffffffff811048ad
     #3 [ffffc90003adb9a0] oops_end at ffffffff8101c19a
     #4 [ffffc90003adb9c0] do_trap at ffffffff81019114
     #5 [ffffc90003adba00] do_error_trap at ffffffff810195d0
     #6 [ffffc90003adbab0] invalid_op at ffffffff81a00a9b
        [exception RIP: btrfs_reloc_cow_block+692]
        RIP: ffffffff8143b614  RSP: ffffc90003adbb68  RFLAGS: 00010246
        RAX: fffffffffffffff7  RBX: ffff8806b9c32000  RCX: ffff8806aad00690
        RDX: ffff880850b295e0  RSI: ffff8806b9c32000  RDI: ffff88084f205bd0
        RBP: ffff880849415000   R8: ffffc90003adbbe0   R9: ffff88085ac90000
        R10: ffff8805f7369140  R11: 0000000000000000  R12: ffff880850b295e0
        R13: ffff88084f205bd0  R14: 0000000000000000  R15: 0000000000000000
        ORIG_RAX: ffffffffffffffff  CS: 0010  SS: 0018
     #7 [ffffc90003adbbb0] __btrfs_cow_block at ffffffff813bf1cd
     #8 [ffffc90003adbc28] btrfs_cow_block at ffffffff813bf4b3
     #9 [ffffc90003adbc78] btrfs_search_slot at ffffffff813c2e6c

    The way relocation moves data extents is by creating a reloc inode and
    preallocating extents in this inode and then copying the data into these
    preallocated extents.  Once we've done this for all of our extents,
    we'll write out these dirty pages, which marks the extent written, and
    goes into btrfs_reloc_cow_block().  From here we get our current
    reloc_control, which _should_ match the reloc_control for the current
    block group we're relocating.

    However if we get an ENOSPC in this path at some point we'll bail out,
    never initiating writeback on this inode.  Not a huge deal, unless we
    happen to be doing relocation on a different block group, and this block
    group is now rc->stage == UPDATE_DATA_PTRS.  This trips the BUG_ON() in
    btrfs_reloc_cow_block(), because we expect to be done modifying the data
    inode.  We are in fact done modifying the metadata for the data inode
    we're currently using, but not the one from the failed block group, and
    thus we BUG_ON().

    (This happens when writeback finishes for extents from the previous
    group, when we are at btrfs_finish_ordered_io() which updates the data
    reloc tree (inode item, drops/adds extent items, etc).)

    Fix this by writing out the reloc data inode always, and then breaking
    out of the loop after that point to keep from tripping this BUG_ON()
    later.

    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    [ add note from Filipe ]
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f033d6a229e56807fe4703c003d2b00043c7d478
Author: Robbie Ko <robbieko@synology.com>
Date:   Tue Mar 26 11:56:11 2019 +0800
```

```
    Btrfs: fix data bytes_may_use underflow with fallocate due to failed quota reserve

    [ Upstream commit 39ad317315887c2cb9a4347a93a8859326ddf136 ]

    When doing fallocate, we first add the range to the reserve_list and
    then reserve the quota.  If quota reservation fails, we'll release all
    reserved parts of reserve_list.

    However, cur_offset is not updated to indicate that this range is
    already been inserted into the list.  Therefore, the same range is freed
    twice.  Once at list_for_each_entry loop, and once at the end of the
    function.  This will result in WARN_ON on bytes_may_use when we free the
    remaining space.

    At the end, under the 'out' label we have a call to:

       btrfs_free_reserved_data_space(inode, data_reserved, alloc_start, alloc_end -
cur_offset);

    The start offset, third argument, should be cur_offset.

    Everything from alloc_start to cur_offset was freed by the
    list_for_each_entry_safe_loop.

    Fixes: 18513091af94 ("btrfs: update btrfs_space_info's bytes_may_use timely")
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: Robbie Ko <robbieko@synology.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eee8c24f4ce72ec5d3319dc73b9f1ebbdc148727
Author: Nadav Amit <namit@vmware.com>
Date:   Thu Apr 25 17:11:31 2019 -0700

    x86/modules: Avoid breaking W^X while loading modules

    [ Upstream commit f2c65fb3221adc6b73b0549fc7ba892022db9797 ]

    When modules and BPF filters are loaded, there is a time window in
    which some memory is both writable and executable. An attacker that has
    already found another vulnerability (e.g., a dangling pointer) might be
    able to exploit this behavior to overwrite kernel code. Prevent having
    writable executable PTEs in this stage.

    In addition, avoiding having W+X mappings can also slightly simplify the
    patching of modules code on initialization (e.g., by alternatives and
    static-key), as would be done in the next patch. This was actually the
    main motivation for this patch.

    To avoid having W+X mappings, set them initially as RW (NX) and after
    they are set as RO set them as X as well. Setting them as executable is
    done as a separate step to avoid one core in which the old PTE is cached
    (hence writable), and another which sees the updated PTE (executable),
    which would break the W^X protection.

    Suggested-by: Thomas Gleixner <tglx@linutronix.de>
    Suggested-by: Andy Lutomirski <luto@amacapital.net>
    Signed-off-by: Nadav Amit <namit@vmware.com>
    Signed-off-by: Rick Edgecombe <rick.p.edgecombe@intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: <akpm@linux-foundation.org>
    Cc: <ard.biesheuvel@linaro.org>
    Cc: <deneen.t.dock@intel.com>
    Cc: <kernel-hardening@lists.openwall.com>
    Cc: <kristen@linux.intel.com>
```

```
      Cc: <linux_dti@icloud.com>
      Cc: <will.deacon@arm.com>
      Cc: Andy Lutomirski <luto@kernel.org>
      Cc: Borislav Petkov <bp@alien8.de>
      Cc: Dave Hansen <dave.hansen@intel.com>
      Cc: H. Peter Anvin <hpa@zytor.com>
      Cc: Jessica Yu <jeyu@kernel.org>
      Cc: Kees Cook <keescook@chromium.org>
      Cc: Linus Torvalds <torvalds@linux-foundation.org>
      Cc: Masami Hiramatsu <mhiramat@kernel.org>
      Cc: Rik van Riel <riel@surriel.com>
      Link: https://lkml.kernel.org/r/20190426001143.4983-12-namit@vmware.com
      Signed-off-by: Ingo Molnar <mingo@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit f4edac5b43c1ed897b8049f7aa37faa32ceba896
  Author: Bart Van Assche <bvanassche@acm.org>
  Date:   Wed Apr 17 14:44:41 2019 -0700

      scsi: qla2xxx: Fix hardirq-unsafe locking

      [ Upstream commit 300ec7415c1fed5c73660f50c8e14a67e236dc0a ]

      Since fc_remote_port_delete() must be called with interrupts enabled, do
      not disable interrupts when calling that function. Remove the lockin calls
      from around the put_sess() call. This is safe because the function that is
      called when the final reference is dropped, qlt_unreg_sess(), grabs the
      proper locks. This patch avoids that lockdep reports the following:

      WARNING: HARDIRQ-safe -> HARDIRQ-unsafe lock order detected
      kworker/2:1/62 [HC0[0]:SC0[0]:HE0:SE1] is trying to acquire:
      0000000009e679b3 (&(&k->k_lock)->rlock){+.+.}, at: klist_next+0x43/0x1d0

      and this task is already holding:
      00000000a033b71c (&(&ha->tgt.sess_lock)->rlock){-...}, at:
  qla24xx_delete_sess_fn+0x55/0xf0 [qla2xxx_scst]
      which would create a new lock dependency:
       (&(&ha->tgt.sess_lock)->rlock){-...} -> (&(&k->k_lock)->rlock){+.+.}

      but this new dependency connects a HARDIRQ-irq-safe lock:
       (&(&ha->tgt.sess_lock)->rlock){-...}

      ... which became HARDIRQ-irq-safe at:
        lock_acquire+0xe3/0x200
        _raw_spin_lock_irqsave+0x3d/0x60
        qla24xx_report_id_acquisition+0xa69/0xe30 [qla2xxx_scst]
        qla24xx_process_response_queue+0x69e/0x1270 [qla2xxx_scst]
        qla24xx_msix_rsp_q+0x79/0xf0 [qla2xxx_scst]
        __handle_irq_event_percpu+0x79/0x3c0
        handle_irq_event_percpu+0x70/0xf0
        handle_irq_event+0x5a/0x8b
        handle_edge_irq+0x12c/0x310
        handle_irq+0x192/0x20a
        do_IRQ+0x73/0x160
        ret_from_intr+0x0/0x1d
        default_idle+0x23/0x1f0
        arch_cpu_idle+0x15/0x20
        default_idle_call+0x35/0x40
        do_idle+0x2bb/0x2e0
        cpu_startup_entry+0x1d/0x20
        start_secondary+0x2a8/0x320
        secondary_startup_64+0xa4/0xb0

      to a HARDIRQ-irq-unsafe lock:
       (&(&k->k_lock)->rlock){+.+.}
```

```
     ... which became HARDIRQ-irq-unsafe at:
     ...
       lock_acquire+0xe3/0x200
       _raw_spin_lock+0x32/0x50
       klist_add_tail+0x33/0xb0
       device_add+0x7e1/0xb50
       device_create_groups_vargs+0x11c/0x150
       device_create_with_groups+0x89/0xb0
       vtconsole_class_init+0xb2/0x124
       do_one_initcall+0xc5/0x3ce
       kernel_init_freeable+0x295/0x32e
       kernel_init+0x11/0x11b
       ret_from_fork+0x3a/0x50

     other info that might help us debug this:

      Possible interrupt unsafe locking scenario:

            CPU0                    CPU1
            ----                    ----
       lock(&(&k->k_lock)->rlock);
                                    local_irq_disable();
                                    lock(&(&ha->tgt.sess_lock)->rlock);
                                    lock(&(&k->k_lock)->rlock);
       <Interrupt>
         lock(&(&ha->tgt.sess_lock)->rlock);

      *** DEADLOCK ***

     3 locks held by kworker/2:1/62:
      #0: 00000000a4319c16 ((wq_completion)"qla2xxx_wq"){+.+.}, at:
 process_one_work+0x437/0xa80
      #1: 00000000ffa34c42 ((work_completion)(&sess->del_work)){+.+.}, at:
 process_one_work+0x437/0xa80
      #2: 00000000a033b71c (&(&ha->tgt.sess_lock)->rlock){-...}, at:
 qla24xx_delete_sess_fn+0x55/0xf0 [qla2xxx_scst]

     the dependencies between HARDIRQ-irq-safe lock and the holding lock:
     -> (&(&ha->tgt.sess_lock)->rlock){-...} ops: 8 {
        IN-HARDIRQ-W at:
                         lock_acquire+0xe3/0x200
                         _raw_spin_lock_irqsave+0x3d/0x60
                         qla24xx_report_id_acquisition+0xa69/0xe30 [qla2xxx_scst]
                         qla24xx_process_response_queue+0x69e/0x1270 [qla2xxx_scst]
                         qla24xx_msix_rsp_q+0x79/0xf0 [qla2xxx_scst]
                         __handle_irq_event_percpu+0x79/0x3c0
                         handle_irq_event_percpu+0x70/0xf0
                         handle_irq_event+0x5a/0x8b
                         handle_edge_irq+0x12c/0x310
                         handle_irq+0x192/0x20a
                         do_IRQ+0x73/0x160
                         ret_from_intr+0x0/0x1d
                         default_idle+0x23/0x1f0
                         arch_cpu_idle+0x15/0x20
                         default_idle_call+0x35/0x40
                         do_idle+0x2bb/0x2e0
                         cpu_startup_entry+0x1d/0x20
                         start_secondary+0x2a8/0x320
                         secondary_startup_64+0xa4/0xb0
        INITIAL USE at:
                        lock_acquire+0xe3/0x200
                        _raw_spin_lock_irqsave+0x3d/0x60
                        qla24xx_report_id_acquisition+0xa69/0xe30 [qla2xxx_scst]
                        qla24xx_process_response_queue+0x69e/0x1270 [qla2xxx_scst]
```

```
                    qla24xx_msix_rsp_q+0x79/0xf0 [qla2xxx_scst]
                    __handle_irq_event_percpu+0x79/0x3c0
                    handle_irq_event_percpu+0x70/0xf0
                    handle_irq_event+0x5a/0x8b
                    handle_edge_irq+0x12c/0x310
                    handle_irq+0x192/0x20a
                    do_IRQ+0x73/0x160
                    ret_from_intr+0x0/0x1d
                    default_idle+0x23/0x1f0
                    arch_cpu_idle+0x15/0x20
                    default_idle_call+0x35/0x40
                    do_idle+0x2bb/0x2e0
                    cpu_startup_entry+0x1d/0x20
                    start_secondary+0x2a8/0x320
                    secondary_startup_64+0xa4/0xb0
 }
 ... key      at: [<ffffffffa0c0d080>] __key.85462+0x0/0xfffffffffff7df80 [qla2xxx_scst]
 ... acquired at:
   lock_acquire+0xe3/0x200
   _raw_spin_lock_irqsave+0x3d/0x60
   klist_next+0x43/0x1d0
   device_for_each_child+0x96/0x110
   scsi_target_block+0x3c/0x40 [scsi_mod]
   fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
   qla2x00_mark_device_lost+0xa0b/0xa30 [qla2xxx_scst]
   qlt_unreg_sess+0x1c6/0x380 [qla2xxx_scst]
   qla24xx_delete_sess_fn+0xe6/0xf0 [qla2xxx_scst]
   process_one_work+0x511/0xa80
   worker_thread+0x67/0x5b0
   kthread+0x1d2/0x1f0
   ret_from_fork+0x3a/0x50

the dependencies between the lock to be acquired
 and HARDIRQ-irq-unsafe lock:
-> (&(&k->k_lock)->rlock){+.+.} ops: 13831 {
   HARDIRQ-ON-W at:
                    lock_acquire+0xe3/0x200
                    _raw_spin_lock+0x32/0x50
                    klist_add_tail+0x33/0xb0
                    device_add+0x7e1/0xb50
                    device_create_groups_vargs+0x11c/0x150
                    device_create_with_groups+0x89/0xb0
                    vtconsole_class_init+0xb2/0x124
                    do_one_initcall+0xc5/0x3ce
                    kernel_init_freeable+0x295/0x32e
                    kernel_init+0x11/0x11b
                    ret_from_fork+0x3a/0x50
   SOFTIRQ-ON-W at:
                    lock_acquire+0xe3/0x200
                    _raw_spin_lock+0x32/0x50
                    klist_add_tail+0x33/0xb0
                    device_add+0x7e1/0xb50
                    device_create_groups_vargs+0x11c/0x150
                    device_create_with_groups+0x89/0xb0
                    vtconsole_class_init+0xb2/0x124
                    do_one_initcall+0xc5/0x3ce
                    kernel_init_freeable+0x295/0x32e
                    kernel_init+0x11/0x11b
                    ret_from_fork+0x3a/0x50
   INITIAL USE at:
                   lock_acquire+0xe3/0x200
                   _raw_spin_lock+0x32/0x50
                   klist_add_tail+0x33/0xb0
                   device_add+0x7e1/0xb50
                   device_create_groups_vargs+0x11c/0x150
```

```
                      device_create_with_groups+0x89/0xb0
                      vtconsole_class_init+0xb2/0x124
                      do_one_initcall+0xc5/0x3ce
                      kernel_init_freeable+0x295/0x32e
                      kernel_init+0x11/0x11b
                      ret_from_fork+0x3a/0x50
     }
     ... key      at: [<ffffffff83ed8780>] __key.15491+0x0/0x40
     ... acquired at:
       lock_acquire+0xe3/0x200
       _raw_spin_lock_irqsave+0x3d/0x60
       klist_next+0x43/0x1d0
       device_for_each_child+0x96/0x110
       scsi_target_block+0x3c/0x40 [scsi_mod]
       fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
       qla2x00_mark_device_lost+0xa0b/0xa30 [qla2xxx_scst]
       qlt_unreg_sess+0x1c6/0x380 [qla2xxx_scst]
       qla24xx_delete_sess_fn+0xe6/0xf0 [qla2xxx_scst]
       process_one_work+0x511/0xa80
       worker_thread+0x67/0x5b0
       kthread+0x1d2/0x1f0
       ret_from_fork+0x3a/0x50

    stack backtrace:
    CPU: 2 PID: 62 Comm: kworker/2:1 Tainted: G           O      5.0.7-dbg+ #8
    Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011
    Workqueue: qla2xxx_wq qla24xx_delete_sess_fn [qla2xxx_scst]
    Call Trace:
     dump_stack+0x86/0xca
     check_usage.cold.52+0x473/0x563
     __lock_acquire+0x11c0/0x23e0
     lock_acquire+0xe3/0x200
     _raw_spin_lock_irqsave+0x3d/0x60
     klist_next+0x43/0x1d0
     device_for_each_child+0x96/0x110
     scsi_target_block+0x3c/0x40 [scsi_mod]
     fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
     qla2x00_mark_device_lost+0xa0b/0xa30 [qla2xxx_scst]
     qlt_unreg_sess+0x1c6/0x380 [qla2xxx_scst]
     qla24xx_delete_sess_fn+0xe6/0xf0 [qla2xxx_scst]
     process_one_work+0x511/0xa80
     worker_thread+0x67/0x5b0
     kthread+0x1d2/0x1f0
     ret_from_fork+0x3a/0x50

    Cc: Himanshu Madhani <hmadhani@marvell.com>
    Cc: Giridhar Malavali <gmalavali@marvell.com>
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Acked-by: Himanshu Madhani <hmadhani@marvell.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b64ba7a584135e90f0517b729d58e74b7f72b209
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Wed Apr 17 14:44:43 2019 -0700

    scsi: qla2xxx: Avoid that lockdep complains about unsafe locking in
tcm_qla2xxx_close_session()

    [ Upstream commit d4023db71108375e4194e92730ba0d32d7f07813 ]

    This patch avoids that lockdep reports the following warning:

    ================================================
    WARNING: HARDIRQ-safe -> HARDIRQ-unsafe lock order detected
```

```
     5.1.0-rc1-dbg+ #11 Tainted: G        W
     -----------------------------------------------------
     rmdir/1478 [HC0[0]:SC0[0]:HE0:SE1] is trying to acquire:
     00000000e7ac4607 (&(&k->k_lock)->rlock){+.+.}, at: klist_next+0x43/0x1d0

     and this task is already holding:
     00000000cf0baf5e (&(&ha->tgt.sess_lock)->rlock){-...}, at:
  tcm_qla2xxx_close_session+0x57/0xb0 [tcm_qla2xxx]
     which would create a new lock dependency:
      (&(&ha->tgt.sess_lock)->rlock){-...} -> (&(&k->k_lock)->rlock){+.+.}

     but this new dependency connects a HARDIRQ-irq-safe lock:
      (&(&ha->tgt.sess_lock)->rlock){-...}

     ... which became HARDIRQ-irq-safe at:
       lock_acquire+0xe3/0x200
       _raw_spin_lock_irqsave+0x3d/0x60
       qla2x00_fcport_event_handler+0x1f3d/0x22b0 [qla2xxx]
       qla2x00_async_login_sp_done+0x1dc/0x1f0 [qla2xxx]
       qla24xx_process_response_queue+0xa37/0x10e0 [qla2xxx]
       qla24xx_msix_rsp_q+0x79/0xf0 [qla2xxx]
       __handle_irq_event_percpu+0x79/0x3c0
       handle_irq_event_percpu+0x70/0xf0
       handle_irq_event+0x5a/0x8b
       handle_edge_irq+0x12c/0x310
       handle_irq+0x192/0x20a
       do_IRQ+0x73/0x160
       ret_from_intr+0x0/0x1d
       default_idle+0x23/0x1f0
       arch_cpu_idle+0x15/0x20
       default_idle_call+0x35/0x40
       do_idle+0x2bb/0x2e0
       cpu_startup_entry+0x1d/0x20
       start_secondary+0x24d/0x2d0
       secondary_startup_64+0xa4/0xb0

     to a HARDIRQ-irq-unsafe lock:
      (&(&k->k_lock)->rlock){+.+.}

     ... which became HARDIRQ-irq-unsafe at:
     ...
       lock_acquire+0xe3/0x200
       _raw_spin_lock+0x32/0x50
       klist_add_tail+0x33/0xb0
       device_add+0x7f4/0xb60
       device_create_groups_vargs+0x11c/0x150
       device_create_with_groups+0x89/0xb0
       vtconsole_class_init+0xb2/0x124
       do_one_initcall+0xc5/0x3ce
       kernel_init_freeable+0x295/0x32e
       kernel_init+0x11/0x11b
       ret_from_fork+0x3a/0x50

     other info that might help us debug this:

      Possible interrupt unsafe locking scenario:

            CPU0                    CPU1
            ----                    ----
       lock(&(&k->k_lock)->rlock);
                                    local_irq_disable();
                                    lock(&(&ha->tgt.sess_lock)->rlock);
                                    lock(&(&k->k_lock)->rlock);
       <Interrupt>
         lock(&(&ha->tgt.sess_lock)->rlock);
```

```
      *** DEADLOCK ***

     4 locks held by rmdir/1478:
      #0: 000000002c7f1ba4 (sb_writers#10){.+.+}, at: mnt_want_write+0x32/0x70
      #1: 00000000c85eb147 (&default_group_class[depth - 1]#2/1){+.+.}, at:
  do_rmdir+0x217/0x2d0
      #2: 000000002b164d6f (&sb->s_type->i_mutex_key#13){++++}, at: vfs_rmdir+0x7e/0x1d0
      #3: 00000000cf0baf5e (&(&ha->tgt.sess_lock)->rlock){-...}, at:
  tcm_qla2xxx_close_session+0x57/0xb0 [tcm_qla2xxx]

     the dependencies between HARDIRQ-irq-safe lock and the holding lock:
     -> (&(&ha->tgt.sess_lock)->rlock){-...} ops: 127 {
        IN-HARDIRQ-W at:
                         lock_acquire+0xe3/0x200
                         _raw_spin_lock_irqsave+0x3d/0x60
                         qla2x00_fcport_event_handler+0x1f3d/0x22b0 [qla2xxx]
                         qla2x00_async_login_sp_done+0x1dc/0x1f0 [qla2xxx]
                         qla24xx_process_response_queue+0xa37/0x10e0 [qla2xxx]
                         qla24xx_msix_rsp_q+0x79/0xf0 [qla2xxx]
                         __handle_irq_event_percpu+0x79/0x3c0
                         handle_irq_event_percpu+0x70/0xf0
                         handle_irq_event+0x5a/0x8b
                         handle_edge_irq+0x12c/0x310
                         handle_irq+0x192/0x20a
                         do_IRQ+0x73/0x160
                         ret_from_intr+0x0/0x1d
                         default_idle+0x23/0x1f0
                         arch_cpu_idle+0x15/0x20
                         default_idle_call+0x35/0x40
                         do_idle+0x2bb/0x2e0
                         cpu_startup_entry+0x1d/0x20
                         start_secondary+0x24d/0x2d0
                         secondary_startup_64+0xa4/0xb0
        INITIAL USE at:
                        lock_acquire+0xe3/0x200
                        _raw_spin_lock_irqsave+0x3d/0x60
                        qla2x00_loop_resync+0xb3d/0x2690 [qla2xxx]
                        qla2x00_do_dpc+0xcee/0xf30 [qla2xxx]
                        kthread+0x1d2/0x1f0
                        ret_from_fork+0x3a/0x50
      }
      ... key      at: [<ffffffffa125f700>] __key.62804+0x0/0xfffffffffff7e900 [qla2xxx]
      ... acquired at:
        __lock_acquire+0x11ed/0x1b60
        lock_acquire+0xe3/0x200
        _raw_spin_lock_irqsave+0x3d/0x60
        klist_next+0x43/0x1d0
        device_for_each_child+0x96/0x110
        scsi_target_block+0x3c/0x40 [scsi_mod]
        fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
        qla2x00_mark_device_lost+0x4d3/0x500 [qla2xxx]
        qlt_unreg_sess+0x104/0x2c0 [qla2xxx]
        tcm_qla2xxx_close_session+0xa2/0xb0 [tcm_qla2xxx]
        target_shutdown_sessions+0x17b/0x190 [target_core_mod]
        core_tpg_del_initiator_node_acl+0xf3/0x1f0 [target_core_mod]
        target_fabric_nacl_base_release+0x25/0x30 [target_core_mod]
        config_item_release+0x9f/0x120 [configfs]
        config_item_put+0x29/0x2b [configfs]
        configfs_rmdir+0x3d2/0x520 [configfs]
        vfs_rmdir+0xb3/0x1d0
        do_rmdir+0x25c/0x2d0
        __x64_sys_rmdir+0x24/0x30
        do_syscall_64+0x77/0x220
        entry_SYSCALL_64_after_hwframe+0x49/0xbe
```

```
      the dependencies between the lock to be acquired
       and HARDIRQ-irq-unsafe lock:
      -> (&(&k->k_lock)->rlock){+.+.} ops: 14568 {
         HARDIRQ-ON-W at:
                           lock_acquire+0xe3/0x200
                           _raw_spin_lock+0x32/0x50
                           klist_add_tail+0x33/0xb0
                           device_add+0x7f4/0xb60
                           device_create_groups_vargs+0x11c/0x150
                           device_create_with_groups+0x89/0xb0
                           vtconsole_class_init+0xb2/0x124
                           do_one_initcall+0xc5/0x3ce
                           kernel_init_freeable+0x295/0x32e
                           kernel_init+0x11/0x11b
                           ret_from_fork+0x3a/0x50
         SOFTIRQ-ON-W at:
                           lock_acquire+0xe3/0x200
                           _raw_spin_lock+0x32/0x50
                           klist_add_tail+0x33/0xb0
                           device_add+0x7f4/0xb60
                           device_create_groups_vargs+0x11c/0x150
                           device_create_with_groups+0x89/0xb0
                           vtconsole_class_init+0xb2/0x124
                           do_one_initcall+0xc5/0x3ce
                           kernel_init_freeable+0x295/0x32e
                           kernel_init+0x11/0x11b
                           ret_from_fork+0x3a/0x50
         INITIAL USE at:
                          lock_acquire+0xe3/0x200
                          _raw_spin_lock+0x32/0x50
                          klist_add_tail+0x33/0xb0
                          device_add+0x7f4/0xb60
                          device_create_groups_vargs+0x11c/0x150
                          device_create_with_groups+0x89/0xb0
                          vtconsole_class_init+0xb2/0x124
                          do_one_initcall+0xc5/0x3ce
                          kernel_init_freeable+0x295/0x32e
                          kernel_init+0x11/0x11b
                          ret_from_fork+0x3a/0x50
       }
       ... key      at: [<ffffffff83f3d900>] __key.15805+0x0/0x40
       ... acquired at:
         __lock_acquire+0x11ed/0x1b60
         lock_acquire+0xe3/0x200
         _raw_spin_lock_irqsave+0x3d/0x60
         klist_next+0x43/0x1d0
         device_for_each_child+0x96/0x110
         scsi_target_block+0x3c/0x40 [scsi_mod]
         fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
         qla2x00_mark_device_lost+0x4d3/0x500 [qla2xxx]
         qlt_unreg_sess+0x104/0x2c0 [qla2xxx]
         tcm_qla2xxx_close_session+0xa2/0xb0 [tcm_qla2xxx]
         target_shutdown_sessions+0x17b/0x190 [target_core_mod]
         core_tpg_del_initiator_node_acl+0xf3/0x1f0 [target_core_mod]
         target_fabric_nacl_base_release+0x25/0x30 [target_core_mod]
         config_item_release+0x9f/0x120 [configfs]
         config_item_put+0x29/0x2b [configfs]
         configfs_rmdir+0x3d2/0x520 [configfs]
         vfs_rmdir+0xb3/0x1d0
         do_rmdir+0x25c/0x2d0
         __x64_sys_rmdir+0x24/0x30
         do_syscall_64+0x77/0x220
         entry_SYSCALL_64_after_hwframe+0x49/0xbe
```

```
     stack backtrace:
     CPU: 7 PID: 1478 Comm: rmdir Tainted: G        W         5.1.0-rc1-dbg+ #11
     Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011
     Call Trace:
      dump_stack+0x86/0xca
      check_usage.cold.59+0x473/0x563
      check_prev_add.constprop.43+0x1f1/0x1170
      __lock_acquire+0x11ed/0x1b60
      lock_acquire+0xe3/0x200
      _raw_spin_lock_irqsave+0x3d/0x60
      klist_next+0x43/0x1d0
      device_for_each_child+0x96/0x110
      scsi_target_block+0x3c/0x40 [scsi_mod]
      fc_remote_port_delete+0xe7/0x1c0 [scsi_transport_fc]
      qla2x00_mark_device_lost+0x4d3/0x500 [qla2xxx]
      qlt_unreg_sess+0x104/0x2c0 [qla2xxx]
      tcm_qla2xxx_close_session+0xa2/0xb0 [tcm_qla2xxx]
      target_shutdown_sessions+0x17b/0x190 [target_core_mod]
      core_tpg_del_initiator_node_acl+0xf3/0x1f0 [target_core_mod]
      target_fabric_nacl_base_release+0x25/0x30 [target_core_mod]
      config_item_release+0x9f/0x120 [configfs]
      config_item_put+0x29/0x2b [configfs]
      configfs_rmdir+0x3d2/0x520 [configfs]
      vfs_rmdir+0xb3/0x1d0
      do_rmdir+0x25c/0x2d0
      __x64_sys_rmdir+0x24/0x30
      do_syscall_64+0x77/0x220
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

     Cc: Himanshu Madhani <hmadhani@marvell.com>
     Cc: Giridhar Malavali <gmalavali@marvell.com>
     Signed-off-by: Bart Van Assche <bvanassche@acm.org>
     Acked-by: Himanshu Madhani <hmadhani@marvell.com>
     Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 8a7f630b651e57988d6291cdda84ec7741807408
 Author: Bart Van Assche <bvanassche@acm.org>
 Date:   Wed Apr 17 14:44:28 2019 -0700

     scsi: qla2xxx: Fix abort handling in tcm_qla2xxx_write_pending()

     [ Upstream commit e209783d66bca04b5fce4429e59338517ffc1a0b ]

     Implementations of the .write_pending() callback functions must guarantee
     that an appropriate LIO core callback function will be called immediately or
     at a later time.  Make sure that this guarantee is met for aborted SCSI
     commands.

     [mkp: typo]

     Cc: Himanshu Madhani <hmadhani@marvell.com>
     Cc: Giridhar Malavali <gmalavali@marvell.com>
     Fixes: 694833ee00c4 ("scsi: tcm_qla2xxx: Do not allow aborted cmd to advance.") # v4.13.
     Fixes: a07100e00ac4 ("qla2xxx: Fix TMR ABORT interaction issue between qla2xxx and TCM")
 # v4.5.
     Signed-off-by: Bart Van Assche <bvanassche@acm.org>
     Acked-by: Himanshu Madhani <hmadhani@marvell.com>
     Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 0c43934021f75d6167aef3421b0df7552abdbb76
 Author: Bart Van Assche <bvanassche@acm.org>
 Date:   Wed Apr 17 14:44:24 2019 -0700
```

```
    scsi: qla2xxx: Fix a qla24xx_enable_msix() error path

    [ Upstream commit 24afabdbd0b3553963a2bbf465895492b14d1107 ]

    Make sure that the allocated interrupts are freed if allocating memory for
    the msix_entries array fails.

    Cc: Himanshu Madhani <hmadhani@marvell.com>
    Cc: Giridhar Malavali <gmalavali@marvell.com>
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Acked-by: Himanshu Madhani <hmadhani@marvell.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e977b1477a6725868302957e6b5c330220391797
Author: Viresh Kumar <viresh.kumar@linaro.org>
Date:   Tue Apr 30 11:35:52 2019 +0530

    cpufreq: Fix kobject memleak

    [ Upstream commit 4ebe36c94aed95de71a8ce6a6762226d31c938ee ]

    Currently the error return path from kobject_init_and_add() is not
    followed by a call to kobject_put() - which means we are leaking the
    kobject.

    Fix it by adding a call to kobject_put() in the error path of
    kobject_init_and_add().

    Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
    Reviewed-by: Tobin C. Harding <tobin@kernel.org>
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 96b7c30e373f6a753214da2f0c2ee7d0ed4dbb87
Author: Nicholas Piggin <npiggin@gmail.com>
Date:   Tue Apr 9 14:40:05 2019 +1000

    powerpc/watchdog: Use hrtimers for per-CPU heartbeat

    [ Upstream commit 7ae3f6e130e8dc6188b59e3b4ebc2f16e9c8d053 ]

    Using a jiffies timer creates a dependency on the tick_do_timer_cpu
    incrementing jiffies. If that CPU has locked up and jiffies is not
    incrementing, the watchdog heartbeat timer for all CPUs stops and
    creates false positives and confusing warnings on local CPUs, and
    also causes the SMP detector to stop, so the root cause is never
    detected.

    Fix this by using hrtimer based timers for the watchdog heartbeat,
    like the generic kernel hardlockup detector.

    Cc: Gautham R. Shenoy <ego@linux.vnet.ibm.com>
    Reported-by: Ravikumar Bangoria <ravi.bangoria@in.ibm.com>
    Signed-off-by: Nicholas Piggin <npiggin@gmail.com>
    Tested-by: Ravi Bangoria <ravi.bangoria@linux.ibm.com>
    Reported-by: Ravi Bangoria <ravi.bangoria@linux.ibm.com>
    Reviewed-by: Gautham R. Shenoy <ego@linux.vnet.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e08f9e6fd4dbae103b3639bc4be2d2a270e68cc5
Author: Nadav Amit <namit@vmware.com>
Date:   Thu Apr 25 17:11:29 2019 -0700
```

```
    x86/ftrace: Set trampoline pages as executable

    [ Upstream commit 3c0dab44e22782359a0a706cbce72de99a22aa75 ]

    Since alloc_module() will not set the pages as executable soon, set
    ftrace trampoline pages as executable after they are allocated.

    For the time being, do not change ftrace to use the text_poke()
    interface. As a result, ftrace still breaks W^X.

    Signed-off-by: Nadav Amit <namit@vmware.com>
    Signed-off-by: Rick Edgecombe <rick.p.edgecombe@intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Reviewed-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Cc: <akpm@linux-foundation.org>
    Cc: <ard.biesheuvel@linaro.org>
    Cc: <deneen.t.dock@intel.com>
    Cc: <kernel-hardening@lists.openwall.com>
    Cc: <kristen@linux.intel.com>
    Cc: <linux_dti@icloud.com>
    Cc: <will.deacon@arm.com>
    Cc: Andy Lutomirski <luto@kernel.org>
    Cc: Borislav Petkov <bp@alien8.de>
    Cc: Dave Hansen <dave.hansen@linux.intel.com>
    Cc: H. Peter Anvin <hpa@zytor.com>
    Cc: Linus Torvalds <torvalds@linux-foundation.org>
    Cc: Rik van Riel <riel@surriel.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lkml.kernel.org/r/20190426001143.4983-10-namit@vmware.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9e5796a9150597c2351ea47f18145c2d63a2f050
Author: Lorenzo Bianconi <lorenzo@kernel.org>
Date:   Sat Mar 2 14:47:38 2019 +0100

    mt76: remove mt76_queue dependency from tx_queue_skb function pointer

    [ Upstream commit 89a37842b0c13c9e568bf12f4fcbe6507147e41d ]

    Remove mt76_queue dependency from tx_queue_skb function pointer and
    rely on mt76_tx_qid instead. This is a preliminary patch to introduce
    mt76_sw_queue support

    Signed-off-by: Lorenzo Bianconi <lorenzo@kernel.org>
    Signed-off-by: Felix Fietkau <nbd@nbd.name>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f2494310358e01a794ec5491f172b76c78f7d0f9
Author: Qian Cai <cai@lca.pw>
Date:   Mon Apr 29 13:37:01 2019 -0400

    arm64: Fix compiler warning from pte_unmap() with -Wunused-but-set-variable

    [ Upstream commit 74dd022f9e6260c3b5b8d15901d27ebcc5f21eda ]

    When building with -Wunused-but-set-variable, the compiler shouts about
    a number of pte_unmap() users, since this expands to an empty macro on
    arm64:

      | mm/gup.c: In function 'gup_pte_range':
      | mm/gup.c:1727:16: warning: variable 'ptem' set but not used
      | [-Wunused-but-set-variable]
      | mm/gup.c: At top level:
      | mm/memory.c: In function 'copy_pte_range':
```

```
       | mm/memory.c:821:24: warning: variable 'orig_dst_pte' set but not used
       | [-Wunused-but-set-variable]
       | mm/memory.c:821:9: warning: variable 'orig_src_pte' set but not used
       | [-Wunused-but-set-variable]
       | mm/swap_state.c: In function 'swap_ra_info':
       | mm/swap_state.c:641:15: warning: variable 'orig_pte' set but not used
       | [-Wunused-but-set-variable]
       | mm/madvise.c: In function 'madvise_free_pte_range':
       | mm/madvise.c:318:9: warning: variable 'orig_pte' set but not used
       | [-Wunused-but-set-variable]

    Rewrite pte_unmap() as a static inline function, which silences the
    warnings.

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

commit 6bce78421ca8c611c63f1bf305d2c6912a818a05
Author: Marc Zyngier <marc.zyngier@arm.com>
Date:   Mon Apr 8 16:49:01 2019 +0100

    ARM: vdso: Remove dependency with the arch_timer driver internals

    [ Upstream commit 1f5b62f09f6b314c8d70b9de5182dae4de1f94da ]

    The VDSO code uses the kernel helper that was originally designed
    to abstract the access between 32 and 64bit systems. It worked so
    far because this function is declared as 'inline'.

    As we're about to revamp that part of the code, the VDSO would
    break. Let's fix it by doing what should have been done from
    the start, a proper system register access.

    Reviewed-by: Mark Rutland <mark.rutland@arm.com>
    Signed-off-by: Marc Zyngier <marc.zyngier@arm.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 10915c3f4d090edf5b57b87c49009c50a05f4d3c
Author: Fabien Dessenne <fabien.dessenne@st.com>
Date:   Wed Apr 24 09:25:44 2019 -0400

    media: stm32-dcmi: return appropriate error codes during probe

    [ Upstream commit b5b5a27bee5884860798ffd0f08e611a3942064b ]

    During probe, return the provided errors value instead of -ENODEV.
    This allows the driver to be deferred probed if needed.

    Signed-off-by: Fabien Dessenne <fabien.dessenne@st.com>
    Acked-by: Hugues Fruchet <hugues.fruchet@st.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ddceee68ce4db2f63b0488e226600c27a536ba8b
Author: Jon Derrick <jonathan.derrick@intel.com>
Date:   Fri Mar 15 18:05:16 2019 -0600

    drm/nouveau/bar/nv50: ensure BAR is mapped

    [ Upstream commit f10b83de1fd49216a4c657816f48001437e4bdd5 ]

    If the BAR is zero size, it indicates it was never successfully mapped.
```

```
    Ensure that the BAR is valid during initialization before attempting to
    use it.

    Signed-off-by: Jon Derrick <jonathan.derrick@intel.com>
    Signed-off-by: Ben Skeggs <bskeggs@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d72be6289f4a56356afe090c5757e129ad5693f5
Author: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
Date:   Tue Apr 30 10:52:29 2019 -0500

    ACPI / property: fix handling of data_nodes in acpi_get_next_subnode()

    [ Upstream commit 23583f7795025e3c783b680d906509366b0906ad ]

    When the DSDT tables expose devices with subdevices and a set of
    hierarchical _DSD properties, the data returned by
    acpi_get_next_subnode() is incorrect, with the results suggesting a bad
    pointer assignment. The parser works fine with device_nodes or
    data_nodes, but not with a combination of the two.

    The problem is traced to an invalid pointer used when jumping from
    handling device_nodes to data nodes. The existing code looks for data
    nodes below the last subdevice found instead of the common root. Fix
    by forcing the acpi_device pointer to be derived from the same fwnode
    for the two types of subnodes.

    This same problem of handling device and data nodes was already fixed
    in a similar way by 'commit bf4703fdd166 ("ACPI / property: fix data
    node parsing in acpi_get_next_subnode()")' but broken later by 'commit
    34055190b19 ("ACPI / property: Add fwnode_get_next_child_node()")', so
    this should probably go to linux-stable all the way to 4.12

    Signed-off-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Reviewed-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8b12f0b6c078fe9d78a4fcfa57ffd0be08836943
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Wed Apr 24 12:52:18 2019 +0300

    brcm80211: potential NULL dereference in brcmf_cfg80211_vndr_cmds_dcmd_handler()

    [ Upstream commit e025da3d7aa4770bb1d1b3b0aa7cc4da1744852d ]

    If "ret_len" is negative then it could lead to a NULL dereference.

    The "ret_len" value comes from nl80211_vendor_cmd(), if it's negative
    then we don't allocate the "dcmd_buf" buffer.  Then we pass "ret_len" to
    brcmf_fil_cmd_data_set() where it is cast to a very high u32 value.
    Most of the functions in that call tree check whether the buffer we pass
    is NULL but there are at least a couple places which don't such as
    brcmf_dbg_hex_dump() and brcmf_msgbuf_query_dcmd().  We memcpy() to and
    from the buffer so it would result in a NULL dereference.

    The fix is to change the types so that "ret_len" can't be negative.  (If
    we memcpy() zero bytes to NULL, that's a no-op and doesn't cause an
    issue).

    Fixes: 1bacb0487d0e ("brcmfmac: replace cfg80211 testmode with vendor command")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit a699aa0d40499c20d16c12bf692b8f839020d5de
Author: Bodong Wang <bodong@mellanox.com>
Date:   Thu Apr 18 18:24:15 2019 -0500

    net/mlx5: E-Switch, Use atomic rep state to serialize state change

    [ Upstream commit 6f4e02193c9a9ea54dd3151cf97489fa787cd0e6 ]

    When the state of rep was introduced, it was also designed to prevent
    duplicate unloading of the same rep. Considering the following two
    flows when an eswitch manager is at switchdev mode with n VF reps loaded.

    +-------------------------------------+--------------------------------+
    | cpu-0                               | cpu-1                          |
    | --------                            | --------                       |
    | mlx5_ib_remove                      | mlx5_eswitch_disable_sriov     |
    |  mlx5_ib_unregister_vport_reps      |  esw_offloads_cleanup          |
    |   mlx5_eswitch_unregister_vport_reps |   esw_offloads_unload_all_reps |
    |    __unload_reps_all_vport          |    __unload_reps_all_vport     |
    +-------------------------------------+--------------------------------+

    These two flows will try to unload the same rep. Per original design,
    once one flow unloads the rep, the state moves to REGISTERED. The 2nd
    flow will no longer needs to do the unload and bails out. However, as
    read and write of the state is not atomic, when 1st flow is doing the
    unload, the state is still LOADED, 2nd flow is able to do the same
    unload action. Kernel crash will happen.

    To solve this, driver should do atomic test-and-set for the state. So
    that only one flow can change the rep state from LOADED to REGISTERED,
    and proceed to do the actual unloading.

    Since the state is changing to atomic type, all other read/write should
    be atomic action as well.

    Fixes: f121e0ea9586 (net/mlx5: E-Switch, Add state to eswitch vport representors)
    Signed-off-by: Bodong Wang <bodong@mellanox.com>
    Reviewed-by: Parav Pandit <parav@mellanox.com>
    Reviewed-by: Vu Pham <vuhuong@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a339b557131c58d151e90baa7567f7f1801a418d
Author: Flavio Suligoi <f.suligoi@asem.it>
Date:   Fri Apr 12 09:32:19 2019 +0200

    spi: pxa2xx: fix SCR (divisor) calculation

    [ Upstream commit 29f2133717c527f492933b0622a4aafe0b3cbe9e ]

    Calculate the divisor for the SCR (Serial Clock Rate), avoiding
    that the SSP transmission rate can be greater than the device rate.

    When the division between the SSP clock and the device rate generates
    a reminder, we have to increment by one the divisor.
    In this way the resulting SSP clock will never be greater than the
    device SPI max frequency.

    For example, with:

     - ssp_clk  = 50 MHz
     - dev freq = 15 MHz

    without this patch the SSP clock will be greater than 15 MHz:
```

```
      - 25 MHz for PXA25x_SSP and CE4100_SSP
      - 16,56 MHz for the others

    Instead, with this patch, we have in both case an SSP clock of 12.5MHz,
    so the max rate of the SPI device clock is respected.

    Signed-off-by: Flavio Suligoi <f.suligoi@asem.it>
    Reviewed-by: Jarkko Nikula <jarkko.nikula@linux.intel.com>
    Reviewed-by: Jarkko Nikula <jarkko.nikula@linux.intel.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 53b314ceb4e48a3a3f34a6379f93ef0821275ae0
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Tue Apr 16 15:12:23 2019 +0200

    ASoC: imx: fix fiq dependencies

    [ Upstream commit ea751227c813ab833609afecfeedaf0aa26f327e ]

    During randconfig builds, I occasionally run into an invalid configuration
    of the freescale FIQ sound support:

    WARNING: unmet direct dependencies detected for SND_SOC_IMX_PCM_FIQ
      Depends on [m]: SOUND [=y] && !UML && SND [=y] && SND_SOC [=y] && SND_IMX_SOC [=m]
      Selected by [y]:
      - SND_SOC_FSL_SPDIF [=y] && SOUND [=y] && !UML && SND [=y] && SND_SOC [=y] &&
SND_IMX_SOC [=m]!=n && (MXC_TZIC [=n] || MXC_AVIC [=y])

    sound/soc/fsl/imx-ssi.o: In function `imx_ssi_remove':
    imx-ssi.c:(.text+0x28): undefined reference to `imx_pcm_fiq_exit'
    sound/soc/fsl/imx-ssi.o: In function `imx_ssi_probe':
    imx-ssi.c:(.text+0xa64): undefined reference to `imx_pcm_fiq_init'

    The Kconfig warning is a result of the symbol being defined inside of
    the "if SND_IMX_SOC" block, and is otherwise harmless. The link error
    is more tricky and happens with SND_SOC_IMX_SSI=y, which may or may not
    imply FIQ support. However, if SND_SOC_FSL_SSI is set to =m at the same
    time, that selects SND_SOC_IMX_PCM_FIQ as a loadable module dependency,
    which then causes a link failure from imx-ssi.

    The solution here is to make SND_SOC_IMX_PCM_FIQ built-in whenever
    one of its potential users is built-in.

    Fixes: ff40260f79dc ("ASoC: fsl: refine DMA/FIQ dependencies")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0da57f6a2988c72c8545ab6417e81f5d1fd791c9
Author: Claudiu Beznea <claudiu.beznea@microchip.com>
Date:   Wed Apr 24 09:17:59 2019 +0000

    spi: atmel-quadspi: fix crash while suspending

    [ Upstream commit e5c27498a0403b270620b1a8a0a66e3efc222fb6 ]

    atmel_qspi objects are kept in spi_controller objects, so, first get
    pointer to spi_controller object and then get atmel_qspi object from
    spi_controller object.

    Fixes: 2d30ac5ed633 ("mtd: spi-nor: atmel-quadspi: Use spi-mem interface for atmel-
quadspi driver")
    Signed-off-by: Claudiu Beznea <claudiu.beznea@microchip.com>
    Reviewed-by: Tudor Ambarus <tudor.ambarus@microchip.com>
```

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

commit ef2c85e8247f0b208818fcf8037d3528c872d165
Author: Anju T Sudhakar <anju@linux.vnet.ibm.com>
Date:   Tue Dec 18 11:50:41 2018 +0530

    powerpc/perf: Fix loop exit condition in nest_imc_event_init

    [ Upstream commit 860b7d2286236170a36f94946d03ca9888d32571 ]

    The data structure (i.e struct imc_mem_info) to hold the memory address
    information for nest imc units is allocated based on the number of nodes
    in the system.

    nest_imc_event_init() traverse this struct array to calculate the memory
    base address for the event-cpu. If we fail to find a match for the event
    cpu's chip-id in imc_mem_info struct array, then the do-while loop will
    iterate until we crash.

    Fix this by changing the loop exit condition based on the number of
    non zero vbase elements in the array, since the allocation is done for
    nr_chips + 1.

    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Fixes: 885dcd709ba91 ("powerpc/perf: Add nest IMC PMU support")
    Signed-off-by: Anju T Sudhakar <anju@linux.vnet.ibm.com>
    Reviewed-by: Madhavan Srinivasan <maddy@linux.vnet.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5d98a98def4b728873af428753cd7b3353630506
Author: Bo YU <tsu.yubo@gmail.com>
Date:   Tue Oct 30 09:21:55 2018 -0400

    powerpc/boot: Fix missing check of lseek() return value

    [ Upstream commit 5d085ec04a000fefb5182d3b03ee46ca96d8389b ]

    This is detected by Coverity scan: CID: 1440481

    Signed-off-by: Bo YU <tsu.yubo@gmail.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f4456a34fce44715688a28eefe75b3dfda7a0656
Author: Anju T Sudhakar <anju@linux.vnet.ibm.com>
Date:   Tue Nov 27 13:54:52 2018 +0530

    powerpc/perf: Return accordingly on invalid chip-id in

    [ Upstream commit a913e5e8b43be1d3897a141ce61c1ec071cad89c ]

    Nest hardware counter memory resides in a per-chip reserve-memory.
    During nest_imc_event_init(), chip-id of the event-cpu is considered to
    calculate the base memory addresss for that cpu. Return, proper error
    condition if the chip_id calculated is invalid.

    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Fixes: 885dcd709ba91 ("powerpc/perf: Add nest IMC PMU support")
    Reviewed-by: Madhavan Srinivasan <maddy@linux.vnet.ibm.com>
    Signed-off-by: Anju T Sudhakar <anju@linux.vnet.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit de7a7a684318d483e5c26b6f7a34c7247e4cfed3
Author: Jerome Brunet <jbrunet@baylibre.com>
Date:   Mon Apr 29 15:29:39 2019 +0200

    ASoC: hdmi-codec: unlock the device on startup errors

    [ Upstream commit 30180e8436046344b12813dc954b2e01dfdcd22d ]

    If the hdmi codec startup fails, it should clear the current_substream
    pointer to free the device. This is properly done for the audio_startup()
    callback but for snd_pcm_hw_constraint_eld().

    Make sure the pointer cleared if an error is reported.

    Signed-off-by: Jerome Brunet <jbrunet@baylibre.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5360b29ee06bcc241770349368c7993e7ad7cb05
Author: Fei Yang <fei.yang@intel.com>
Date:   Tue Mar 19 22:32:20 2019 -0700

    usb: gadget: f_fs: don't free buffer prematurely

    [ Upstream commit 73103c7f958b99561555c3bd1bc1a0809e0b7d61 ]

    The following kernel panic happens due to the io_data buffer gets deallocated
    before the async io is completed. Add a check for the case where io_data buffer
    should be deallocated by ffs_user_copy_worker.

    [   41.663334] BUG: unable to handle kernel NULL pointer dereference at 0000000000000048
    [   41.672099] #PF error: [normal kernel read fault]
    [   41.677356] PGD 20c974067 P4D 20c974067 PUD 20c973067 PMD 0
    [   41.683687] Oops: 0000 [#1] PREEMPT SMP
    [   41.687976] CPU: 1 PID: 7 Comm: kworker/u8:0 Tainted: G     U            5.0.0-quilt-
2e5dc0ac-00790-gd8c79f2-dirty #2
    [   41.705309] Workqueue: adb ffs_user_copy_worker
    [   41.705316] RIP: 0010:__vunmap+0x2a/0xc0
    [   41.705318] Code: 0f 1f 44 00 00 48 85 ff 0f 84 87 00 00 00 55 f7 c7 ff 0f 00 00 48
89 e5 41 55 41 89 f5 41 54 53 48 89 fb 75 71 e8 56 d7 ff ff <4c> 8b 60 48 4d 85 e4 74 76 48
89 df e8 25 ff ff ff 45 85 ed 74 46
    [   41.705320] RSP: 0018:ffffbc3a40053df0 EFLAGS: 00010286
    [   41.705322] RAX: 0000000000000000 RBX: ffffbc3a406f1000 RCX: 0000000000000000
    [   41.705323] RDX: 0000000000000001 RSI: 0000000000000001 RDI: 00000000ffffffff
    [   41.705324] RBP: ffffbc3a40053e08 R08: 000000000001fb79 R09: 0000000000000037
    [   41.705325] R10: ffffbc3a40053b68 R11: ffffbc3a40053cad R12: fffffffffffffff2
    [   41.705326] R13: 0000000000000001 R14: 0000000000000000 R15: ffffffffffffffff
    [   41.705328] FS:  0000000000000000(0000) GS:ffff9e2977a80000(0000)
knlGS:0000000000000000
    [   41.705329] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [   41.705330] CR2: 0000000000000048 CR3: 000000020c994000 CR4: 00000000003406e0
    [   41.705331] Call Trace:
    [   41.705338]  vfree+0x50/0xb0
    [   41.705341]  ffs_user_copy_worker+0xe9/0x1c0
    [   41.705344]  process_one_work+0x19f/0x3e0
    [   41.705348]  worker_thread+0x3f/0x3b0
    [   41.829766]  kthread+0x12b/0x150
    [   41.833371]  ? process_one_work+0x3e0/0x3e0
    [   41.838045]  ? kthread_create_worker_on_cpu+0x70/0x70
    [   41.843695]  ret_from_fork+0x3a/0x50
    [   41.847689] Modules linked in: hci_uart bluetooth ecdh_generic rfkill_gpio dwc3_pci
dwc3 snd_usb_audio mei_me tpm_crb snd_usbmidi_lib xhci_pci xhci_hcd mei tpm snd_hwdep
cfg80211 snd_soc_skl snd_soc_skl_ipc snd_soc_sst_ipc snd_soc_sst_dsp snd_hda_ext_core
snd_hda_core videobuf2_dma_sg crlmodule
    [   41.876880] CR2: 0000000000000048
```

```
    [   41.880584] ---[ end trace 2bc4addff0f2e673 ]---
    [   41.891346] RIP: 0010:__vunmap+0x2a/0xc0
    [   41.895734] Code: 0f 1f 44 00 00 48 85 ff 0f 84 87 00 00 00 55 f7 c7 ff 0f 00 00 48
 89 e5 41 55 41 89 f5 41 54 53 48 89 fb 75 71 e8 56 d7 ff ff <4c> 8b 60 48 4d 85 e4 74 76 48
 89 df e8 25 ff ff ff 45 85 ed 74 46
    [   41.916740] RSP: 0018:ffffbc3a40053df0 EFLAGS: 00010286
    [   41.922583] RAX: 0000000000000000 RBX: ffffbc3a406f1000 RCX: 0000000000000000
    [   41.930563] RDX: 0000000000000001 RSI: 0000000000000001 RDI: 00000000ffffffff
    [   41.938540] RBP: ffffbc3a40053e08 R08: 000000000001fb79 R09: 0000000000000037
    [   41.946520] R10: ffffbc3a40053b68 R11: ffffbc3a40053cad R12: fffffffffffffff2
    [   41.954502] R13: 0000000000000001 R14: 0000000000000000 R15: ffffffffffffffff
    [   41.962482] FS:  0000000000000000(0000) GS:ffff9e2977a80000(0000)
 knlGS:0000000000000000
    [   41.971536] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [   41.977960] CR2: 0000000000000048 CR3: 000000020c994000 CR4: 00000000003406e0
    [   41.985930] Kernel panic - not syncing: Fatal exception
    [   41.991817] Kernel Offset: 0x16000000 from 0xffffffff81000000 (relocation range:
 0xffffffff80000000-0xffffffffbfffffff)
    [   42.009525] Rebooting in 10 seconds..
    [   52.014376] ACPI MEMORY or I/O RESET_REG.

    Fixes: 772a7a724f69 ("usb: gadget: f_fs: Allow scatter-gather buffers")
    Signed-off-by: Fei Yang <fei.yang@intel.com>
    Reviewed-by: Manu Gautam <mgautam@codeaurora.org>
    Tested-by: John Stultz <john.stultz@linaro.org>
    Signed-off-by: Felipe Balbi <felipe.balbi@linux.intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 79137ab40c3b7cc2fbaefbc3f9281c47afb95cfc
 Author: Marek Szyprowski <m.szyprowski@samsung.com>
 Date:   Wed Mar 27 10:56:08 2019 +0100

    usb: dwc3: move synchronize_irq() out of the spinlock protected block

    [ Upstream commit 41a91c606e7d2b74358a944525267cc451c271e8 ]

    dwc3_gadget_suspend() is called under dwc->lock spinlock. In such context
    calling synchronize_irq() is not allowed. Move the problematic call out
    of the protected block to fix the following kernel BUG during system
    suspend:

    BUG: sleeping function called from invalid context at kernel/irq/manage.c:112
    in_atomic(): 1, irqs_disabled(): 128, pid: 1601, name: rtcwake
    6 locks held by rtcwake/1601:
     #0: f70ac2a2 (sb_writers#7){.+.+}, at: vfs_write+0x130/0x16c
     #1: b5fe1270 (&of->mutex){+.+.}, at: kernfs_fop_write+0xc0/0x1e4
     #2: 7e597705 (kn->count#60){.+.+}, at: kernfs_fop_write+0xc8/0x1e4
     #3: 8b3527d0 (system_transition_mutex){+.+.}, at: pm_suspend+0xc4/0xc04
     #4: fc7f1c42 (&dev->mutex){....}, at: __device_suspend+0xd8/0x74c
     #5: 4b36507e (&(&dwc->lock)->rlock){....}, at: dwc3_gadget_suspend+0x24/0x3c
    irq event stamp: 11252
    hardirqs last  enabled at (11251): [<c09c54a4>] _raw_spin_unlock_irqrestore+0x6c/0x74
    hardirqs last disabled at (11252): [<c09c4d44>] _raw_spin_lock_irqsave+0x1c/0x5c
    softirqs last  enabled at (9744): [<c0102564>] __do_softirq+0x3a4/0x66c
    softirqs last disabled at (9737): [<c0128528>] irq_exit+0x140/0x168
    Preemption disabled at:
    [<00000000>]   (null)
    CPU: 7 PID: 1601 Comm: rtcwake Not tainted
    5.0.0-rc3-next-20190122-00039-ga3f4ee4f8a52 #5252
    Hardware name: SAMSUNG EXYNOS (Flattened Device Tree)
    [<c01110f0>] (unwind_backtrace) from [<c010d120>] (show_stack+0x10/0x14)
    [<c010d120>] (show_stack) from [<c09a4d04>] (dump_stack+0x90/0xc8)
    [<c09a4d04>] (dump_stack) from [<c014c700>] (___might_sleep+0x22c/0x2c8)
    [<c014c700>] (___might_sleep) from [<c0189d68>] (synchronize_irq+0x28/0x84)
    [<c0189d68>] (synchronize_irq) from [<c05cbbf8>] (dwc3_gadget_suspend+0x34/0x3c)
```

```
     [<c05cbbf8>] (dwc3_gadget_suspend) from [<c05bd020>] (dwc3_suspend_common+0x154/0x410)
     [<c05bd020>] (dwc3_suspend_common) from [<c05bd34c>] (dwc3_suspend+0x14/0x2c)
     [<c05bd34c>] (dwc3_suspend) from [<c051c730>] (platform_pm_suspend+0x2c/0x54)
     [<c051c730>] (platform_pm_suspend) from [<c05285d4>] (dpm_run_callback+0xa4/0x3dc)
     [<c05285d4>] (dpm_run_callback) from [<c0528a40>] (__device_suspend+0x134/0x74c)
     [<c0528a40>] (__device_suspend) from [<c052c508>] (dpm_suspend+0x174/0x588)
     [<c052c508>] (dpm_suspend) from [<c0182134>] (suspend_devices_and_enter+0xc0/0xe74)
     [<c0182134>] (suspend_devices_and_enter) from [<c0183658>] (pm_suspend+0x770/0xc04)
     [<c0183658>] (pm_suspend) from [<c0180ddc>] (state_store+0x6c/0xcc)
     [<c0180ddc>] (state_store) from [<c09a9a70>] (kobj_attr_store+0x14/0x20)
     [<c09a9a70>] (kobj_attr_store) from [<c02d6800>] (sysfs_kf_write+0x4c/0x50)
     [<c02d6800>] (sysfs_kf_write) from [<c02d594c>] (kernfs_fop_write+0xfc/0x1e4)
     [<c02d594c>] (kernfs_fop_write) from [<c02593d8>] (__vfs_write+0x2c/0x160)
     [<c02593d8>] (__vfs_write) from [<c0259694>] (vfs_write+0xa4/0x16c)
     [<c0259694>] (vfs_write) from [<c0259870>] (ksys_write+0x40/0x8c)
     [<c0259870>] (ksys_write) from [<c0101000>] (ret_fast_syscall+0x0/0x28)
     Exception stack(0xed55ffa8 to 0xed55fff0)
     ...

     Fixes: 01c10880d242 ("usb: dwc3: gadget: synchronize_irq dwc irq in suspend")
     Signed-off-by: Marek Szyprowski <m.szyprowski@samsung.com>
     Signed-off-by: Felipe Balbi <felipe.balbi@linux.intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 489dc4d1616dbcdeb9eba0d20cdde98d6646fb1e
  Author: Minas Harutyunyan <minas.harutyunyan@synopsys.com>
  Date:   Mon Mar 18 14:24:30 2019 +0400

     usb: dwc2: gadget: Increase descriptors count for ISOC's

     [ Upstream commit 54f37f56631747075f1f9a2f0edf6ba405e3e66c ]

     Some function drivers queueing more than 128 ISOC requests at a time.
     To avoid "descriptor chain full" cases, increasing descriptors count
     from MAX_DMA_DESC_NUM_GENERIC to MAX_DMA_DESC_NUM_HS_ISOC for ISOC's
     only.

     Signed-off-by: Minas Harutyunyan <hminas@synopsys.com>
     Signed-off-by: Felipe Balbi <felipe.balbi@linux.intel.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 5b3f97ce4efdc6ec696da177e9d0440d9bbfe200
  Author: Mac Chiang <mac.chiang@intel.com>
  Date:   Thu May 2 14:12:04 2019 +0800

     ASoC: Intel: kbl_da7219_max98357a: Map BTN_0 to KEY_PLAYPAUSE

     [ Upstream commit 16ec5dfe0327ddcf279957bffe4c8fe527088c63 ]

     On kbl_rt5663_max98927, commit 38a5882e4292
         ("ASoC: Intel: kbl_rt5663_max98927: Map BTN_0 to KEY_PLAYPAUSE")
         This key pair mapping to play/pause when playing Youtube

     The Android 3.5mm Headset jack specification mentions that BTN_0 should
     be mapped to KEY_MEDIA, but this is less logical than KEY_PLAYPAUSE,
     which has much broader userspace support.

     For example, the Chrome OS userspace now supports KEY_PLAYPAUSE to toggle
     play/pause of videos and audio, but does not handle KEY_MEDIA.

     Furthermore, Android itself now supports KEY_PLAYPAUSE equivalently, as the
     new USB headset spec requires KEY_PLAYPAUSE for BTN_0.
     https://source.android.com/devices/accessories/headset/usb-headset-spec

     The same fix is required on Chrome kbl_da7219_max98357a.
```

```
    Signed-off-by: Mac Chiang <mac.chiang@intel.com>
    Reviewed-by: Benson Leung <bleung@chromium.org>
    Acked-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9c3a91551d182d2ee730963a26e0ae5031935fd7
Author: Wen Yang <wen.yang99@zte.com.cn>
Date:   Fri Apr 12 14:02:23 2019 +0800

    pinctrl: zte: fix leaked of_node references

    [ Upstream commit 02d15f0d80720545f1f4922a1550ea4aaad4e152 ]

    The call to of_parse_phandle returns a node pointer with refcount
    incremented thus it must be explicitly decremented after the last
    usage.

    Detected by coccinelle with the following warnings:
    ./drivers/pinctrl/zte/pinctrl-zx.c:415:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 407, but without a corresponding object release
within this function.
    ./drivers/pinctrl/zte/pinctrl-zx.c:422:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 407, but without a corresponding object release
within this function.
    ./drivers/pinctrl/zte/pinctrl-zx.c:436:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 407, but without a corresponding object release
within this function.
    ./drivers/pinctrl/zte/pinctrl-zx.c:444:2-8: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 407, but without a corresponding object release
within this function.
    ./drivers/pinctrl/zte/pinctrl-zx.c:448:1-7: ERROR: missing of_node_put; acquired a node
pointer with refcount incremented on line 407, but without a corresponding object release
within this function.

    Signed-off-by: Wen Yang <wen.yang99@zte.com.cn>
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Cc: Jun Nie <jun.nie@linaro.org>
    Cc: Linus Walleij <linus.walleij@linaro.org>
    Cc: linux-gpio@vger.kernel.org
    Cc: linux-kernel@vger.kernel.org
    Acked-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a3d50fb7ce931f08a2c6194fdc4c9a56c9ed4153
Author: João Paulo Rechi Vita <jprvita@gmail.com>
Date:   Thu May 2 10:01:52 2019 +0800

    Bluetooth: Ignore CC events not matching the last HCI command

    [ Upstream commit f80c5dad7b6467b884c445ffea45985793b4b2d0 ]

    This commit makes the kernel not send the next queued HCI command until
    a command complete arrives for the last HCI command sent to the
    controller. This change avoids a problem with some buggy controllers
    (seen on two SKUs of QCA9377) that send an extra command complete event
    for the previous command after the kernel had already sent a new HCI
    command to the controller.

    The problem was reproduced when starting an active scanning procedure,
    where an extra command complete event arrives for the LE_SET_RANDOM_ADDR
    command. When this happends the kernel ends up not processing the
    command complete for the following commmand, LE_SET_SCAN_PARAM, and
```

```
ultimately behaving as if a passive scanning procedure was being
performed, when in fact controller is performing an active scanning
procedure. This makes it impossible to discover BLE devices as no device
found events are sent to userspace.

This problem is reproducible on 100% of the attempts on the affected
controllers. The extra command complete event can be seen at timestamp
27.420131 on the btmon logs bellow.

Bluetooth monitor ver 5.50
= Note: Linux version 5.0.0+ (x86_64)                                 0.352340
= Note: Bluetooth subsystem version 2.22                              0.352343
= New Index: 80:C5:F2:8F:87:84 (Primary,USB,hci0)              [hci0] 0.352344
= Open Index: 80:C5:F2:8F:87:84                                [hci0] 0.352345
= Index Info: 80:C5:F2:8F:87:84 (Qualcomm)                     [hci0] 0.352346
@ MGMT Open: bluetoothd (privileged) version 1.14             {0x0001} 0.352347
@ MGMT Open: btmon (privileged) version 1.14                  {0x0002} 0.352366
@ MGMT Open: btmgmt (privileged) version 1.14                {0x0003} 27.302164
@ MGMT Command: Start Discovery (0x0023) plen 1       {0x0003} [hci0] 27.302310
        Address type: 0x06
          LE Public
          LE Random
< HCI Command: LE Set Random Address (0x08|0x0005) plen 6   #1 [hci0] 27.302496
        Address: 15:60:F2:91:B2:24 (Non-Resolvable)
> HCI Event: Command Complete (0x0e) plen 4                 #2 [hci0] 27.419117
      LE Set Random Address (0x08|0x0005) ncmd 1
        Status: Success (0x00)
< HCI Command: LE Set Scan Parameters (0x08|0x000b) plen 7  #3 [hci0] 27.419244
        Type: Active (0x01)
        Interval: 11.250 msec (0x0012)
        Window: 11.250 msec (0x0012)
        Own address type: Random (0x01)
        Filter policy: Accept all advertisement (0x00)
> HCI Event: Command Complete (0x0e) plen 4                 #4 [hci0] 27.420131
      LE Set Random Address (0x08|0x0005) ncmd 1
        Status: Success (0x00)
< HCI Command: LE Set Scan Enable (0x08|0x000c) plen 2      #5 [hci0] 27.420259
        Scanning: Enabled (0x01)
        Filter duplicates: Enabled (0x01)
> HCI Event: Command Complete (0x0e) plen 4                 #6 [hci0] 27.420969
      LE Set Scan Parameters (0x08|0x000b) ncmd 1
        Status: Success (0x00)
> HCI Event: Command Complete (0x0e) plen 4                 #7 [hci0] 27.421983
      LE Set Scan Enable (0x08|0x000c) ncmd 1
        Status: Success (0x00)
@ MGMT Event: Command Complete (0x0001) plen 4        {0x0003} [hci0] 27.422059
      Start Discovery (0x0023) plen 1
        Status: Success (0x00)
        Address type: 0x06
          LE Public
          LE Random
@ MGMT Event: Discovering (0x0013) plen 2             {0x0003} [hci0] 27.422067
        Address type: 0x06
          LE Public
          LE Random
        Discovery: Enabled (0x01)
@ MGMT Event: Discovering (0x0013) plen 2             {0x0002} [hci0] 27.422067
        Address type: 0x06
          LE Public
          LE Random
        Discovery: Enabled (0x01)
@ MGMT Event: Discovering (0x0013) plen 2             {0x0001} [hci0] 27.422067
        Address type: 0x06
          LE Public
          LE Random
```

```
             Discovery: Enabled (0x01)

    Signed-off-by: João Paulo Rechi Vita <jprvita@endlessm.com>
    Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b7aa43b40b35e7d085c5245722b5ed41d99975d3
Author: Haiyang Zhang <haiyangz@microsoft.com>
Date:   Tue Apr 30 19:29:07 2019 +0000

    hv_netvsc: fix race that may miss tx queue wakeup

    [ Upstream commit 93aa4792c3908eac87ddd368ee0fe0564148232b ]

    When the ring buffer is almost full due to RX completion messages, a
    TX packet may reach the "low watermark" and cause the queue stopped.
    If the TX completion arrives earlier than queue stopping, the wakeup
    may be missed.

    This patch moves the check for the last pending packet to cover both
    EAGAIN and success cases, so the queue will be reliably waked up when
    necessary.

    Reported-and-tested-by: Stephan Klein <stephan.klein@wegfinder.at>
    Signed-off-by: Haiyang Zhang <haiyangz@microsoft.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7a154742b667b29e93becb7ce19b8dd893b66109
Author: Sameeh Jubran <sameehj@amazon.com>
Date:   Wed May 1 16:47:04 2019 +0300

    net: ena: fix: set freed objects to NULL to avoid failing future allocations

    [ Upstream commit 8ee8ee7fe87bf64738ab4e31be036a7165608b27 ]

    In some cases when a queue related allocation fails, successful past
    allocations are freed but the pointer that pointed to them is not
    set to NULL. This is a problem for 2 reasons:
    1. This is generally a bad practice since this pointer might be
    accidentally accessed in the future.
    2. Future allocations using the same pointer check if the pointer
    is NULL and fail if it is not.

    Fixed this by setting such pointers to NULL in the allocation of
    queue related objects.

    Also refactored the code of ena_setup_tx_resources() to goto-style
    error handling to avoid code duplication of resource freeing.

    Fixes: 1738cd3ed342 ("net: ena: Add a driver for Amazon Elastic Network Adapters (ENA)")
    Signed-off-by: Arthur Kiyanovski <akiyano@amazon.com>
    Signed-off-by: Sameeh Jubran <sameehj@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 93504109f9ffc222ad585efbd830b55325cff47f
Author: Sameeh Jubran <sameehj@amazon.com>
Date:   Wed May 1 16:47:10 2019 +0300

    net: ena: gcc 8: fix compilation warning

    [ Upstream commit f913308879bc6ae437ce64d878c7b05643ddea44 ]

    GCC 8 contains a number of new warnings as well as enhancements to existing
```

```
    checkers. The warning - Wstringop-truncation - warns for calls to bounded
    string manipulation functions such as strncat, strncpy, and stpncpy that
    may either truncate the copied string or leave the destination unchanged.

    In our case the destination string length (32 bytes) is much shorter than
    the source string (64 bytes) which causes this warning to show up. In
    general the destination has to be at least a byte larger than the length
    of the source string with strncpy for this warning not to showup.

    This can be easily fixed by using strlcpy instead which already does the
    truncation to the string. Documentation for this function can be
    found here:

    https://elixir.bootlin.com/linux/latest/source/lib/string.c#L141

    Fixes: 1738cd3ed342 ("net: ena: Add a driver for Amazon Elastic Network Adapters (ENA)")
    Signed-off-by: Sameeh Jubran <sameehj@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ec148de168cebff47811cd3403cea5d422e4a25b
Author: Sameer Pujar <spujar@nvidia.com>
Date:   Thu May 2 18:25:16 2019 +0530

    dmaengine: tegra210-dma: free dma controller in remove()

    [ Upstream commit f030e419501cb95e961e9ed35c493b5d46a04eca ]

    Following kernel panic is seen during DMA driver unload->load sequence
    ================================================================================
    Unable to handle kernel paging request at virtual address ffffff8001198880
    Internal error: Oops: 86000007 [#1] PREEMPT SMP
    CPU: 0 PID: 5907 Comm: HwBinder:4123_1 Tainted: G C 4.9.128-tegra-g065839f
    Hardware name: galen (DT)
    task: ffffffc3590d1a80 task.stack: ffffffc3d0678000
    PC is at 0xffffff8001198880
    LR is at of_dma_request_slave_channel+0xd8/0x1f8
    pc : [<ffffff8001198880>] lr : [<ffffff8008746f30>] pstate: 60400045
    sp : ffffffc3d067b710
    x29: ffffffc3d067b710 x28: 000000000000002f
    x27: ffffff800949e000 x26: ffffff800949e750
    x25: ffffff800949e000 x24: ffffffbefe817d84
    x23: ffffff8009f77cb0 x22: 0000000000000028
    x21: ffffffc3ffda49c8 x20: 0000000000000029
    x19: 0000000000000001 x18: ffffffffffffffff
    x17: 0000000000000000 x16: ffffff80082b66a0
    x15: ffffff8009e78250 x14: 000000000000000a
    x13: 0000000000000038 x12: 0101010101010101
    x11: 0000000000000030 x10: 0101010101010101
    x9 : fffffffffffffffc x8 : 7f7f7f7f7f7f7f7f
    x7 : 62ff726b6b64622c x6 : 0000000000008064
    x5 : 6400000000000000 x4 : ffffffbefe817c44
    x3 : ffffffc3ffda3e08 x2 : ffffff8001198880
    x1 : ffffffc3d48323c0 x0 : ffffffc3d067b788

    Process HwBinder:4123_1 (pid: 5907, stack limit = 0xffffffc3d0678028)
    Call trace:
    [<ffffff8001198880>] 0xffffff8001198880
    [<ffffff80087459f8>] dma_request_chan+0x50/0x1f0
    [<ffffff8008745bc0>] dma_request_slave_channel+0x28/0x40
    [<ffffff8001552c44>] tegra_alt_pcm_open+0x114/0x170
    [<ffffff8008d65fa4>] soc_pcm_open+0x10c/0x878
    [<ffffff8008d18618>] snd_pcm_open_substream+0xc0/0x170
    [<ffffff8008d1878c>] snd_pcm_open+0xc4/0x240
    [<ffffff8008d189e0>] snd_pcm_playback_open+0x58/0x80
```

```
      [<ffffff8008cfc6d4>] snd_open+0xb4/0x178
      [<ffffff8008250628>] chrdev_open+0xb8/0x1d0
      [<ffffff8008246fdc>] do_dentry_open+0x214/0x318
      [<ffffff80082485d0>] vfs_open+0x58/0x88
      [<ffffff800825bce0>] do_last+0x450/0xde0
      [<ffffff800825c718>] path_openat+0xa8/0x368
      [<ffffff800825dd84>] do_filp_open+0x8c/0x110
      [<ffffff8008248a74>] do_sys_open+0x164/0x220
      [<ffffff80082b66dc>] compat_SyS_openat+0x3c/0x50
      [<ffffff8008083040>] el0_svc_naked+0x34/0x38
      ---[ end trace 67e6d544e65b5145 ]---
      Kernel panic - not syncing: Fatal exception
      ====================================================================

      In device probe(), of_dma_controller_register() registers DMA controller.
      But when driver is removed, this is not freed. During driver reload this
      results in data abort and kernel panic. Add of_dma_controller_free() in
      driver remove path to fix the issue.

      Fixes: f46b195799b5 ("dmaengine: tegra-adma: Add support for Tegra210 ADMA")
      Signed-off-by: Sameer Pujar <spujar@nvidia.com>
      Reviewed-by: Jon Hunter <jonathanh@nvidia.com>
      Signed-off-by: Vinod Koul <vkoul@kernel.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 1ba71e3046cb28990597d818f9f442066c645652
  Author: Ming Lei <ming.lei@redhat.com>
  Date:   Tue Apr 30 09:52:23 2019 +0800

      blk-mq: grab .q_usage_counter when queuing request from plug code path

      [ Upstream commit e87eb301bee183d82bb3d04bd71b6660889a2588 ]

      Just like aio/io_uring, we need to grab 2 refcount for queuing one
      request, one is for submission, another is for completion.

      If the request isn't queued from plug code path, the refcount grabbed
      in generic_make_request() serves for submission. In theroy, this
      refcount should have been released after the sumission(async run queue)
      is done. blk_freeze_queue() works with blk_sync_queue() together
      for avoiding race between cleanup queue and IO submission, given async
      run queue activities are canceled because hctx->run_work is scheduled with
      the refcount held, so it is fine to not hold the refcount when
      running the run queue work function for dispatch IO.

      However, if request is staggered into plug list, and finally queued
      from plug code path, the refcount in submission side is actually missed.
      And we may start to run queue after queue is removed because the queue's
      kobject refcount isn't guaranteed to be grabbed in flushing plug list
      context, then kernel oops is triggered, see the following race:

      blk_mq_flush_plug_list():
              blk_mq_sched_insert_requests()
                      insert requests to sw queue or scheduler queue
                      blk_mq_run_hw_queue

      Because of concurrent run queue, all requests inserted above may be
      completed before calling the above blk_mq_run_hw_queue. Then queue can
      be freed during the above blk_mq_run_hw_queue().

      Fixes the issue by grab .q_usage_counter before calling
      blk_mq_sched_insert_requests() in blk_mq_flush_plug_list(). This way is
      safe because the queue is absolutely alive before inserting request.

      Cc: Dongli Zhang <dongli.zhang@oracle.com>
```

```
    Cc: James Smart <james.smart@broadcom.com>
    Cc: linux-scsi@vger.kernel.org,
    Cc: Martin K . Petersen <martin.petersen@oracle.com>,
    Cc: Christoph Hellwig <hch@lst.de>,
    Cc: James E . J . Bottomley <jejb@linux.vnet.ibm.com>,
    Reviewed-by: Bart Van Assche <bvanassche@acm.org>
    Tested-by: James Smart <james.smart@broadcom.com>
    Signed-off-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c415ae66aa1fb617149d31edbee25d4b698cc276
Author: Ming Lei <ming.lei@redhat.com>
Date:   Tue Apr 30 09:52:26 2019 +0800

    blk-mq: split blk_mq_alloc_and_init_hctx into two parts

    [ Upstream commit 7c6c5b7c9186e3fb5b10afb8e5f710ae661144c6 ]

    Split blk_mq_alloc_and_init_hctx into two parts, and one is
    blk_mq_alloc_hctx() for allocating all hctx resources, another
    is blk_mq_init_hctx() for initializing hctx, which serves as
    counter-part of blk_mq_exit_hctx().

    Cc: Dongli Zhang <dongli.zhang@oracle.com>
    Cc: James Smart <james.smart@broadcom.com>
    Cc: Bart Van Assche <bart.vanassche@wdc.com>
    Cc: linux-scsi@vger.kernel.org
    Cc: Martin K . Petersen <martin.petersen@oracle.com>
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: James E . J . Bottomley <jejb@linux.vnet.ibm.com>
    Reviewed-by: Hannes Reinecke <hare@suse.com>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Tested-by: James Smart <james.smart@broadcom.com>
    Signed-off-by: Ming Lei <ming.lei@redhat.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 75dcfe5c969a165b9f95f575c9bb09be624c025f
Author: Tony Nguyen <anthony.l.nguyen@intel.com>
Date:   Tue Apr 16 10:21:15 2019 -0700

    ice: Preserve VLAN Rx stripping settings

    [ Upstream commit e80e76db6c5bbc7a8f8512f3dc630a2170745b0b ]

    When Tx insertion is set, we are not accounting for the state of Rx
    stripping.  This causes Rx stripping to be enabled any time Tx
    insertion is changed, even when it's supposed to be disabled.

    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dd08b6d457f5dabcda5120cff0639ac20cfd11c3
Author: Tony Nguyen <anthony.l.nguyen@intel.com>
Date:   Tue Apr 16 10:21:23 2019 -0700

    ice: Separate if conditions for ice_set_features()

    [ Upstream commit 8f529ff912073f778e3cd74e87fb69a36499fc2f ]

    Set features can have multiple features turned on|off in a single
```

```
    call.  Grouping these all in an if/else means after one condition
    is met, other conditions/features will not be evaluated.  Break
    the if/else statements by feature to ensure all features will be
    handled properly.

    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Anirudh Venkataramanan <anirudh.venkataramanan@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 16e03948c954ebbec44623da4193e263837d80ae
Author: Masahiro Yamada <yamada.masahiro@socionext.com>
Date:   Wed May 1 22:45:59 2019 +0900

    bpftool: exclude bash-completion/bpftool from .gitignore pattern

    [ Upstream commit a7d006714724de4334c5e3548701b33f7b12ca96 ]

    tools/bpf/bpftool/.gitignore has the "bpftool" pattern, which is
    intended to ignore the following build artifact:

      tools/bpf/bpftool/bpftool

    However, the .gitignore entry is effective not only for the current
    directory, but also for any sub-directories.

    So, from the point of .gitignore grammar, the following check-in file
    is also considered to be ignored:

      tools/bpf/bpftool/bash-completion/bpftool

    As the manual gitignore(5) says "Files already tracked by Git are not
    affected", this is not a problem as far as Git is concerned.

    However, Git is not the only program that parses .gitignore because
    .gitignore is useful to distinguish build artifacts from source files.

    For example, tar(1) supports the --exclude-vcs-ignore option. As of
    writing, this option does not work perfectly, but it intends to create
    a tarball excluding files specified by .gitignore.

    So, I believe it is better to fix this issue.

    You can fix it by prefixing the pattern with a slash; the leading slash
    means the specified pattern is relative to the current directory.

    Signed-off-by: Masahiro Yamada <yamada.masahiro@socionext.com>
    Reviewed-by: Quentin Monnet <quentin.monnet@netronome.com>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 85af86c03dbbad3e0549c21abcff5c993500318c
Author: Yonghong Song <yhs@fb.com>
Date:   Mon Apr 29 16:59:38 2019 -0700

    selftests/bpf: set RLIMIT_MEMLOCK properly for test_libbpf_open.c

    [ Upstream commit 6cea33701eb024bc6c920ab83940ee22afd29139 ]

    Test test_libbpf.sh failed on my development server with failure
      -bash-4.4$ sudo ./test_libbpf.sh
      [0] libbpf: Error in bpf_object__probe_name():Operation not permitted(1).
          Couldn't load basic 'r0 = 0' BPF program.
      test_libbpf: failed at file test_l4lb.o
```

```
       selftests: test_libbpf [FAILED]
       -bash-4.4$

    The reason is because my machine has 64KB locked memory by default which
    is not enough for this program to get locked memory.
    Similar to other bpf selftests, let us increase RLIMIT_MEMLOCK
    to infinity, which fixed the issue.

    Signed-off-by: Yonghong Song <yhs@fb.com>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4b9c2fdd672b7a55fed68300b61e58a5e6ab820b
Author: Björn Töpel <bjorn.topel@intel.com>
Date:   Tue Apr 30 14:45:35 2019 +0200

    libbpf: fix invalid munmap call

    [ Upstream commit 0e6741f092979535d159d5a851f12c88bfb7cb9a ]

    When unmapping the AF_XDP memory regions used for the rings, an
    invalid address was passed to the munmap() calls. Instead of passing
    the beginning of the memory region, the descriptor region was passed
    to munmap.

    When the userspace application tried to tear down an AF_XDP socket,
    the operation failed and the application would still have a reference
    to socket it wished to get rid of.

    Reported-by: William Tu <u9012063@gmail.com>
    Fixes: 1cad07884239 ("libbpf: add support for using AF_XDP sockets")
    Signed-off-by: Björn Töpel <bjorn.topel@intel.com>
    Tested-by: William Tu <u9012063@gmail.com>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2d089a4aad52abe13b487b64016f3f36df0795a
Author: Martyna Szapar <martyna.szapar@intel.com>
Date:   Mon Apr 15 14:43:07 2019 -0700

    i40e: Fix of memory leak and integer truncation in i40e_virtchnl.c

    [ Upstream commit 24474f2709af6729b9b1da1c5e160ab62e25e3a4 ]

    Fixed possible memory leak in i40e_vc_add_cloud_filter function:
    cfilter is being allocated and in some error conditions
    the function returns without freeing the memory.

    Fix of integer truncation from u16 (type of queue_id value) to u8
    when calling i40e_vc_isvalid_queue_id function.

    Signed-off-by: Martyna Szapar <martyna.szapar@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 36cd1815fd109860ec7681e41508146cd32d9b26
Author: Vineet Gupta <Vineet.Gupta1@synopsys.com>
Date:   Thu May 2 08:56:50 2019 -0700

    tools/bpf: fix perf build error with uClibc (seen on ARC)

    [ Upstream commit ca31ca8247e2d3807ff5fa1d1760616a2292001c ]

    When build perf for ARC recently, there was a build failure due to lack
    of __NR_bpf.
```

```
    | Auto-detecting system features:
    |
    | ...                     get_cpuid: [ OFF ]
    | ...                           bpf: [ on  ]
    |
    | #  error __NR_bpf not defined. libbpf does not support your arch.
       ^~~~~
    | bpf.c: In function 'sys_bpf':
    | bpf.c:66:17: error: '__NR_bpf' undeclared (first use in this function)
    |  return syscall(__NR_bpf, cmd, attr, size);
    |                 ^~~~~~~~
    |                 sys_bpf
```

```
    Signed-off-by: Vineet Gupta <vgupta@synopsys.com>
    Acked-by: Yonghong Song <yhs@fb.com>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit aeae78ba0a0e6138b86589da6cf32a355c662a63
Author: Raul E Rangel <rrangel@chromium.org>
Date:   Mon Apr 29 11:32:39 2019 -0600

    mmc: core: Verify SD bus width

    [ Upstream commit 9e4be8d03f50d1b25c38e2b59e73b194c130df7d ]

    The SD Physical Layer Spec says the following: Since the SD Memory Card
    shall support at least the two bus modes 1-bit or 4-bit width, then any SD
    Card shall set at least bits 0 and 2 (SD_BUS_WIDTH="0101").

    This change verifies the card has specified a bus width.

    AMD SDHC Device 7806 can get into a bad state after a card disconnect
    where anything transferred via the DATA lines will always result in a
    zero filled buffer. Currently the driver will continue without error if
    the HC is in this condition. A block device will be created, but reading
    from it will result in a zero buffer. This makes it seem like the SD
    device has been erased, when in actuality the data is never getting
    copied from the DATA lines to the data buffer.

    SCR is the first command in the SD initialization sequence that uses the
    DATA lines. By checking that the response was invalid, we can abort
    mounting the card.

    Reviewed-by: Avri Altman <avri.altman@wdc.com>
    Signed-off-by: Raul E Rangel <rrangel@chromium.org>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fb9132e804c00aa583427dfe5191ecb2d360e9a2
Author: Andreas Gruenbacher <agruenba@redhat.com>
Date:   Thu Apr 4 21:11:11 2019 +0100

    gfs2: Fix occasional glock use-after-free

    [ Upstream commit 9287c6452d2b1f24ea8e84bd3cf6f3c6f267f712 ]

    This patch has to do with the life cycle of glocks and buffers.  When
    gfs2 metadata or journaled data is queued to be written, a gfs2_bufdata
    object is assigned to track the buffer, and that is queued to various
    lists, including the glock's gl_ail_list to indicate it's on the active
    items list.  Once the page associated with the buffer has been written,
    it is removed from the ail list, but its life isn't over until a revoke
    has been successfully written.
```

```
    So after the block is written, its bufdata object is moved from the
    glock's gl_ail_list to a file-system-wide list of pending revokes,
    sd_log_le_revoke.  At that point the glock still needs to track how many
    revokes it contributed to that list (in gl_revokes) so that things like
    glock go_sync can ensure all the metadata has been not only written, but
    also revoked before the glock is granted to a different node.  This is
    to guarantee journal replay doesn't replay the block once the glock has
    been granted to another node.

    Ross Lagerwall recently discovered a race in which an inode could be
    evicted, and its glock freed after its ail list had been synced, but
    while it still had unwritten revokes on the sd_log_le_revoke list.  The
    evict decremented the glock reference count to zero, which allowed the
    glock to be freed.  After the revoke was written, function
    revoke_lo_after_commit tried to adjust the glock's gl_revokes counter
    and clear its GLF_LFLUSH flag, at which time it referenced the freed
    glock.

    This patch fixes the problem by incrementing the glock reference count
    in gfs2_add_revoke when the glock's first bufdata object is moved from
    the glock to the global revokes list. Later, when the glock's last such
    bufdata object is freed, the reference count is decremented. This
    guarantees that whichever process finishes last (the revoke writing or
    the evict) will properly free the glock, and neither will reference the
    glock after it has been freed.

    Reported-by: Ross Lagerwall <ross.lagerwall@citrix.com>
    Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
    Signed-off-by: Bob Peterson <rpeterso@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 652fa249b99f96fe3423a0712aa87ab433208551
Author: Mike Marciniszyn <mike.marciniszyn@intel.com>
Date:   Mon Mar 18 09:55:09 2019 -0700

    IB/hfi1: Fix WQ_MEM_RECLAIM warning

    [ Upstream commit 4c4b1996b5db688e2dcb8242b0a3bf7b1e845e42 ]

    The work_item cancels that occur when a QP is destroyed can elicit the
    following trace:

     workqueue: WQ_MEM_RECLAIM ipoib_wq:ipoib_cm_tx_reap [ib_ipoib] is flushing
!WQ_MEM_RECLAIM hfi0_0:_hfi1_do_send [hfi1]
     WARNING: CPU: 7 PID: 1403 at kernel/workqueue.c:2486 check_flush_dependency+0xb1/0x100
     Call Trace:
      __flush_work.isra.29+0x8c/0x1a0
      ? __switch_to_asm+0x40/0x70
      __cancel_work_timer+0x103/0x190
      ? schedule+0x32/0x80
      iowait_cancel_work+0x15/0x30 [hfi1]
      rvt_reset_qp+0x1f8/0x3e0 [rdmavt]
      rvt_destroy_qp+0x65/0x1f0 [rdmavt]
      ? _cond_resched+0x15/0x30
      ib_destroy_qp+0xe9/0x230 [ib_core]
      ipoib_cm_tx_reap+0x21c/0x560 [ib_ipoib]
      process_one_work+0x171/0x370
      worker_thread+0x49/0x3f0
      kthread+0xf8/0x130
      ? max_active_store+0x80/0x80
      ? kthread_bind+0x10/0x10
      ret_from_fork+0x35/0x40

    Since QP destruction frees memory, hfi1_wq should have the WQ_MEM_RECLAIM.
```

```
    The hfi1_wq does not allocate memory with GFP_KERNEL or otherwise become
    entangled with memory reclaim, so this flag is appropriate.

    Fixes: 0a226edd203f ("staging/rdma/hfi1: Use parallel workqueue for SDMA engines")
    Reviewed-by: Michael J. Ruhl <michael.j.ruhl@intel.com>
    Signed-off-by: Mike Marciniszyn <mike.marciniszyn@intel.com>
    Signed-off-by: Dennis Dalessandro <dennis.dalessandro@intel.com>
    Signed-off-by: Jason Gunthorpe <jgg@mellanox.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b6e7eed82f14b2fc1142027f88d6499c055f66ce
Author: Shenghui Wang <shhuiw@foxmail.com>
Date:   Tue May 7 16:03:19 2019 +0800

    io_uring: use cpu_online() to check p->sq_thread_cpu instead of cpu_possible()

    [ Upstream commit 7889f44dd9cee15aff1c3f7daf81ca4dfed48fc7 ]

    This issue is found by running liburing/test/io_uring_setup test.

    When test run, the testcase "attempt to bind to invalid cpu" would not
    pass with messages like:
       io_uring_setup(1, 0xbfc2f7c8), \
    flags: IORING_SETUP_SQPOLL|IORING_SETUP_SQ_AFF, \
    resv: 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000, \
    sq_thread_cpu: 2
       expected -1, got 3
       FAIL

    On my system, there is:
       CPU(s) possible : 0-3
       CPU(s) online   : 0-1
       CPU(s) offline  : 2-3
       CPU(s) present  : 0-1

    The sq_thread_cpu 2 is offline on my system, so the bind should fail.
    But cpu_possible() will pass the check. We shouldn't be able to bind
    to an offline cpu. Use cpu_online() to do the check.

    After the change, the testcase run as expected: EINVAL will be returned
    for cpu offlined.

    Reviewed-by: Jeff Moyer <jmoyer@redhat.com>
    Signed-off-by: Shenghui Wang <shhuiw@foxmail.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e983250f4b27142298bbf766fe8cdfcdde038e78
Author: Abhi Das <adas@redhat.com>
Date:   Tue Apr 30 16:53:47 2019 -0500

    gfs2: fix race between gfs2_freeze_func and unmount

    [ Upstream commit 8f91821990fd6f170a5dca79697a441181a41b16 ]

    As part of the freeze operation, gfs2_freeze_func() is left blocking
    on a request to hold the sd_freeze_gl in SH. This glock is held in EX
    by the gfs2_freeze() code.

    A subsequent call to gfs2_unfreeze() releases the EXclusively held
    sd_freeze_gl, which allows gfs2_freeze_func() to acquire it in SH and
    resume its operation.

    gfs2_unfreeze(), however, doesn't wait for gfs2_freeze_func() to complete.
```

```
    If a umount is issued right after unfreeze, it could result in an
    inconsistent filesystem because some journal data (statfs update) isn't
    written out.

    Refer to commit 24972557b12c for a more detailed explanation of how
    freeze/unfreeze work.

    This patch causes gfs2_unfreeze() to wait for gfs2_freeze_func() to
    complete before returning to the user.

    Signed-off-by: Abhi Das <adas@redhat.com>
    Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dab2c7d98f377202c66f948bffe4be637246758a
Author: Roberto Bergantinos Corpas <rbergant@redhat.com>
Date:   Thu Apr 25 15:36:51 2019 +0200

    NFS: make nfs_match_client killable

    [ Upstream commit 950a578c6128c2886e295b9c7ecb0b6b22fcc92b ]

        Actually we don't do anything with return value from
        nfs_wait_client_init_complete in nfs_match_client, as a
        consequence if we get a fatal signal and client is not
        fully initialised, we'll loop to "again" label

        This has been proven to cause soft lockups on some scenarios
        (no-carrier but configured network interfaces)

    Signed-off-by: Roberto Bergantinos Corpas <rbergant@redhat.com>
    Reviewed-by: Benjamin Coddington <bcodding@redhat.com>
    Signed-off-by: Anna Schumaker <Anna.Schumaker@Netapp.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 86edf5af379343f5b82085d924437d48e774e40e
Author: David Howells <dhowells@redhat.com>
Date:   Wed May 1 15:58:24 2019 +0100

    afs: Fix getting the afs.fid xattr

    [ Upstream commit a2f611a3dc317d8ea1c98ad6c54b911cf7f93193 ]

    The AFS3 FID is three 32-bit unsigned numbers and is represented as three
    up-to-8-hex-digit numbers separated by colons to the afs.fid xattr.
    However, with the advent of support for YFS, the FID is now a 64-bit volume
    number, a 96-bit vnode/inode number and a 32-bit uniquifier (as before).
    Whilst the sprintf in afs_xattr_get_fid() has been partially updated (it
    currently ignores the upper 32 bits of the 96-bit vnode number), the size
    of the stack-based buffer has not been increased to match, thereby allowing
    stack corruption to occur.

    Fix this by increasing the buffer size appropriately and conditionally
    including the upper part of the vnode number if it is non-zero.  The latter
    requires the lower part to be zero-padded if the upper part is non-zero.

    Fixes: 3b6492df4153 ("afs: Increase to 64-bit volume ID and 96-bit vnode ID for YFS")
    Signed-off-by: David Howells <dhowells@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 49842a9e1f393c2fb585920e9ef701f057d32b9d
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Mon May 6 23:57:54 2019 +0800

    cxgb4: Fix error path in cxgb4_init_module
```

```
    [ Upstream commit a3147770bea76c8dbad73eca3a24c2118da5e719 ]

    BUG: unable to handle kernel paging request at ffffffffa016a270
    PGD 3270067 P4D 3270067 PUD 3271063 PMD 230bbd067 PTE 0
    Oops: 0000 [#1
    CPU: 0 PID: 6134 Comm: modprobe Not tainted 5.1.0+ #33
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.9.3-0-ge2fc41e-
prebuilt.qemu-project.org 04/01/2014
    RIP: 0010:atomic_notifier_chain_register+0x24/0x60
    Code: 1f 80 00 00 00 00 55 48 89 e5 41 54 49 89 f4 53 48 89 fb e8 ae b4 38 01 48 8b 53
38 48 8d 4b 38 48 85 d2 74 20 45 8b 44 24 10 <44> 3b 42 10 7e 08 eb 13 44 39 42 10 7c 0d 48
8d 4a 08 48 8b 52 08
    RSP: 0018:ffffc90000e2bc60 EFLAGS: 00010086
    RAX: 0000000000000292 RBX: ffffffff83467240 RCX: ffffffff83467278
    RDX: ffffffffa016a260 RSI: ffffffff83752140 RDI: ffffffff83467240
    RBP: ffffc90000e2bc70 R08: 0000000000000000 R09: 0000000000000001
    R10: 0000000000000000 R11: 00000000014fa61f R12: ffffffffa01c8260
    R13: ffff888231091e00 R14: 0000000000000000 R15: ffffc90000e2be78
    FS:  00007fbd8d7cd540(0000) GS:ffff888237a00000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: ffffffffa016a270 CR3: 000000022c7e3000 CR4: 00000000000006f0
    Call Trace:
     register_inet6addr_notifier+0x13/0x20
     cxgb4_init_module+0x6c/0x1000 [cxgb4
     ? 0xffffffffa01d7000
     do_one_initcall+0x6c/0x3cc
     ? do_init_module+0x22/0x1f1
     ? rcu_read_lock_sched_held+0x97/0xb0
     ? kmem_cache_alloc_trace+0x325/0x3b0
     do_init_module+0x5b/0x1f1
     load_module+0x1db1/0x2690
     ? m_show+0x1d0/0x1d0
     __do_sys_finit_module+0xc5/0xd0
     __x64_sys_finit_module+0x15/0x20
     do_syscall_64+0x6b/0x1d0
     entry_SYSCALL_64_after_hwframe+0x49/0xbe

    If pci_register_driver fails, register inet6addr_notifier is
    pointless. This patch fix the error path in cxgb4_init_module.

    Fixes: b5a02f503caa ("cxgb4 : Update ipv6 address handling api")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dcbdbe13796ffa561510d2686a6318564e728bbd
Author: Ross Lagerwall <ross.lagerwall@citrix.com>
Date:   Wed Mar 27 17:09:17 2019 +0000

    gfs2: Fix lru_count going negative

    [ Upstream commit 7881ef3f33bb80f459ea6020d1e021fc524a6348 ]

    Under certain conditions, lru_count may drop below zero resulting in
    a large amount of log spam like this:

    vmscan: shrink_slab: gfs2_dump_glock+0x3b0/0x630 [gfs2] \
        negative objects to delete nr=-1

    This happens as follows:
    1) A glock is moved from lru_list to the dispose list and lru_count is
       decremented.
    2) The dispose function calls cond_resched() and drops the lru lock.
    3) Another thread takes the lru lock and tries to add the same glock to
```

```
       lru_list, checking if the glock is on an lru list.
    4) It is on a list (actually the dispose list) and so it avoids
       incrementing lru_count.
    5) The glock is moved to lru_list.
    5) The original thread doesn't dispose it because it has been re-added
       to the lru list but the lru_count has still decreased by one.

    Fix by checking if the LRU flag is set on the glock rather than checking
    if the glock is on some list and rearrange the code so that the LRU flag
    is added/removed precisely when the glock is added/removed from lru_list.

    Signed-off-by: Ross Lagerwall <ross.lagerwall@citrix.com>
    Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 93d91ef889548c3527846fb469b4ec6d8ffa4ad9
Author: David Sterba <dsterba@suse.com>
Date:   Wed May 29 19:25:47 2019 +0200

    Revert "btrfs: Honour FITRIM range constraints during free space trim"

    This reverts commit eb432217d775a90c061681c0dfa3c7abfba75123.

    There is currently no corresponding patch in master due to additional
    changes that would be significantly different from plain revert in the
    respective stable branch.

    The range argument was not handled correctly and could cause trim to
    overlap allocated areas or reach beyond the end of the device. The
    address space that fitrim normally operates on is in logical
    coordinates, while the discards are done on the physical device extents.
    This distinction cannot be made with the current ioctl interface and
    caused the confusion.

    The bug depends on the layout of block groups and does not always
    happen. The whole-fs trim (run by default by the fstrim tool) is not
    affected.

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

commit dc9ddd15e921697e05d37af38752091a90f6c5b3
Author: Kristian Evensen <kristian.evensen@gmail.com>
Date:   Fri May 3 17:40:07 2019 +0200

    netfilter: ctnetlink: Resolve conntrack L3-protocol flush regression

    commit f8e608982022fad035160870f5b06086d3cba54d upstream.

    Commit 59c08c69c278 ("netfilter: ctnetlink: Support L3 protocol-filter
    on flush") introduced a user-space regression when flushing connection
    track entries. Before this commit, the nfgen_family field was not used
    by the kernel and all entries were removed. Since this commit,
    nfgen_family is used to filter out entries that should not be removed.
    One example a broken tool is conntrack. conntrack always sets
    nfgen_family to AF_INET, so after 59c08c69c278 only IPv4 entries were
    removed with the -F parameter.

    Pablo Neira Ayuso suggested using nfgenmsg->version to resolve the
    regression, and this commit implements his suggestion. nfgenmsg->version
    is so far set to zero, so it is well-suited to be used as a flag for
    selecting old or new flush behavior. If version is 0, nfgen_family is
    ignored and all entries are used. If user-space sets the version to one
    (or any other value than 0), then the new behavior is used. As version
    only can have two valid values, I chose not to add a new
```

```
    NFNETLINK_VERSION-constant.

    Fixes: 59c08c69c278 ("netfilter: ctnetlink: Support L3 protocol-filter on flush")
    Reported-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Suggested-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Kristian Evensen <kristian.evensen@gmail.com>
    Tested-by: Nicolas Dichtel <nicolas.dichtel@6wind.com>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a71be911c15a7365aeb3a74a9d5086a483ce5a9f
Author: Al Viro <viro@zeniv.linux.org.uk>
Date:   Thu Apr 4 21:04:13 2019 -0400

    acct_on(): don't mess with freeze protection

    commit 9419a3191dcb27f24478d288abaab697228d28e6 upstream.

    What happens there is that we are replacing file->path.mnt of
    a file we'd just opened with a clone and we need the write
    count contribution to be transferred from original mount to
    new one.  That's it.  We do *NOT* want any kind of freeze
    protection for the duration of switchover.

    IOW, we should just use __mnt_{want,drop}_write() for that
    switchover; no need to bother with mnt_{want,drop}_write()
    there.

    Tested-by: Amir Goldstein <amir73il@gmail.com>
    Reported-by: syzbot+2a73a6ea9507b7112141@syzkaller.appspotmail.com
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f3919070a3aadb907c2de6c6dda71f60aaf313a0
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Mon Apr 8 11:45:29 2019 +0800

    at76c50x-usb: Don't register led_trigger if usb_register_driver failed

    commit 09ac2694b0475f96be895848687ebcbba97eeecf upstream.

    Syzkaller report this:

    [ 1213.468581] BUG: unable to handle kernel paging request at fffffbfff83bf338
    [ 1213.469530] #PF error: [normal kernel read fault]
    [ 1213.469530] PGD 237fe4067 P4D 237fe4067 PUD 237e60067 PMD 1c868b067 PTE 0
    [ 1213.473514] Oops: 0000 [#1] SMP KASAN PTI
    [ 1213.473514] CPU: 0 PID: 6321 Comm: syz-executor.0 Tainted: G         C        5.1.0-
rc3+ #8
    [ 1213.473514] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-
1ubuntu1 04/01/2014
    [ 1213.473514] RIP: 0010:strcmp+0x31/0xa0
    [ 1213.473514] Code: 00 00 00 00 fc ff df 55 53 48 83 ec 08 eb 0a 84 db 48 89 ef 74 5a
4c 89 e6 48 89 f8 48 89 fa 48 8d 6f 01 48 c1 e8 03 83 e2 07 <42> 0f b6 04 28 38 d0 7f 04 84
c0 75 50 48 89 f0 48 89 f2 0f b6 5d
    [ 1213.473514] RSP: 0018:ffff8881f2b7f950 EFLAGS: 00010246
    [ 1213.473514] RAX: 1ffffffff83bf338 RBX: ffff8881ea6f7240 RCX: ffffffff825350c6
    [ 1213.473514] RDX: 0000000000000000 RSI: ffffffffc1ee19c0 RDI: ffffffffc1df99c0
    [ 1213.473514] RBP: ffffffffc1df99c1 R08: 0000000000000001 R09: 0000000000000004
    [ 1213.473514] R10: 0000000000000000 R11: ffff8881de353f00 R12: ffff8881ee727900
    [ 1213.473514] R13: dfffffc0000000000 R14: 0000000000000001 R15: ffffffffc1eeaaf0
    [ 1213.473514] FS:  00007fa66fa01700(0000) GS:ffff8881f7200000(0000)
knlGS:0000000000000000
    [ 1213.473514] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [ 1213.473514] CR2: fffffbfff83bf338 CR3: 00000001ebb9e005 CR4: 00000000007606f0
```

```
    [ 1213.473514] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [ 1213.473514] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [ 1213.473514] PKRU: 55555554
    [ 1213.473514] Call Trace:
    [ 1213.473514]  led_trigger_register+0x112/0x3f0
    [ 1213.473514]  led_trigger_register_simple+0x7a/0x110
    [ 1213.473514]  ? 0xffffffffc1c10000
    [ 1213.473514]  at76_mod_init+0x77/0x1000 [at76c50x_usb]
    [ 1213.473514]  do_one_initcall+0xbc/0x47d
    [ 1213.473514]  ? perf_trace_initcall_level+0x3a0/0x3a0
    [ 1213.473514]  ? kasan_unpoison_shadow+0x30/0x40
    [ 1213.473514]  ? kasan_unpoison_shadow+0x30/0x40
    [ 1213.473514]  do_init_module+0x1b5/0x547
    [ 1213.473514]  load_module+0x6405/0x8c10
    [ 1213.473514]  ? module_frob_arch_sections+0x20/0x20
    [ 1213.473514]  ? kernel_read_file+0x1e6/0x5d0
    [ 1213.473514]  ? find_held_lock+0x32/0x1c0
    [ 1213.473514]  ? cap_capable+0x1ae/0x210
    [ 1213.473514]  ? __do_sys_finit_module+0x162/0x190
    [ 1213.473514]  __do_sys_finit_module+0x162/0x190
    [ 1213.473514]  ? __ia32_sys_init_module+0xa0/0xa0
    [ 1213.473514]  ? __mutex_unlock_slowpath+0xdc/0x690
    [ 1213.473514]  ? wait_for_completion+0x370/0x370
    [ 1213.473514]  ? vfs_write+0x204/0x4a0
    [ 1213.473514]  ? do_syscall_64+0x18/0x450
    [ 1213.473514]  do_syscall_64+0x9f/0x450
    [ 1213.473514]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
    [ 1213.473514] RIP: 0033:0x462e99
    [ 1213.473514] Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89
f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48
c7 c1 bc ff ff ff f7 d8 64 89 01 48
    [ 1213.473514] RSP: 002b:00007fa66fa00c58 EFLAGS: 00000246 ORIG_RAX: 0000000000000139
    [ 1213.473514] RAX: ffffffffffffffda RBX: 000000000073bf00 RCX: 0000000000462e99
    [ 1213.473514] RDX: 0000000000000000 RSI: 0000000020000300 RDI: 0000000000000003
    [ 1213.473514] RBP: 00007fa66fa00c70 R08: 0000000000000000 R09: 0000000000000000
    [ 1213.473514] R10: 0000000000000000 R11: 0000000000000246 R12: 00007fa66fa016bc
    [ 1213.473514] R13: 00000000004bcefa R14: 00000000006f6fb0 R15: 0000000000000004

    If usb_register failed, no need to call led_trigger_register_simple.

    Reported-by: Hulk Robot <hulkci@huawei.com>
    Fixes: 1264b951463a ("at76c50x-usb: add driver")
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2eec11f7e066d5559ec6b37c029941f23137eb8b
Author: Linus Lüssing <linus.luessing@c0d3.blue>
Date:   Wed Apr 24 03:19:14 2019 +0200

    batman-adv: mcast: fix multicast tt/tvlv worker locking

    commit a3c7cd0cdf1107f891aff847ad481e34df727055 upstream.

    Syzbot has reported some issues with the locking assumptions made for
    the multicast tt/tvlv worker: It was able to trigger the WARN_ON() in
    batadv_mcast_mla_tt_retract() and batadv_mcast_mla_tt_add().
    While hard/not reproduceable for us so far it seems that the
    delayed_work_pending() we use might not be quite safe from reordering.

    Therefore this patch adds an explicit, new spinlock to protect the
    update of the mla_list and flags in bat_priv and then removes the
    WARN_ON(delayed_work_pending()).

    Reported-by: syzbot+83f2d54ec6b7e417e13f@syzkaller.appspotmail.com
```

```
    Reported-by: syzbot+050927a651272b145a5d@syzkaller.appspotmail.com
    Reported-by: syzbot+979ffc89b87309b1b94b@syzkaller.appspotmail.com
    Reported-by: syzbot+f9f3f388440283da2965@syzkaller.appspotmail.com
    Fixes: cbebd363b2e9 ("batman-adv: Use own timer for multicast TT and TVLV updates")
    Signed-off-by: Linus Lüssing <linus.luessing@c0d3.blue>
    Signed-off-by: Sven Eckelmann <sven@narfation.org>
    Signed-off-by: Simon Wunderlich <sw@simonwunderlich.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 45d7cd7cd0d8f994680aaaa7a3c8ab065d0e295d
Author: Eric Dumazet <edumazet@google.com>
Date:   Mon May 13 09:59:16 2019 -0700

    bpf: devmap: fix use-after-free Read in __dev_map_entry_free

    commit 2baae3545327632167c0180e9ca1d467416f1919 upstream.

    synchronize_rcu() is fine when the rcu callbacks only need
    to free memory (kfree_rcu() or direct kfree() call rcu call backs)

    __dev_map_entry_free() is a bit more complex, so we need to make
    sure that call queued __dev_map_entry_free() callbacks have completed.

    sysbot report:

    BUG: KASAN: use-after-free in dev_map_flush_old kernel/bpf/devmap.c:365
    [inline]
    BUG: KASAN: use-after-free in __dev_map_entry_free+0x2a8/0x300
    kernel/bpf/devmap.c:379
    Read of size 8 at addr ffff8801b8da38c8 by task ksoftirqd/1/18

    CPU: 1 PID: 18 Comm: ksoftirqd/1 Not tainted 4.17.0+ #39
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS
    Google 01/01/2011
    Call Trace:
      __dump_stack lib/dump_stack.c:77 [inline]
      dump_stack+0x1b9/0x294 lib/dump_stack.c:113
      print_address_description+0x6c/0x20b mm/kasan/report.c:256
      kasan_report_error mm/kasan/report.c:354 [inline]
      kasan_report.cold.7+0x242/0x2fe mm/kasan/report.c:412
      __asan_report_load8_noabort+0x14/0x20 mm/kasan/report.c:433
      dev_map_flush_old kernel/bpf/devmap.c:365 [inline]
      __dev_map_entry_free+0x2a8/0x300 kernel/bpf/devmap.c:379
      __rcu_reclaim kernel/rcu/rcu.h:178 [inline]
      rcu_do_batch kernel/rcu/tree.c:2558 [inline]
      invoke_rcu_callbacks kernel/rcu/tree.c:2818 [inline]
      __rcu_process_callbacks kernel/rcu/tree.c:2785 [inline]
      rcu_process_callbacks+0xe9d/0x1760 kernel/rcu/tree.c:2802
      __do_softirq+0x2e0/0xaf5 kernel/softirq.c:284
      run_ksoftirqd+0x86/0x100 kernel/softirq.c:645
      smpboot_thread_fn+0x417/0x870 kernel/smpboot.c:164
      kthread+0x345/0x410 kernel/kthread.c:240
      ret_from_fork+0x3a/0x50 arch/x86/entry/entry_64.S:412

    Allocated by task 6675:
      save_stack+0x43/0xd0 mm/kasan/kasan.c:448
      set_track mm/kasan/kasan.c:460 [inline]
      kasan_kmalloc+0xc4/0xe0 mm/kasan/kasan.c:553
      kmem_cache_alloc_trace+0x152/0x780 mm/slab.c:3620
      kmalloc include/linux/slab.h:513 [inline]
      kzalloc include/linux/slab.h:706 [inline]
      dev_map_alloc+0x208/0x7f0 kernel/bpf/devmap.c:102
      find_and_alloc_map kernel/bpf/syscall.c:129 [inline]
      map_create+0x393/0x1010 kernel/bpf/syscall.c:453
      __do_sys_bpf kernel/bpf/syscall.c:2351 [inline]
```

```
        __se_sys_bpf kernel/bpf/syscall.c:2328 [inline]
        __x64_sys_bpf+0x303/0x510 kernel/bpf/syscall.c:2328
        do_syscall_64+0x1b1/0x800 arch/x86/entry/common.c:290
        entry_SYSCALL_64_after_hwframe+0x49/0xbe

      Freed by task 26:
        save_stack+0x43/0xd0 mm/kasan/kasan.c:448
        set_track mm/kasan/kasan.c:460 [inline]
        __kasan_slab_free+0x11a/0x170 mm/kasan/kasan.c:521
        kasan_slab_free+0xe/0x10 mm/kasan/kasan.c:528
        __cache_free mm/slab.c:3498 [inline]
        kfree+0xd9/0x260 mm/slab.c:3813
        dev_map_free+0x4fa/0x670 kernel/bpf/devmap.c:191
        bpf_map_free_deferred+0xba/0xf0 kernel/bpf/syscall.c:262
        process_one_work+0xc64/0x1b70 kernel/workqueue.c:2153
        worker_thread+0x181/0x13a0 kernel/workqueue.c:2296
        kthread+0x345/0x410 kernel/kthread.c:240
        ret_from_fork+0x3a/0x50 arch/x86/entry/entry_64.S:412

      The buggy address belongs to the object at ffff8801b8da37c0
        which belongs to the cache kmalloc-512 of size 512
      The buggy address is located 264 bytes inside of
        512-byte region [ffff8801b8da37c0, ffff8801b8da39c0)
      The buggy address belongs to the page:
      page:ffffea0006e368c0 count:1 mapcount:0 mapping:ffff8801da800940
      index:0xffff8801b8da3540
      flags: 0x2fffc0000000100(slab)
      raw: 02fffc0000000100 ffffea0007217b88 ffffea0006e30cc8 ffff8801da800940
      raw: ffff8801b8da3540 ffff8801b8da3040 0000000100000004 0000000000000000
      page dumped because: kasan: bad access detected

      Memory state around the buggy address:
        ffff8801b8da3780: fc fc fc fc fc fc fc fc fb fb fb fb fb fb fb fb
        ffff8801b8da3800: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
      > ffff8801b8da3880: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
                                                  ^
        ffff8801b8da3900: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
        ffff8801b8da3980: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc

      Fixes: 546ac1ffb70d ("bpf: add devmap, a map for storing net device references")
      Signed-off-by: Eric Dumazet <edumazet@google.com>
      Reported-by: syzbot+457d3e2ffbcf31aee5c0@syzkaller.appspotmail.com
      Acked-by: Toke Høiland-Jørgensen <toke@redhat.com>
      Acked-by: Jesper Dangaard Brouer <brouer@redhat.com>
      Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 3b924789e9f88e4f748aabb701216966eb836967
  Author: YueHaibing <yuehaibing@huawei.com>
  Date:   Wed Mar 6 19:56:58 2019 +0800

      ssb: Fix possible NULL pointer dereference in ssb_host_pcmcia_exit

      commit b2c01aab9646ed8ffb7c549afe55d5349c482425 upstream.

      Syzkaller report this:

      kasan: GPF could be caused by NULL-ptr deref or user memory access
      general protection fault: 0000 [#1] SMP KASAN PTI
      CPU: 0 PID: 4492 Comm: syz-executor.0 Not tainted 5.0.0-rc7+ #45
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014
      RIP: 0010:sysfs_remove_file_ns+0x27/0x70 fs/sysfs/file.c:468
      Code: 00 00 00 41 54 55 48 89 fd 53 49 89 d4 48 89 f3 e8 ee 76 9c ff 48 8d 7d 30 48 b8
  00 00 00 00 00 fc ff df 48 89 fa 48 c1 ea 03 <80> 3c 02 00 75 2d 48 89 da 48 b8 00 00 00 00
  00 fc ff df 48 8b 6d
```

```
     RSP: 0018:ffff8881e9d9fc00 EFLAGS: 00010206
     RAX: dfffc0000000000 RBX: ffffffff900367e0 RCX: ffffffff81a95952
     RDX: 0000000000000006 RSI: ffffc90001405000 RDI: 0000000000000030
     RBP: 0000000000000000 R08: fffffbfff1fa22ed R09: fffffbfff1fa22ed
     R10: 0000000000000001 R11: fffffbfff1fa22ec R12: 0000000000000000
     R13: ffffffffc1abdac0 R14: 1ffff1103d3b3f8b R15: 0000000000000000
     FS:  00007fe409dc1700(0000) GS:ffff8881f1200000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     CR2: 0000001b2d721000 CR3: 00000001e98b6005 CR4: 00000000007606f0
     DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     PKRU: 55555554
     Call Trace:
      sysfs_remove_file include/linux/sysfs.h:519 [inline]
      driver_remove_file+0x40/0x50 drivers/base/driver.c:122
      pcmcia_remove_newid_file drivers/pcmcia/ds.c:163 [inline]
      pcmcia_unregister_driver+0x7d/0x2b0 drivers/pcmcia/ds.c:209
      ssb_modexit+0xa/0x1b [ssb]
      __do_sys_delete_module kernel/module.c:1018 [inline]
      __se_sys_delete_module kernel/module.c:961 [inline]
      __x64_sys_delete_module+0x3dc/0x5e0 kernel/module.c:961
      do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
     RIP: 0033:0x462e99
     Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48
  89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff
  ff f7 d8 64 89 01 48
     RSP: 002b:00007fe409dc0c58 EFLAGS: 00000246 ORIG_RAX: 00000000000000b0
     RAX: ffffffffffffffda RBX: 000000000073bf00 RCX: 0000000000462e99
     RDX: 0000000000000000 RSI: 0000000000000000 RDI: 00000000200000c0
     RBP: 0000000000000002 R08: 0000000000000000 R09: 0000000000000000
     R10: 0000000000000000 R11: 0000000000000246 R12: 00007fe409dc16bc
     R13: 00000000004bccaa R14: 00000000006f6bc8 R15: 00000000ffffffff
     Modules linked in: ssb(-) 3c59x nvme_core macvlan tap pata_hpt3x3 rt2x00pci null_blk
  tsc40 pm_notifier_error_inject notifier_error_inject mdio cdc_wdm nf_reject_ipv4
  ath9k_common ath9k_hw ath pppox ppp_generic slhc ehci_platform wl12xx wlcore tps6507x_ts
  ioc4 nf_synproxy_core ide_gd_mod ax25 can_dev iwlwifi can_raw atm tm2_touchkey can_gw can
  sundance adp5588_keys rt2800mmio rt2800lib rt2x00mmio rt2x00lib eeprom_93cx6 pn533 lru_cache
  elants_i2c ip_set nfnetlink gameport tipc hampshire nhc_ipv6 nhc_hop nhc_udp nhc_fragment
  nhc_routing nhc_mobility nhc_dest 6lowpan silead brcmutil nfc mt76_usb mt76 mac80211
  iptable_security iptable_raw iptable_mangle iptable_nat nf_nat_ipv4 nf_nat nf_conntrack
  nf_defrag_ipv6 nf_defrag_ipv4 iptable_filter bpfilter ip6_vti ip_gre sit hsr veth vxcan
  batman_adv cfg80211 rfkill chnl_net caif nlmon vcan bridge stp llc ip6_gre ip6_tunnel
  tunnel6 tun joydev mousedev serio_raw ide_pci_generic piix floppy ide_core sch_fq_codel
  ip_tables x_tables ipv6
      [last unloaded: 3c59x]
     Dumping ftrace buffer:
        (ftrace buffer empty)
     ---[ end trace 3913cbf8011e1c05 ]---

     In ssb_modinit, it does not fail SSB init when ssb_host_pcmcia_init failed,
     however in ssb_modexit, ssb_host_pcmcia_exit calls pcmcia_unregister_driver
     unconditionally, which may tigger a NULL pointer dereference issue as above.

     Reported-by: Hulk Robot <hulkci@huawei.com>
     Fixes: 399500da18f7 ("ssb: pick PCMCIA host code support from b43 driver")
     Signed-off-by: YueHaibing <yuehaibing@huawei.com>
     Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit b82e0f501b390ea6c30629d026289e314a3eb342
  Author: Alexander Potapenko <glider@google.com>
  Date:   Thu Apr 4 10:56:46 2019 -0400

     media: vivid: use vfree() instead of kfree() for dev->bitmap_cap
```

```
    commit dad7e270ba712ba1c99cd2d91018af6044447a06 upstream.

    syzkaller reported crashes on kfree() called from
    vivid_vid_cap_s_selection(). This looks like a simple typo, as
    dev->bitmap_cap is allocated with vzalloc() throughout the file.

    Fixes: ef834f7836ec0 ("[media] vivid: add the video capture and output
    parts")

    Signed-off-by: Alexander Potapenko <glider@google.com>
    Reported-by: Syzbot <syzbot+6c0effb5877f6b0344e2@syzkaller.appspotmail.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit da757ae6d328841bd34a57a454b99c6ee725bc4a
Author: Hans Verkuil <hverkuil@xs4all.nl>
Date:   Mon Nov 19 06:09:00 2018 -0500

    media: vb2: add waiting_in_dqbuf flag

    commit d65842f7126aa1a87fb44b7c9980c12630ed4f33 upstream.

    Calling VIDIOC_DQBUF can release the core serialization lock pointed to
    by vb2_queue->lock if it has to wait for a new buffer to arrive.

    However, if userspace dup()ped the video device filehandle, then it is
    possible to read or call DQBUF from two filehandles at the same time.

    It is also possible to call REQBUFS from one filehandle while the other
    is waiting for a buffer. This will remove all the buffers and reallocate
    new ones. Removing all the buffers isn't the problem here (that's already
    handled correctly by DQBUF), but the reallocating part is: DQBUF isn't
    aware that the buffers have changed.

    This is fixed by setting a flag whenever the lock is released while waiting
    for a buffer to arrive. And checking the flag where needed so we can return
    -EBUSY.

    Signed-off-by: Hans Verkuil <hverkuil@xs4all.nl>
    Reported-by: Syzbot <syzbot+4180ff9ca6810b06c1e9@syzkaller.appspotmail.com>
    Reviewed-by: Tomasz Figa <tfiga@chromium.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0270d8b8d8ba03f5a7bfc3a3b89b0c9a7d38fe15
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Tue Mar 5 00:40:26 2019 -0500

    media: serial_ir: Fix use-after-free in serial_ir_init_module

    commit 56cd26b618855c9af48c8301aa6754ced8dd0beb upstream.

    Syzkaller report this:

    BUG: KASAN: use-after-free in sysfs_remove_file_ns+0x5f/0x70 fs/sysfs/file.c:468
    Read of size 8 at addr ffff8881dc7ae030 by task syz-executor.0/6249

    CPU: 1 PID: 6249 Comm: syz-executor.0 Not tainted 5.0.0-rc8+ #3
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014
    Call Trace:
     __dump_stack lib/dump_stack.c:77 [inline]
     dump_stack+0xfa/0x1ce lib/dump_stack.c:113
```

```
      print_address_description+0x65/0x270 mm/kasan/report.c:187
      kasan_report+0x149/0x18d mm/kasan/report.c:317
      ? 0xffffffffc1728000
      sysfs_remove_file_ns+0x5f/0x70 fs/sysfs/file.c:468
      sysfs_remove_file include/linux/sysfs.h:519 [inline]
      driver_remove_file+0x40/0x50 drivers/base/driver.c:122
      remove_bind_files drivers/base/bus.c:585 [inline]
      bus_remove_driver+0x186/0x220 drivers/base/bus.c:725
      driver_unregister+0x6c/0xa0 drivers/base/driver.c:197
      serial_ir_init_module+0x169/0x1000 [serial_ir]
      do_one_initcall+0xfa/0x5ca init/main.c:887
      do_init_module+0x204/0x5f6 kernel/module.c:3460
      load_module+0x66b2/0x8570 kernel/module.c:3808
      __do_sys_finit_module+0x238/0x2a0 kernel/module.c:3902
      do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
     RIP: 0033:0x462e99
     Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48
  89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff
  ff f7 d8 64 89 01 48
     RSP: 002b:00007f9450132c58 EFLAGS: 00000246 ORIG_RAX: 0000000000000139
     RAX: ffffffffffffffda RBX: 000000000073bf00 RCX: 0000000000462e99
     RDX: 0000000000000000 RSI: 0000000020000100 RDI: 0000000000000003
     RBP: 00007f9450132c70 R08: 0000000000000000 R09: 0000000000000000
     R10: 0000000000000000 R11: 0000000000000246 R12: 00007f94501336bc
     R13: 00000000004bcefa R14: 00000000006f6fb0 R15: 0000000000000004

     Allocated by task 6249:
      set_track mm/kasan/common.c:85 [inline]
      __kasan_kmalloc.constprop.3+0xa0/0xd0 mm/kasan/common.c:495
      kmalloc include/linux/slab.h:545 [inline]
      kzalloc include/linux/slab.h:740 [inline]
      bus_add_driver+0xc0/0x610 drivers/base/bus.c:651
      driver_register+0x1bb/0x3f0 drivers/base/driver.c:170
      serial_ir_init_module+0xe8/0x1000 [serial_ir]
      do_one_initcall+0xfa/0x5ca init/main.c:887
      do_init_module+0x204/0x5f6 kernel/module.c:3460
      load_module+0x66b2/0x8570 kernel/module.c:3808
      __do_sys_finit_module+0x238/0x2a0 kernel/module.c:3902
      do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

     Freed by task 6249:
      set_track mm/kasan/common.c:85 [inline]
      __kasan_slab_free+0x130/0x180 mm/kasan/common.c:457
      slab_free_hook mm/slub.c:1430 [inline]
      slab_free_freelist_hook mm/slub.c:1457 [inline]
      slab_free mm/slub.c:3005 [inline]
      kfree+0xe1/0x270 mm/slub.c:3957
      kobject_cleanup lib/kobject.c:662 [inline]
      kobject_release lib/kobject.c:691 [inline]
      kref_put include/linux/kref.h:67 [inline]
      kobject_put+0x146/0x240 lib/kobject.c:708
      bus_remove_driver+0x10e/0x220 drivers/base/bus.c:732
      driver_unregister+0x6c/0xa0 drivers/base/driver.c:197
      serial_ir_init_module+0x14c/0x1000 [serial_ir]
      do_one_initcall+0xfa/0x5ca init/main.c:887
      do_init_module+0x204/0x5f6 kernel/module.c:3460
      load_module+0x66b2/0x8570 kernel/module.c:3808
      __do_sys_finit_module+0x238/0x2a0 kernel/module.c:3902
      do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
      entry_SYSCALL_64_after_hwframe+0x49/0xbe

     The buggy address belongs to the object at ffff8881dc7ae000
      which belongs to the cache kmalloc-256 of size 256
```

```
    The buggy address is located 48 bytes inside of
     256-byte region [ffff8881dc7ae000, ffff8881dc7ae100)
    The buggy address belongs to the page:
    page:ffffea000771eb80 count:1 mapcount:0 mapping:ffff8881f6c02e00 index:0x0
    flags: 0x2fffc0000000200(slab)
    raw: 02fffc0000000200 ffffea0007d14800 0000000400000002 ffff8881f6c02e00
    raw: 0000000000000000 00000000800c000c 00000001ffffffff 0000000000000000
    page dumped because: kasan: bad access detected

    Memory state around the buggy address:
     ffff8881dc7adf00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
     ffff8881dc7adf80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    >ffff8881dc7ae000: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
                                  ^
     ffff8881dc7ae080: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     ffff8881dc7ae100: fc fc fc fc fc fc fc fc 00 00 00 00 00 00 00 00

    There are already cleanup handlings in serial_ir_init error path,
    no need to call serial_ir_exit do it again in serial_ir_init_module,
    otherwise will trigger a use-after-free issue.

    Fixes: fa5dc29c1fcc ("[media] lirc_serial: move out of staging and rename to serial_ir")

    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8c103d2ba1c69a3ef35dc807de14bc3ab29f0029
Author: YueHaibing <yuehaibing@huawei.com>
Date:   Wed Mar 6 07:45:08 2019 -0500

    media: cpia2: Fix use-after-free in cpia2_exit

    commit dea37a97265588da604c6ba80160a287b72c7bfd upstream.

    Syzkaller report this:

    BUG: KASAN: use-after-free in sysfs_remove_file_ns+0x5f/0x70 fs/sysfs/file.c:468
    Read of size 8 at addr ffff8881f59a6b70 by task syz-executor.0/8363

    CPU: 0 PID: 8363 Comm: syz-executor.0 Not tainted 5.0.0-rc8+ #3
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014
    Call Trace:
     __dump_stack lib/dump_stack.c:77 [inline]
     dump_stack+0xfa/0x1ce lib/dump_stack.c:113
     print_address_description+0x65/0x270 mm/kasan/report.c:187
     kasan_report+0x149/0x18d mm/kasan/report.c:317
     sysfs_remove_file_ns+0x5f/0x70 fs/sysfs/file.c:468
     sysfs_remove_file include/linux/sysfs.h:519 [inline]
     driver_remove_file+0x40/0x50 drivers/base/driver.c:122
     usb_remove_newid_files drivers/usb/core/driver.c:212 [inline]
     usb_deregister+0x12a/0x3b0 drivers/usb/core/driver.c:1005
     cpia2_exit+0xa/0x16 [cpia2]
     __do_sys_delete_module kernel/module.c:1018 [inline]
     __se_sys_delete_module kernel/module.c:961 [inline]
     __x64_sys_delete_module+0x3dc/0x5e0 kernel/module.c:961
     do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
     entry_SYSCALL_64_after_hwframe+0x49/0xbe
    RIP: 0033:0x462e99
    Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48
 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff
 ff f7 d8 64 89 01 48
    RSP: 002b:00007f86f3754c58 EFLAGS: 00000246 ORIG_RAX: 00000000000000b0
```

```
RAX: ffffffffffffffda RBX: 000000000073bf00 RCX: 0000000000462e99
RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000020000300
RBP: 0000000000000002 R08: 0000000000000000 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000246 R12: 00007f86f37556bc
R13: 00000000004bcca9 R14: 00000000006f6b48 R15: 00000000ffffffff

Allocated by task 8363:
 set_track mm/kasan/common.c:85 [inline]
 __kasan_kmalloc.constprop.3+0xa0/0xd0 mm/kasan/common.c:495
 kmalloc include/linux/slab.h:545 [inline]
 kzalloc include/linux/slab.h:740 [inline]
 bus_add_driver+0xc0/0x610 drivers/base/bus.c:651
 driver_register+0x1bb/0x3f0 drivers/base/driver.c:170
 usb_register_driver+0x267/0x520 drivers/usb/core/driver.c:965
 0xffffffffc1b4817c
 do_one_initcall+0xfa/0x5ca init/main.c:887
 do_init_module+0x204/0x5f6 kernel/module.c:3460
 load_module+0x66b2/0x8570 kernel/module.c:3808
 __do_sys_finit_module+0x238/0x2a0 kernel/module.c:3902
 do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
 entry_SYSCALL_64_after_hwframe+0x49/0xbe

Freed by task 8363:
 set_track mm/kasan/common.c:85 [inline]
 __kasan_slab_free+0x130/0x180 mm/kasan/common.c:457
 slab_free_hook mm/slub.c:1430 [inline]
 slab_free_freelist_hook mm/slub.c:1457 [inline]
 slab_free mm/slub.c:3005 [inline]
 kfree+0xe1/0x270 mm/slub.c:3957
 kobject_cleanup lib/kobject.c:662 [inline]
 kobject_release lib/kobject.c:691 [inline]
 kref_put include/linux/kref.h:67 [inline]
 kobject_put+0x146/0x240 lib/kobject.c:708
 bus_remove_driver+0x10e/0x220 drivers/base/bus.c:732
 driver_unregister+0x6c/0xa0 drivers/base/driver.c:197
 usb_register_driver+0x341/0x520 drivers/usb/core/driver.c:980
 0xffffffffc1b4817c
 do_one_initcall+0xfa/0x5ca init/main.c:887
 do_init_module+0x204/0x5f6 kernel/module.c:3460
 load_module+0x66b2/0x8570 kernel/module.c:3808
 __do_sys_finit_module+0x238/0x2a0 kernel/module.c:3902
 do_syscall_64+0x147/0x600 arch/x86/entry/common.c:290
 entry_SYSCALL_64_after_hwframe+0x49/0xbe

The buggy address belongs to the object at ffff8881f59a6b40
 which belongs to the cache kmalloc-256 of size 256
The buggy address is located 48 bytes inside of
 256-byte region [ffff8881f59a6b40, ffff8881f59a6c40)
The buggy address belongs to the page:
page:ffffea0007d66980 count:1 mapcount:0 mapping:ffff8881f6c02e00 index:0x0
flags: 0x2fffc0000000200(slab)
raw: 02fffc0000000200 dead000000000100 dead000000000200 ffff8881f6c02e00
raw: 0000000000000000 00000000800c000c 00000001ffffffff 0000000000000000
page dumped because: kasan: bad access detected

Memory state around the buggy address:
 ffff8881f59a6a00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 ffff8881f59a6a80: 00 00 00 00 00 00 00 00 00 00 fc fc fc fc fc fc
>ffff8881f59a6b00: fc fc fc fc fc fc fc fc fb fb fb fb fb fb fb fb
                                                              ^
 ffff8881f59a6b80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
 ffff8881f59a6c00: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc

cpia2_init does not check return value of cpia2_init, if it failed
in usb_register_driver, there is already cleanup using driver_unregister.
```

```
    No need call cpia2_usb_cleanup on module exit.

    Reported-by: Hulk Robot <hulkci@huawei.com>
    Signed-off-by: YueHaibing <yuehaibing@huawei.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+samsung@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e630b9ac9dce61e080d1cea80819f99c67892a48
Author: Jiufei Xue <jiufei.xue@linux.alibaba.com>
Date:   Thu Apr 11 19:25:12 2019 +0200

    fbdev: fix WARNING in __alloc_pages_nodemask bug

    commit 8c40292be9169a9cbe19aadd1a6fc60cbd1af82f upstream.

    Syzkaller hit 'WARNING in __alloc_pages_nodemask' bug.

    WARNING: CPU: 1 PID: 1473 at mm/page_alloc.c:4377
    __alloc_pages_nodemask+0x4da/0x2130
    Kernel panic - not syncing: panic_on_warn set ...

    Call Trace:
     alloc_pages_current+0xb1/0x1e0
     kmalloc_order+0x1f/0x60
     kmalloc_order_trace+0x1d/0x120
     fb_alloc_cmap_gfp+0x85/0x2b0
     fb_set_user_cmap+0xff/0x370
     do_fb_ioctl+0x949/0xa20
     fb_ioctl+0xdd/0x120
     do_vfs_ioctl+0x186/0x1070
     ksys_ioctl+0x89/0xa0
     __x64_sys_ioctl+0x74/0xb0
     do_syscall_64+0xc8/0x550
     entry_SYSCALL_64_after_hwframe+0x49/0xbe

    This is a warning about order >= MAX_ORDER and the order is from
    userspace ioctl. Add flag __NOWARN to silence this warning.

    Signed-off-by: Jiufei Xue <jiufei.xue@linux.alibaba.com>
    Signed-off-by: Bartlomiej Zolnierkiewicz <b.zolnierkie@samsung.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ebe6000e90edd5ee4ad8e88f811f42dd800e17f9
Author: Amir Goldstein <amir73il@gmail.com>
Date:   Thu Mar 28 17:38:29 2019 +0200

    ovl: relax WARN_ON() for overlapping layers use case

    commit acf3062a7e1ccf67c6f7e7c28671a6708fde63b0 upstream.

    This nasty little syzbot repro:
    https://syzkaller.appspot.com/x/repro.syz?x=12c7a94f400000

    Creates overlay mounts where the same directory is both in upper and lower
    layers. Simplified example:

      mkdir foo work
      mount -t overlay none foo -o"lowerdir=.,upperdir=foo,workdir=work"

    The repro runs several threads in parallel that attempt to chdir into foo
    and attempt to symlink/rename/exec/mkdir the file bar.

    The repro hits a WARN_ON() I placed in ovl_instantiate(), which suggests
    that an overlay inode already exists in cache and is hashed by the pointer
```

```
    of the real upper dentry that ovl_create_real() has just created. At the
    point of the WARN_ON(), for overlay dir inode lock is held and upper dir
    inode lock, so at first, I did not see how this was possible.

    On a closer look, I see that after ovl_create_real(), because of the
    overlapping upper and lower layers, a lookup by another thread can find the
    file foo/bar that was just created in upper layer, at overlay path
    foo/foo/bar and hash the an overlay inode with the new real dentry as lower
    dentry. This is possible because the overlay directory foo/foo is not
    locked and the upper dentry foo/bar is in dcache, so ovl_lookup() can find
    it without taking upper dir inode shared lock.

    Overlapping layers is considered a wrong setup which would result in
    unexpected behavior, but it shouldn't crash the kernel and it shouldn't
    trigger WARN_ON() either, so relax this WARN_ON() and leave a pr_warn()
    instead to cover all cases of failure to get an overlay inode.

    The error returned from failure to insert new inode to cache with
    inode_insert5() was changed to -EEXIST, to distinguish from the error
    -ENOMEM returned on failure to get/allocate inode with iget5_locked().

    Reported-by: syzbot+9c69c282adc4edd2b540@syzkaller.appspotmail.com
    Fixes: 01b39dcc9568 ("ovl: use inode_insert5() to hash a newly...")
    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 2f71a4e3efa216ed4a3579373fe65beec3e8853d
Author: Will Deacon <will.deacon@arm.com>
Date:   Mon Apr 29 13:03:57 2019 +0100

    arm64: errata: Add workaround for Cortex-A76 erratum #1463225

    commit 969f5ea627570e91c9d54403287ee3ed657f58fe upstream.

    Revisions of the Cortex-A76 CPU prior to r4p0 are affected by an erratum
    that can prevent interrupts from being taken when single-stepping.

    This patch implements a software workaround to prevent userspace from
    effectively being able to disable interrupts.

    Cc: <stable@vger.kernel.org>
    Cc: Marc Zyngier <marc.zyngier@arm.com>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4aeac8596745be15ff3f5bd98e3489e2a03575ec
Author: Shile Zhang <shile.zhang@linux.alibaba.com>
Date:   Mon Apr 1 17:47:00 2019 +0200

    fbdev: fix divide error in fb_var_to_videomode

    commit cf84807f6dd0be5214378e66460cfc9187f532f9 upstream.

    To fix following divide-by-zero error found by Syzkaller:

      divide error: 0000 [#1] SMP PTI
      CPU: 7 PID: 8447 Comm: test Kdump: loaded Not tainted 4.19.24-8.al7.x86_64 #1
      Hardware name: Alibaba Cloud Alibaba Cloud ECS, BIOS rel-1.12.0-0-ga698c8995f-
prebuilt.qemu.org 04/01/2014
      RIP: 0010:fb_var_to_videomode+0xae/0xc0
      Code: 04 44 03 46 78 03 4e 7c 44 03 46 68 03 4e 70 89 ce d1 ee 69 c0 e8 03 00 00 f6 c2
01 0f 45 ce 83 e2 02 8d 34 09 0f 45 ce 31 d2 <41> f7 f0 31 d2 f7 f1 89 47 08 f3 c3 66 0f 1f
44 00 00 0f 1f 44 00
```

```
       RSP: 0018:ffffb7e189347bf0 EFLAGS: 00010246
       RAX: 00000000e1692410 RBX: ffffb7e189347d60 RCX: 0000000000000000
       RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffffb7e189347c10
       RBP: ffff99972a091c00 R08: 0000000000000000 R09: 0000000000000000
       R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000100
       R13: 0000000000010000 R14: 00007ffd66baf6d0 R15: 0000000000000000
       FS:  00007f2054d11740(0000) GS:ffff99972fbc0000(0000) knlGS:0000000000000000
       CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
       CR2: 00007f205481fd20 CR3: 00000004288a0001 CR4: 00000000001606a0
       Call Trace:
        fb_set_var+0x257/0x390
        ? lookup_fast+0xbb/0x2b0
        ? fb_open+0xc0/0x140
        ? chrdev_open+0xa6/0x1a0
        do_fb_ioctl+0x445/0x5a0
        do_vfs_ioctl+0x92/0x5f0
        ? __alloc_fd+0x3d/0x160
        ksys_ioctl+0x60/0x90
        __x64_sys_ioctl+0x16/0x20
        do_syscall_64+0x5b/0x190
        entry_SYSCALL_64_after_hwframe+0x44/0xa9
       RIP: 0033:0x7f20548258d7
       Code: 44 00 00 48 8b 05 b9 15 2d 00 64 c7 00 26 00 00 00 48 c7 c0 ff ff ff ff c3 66 2e
 0f 1f 84 00 00 00 00 00 b8 10 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 89 15 2d
 00 f7 d8 64 89 01 48

    It can be triggered easily with following test code:

      #include <linux/fb.h>
      #include <fcntl.h>
      #include <sys/ioctl.h>
      int main(void)
      {
              struct fb_var_screeninfo var = {.activate = 0x100, .pixclock = 60};
              int fd = open("/dev/fb0", O_RDWR);
              if (fd < 0)
                      return 1;

              if (ioctl(fd, FBIOPUT_VSCREENINFO, &var))
                      return 1;

              return 0;
      }

    Signed-off-by: Shile Zhang <shile.zhang@linux.alibaba.com>
    Cc: Fredrik Noring <noring@nocrew.org>
    Cc: Daniel Vetter <daniel.vetter@ffwll.ch>
    Reviewed-by: Mukesh Ojha <mojha@codeaurora.org>
    Signed-off-by: Bartlomiej Zolnierkiewicz <b.zolnierkie@samsung.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 53804824ba927ecd4b32766ef5f34636880f29a5
 Author: Tobin C. Harding <tobin@kernel.org>
 Date:   Mon May 13 13:39:12 2019 +1000

    btrfs: sysfs: don't leak memory when failing add fsid

    commit e32773357d5cc271b1d23550b3ed026eb5c2a468 upstream.

    A failed call to kobject_init_and_add() must be followed by a call to
    kobject_put().  Currently in the error path when adding fs_devices we
    are missing this call.  This could be fixed by calling
    btrfs_sysfs_remove_fsid() if btrfs_sysfs_add_fsid() returns an error or
    by adding a call to kobject_put() directly in btrfs_sysfs_add_fsid().
    Here we choose the second option because it prevents the slightly
```

```
    unusual error path handling requirements of kobject from leaking out
    into btrfs functions.

    Add a call to kobject_put() in the error path of kobject_add_and_init().
    This causes the release method to be called if kobject_init_and_add()
    fails.  open_tree() is the function that calls btrfs_sysfs_add_fsid()
    and the error code in this function is already written with the
    assumption that the release method is called during the error path of
    open_tree() (as seen by the call to btrfs_sysfs_remove_fsid() under the
    fail_fsdev_sysfs label).

    Cc: stable@vger.kernel.org # v4.4+
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Tobin C. Harding <tobin@kernel.org>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2a48467bfdfa0a08ff8731241f54d59a4440748
Author: Tobin C. Harding <tobin@kernel.org>
Date:   Mon May 13 13:39:11 2019 +1000

    btrfs: sysfs: Fix error path kobject memory leak

    commit 450ff8348808a89cc27436771aa05c2b90c0eef1 upstream.

    If a call to kobject_init_and_add() fails we must call kobject_put()
    otherwise we leak memory.

    Calling kobject_put() when kobject_init_and_add() fails drops the
    refcount back to 0 and calls the ktype release method (which in turn
    calls the percpu destroy and kfree).

    Add call to kobject_put() in the error path of call to
    kobject_init_and_add().

    Cc: stable@vger.kernel.org # v4.4+
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Tobin C. Harding <tobin@kernel.org>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ffd658adf684aa47b73229ab12b9adc8f4dc18f6
Author: Filipe Manana <fdmanana@suse.com>
Date:   Mon May 6 16:44:02 2019 +0100

    Btrfs: fix race between ranged fsync and writeback of adjacent ranges

    commit 0c713cbab6200b0ab6473b50435e450a6e1de85d upstream.

    When we do a full fsync (the bit BTRFS_INODE_NEEDS_FULL_SYNC is set in the
    inode) that happens to be ranged, which happens during a msync() or writes
    for files opened with O_SYNC for example, we can end up with a corrupt log,
    due to different file extent items representing ranges that overlap with
    each other, or hit some assertion failures.

    When doing a ranged fsync we only flush delalloc and wait for ordered
    exents within that range. If while we are logging items from our inode
    ordered extents for adjacent ranges complete, we end up in a race that can
    make us insert the file extent items that overlap with others we logged
    previously and the assertion failures.

    For example, if tree-log.c:copy_items() receives a leaf that has the
    following file extents items, all with a length of 4K and therefore there
```

```
    is an implicit hole in the range 68K to 72K - 1:

      (257 EXTENT_ITEM 64K), (257 EXTENT_ITEM 72K), (257 EXTENT_ITEM 76K), ...

    It copies them to the log tree. However due to the need to detect implicit
    holes, it may release the path, in order to look at the previous leaf to
    detect an implicit hole, and then later it will search again in the tree
    for the first file extent item key, with the goal of locking again the
    leaf (which might have changed due to concurrent changes to other inodes).

    However when it locks again the leaf containing the first key, the key
    corresponding to the extent at offset 72K may not be there anymore since
    there is an ordered extent for that range that is finishing (that is,
    somewhere in the middle of btrfs_finish_ordered_io()), and it just
    removed the file extent item but has not yet replaced it with a new file
    extent item, so the part of copy_items() that does hole detection will
    decide that there is a hole in the range starting from 68K to 76K - 1,
    and therefore insert a file extent item to represent that hole, having
    a key offset of 68K. After that we now have a log tree with 2 different
    extent items that have overlapping ranges:

     1) The file extent item copied before copy_items() released the path,
        which has a key offset of 72K and a length of 4K, representing the
        file range 72K to 76K - 1.

     2) And a file extent item representing a hole that has a key offset of
        68K and a length of 8K, representing the range 68K to 76K - 1. This
        item was inserted after releasing the path, and overlaps with the
        extent item inserted before.

    The overlapping extent items can cause all sorts of unpredictable and
    incorrect behaviour, either when replayed or if a fast (non full) fsync
    happens later, which can trigger a BUG_ON() when calling
    btrfs_set_item_key_safe() through __btrfs_drop_extents(), producing a
    trace like the following:

      [61666.783269] ------------[ cut here ]------------
      [61666.783943] kernel BUG at fs/btrfs/ctree.c:3182!
      [61666.784644] invalid opcode: 0000 [#1] PREEMPT SMP
      (...)
      [61666.786253] task: ffff880117b88c40 task.stack: ffffc90008168000
      [61666.786253] RIP: 0010:btrfs_set_item_key_safe+0x7c/0xd2 [btrfs]
      [61666.786253] RSP: 0018:ffffc9000816b958 EFLAGS: 00010246
      [61666.786253] RAX: 0000000000000000 RBX: 000000000000000f RCX: 0000000000030000
      [61666.786253] RDX: 0000000000000000 RSI: ffffc9000816ba4f RDI: ffffc9000816b937
      [61666.786253] RBP: ffffc9000816b998 R08: ffff88011dae2428 R09: 0000000000001000
      [61666.786253] R10: 0000160000000000 R11: 6db6db6db6db6db7 R12: ffff88011dae2418
      [61666.786253] R13: ffffc9000816ba4f R14: ffff8801e10c4118 R15: ffff8801e715c000
      [61666.786253] FS:  00007f6060a18700(0000) GS:ffff88023f5c0000(0000)
knlGS:0000000000000000
      [61666.786253] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      [61666.786253] CR2: 00007f6060a28000 CR3: 0000000213e69000 CR4: 00000000000006e0
      [61666.786253] Call Trace:
      [61666.786253]  __btrfs_drop_extents+0x5e3/0xaad [btrfs]
      [61666.786253]  ? time_hardirqs_on+0x9/0x14
      [61666.786253]  btrfs_log_changed_extents+0x294/0x4e0 [btrfs]
      [61666.786253]  ? release_extent_buffer+0x38/0xb4 [btrfs]
      [61666.786253]  btrfs_log_inode+0xb6e/0xcdc [btrfs]
      [61666.786253]  ? lock_acquire+0x131/0x1c5
      [61666.786253]  ? btrfs_log_inode_parent+0xee/0x659 [btrfs]
      [61666.786253]  ? arch_local_irq_save+0x9/0xc
      [61666.786253]  ? btrfs_log_inode_parent+0x1f5/0x659 [btrfs]
      [61666.786253]  btrfs_log_inode_parent+0x223/0x659 [btrfs]
      [61666.786253]  ? arch_local_irq_save+0x9/0xc
      [61666.786253]  ? lockref_get_not_zero+0x2c/0x34
```

```
  [61666.786253]  ? rcu_read_unlock+0x3e/0x5d
  [61666.786253]  btrfs_log_dentry_safe+0x60/0x7b [btrfs]
  [61666.786253]  btrfs_sync_file+0x317/0x42c [btrfs]
  [61666.786253]  vfs_fsync_range+0x8c/0x9e
  [61666.786253]  SyS_msync+0x13c/0x1c9
  [61666.786253]  entry_SYSCALL_64_fastpath+0x18/0xad
```

A sample of a corrupt log tree leaf with overlapping extents I got from running btrfs/072:

```
      item 14 key (295 108 200704) itemoff 2599 itemsize 53
              extent data disk bytenr 0 nr 0
              extent data offset 0 nr 458752 ram 458752
      item 15 key (295 108 659456) itemoff 2546 itemsize 53
              extent data disk bytenr 4343541760 nr 770048
              extent data offset 606208 nr 163840 ram 770048
      item 16 key (295 108 663552) itemoff 2493 itemsize 53
              extent data disk bytenr 4343541760 nr 770048
              extent data offset 610304 nr 155648 ram 770048
      item 17 key (295 108 819200) itemoff 2440 itemsize 53
              extent data disk bytenr 4334788608 nr 4096
              extent data offset 0 nr 4096 ram 4096
```

The file extent item at offset 659456 (item 15) ends at offset 823296 (659456 + 163840) while the next file extent item (item 16) starts at offset 663552.

Another different problem that the race can trigger is a failure in the assertions at tree-log.c:copy_items(), which expect that the first file extent item key we found before releasing the path exists after we have released path and that the last key we found before releasing the path also exists after releasing the path:

```
  $ cat -n fs/btrfs/tree-log.c
  4080          if (need_find_last_extent) {
  4081                  /* btrfs_prev_leaf could return 1 without releasing the path */
  4082                  btrfs_release_path(src_path);
  4083                  ret = btrfs_search_slot(NULL, inode->root, &first_key,
  4084                                          src_path, 0, 0);
  4085                  if (ret < 0)
  4086                          return ret;
  4087                  ASSERT(ret == 0);
  (...)
  4103                  if (i >= btrfs_header_nritems(src_path->nodes[0])) {
  4104                          ret = btrfs_next_leaf(inode->root, src_path);
  4105                          if (ret < 0)
  4106                                  return ret;
  4107                          ASSERT(ret == 0);
  4108                          src = src_path->nodes[0];
  4109                          i = 0;
  4110                          need_find_last_extent = true;
  4111                  }
  (...)
```

The second assertion implicitly expects that the last key before the path release still exists, because the surrounding while loop only stops after we have found that key. When this assertion fails it produces a stack like this:

```
  [139590.037075] assertion failed: ret == 0, file: fs/btrfs/tree-log.c, line: 4107
  [139590.037406] ------------[ cut here ]------------
  [139590.037707] kernel BUG at fs/btrfs/ctree.h:3546!
  [139590.038034] invalid opcode: 0000 [#1] SMP DEBUG_PAGEALLOC PTI
  [139590.038340] CPU: 1 PID: 31841 Comm: fsstress Tainted: G        W         5.0.0-
btrfs-next-46 #1
```

```
       (...)
       [139590.039354] RIP: 0010:assfail.constprop.24+0x18/0x1a [btrfs]
       (...)
       [139590.040397] RSP: 0018:ffffa27f48f2b9b0 EFLAGS: 00010282
       [139590.040730] RAX: 0000000000000041 RBX: ffff897c635d92c8 RCX: 0000000000000000
       [139590.041105] RDX: 0000000000000000 RSI: ffff897d36a96868 RDI: ffff897d36a96868
       [139590.041470] RBP: ffff897d1b9a0708 R08: 0000000000000000 R09: 0000000000000000
       [139590.041815] R10: 0000000000000008 R11: 0000000000000000 R12: 0000000000000013
       [139590.042159] R13: 0000000000000227 R14: ffff897cffcbba88 R15: 0000000000000001
       [139590.042501] FS:  00007f2efc8dee80(0000) GS:ffff897d36a80000(0000)
 knlGS:0000000000000000
       [139590.042847] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
       [139590.043199] CR2: 00007f8c064935e0 CR3: 0000000232252002 CR4: 00000000003606e0
       [139590.043547] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
       [139590.043899] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
       [139590.044250] Call Trace:
       [139590.044631]  copy_items+0xa3f/0x1000 [btrfs]
       [139590.045009]  ? generic_bin_search.constprop.32+0x61/0x200 [btrfs]
       [139590.045396]  btrfs_log_inode+0x7b3/0xd70 [btrfs]
       [139590.045773]  btrfs_log_inode_parent+0x2b3/0xce0 [btrfs]
       [139590.046143]  ? do_raw_spin_unlock+0x49/0xc0
       [139590.046510]  btrfs_log_dentry_safe+0x4a/0x70 [btrfs]
       [139590.046872]  btrfs_sync_file+0x3b6/0x440 [btrfs]
       [139590.047243]  btrfs_file_write_iter+0x45b/0x5c0 [btrfs]
       [139590.047592]  __vfs_write+0x129/0x1c0
       [139590.047932]  vfs_write+0xc2/0x1b0
       [139590.048270]  ksys_write+0x55/0xc0
       [139590.048608]  do_syscall_64+0x60/0x1b0
       [139590.048946]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
       [139590.049287] RIP: 0033:0x7f2efc4be190
       (...)
       [139590.050342] RSP: 002b:00007ffe743243a8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
       [139590.050701] RAX: ffffffffffffffda RBX: 0000000000008d58 RCX: 00007f2efc4be190
       [139590.051067] RDX: 0000000000008d58 RSI: 00005567eca0f370 RDI: 0000000000000003
       [139590.051459] RBP: 0000000000000024 R08: 0000000000000003 R09: 0000000000008d60
       [139590.051863] R10: 0000000000000078 R11: 0000000000000246 R12: 0000000000000003
       [139590.052252] R13: 00000000003d3507 R14: 00005567eca0f370 R15: 0000000000000000
       (...)
       [139590.055128] ---[ end trace 193f35d0215cdeeb ]---

    So fix this race between a full ranged fsync and writeback of adjacent
    ranges by flushing all delalloc and waiting for all ordered extents to
    complete before logging the inode. This is the simplest way to solve the
    problem because currently the full fsync path does not deal with ranges
    at all (it assumes a full range from 0 to LLONG_MAX) and it always needs
    to look at adjacent ranges for hole detection. For use cases of ranged
    fsyncs this can make a few fsyncs slower but on the other hand it can
    make some following fsyncs to other ranges do less work or no need to do
    anything at all. A full fsync is rare anyway and happens only once after
    loading/creating an inode and once after less common operations such as a
    shrinking truncate.

    This is an issue that exists for a long time, and was often triggered by
    generic/127, because it does mmap'ed writes and msync (which triggers a
    ranged fsync). Adding support for the tree checker to detect overlapping
    extents (next patch in the series) and trigger a WARN() when such cases
    are found, and then calling btrfs_check_leaf_full() at the end of
    btrfs_insert_file_extent() made the issue much easier to detect. Running
    btrfs/072 with that change to the tree checker and making fsstress open
    files always with O_SYNC made it much easier to trigger the issue (as
    triggering it with generic/127 is very rare).

    CC: stable@vger.kernel.org # 3.16+
    Reviewed-by: Josef Bacik <josef@toxicpanda.com>
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
```

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

commit fb4bdda0f904dc87e51e51383d7351a887c87959
Author: Filipe Manana <fdmanana@suse.com>
Date:   Mon May 6 16:43:51 2019 +0100

    Btrfs: avoid fallback to transaction commit during fsync of files with holes

    commit ebb929060aeb162417b4c1307e63daee47b208d9 upstream.

    When we are doing a full fsync (bit BTRFS_INODE_NEEDS_FULL_SYNC set) of a
    file that has holes and has file extent items spanning two or more leafs,
    we can end up falling to back to a full transaction commit due to a logic
    bug that leads to failure to insert a duplicate file extent item that is
    meant to represent a hole between the last file extent item of a leaf and
    the first file extent item in the next leaf. The failure (EEXIST error)
    leads to a transaction commit (as most errors when logging an inode do).

    For example, we have the two following leafs:

    Leaf N:

      ------------------------------------------------
      | ..., ..., ..., (257, FILE_EXTENT_ITEM, 64K) |
      ------------------------------------------------
      The file extent item at the end of leaf N has a length of 4Kb,
      representing the file range from 64K to 68K - 1.

    Leaf N + 1:

      ------------------------------------------------
      | (257, FILE_EXTENT_ITEM, 72K), ..., ..., ... |
      ------------------------------------------------
      The file extent item at the first slot of leaf N + 1 has a length of
      4Kb too, representing the file range from 72K to 76K - 1.

    During the full fsync path, when we are at tree-log.c:copy_items() with
    leaf N as a parameter, after processing the last file extent item, that
    represents the extent at offset 64K, we take a look at the first file
    extent item at the next leaf (leaf N + 1), and notice there's a 4K hole
    between the two extents, and therefore we insert a file extent item
    representing that hole, starting at file offset 68K and ending at offset
    72K - 1. However we don't update the value of *last_extent, which is used
    to represent the end offset (plus 1, non-inclusive end) of the last file
    extent item inserted in the log, so it stays with a value of 68K and not
    with a value of 72K.

    Then, when copy_items() is called for leaf N + 1, because the value of
    *last_extent is smaller then the offset of the first extent item in the
    leaf (68K < 72K), we look at the last file extent item in the previous
    leaf (leaf N) and see it there's a 4K gap between it and our first file
    extent item (again, 68K < 72K), so we decide to insert a file extent item
    representing the hole, starting at file offset 68K and ending at offset
    72K - 1, this insertion will fail with -EEXIST being returned from
    btrfs_insert_file_extent() because we already inserted a file extent item
    representing a hole for this offset (68K) in the previous call to
    copy_items(), when processing leaf N.

    The -EEXIST error gets propagated to the fsync callback, btrfs_sync_file(),
    which falls back to a full transaction commit.

    Fix this by adjusting *last_extent after inserting a hole when we had to
    look at the next leaf.
```

```
    Fixes: 4ee3fad34a9c ("Btrfs: fix fsync after hole punching when using no-holes feature")
    Cc: stable@vger.kernel.org # 4.14+
    Reviewed-by: Josef Bacik <josef@toxicpanda.com>
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit be69efb38e07a69d809b4ab698a8f477c78b1e2a
Author: Filipe Manana <fdmanana@suse.com>
Date:   Mon Apr 29 13:08:14 2019 +0100

    Btrfs: do not abort transaction at btrfs_update_root() after failure to COW path

    commit 72bd2323ec87722c115a5906bc6a1b31d11e8f54 upstream.

    Currently when we fail to COW a path at btrfs_update_root() we end up
    always aborting the transaction. However all the current callers of
    btrfs_update_root() are able to deal with errors returned from it, many do
    end up aborting the transaction themselves (directly or not, such as the
    transaction commit path), other BUG_ON() or just gracefully cancel whatever
    they were doing.

    When syncing the fsync log, we call btrfs_update_root() through
    tree-log.c:update_log_root(), and if it returns an -ENOSPC error, the log
    sync code does not abort the transaction, instead it gracefully handles
    the error and returns -EAGAIN to the fsync handler, so that it falls back
    to a transaction commit. Any other error different from -ENOSPC, makes the
    log sync code abort the transaction.

    So remove the transaction abort from btrfs_update_log() when we fail to
    COW a path to update the root item, so that if an -ENOSPC failure happens
    we avoid aborting the current transaction and have a chance of the fsync
    succeeding after falling back to a transaction commit.

    Bugzilla: https://bugzilla.kernel.org/show_bug.cgi?id=203413
    Fixes: 79787eaab46121 ("btrfs: replace many BUG_ONs with proper error handling")
    Cc: stable@vger.kernel.org # 4.4+
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
    Reviewed-by: Anand Jain <anand.jain@oracle.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 69bb50795ac899971b27c4c150948e2326e98c9f
Author: Johnny Chang <johnnyc@synology.com>
Date:   Fri Apr 26 11:01:05 2019 +0800

    btrfs: Check the compression level before getting a workspace

    commit 2b90883c561ddcc641741c2e4df1f702a4f2acb8 upstream.

    When a file's compression property is set as zlib or zstd but leave
    the compression mount option not be set, that means btrfs will try
    to compress the file with default compression level. But in
    btrfs_compress_pages(), it calls get_workspace() with level = 0.
    This will return a workspace with a wrong compression level.
    For zlib, the compression level in the workspace will be 0
    (that means "store only"). And for zstd, the compression in the
    workspace will be 1, not the default level 3.

    How to reproduce:
      mkfs -t btrfs /dev/sdb
      mount /dev/sdb /mnt/
      mkdir /mnt/zlib
      btrfs property set /mnt/zlib/ compression zlib
      dd if=/dev/zero of=/mnt/zlib/compression-friendly-file-10M bs=1M count=10
```

```
       sync
       btrfs-debugfs -f /mnt/zlib/compression-friendly-file-10M

     btrfs-debugfs output:
     * before:
       ...
       (258 9961472): ram 524288 disk 1106247680 disk_size 524288
       file: ... extents 20 disk size 10485760 logical size 10485760 ratio 1.00

     * after:
      ...
      (258 10354688): ram 131072 disk 14217216 disk_size 4096
      file: ... extents 80 disk size 327680 logical size 10485760 ratio 32.00

     The steps for zstd are similar, but need to put a debugging message to
     show the level of the return workspace in zstd_get_workspace().

     This commit adds a check of the compression level before getting a
     workspace by set_level().

     CC: stable@vger.kernel.org # 5.1+
     Signed-off-by: Johnny Chang <johnnyc@synology.com>
     Reviewed-by: David Sterba <dsterba@suse.com>
     Signed-off-by: David Sterba <dsterba@suse.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 38bf3e22d058202f7fe8121845a0b304c0d691f6
 Author: Josef Bacik <josef@toxicpanda.com>
 Date:   Fri May 3 11:10:06 2019 -0400

     btrfs: don't double unlock on error in btrfs_punch_hole

     commit 8fca955057b9c58467d1b231e43f19c4cf26ae8c upstream.

     If we have an error writing out a delalloc range in
     btrfs_punch_hole_lock_range we'll unlock the inode and then goto
     out_only_mutex, where we will again unlock the inode.  This is bad,
     don't do this.

     Fixes: f27451f22996 ("Btrfs: add support for fallocate's zero range operation")
     CC: stable@vger.kernel.org # 4.19+
     Reviewed-by: Filipe Manana <fdmanana@suse.com>
     Signed-off-by: Josef Bacik <josef@toxicpanda.com>
     Reviewed-by: David Sterba <dsterba@suse.com>
     Signed-off-by: David Sterba <dsterba@suse.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 873aac4c3d7a2fd7e4b76e5142235cbc474de4d7
 Author: Andreas Gruenbacher <agruenba@redhat.com>
 Date:   Fri May 17 19:18:43 2019 +0100

     gfs2: Fix sign extension bug in gfs2_update_stats

     commit 5a5ec83d6ac974b12085cd99b196795f14079037 upstream.

     Commit 4d207133e9c3 changed the types of the statistic values in struct
     gfs2_lkstats from s64 to u64.  Because of that, what should be a signed
     value in gfs2_update_stats turned into an unsigned value.  When shifted
     right, we end up with a large positive value instead of a small negative
     value, which results in an incorrect variance estimate.

     Fixes: 4d207133e9c3 ("gfs2: Make statistics unsigned, suitable for use with do_div()")
     Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>
     Cc: stable@vger.kernel.org # v4.4+
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit b232deedaa600d39bdd869f5fb046c0ae3afdaff
Author: Christoph Hellwig <hch@lst.de>
Date:   Tue Apr 30 06:51:50 2019 -0400

    arm64/iommu: handle non-remapped addresses in ->mmap and ->get_sgtable

    commit a98d9ae937d256ed679a935fc82d9deaa710d98e upstream.

    DMA allocations that can't sleep may return non-remapped addresses, but
    we do not properly handle them in the mmap and get_sgtable methods.
    Resolve non-vmalloc addresses using virt_to_page to handle this corner
    case.

    Cc: <stable@vger.kernel.org>
    Acked-by: Catalin Marinas <catalin.marinas@arm.com>
    Reviewed-by: Robin Murphy <robin.murphy@arm.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4101ec81b182c10fae26f06e995e32a628825cc0
Author: Will Deacon <will.deacon@arm.com>
Date:   Fri May 24 14:15:34 2019 +0100

    arm64: Kconfig: Make ARM64_PSEUDO_NMI depend on BROKEN for now

    commit 96a13f57b946be7a6c10405e4bd780c0b6b6fe63 upstream.

    Although we merged support for pseudo-nmi using interrupt priority
    masking in 5.1, we've since uncovered a number of non-trivial issues
    with the implementation. Although there are patches pending to address
    these problems, we're facing issues that prevent us from merging them at
    this current time:

      https://lkml.kernel.org/r/1556553607-46531-1-git-send-email-julien.thierry@arm.com

    For now, simply mark this optional feature as BROKEN in the hope that we
    can fix things properly in the near future.

    Cc: <stable@vger.kernel.org> # 5.1
    Cc: Julien Thierry <julien.thierry@arm.com>
    Acked-by: Marc Zyngier <marc.zyngier@arm.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2c2104890d31e3bd52ac4fd661d6dc9026238b0c
Author: Ard Biesheuvel <ard.biesheuvel@arm.com>
Date:   Thu May 23 10:17:37 2019 +0100

    arm64/kernel: kaslr: reduce module randomization range to 2 GB

    commit b2eed9b58811283d00fa861944cb75797d4e52a7 upstream.

    The following commit

      7290d5809571 ("module: use relative references for __ksymtab entries")

    updated the ksymtab handling of some KASLR capable architectures
    so that ksymtab entries are emitted as pairs of 32-bit relative
    references. This reduces the size of the entries, but more
    importantly, it gets rid of statically assigned absolute
    addresses, which require fixing up at boot time if the kernel
    is self relocating (which takes a 24 byte RELA entry for each
    member of the ksymtab struct).
```

```
    Since ksymtab entries are always part of the same module as the
    symbol they export, it was assumed at the time that a 32-bit
    relative reference is always sufficient to capture the offset
    between a ksymtab entry and its target symbol.

    Unfortunately, this is not always true: in the case of per-CPU
    variables, a per-CPU variable's base address (which usually differs
    from the actual address of any of its per-CPU copies) is allocated
    in the vicinity of the ..data.percpu section in the core kernel
    (i.e., in the per-CPU reserved region which follows the section
    containing the core kernel's statically allocated per-CPU variables).

    Since we randomize the module space over a 4 GB window covering
    the core kernel (based on the -/+ 4 GB range of an ADRP/ADD pair),
    we may end up putting the core kernel out of the -/+ 2 GB range of
    32-bit relative references of module ksymtab entries that refer to
    per-CPU variables.

    So reduce the module randomization range a bit further. We lose
    1 bit of randomization this way, but this is something we can
    tolerate.

    Cc: <stable@vger.kernel.org> # v4.19+
    Signed-off-by: Ard Biesheuvel <ard.biesheuvel@arm.com>
    Signed-off-by: Will Deacon <will.deacon@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e9e27bfce65c3cf463b1d0554221c67d1b869328
Author: Dan Williams <dan.j.williams@intel.com>
Date:   Thu May 16 17:05:21 2019 -0700

    libnvdimm/pmem: Bypass CONFIG_HARDENED_USERCOPY overhead

    commit 52f476a323f9efc959be1c890d0cdcf12e1582e0 upstream.

    Jeff discovered that performance improves from ~375K iops to ~519K iops
    on a simple psync-write fio workload when moving the location of 'struct
    page' from the default PMEM location to DRAM. This result is surprising
    because the expectation is that 'struct page' for dax is only needed for
    third party references to dax mappings. For example, a dax-mapped buffer
    passed to another system call for direct-I/O requires 'struct page' for
    sending the request down the driver stack and pinning the page. There is
    no usage of 'struct page' for first party access to a file via
    read(2)/write(2) and friends.

    However, this "no page needed" expectation is violated by
    CONFIG_HARDENED_USERCOPY and the check_copy_size() performed in
    copy_from_iter_full_nocache() and copy_to_iter_mcsafe(). The
    check_heap_object() helper routine assumes the buffer is backed by a
    slab allocator (DRAM) page and applies some checks.  Those checks are
    invalid, dax pages do not originate from the slab, and redundant,
    dax_iomap_actor() has already validated that the I/O is within bounds.
    Specifically that routine validates that the logical file offset is
    within bounds of the file, then it does a sector-to-pfn translation
    which validates that the physical mapping is within bounds of the block
    device.

    Bypass additional hardened usercopy overhead and call the 'no check'
    versions of the copy_{to,from}_iter operations directly.

    Fixes: 0aed55af8834 ("x86, uaccess: introduce copy_from_iter_flushcache...")
    Cc: <stable@vger.kernel.org>
    Cc: Jeff Moyer <jmoyer@redhat.com>
    Cc: Ingo Molnar <mingo@redhat.com>
```

```
    Cc: Christoph Hellwig <hch@lst.de>
    Cc: Al Viro <viro@zeniv.linux.org.uk>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Matthew Wilcox <willy@infradead.org>
    Reported-and-tested-by: Jeff Smits <jeff.smits@intel.com>
    Acked-by: Kees Cook <keescook@chromium.org>
    Acked-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Dan Williams <dan.j.williams@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e3feb4afa8754129929febcee8798782be97f956
Author: Wanpeng Li <wanpengli@tencent.com>
Date:   Fri May 17 16:49:50 2019 +0800

    KVM: nVMX: Fix using __this_cpu_read() in preemptible context

    commit 541e886f7972cc647804dbb4909189e67987a945 upstream.

     BUG: using __this_cpu_read() in preemptible [00000000] code: qemu-system-x86/4590
      caller is nested_vmx_enter_non_root_mode+0xebd/0x1790 [kvm_intel]
      CPU: 4 PID: 4590 Comm: qemu-system-x86 Tainted: G           OE     5.1.0-rc4+ #1
      Call Trace:
       dump_stack+0x67/0x95
       __this_cpu_preempt_check+0xd2/0xe0
       nested_vmx_enter_non_root_mode+0xebd/0x1790 [kvm_intel]
       nested_vmx_run+0xda/0x2b0 [kvm_intel]
       handle_vmlaunch+0x13/0x20 [kvm_intel]
       vmx_handle_exit+0xbd/0x660 [kvm_intel]
       kvm_arch_vcpu_ioctl_run+0xa2c/0x1e50 [kvm]
       kvm_vcpu_ioctl+0x3ad/0x6d0 [kvm]
       do_vfs_ioctl+0xa5/0x6e0
       ksys_ioctl+0x6d/0x80
       __x64_sys_ioctl+0x1a/0x20
       do_syscall_64+0x6f/0x6c0
       entry_SYSCALL_64_after_hwframe+0x49/0xbe

    Accessing per-cpu variable should disable preemption, this patch extends the
    preemption disable region for __this_cpu_read().

    Cc: Paolo Bonzini <pbonzini@redhat.com>
    Cc: Radim Krčmář <rkrcmar@redhat.com>
    Signed-off-by: Wanpeng Li <wanpengli@tencent.com>
    Fixes: 52017608da33 ("KVM: nVMX: add option to perform early consistency checks via
H/W")
    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 4a4c222ecf3d0e01511b4a9a0ad66f9d48aa6e80
Author: Suthikulpanit, Suravee <Suravee.Suthikulpanit@amd.com>
Date:   Tue May 14 15:49:52 2019 +0000

    kvm: svm/avic: fix off-by-one in checking host APIC ID

    commit c9bcd3e3335d0a29d89fabd2c385e1b989e6f1b0 upstream.

    Current logic does not allow VCPU to be loaded onto CPU with
    APIC ID 255. This should be allowed since the host physical APIC ID
    field in the AVIC Physical APIC table entry is an 8-bit value,
    and APIC ID 255 is valid in system with x2APIC enabled.
    Instead, do not allow VCPU load if the host APIC ID cannot be
    represented by an 8-bit value.

    Also, use the more appropriate AVIC_PHYSICAL_ID_ENTRY_HOST_PHYSICAL_ID_MASK
```

```
    instead of AVIC_MAX_PHYSICAL_ID_COUNT.

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

commit baaee95674c8d7e60120bb2beef6e730f670380f
Author: Peter Xu <peterx@redhat.com>
Date:   Sun May 5 16:56:42 2019 +0800

    kvm: Check irqchip mode before assign irqfd

    commit 654f1f13ea56b92bacade8ce2725aea0457f91c0 upstream.

    When assigning kvm irqfd we didn't check the irqchip mode but we allow
    KVM_IRQFD to succeed with all the irqchip modes.  However it does not
    make much sense to create irqfd even without the kernel chips.  Let's
    provide a arch-dependent helper to check whether a specific irqfd is
    allowed by the arch.  At least for x86, it should make sense to check:

    - when irqchip mode is NONE, all irqfds should be disallowed, and,

    - when irqchip mode is SPLIT, irqfds that are with resamplefd should
      be disallowed.

    For either of the case, previously we'll silently ignore the irq or
    the irq ack event if the irqchip mode is incorrect.  However that can
    cause misterious guest behaviors and it can be hard to triage.  Let's
    fail KVM_IRQFD even earlier to detect these incorrect configurations.

    CC: Paolo Bonzini <pbonzini@redhat.com>
    CC: Radim Krčmář <rkrcmar@redhat.com>
    CC: Alex Williamson <alex.williamson@redhat.com>
    CC: Eduardo Habkost <ehabkost@redhat.com>
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e00303be84b2bbbd1876116d0544d65446deadc2
Author: Dan Williams <dan.j.williams@intel.com>
Date:   Thu May 16 13:26:29 2019 -0700

    dax: Arrange for dax_supported check to span multiple devices

    commit 7bf7eac8d648057519adb6fce1e31458c902212c upstream.

    Pankaj reports that starting with commit ad428cdb525a "dax: Check the
    end of the block-device capacity with dax_direct_access()" device-mapper
    no longer allows dax operation. This results from the stricter checks in
    __bdev_dax_supported() that validate that the start and end of a
    block-device map to the same 'pagemap' instance.

    Teach the dax-core and device-mapper to validate the 'pagemap' on a
    per-target basis. This is accomplished by refactoring the
    bdev_dax_supported() internals into generic_fsdax_supported() which
    takes a sector range to validate. Consequently generic_fsdax_supported()
    is suitable to be used in a device-mapper ->iterate_devices() callback.
    A new ->dax_supported() operation is added to allow composite devices to
    split and route upper-level bdev_dax_supported() requests.

    Fixes: ad428cdb525a ("dax: Check the end of the block-device...")
    Cc: <stable@vger.kernel.org>
    Cc: Ira Weiny <ira.weiny@intel.com>
```

```
      Cc: Dave Jiang <dave.jiang@intel.com>
      Cc: Keith Busch <keith.busch@intel.com>
      Cc: Matthew Wilcox <willy@infradead.org>
      Cc: Vishal Verma <vishal.l.verma@intel.com>
      Cc: Heiko Carstens <heiko.carstens@de.ibm.com>
      Cc: Martin Schwidefsky <schwidefsky@de.ibm.com>
      Reviewed-by: Jan Kara <jack@suse.cz>
      Reported-by: Pankaj Gupta <pagupta@redhat.com>
      Reviewed-by: Pankaj Gupta <pagupta@redhat.com>
      Tested-by: Pankaj Gupta <pagupta@redhat.com>
      Tested-by: Vaibhav Jain <vaibhav@linux.ibm.com>
      Reviewed-by: Mike Snitzer <snitzer@redhat.com>
      Signed-off-by: Dan Williams <dan.j.williams@intel.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 269360f112059c7c0fe6cf1788215c331c3aad55
  Author: Tom Zanussi <tom.zanussi@linux.intel.com>
  Date:   Thu Apr 18 10:18:52 2019 -0500

      tracing: Add a check_val() check before updating cond_snapshot() track_val

      commit 9b2ca371b1505a547217b244f903ad3fb86fa5b4 upstream.

      Without this check a snapshot is taken whenever a bucket's max is hit,
      rather than only when the global max is hit, as it should be.

      Before:

        In this example, we do a first run of the workload (cyclictest),
        examine the output, note the max ('triggering value') (347), then do
        a second run and note the max again.

        In this case, the max in the second run (39) is below the max in the
        first run, but since we haven't cleared the histogram, the first max
        is still in the histogram and is higher than any other max, so it
        should still be the max for the snapshot.  It isn't however - the
        value should still be 347 after the second run.

        # echo 'hist:keys=pid:ts0=common_timestamp.usecs if comm=="cyclictest"' >>
  /sys/kernel/debug/tracing/events/sched/sched_waking/trigger
        # echo
  'hist:keys=next_pid:wakeup_lat=common_timestamp.usecs-$ts0:onmax($wakeup_lat).save(next_prio
  ,next_comm,prev_pid,prev_prio,prev_comm):onmax($wakeup_lat).snapshot() if
  next_comm=="cyclictest"' >> /sys/kernel/debug/tracing/events/sched/sched_switch/trigger

        # cyclictest -p 80 -n -s -t 2 -D 2

        # cat /sys/kernel/debug/tracing/events/sched/sched_switch/hist

        { next_pid:       2143 } hitcount:        199
          max:         44  next_prio:        120  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/4

        { next_pid:       2145 } hitcount:       1325
          max:         38  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/2

        { next_pid:       2144 } hitcount:       1982
          max:        347  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/6

        Snapshot taken (see tracing/snapshot).  Details:
            triggering value { onmax($wakeup_lat) }:        347
            triggered by event with key: { next_pid:       2144 }
```

```
        # cyclictest -p 80 -n -s -t 2 -D 2

        # cat /sys/kernel/debug/tracing/events/sched/sched_switch/hist

        { next_pid:       2143 } hitcount:        199
          max:         44  next_prio:        120  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/4

        { next_pid:       2148 } hitcount:        199
          max:         16  next_prio:        120  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/1

        { next_pid:       2145 } hitcount:       1325
          max:         38  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/2

        { next_pid:       2150 } hitcount:       1326
          max:         39  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/4

        { next_pid:       2144 } hitcount:       1982
          max:        347  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/6

        { next_pid:       2149 } hitcount:       1983
          max:        130  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/0

        Snapshot taken (see tracing/snapshot).  Details:
          triggering value { onmax($wakeup_lat) }:    39
          triggered by event with key: { next_pid:       2150 }

      After:

        In this example, we do a first run of the workload (cyclictest),
        examine the output, note the max ('triggering value') (375), then do
        a second run and note the max again.

        In this case, the max in the second run is still 375, the highest in
        any bucket, as it should be.

        # cyclictest -p 80 -n -s -t 2 -D 2

        # cat /sys/kernel/debug/tracing/events/sched/sched_switch/hist

        { next_pid:       2072 } hitcount:        200
          max:         28  next_prio:        120  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/5

        { next_pid:       2074 } hitcount:       1323
          max:        375  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/2

        { next_pid:       2073 } hitcount:       1980
          max:        153  next_prio:         19  next_comm: cyclictest
          prev_pid:          0  prev_prio:        120  prev_comm: swapper/6

        Snapshot taken (see tracing/snapshot).  Details:
          triggering value { onmax($wakeup_lat) }:        375
          triggered by event with key: { next_pid:       2074 }

        # cyclictest -p 80 -n -s -t 2 -D 2

        # cat /sys/kernel/debug/tracing/events/sched/sched_switch/hist
```

```
       { next_pid:       2101 } hitcount:         199
         max:         49  next_prio:        120  next_comm: cyclictest
         prev_pid:          0  prev_prio:        120  prev_comm: swapper/6

       { next_pid:       2072 } hitcount:         200
         max:         28  next_prio:        120  next_comm: cyclictest
         prev_pid:          0  prev_prio:        120  prev_comm: swapper/5

       { next_pid:       2074 } hitcount:        1323
         max:        375  next_prio:         19  next_comm: cyclictest
         prev_pid:          0  prev_prio:        120  prev_comm: swapper/2

       { next_pid:       2103 } hitcount:        1325
         max:         74  next_prio:         19  next_comm: cyclictest
         prev_pid:          0  prev_prio:        120  prev_comm: swapper/4

       { next_pid:       2073 } hitcount:        1980
         max:        153  next_prio:         19  next_comm: cyclictest
         prev_pid:          0  prev_prio:        120  prev_comm: swapper/6

       { next_pid:       2102 } hitcount:        1981
         max:         84  next_prio:         19  next_comm: cyclictest
         prev_pid:         12  prev_prio:        120  prev_comm: kworker/0:1

       Snapshot taken (see tracing/snapshot).  Details:
         triggering value { onmax($wakeup_lat) }:        375
         triggered by event with key: { next_pid:       2074 }
```

    Link:
http://lkml.kernel.org/r/95958351329f129c07504b4d1769c47a97b70d65.1555597045.git.tom.zanussi
@linux.intel.com

    Cc: stable@vger.kernel.org
    Fixes: a3785b7eca8fd ("tracing: Add hist trigger snapshot() action")
    Signed-off-by: Tom Zanussi <tom.zanussi@linux.intel.com>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit acf49fa4ee4ecec2205c112a0a50ce22c8d2c9ff
Author: Trac Hoang <trac.hoang@broadcom.com>
Date: Thu May 9 10:24:27 2019 -0700

    mmc: sdhci-iproc: Set NO_HISPD bit to fix HS50 data hold time problem

    commit ec0970e0a1b2c807c908d459641a9f9a1be3e130 upstream.

    The iproc host eMMC/SD controller hold time does not meet the
    specification in the HS50 mode. This problem can be mitigated
    by disabling the HISPD bit; thus forcing the controller output
    data to be driven on the falling clock edges rather than the
    rising clock edges.

    Stable tag (v4.12+) chosen to assist stable kernel maintainers so that
    the change does not produce merge conflicts backporting to older kernel
    versions. In reality, the timing bug existed since the driver was first
    introduced but there is no need for this driver to be supported in kernel
    versions that old.

    Cc: stable@vger.kernel.org # v4.12+
    Signed-off-by: Trac Hoang <trac.hoang@broadcom.com>
    Signed-off-by: Scott Branden <scott.branden@broadcom.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

```
commit a0514c0a450ac13078f59ad50e1c65b7748f833a
Author: Trac Hoang <trac.hoang@broadcom.com>
Date:   Thu May 9 10:24:26 2019 -0700

    mmc: sdhci-iproc: cygnus: Set NO_HISPD bit to fix HS50 data hold time problem

    commit b7dfa695afc40d5396ed84b9f25aa3754de23e39 upstream.

    The iproc host eMMC/SD controller hold time does not meet the
    specification in the HS50 mode. This problem can be mitigated
    by disabling the HISPD bit; thus forcing the controller output
    data to be driven on the falling clock edges rather than the
    rising clock edges.

    This change applies only to the Cygnus platform.

    Stable tag (v4.12+) chosen to assist stable kernel maintainers so that
    the change does not produce merge conflicts backporting to older kernel
    versions. In reality, the timing bug existed since the driver was first
    introduced but there is no need for this driver to be supported in kernel
    versions that old.

    Cc: stable@vger.kernel.org # v4.12+
    Signed-off-by: Trac Hoang <trac.hoang@broadcom.com>
    Signed-off-by: Scott Branden <scott.branden@broadcom.com>
    Acked-by: Adrian Hunter <adrian.hunter@intel.com>
    Signed-off-by: Ulf Hansson <ulf.hansson@linaro.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1860a5577fa9dab29d0f621f7a554b10fdc1f680
Author: Daniel Axtens <dja@axtens.net>
Date:   Wed May 15 20:24:50 2019 +1000

    crypto: vmx - CTR: always increment IV as quadword

    commit 009b30ac7444c17fae34c4f435ebce8e8e2b3250 upstream.

    The kernel self-tests picked up an issue with CTR mode:
    alg: skcipher: p8_aes_ctr encryption test failed (wrong result) on test vector 3,
cfg="uneven misaligned splits, may sleep"

    Test vector 3 has an IV of FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD, so
    after 3 increments it should wrap around to 0.

    In the aesp8-ppc code from OpenSSL, there are two paths that
    increment IVs: the bulk (8 at a time) path, and the individual
    path which is used when there are fewer than 8 AES blocks to
    process.

    In the bulk path, the IV is incremented with vadduqm: "Vector
    Add Unsigned Quadword Modulo", which does 128-bit addition.

    In the individual path, however, the IV is incremented with
    vadduwm: "Vector Add Unsigned Word Modulo", which instead
    does 4 32-bit additions. Thus the IV would instead become
    FFFFFFFFFFFFFFFFFFFFFFFF00000000, throwing off the result.

    Use vadduqm.

    This was probably a typo originally, what with q and w being
    adjacent. It is a pretty narrow edge case: I am really
    impressed by the quality of the kernel self-tests!

    Fixes: 5c380d623ed3 ("crypto: vmx - Add support for VMS instructions by ASM")
    Cc: stable@vger.kernel.org
```

```
    Signed-off-by: Daniel Axtens <dja@axtens.net>
    Acked-by: Nayna Jain <nayna@linux.ibm.com>
    Tested-by: Nayna Jain <nayna@linux.ibm.com>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6920fcd3eebcae2eccbe1e488ef19c923ab197fd
Author: Eric Biggers <ebiggers@google.com>
Date:   Tue May 14 16:13:15 2019 -0700

    crypto: hash - fix incorrect HASH_MAX_DESCSIZE

    commit e1354400b25da645c4764ed6844d12f1582c3b66 upstream.

    The "hmac(sha3-224-generic)" algorithm has a descsize of 368 bytes,
    which is greater than HASH_MAX_DESCSIZE (360) which is only enough for
    sha3-224-generic.  The check in shash_prepare_alg() doesn't catch this
    because the HMAC template doesn't set descsize on the algorithms, but
    rather sets it on each individual HMAC transform.

    This causes a stack buffer overflow when SHASH_DESC_ON_STACK() is used
    with hmac(sha3-224-generic).

    Fix it by increasing HASH_MAX_DESCSIZE to the real maximum.  Also add a
    sanity check to hmac_init().

    This was detected by the improved crypto self-tests in v5.2, by loading
    the tcrypt module with CONFIG_CRYPTO_MANAGER_EXTRA_TESTS=y enabled.  I
    didn't notice this bug when I ran the self-tests by requesting the
    algorithms via AF_ALG (i.e., not using tcrypt), probably because the
    stack layout differs in the two cases and that made a difference here.

    KASAN report:

        BUG: KASAN: stack-out-of-bounds in memcpy include/linux/string.h:359 [inline]
        BUG: KASAN: stack-out-of-bounds in shash_default_import+0x52/0x80 crypto/shash.c:223
        Write of size 360 at addr ffff8880651defc8 by task insmod/3689

        CPU: 2 PID: 3689 Comm: insmod Tainted: G            E     5.1.0-10741-g35c99ffa20edd
#11
        Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1 04/01/2014
        Call Trace:
         __dump_stack lib/dump_stack.c:77 [inline]
         dump_stack+0x86/0xc5 lib/dump_stack.c:113
         print_address_description+0x7f/0x260 mm/kasan/report.c:188
         __kasan_report+0x144/0x187 mm/kasan/report.c:317
         kasan_report+0x12/0x20 mm/kasan/common.c:614
         check_memory_region_inline mm/kasan/generic.c:185 [inline]
         check_memory_region+0x137/0x190 mm/kasan/generic.c:191
         memcpy+0x37/0x50 mm/kasan/common.c:125
         memcpy include/linux/string.h:359 [inline]
         shash_default_import+0x52/0x80 crypto/shash.c:223
         crypto_shash_import include/crypto/hash.h:880 [inline]
         hmac_import+0x184/0x240 crypto/hmac.c:102
         hmac_init+0x96/0xc0 crypto/hmac.c:107
         crypto_shash_init include/crypto/hash.h:902 [inline]
         shash_digest_unaligned+0x9f/0xf0 crypto/shash.c:194
         crypto_shash_digest+0xe9/0x1b0 crypto/shash.c:211
         generate_random_hash_testvec.constprop.11+0x1ec/0x5b0 crypto/testmgr.c:1331
         test_hash_vs_generic_impl+0x3f7/0x5c0 crypto/testmgr.c:1420
         __alg_test_hash+0x26d/0x340 crypto/testmgr.c:1502
         alg_test_hash+0x22e/0x330 crypto/testmgr.c:1552
         alg_test.part.7+0x132/0x610 crypto/testmgr.c:4931
         alg_test+0x1f/0x40 crypto/testmgr.c:4952
```

```
    Fixes: b68a7ec1e9a3 ("crypto: hash - Remove VLA usage")
    Reported-by: Corentin Labbe <clabbe.montjoie@gmail.com>
    Cc: <stable@vger.kernel.org> # v4.20+
    Cc: Kees Cook <keescook@chromium.org>
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Reviewed-by: Kees Cook <keescook@chromium.org>
    Tested-by: Corentin Labbe <clabbe.montjoie@gmail.com>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 204d535040fad5192086852650e9ccfd449087f4
Author: Martin K. Petersen <martin.petersen@oracle.com>
Date:   Mon May 20 10:57:18 2019 -0400

    Revert "scsi: sd: Keep disk read-only when re-reading partition"

    commit 8acf608e602f6ec38b7cc37b04c80f1ce9a1a6cc upstream.

    This reverts commit 20bd1d026aacc5399464f8328f305985c493cde3.

    This patch introduced regressions for devices that come online in
    read-only state and subsequently switch to read-write.

    Given how the partition code is currently implemented it is not
    possible to persist the read-only flag across a device revalidate
    call. This may need to get addressed in the future since it is common
    for user applications to proactively call BLKRRPART.

    Reverting this commit will re-introduce a regression where a
    device-initiated revalidate event will cause the admin state to be
    forgotten. A separate patch will address this issue.

    Fixes: 20bd1d026aac ("scsi: sd: Keep disk read-only when re-reading partition")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 15e5e4b9ff861952765cce30d1ed78bf22869ff8
Author: Andrea Parri <andrea.parri@amarulasolutions.com>
Date:   Mon May 20 19:23:57 2019 +0200

    sbitmap: fix improper use of smp_mb__before_atomic()

    commit a0934fd2b1208458e55fc4b48f55889809fce666 upstream.

    This barrier only applies to the read-modify-write operations; in
    particular, it does not apply to the atomic_set() primitive.

    Replace the barrier with an smp_mb().

    Fixes: 6c0ca7ae292ad ("sbitmap: fix wakeup hang after sbq resize")
    Cc: stable@vger.kernel.org
    Reported-by: "Paul E. McKenney" <paulmck@linux.ibm.com>
    Reported-by: Peter Zijlstra <peterz@infradead.org>
    Signed-off-by: Andrea Parri <andrea.parri@amarulasolutions.com>
    Reviewed-by: Ming Lei <ming.lei@redhat.com>
    Cc: Jens Axboe <axboe@kernel.dk>
    Cc: Omar Sandoval <osandov@fb.com>
    Cc: Ming Lei <ming.lei@redhat.com>
    Cc: linux-block@vger.kernel.org
    Cc: "Paul E. McKenney" <paulmck@linux.ibm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 01b5e7f86d86d188e487d4991053d3087c336c9c
Author: Andrea Parri <andrea.parri@amarulasolutions.com>
Date:   Mon May 20 19:23:56 2019 +0200

    bio: fix improper use of smp_mb__before_atomic()

    commit f381c6a4bd0ae0fde2d6340f1b9bb0f58d915de6 upstream.

    This barrier only applies to the read-modify-write operations; in
    particular, it does not apply to the atomic_set() primitive.

    Replace the barrier with an smp_mb().

    Fixes: dac56212e8127 ("bio: skip atomic inc/dec of ->bi_cnt for most use cases")
    Cc: stable@vger.kernel.org
    Reported-by: "Paul E. McKenney" <paulmck@linux.ibm.com>
    Reported-by: Peter Zijlstra <peterz@infradead.org>
    Signed-off-by: Andrea Parri <andrea.parri@amarulasolutions.com>
    Reviewed-by: Ming Lei <ming.lei@redhat.com>
    Cc: Jens Axboe <axboe@kernel.dk>
    Cc: Ming Lei <ming.lei@redhat.com>
    Cc: linux-block@vger.kernel.org
    Cc: "Paul E. McKenney" <paulmck@linux.ibm.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e83d85e7d8bf74d5ce6b64f2235ea7be6eee990f
Author: Borislav Petkov <bp@suse.de>
Date:   Wed May 8 19:02:48 2019 +0200

    x86/kvm/pmu: Set AMD's virt PMU version to 1

    commit a80c4ec10ed9632c44c829452dc40a0443ff4e85 upstream.

    After commit:

      672ff6cff80c ("KVM: x86: Raise #GP when guest vCPU do not support PMU")

    my AMD guests started #GPing like this:

      general protection fault: 0000 [#1] PREEMPT SMP
      CPU: 1 PID: 4355 Comm: bash Not tainted 5.1.0-rc6+ #3
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014
      RIP: 0010:x86_perf_event_update+0x3b/0xa0

    with Code: pointing to RDPMC. It is RDPMC because the guest has the
    hardware watchdog CONFIG_HARDLOCKUP_DETECTOR_PERF enabled which uses
    perf. Instrumenting kvm_pmu_rdpmc() some, showed that it fails due to:

      if (!pmu->version)
            return 1;

    which the above commit added. Since AMD's PMU leaves the version at 0,
    that causes the #GP injection into the guest.

    Set pmu->version arbitrarily to 1 and move it above the non-applicable
    struct kvm_pmu members.

    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Janakarajan Natarajan <Janakarajan.Natarajan@amd.com>
    Cc: kvm@vger.kernel.org
    Cc: Liran Alon <liran.alon@oracle.com>
```

```
    Cc: Mihai Carabas <mihai.carabas@oracle.com>
    Cc: Paolo Bonzini <pbonzini@redhat.com>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: "Radim Krčmář" <rkrcmar@redhat.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Tom Lendacky <thomas.lendacky@amd.com>
    Cc: x86@kernel.org
    Cc: stable@vger.kernel.org
    Fixes: 672ff6cff80c ("KVM: x86: Raise #GP when guest vCPU do not support PMU")
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d7a74fbae714347c2dd8931ddff724db03855592
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Fri May 24 21:52:46 2019 +0200

    KVM: x86: fix return value for reserved EFER

    commit 66f61c92889ff3ca365161fb29dd36d6354682ba upstream.

    Commit 11988499e62b ("KVM: x86: Skip EFER vs. guest CPUID checks for
    host-initiated writes", 2019-04-02) introduced a "return false" in a
    function returning int, and anyway set_efer has a "nonzero on error"
    conventon so it should be returning 1.

    Reported-by: Pavel Machek <pavel@denx.de>
    Fixes: 11988499e62b ("KVM: x86: Skip EFER vs. guest CPUID checks for host-initiated
writes")
    Cc: Sean Christopherson <sean.j.christopherson@intel.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 824adfb226a8d3fd0f1cd954dfb2807cbe7766f5
Author: Jan Kara <jack@suse.cz>
Date:   Thu May 23 23:07:08 2019 -0400

    ext4: wait for outstanding dio during truncate in nojournal mode

    commit 82a25b027ca48d7ef197295846b352345853dfa8 upstream.

    We didn't wait for outstanding direct IO during truncate in nojournal
    mode (as we skip orphan handling in that case). This can lead to fs
    corruption or stale data exposure if truncate ends up freeing blocks
    and these get reallocated before direct IO finishes. Fix the condition
    determining whether the wait is necessary.

    CC: stable@vger.kernel.org
    Fixes: 1c9114f9c0f1 ("ext4: serialize unlocked dio reads with truncate")
    Reviewed-by: Ira Weiny <ira.weiny@intel.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5f2e67d310fffc2519baea2f41bbb21b1b55c59d
Author: Jan Kara <jack@suse.cz>
Date:   Thu May 23 23:35:28 2019 -0400

    ext4: do not delete unlinked inode from orphan list on failed truncate

    commit ee0ed02ca93ef1ecf8963ad96638795d55af2c14 upstream.

    It is possible that unlinked inode enters ext4_setattr() (e.g. if
    somebody calls ftruncate(2) on unlinked but still open file). In such
    case we should not delete the inode from the orphan list if truncate
```

```
    fails. Note that this is mostly a theoretical concern as filesystem is
    corrupted if we reach this path anyway but let's be consistent in our
    orphan handling.

    Reviewed-by: Ira Weiny <ira.weiny@intel.com>
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 680ae6bae20a0a0a984ec1b7559b5da0942be7a5
Author: Steven Rostedt (VMware) <rostedt@goodmis.org>
Date:   Sat May 11 08:32:40 2019 -0400

    x86: Hide the int3_emulate_call/jmp functions from UML

    commit 693713cbdb3a4bda5a8a678c31f06560bbb14657 upstream.

    User Mode Linux does not have access to the ip or sp fields of the pt_regs,
    and accessing them causes UML to fail to build. Hide the int3_emulate_jmp()
    and int3_emulate_call() instructions from UML, as it doesn't need them
    anyway.

    Reported-by: kbuild test robot <lkp@intel.com>
    Signed-off-by: Steven Rostedt (VMware) <rostedt@goodmis.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```