```
commit b269b0268d4121d033721775d6e0c1114acfe50b
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Aug 11 12:14:29 2023 +0200

    Linux 6.4.10

    Link: https://lore.kernel.org/r/20230809103642.720851262@linuxfoundation.org
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Joel Fernandes (Google) <joel@joelfernandes.org>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Tested-by: SeongJae Park <sj@kernel.org>
    Tested-by: Salvatore Bonaccorso <carnil@debian.org>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Conor Dooley <conor.dooley@microchip.com>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Miguel Ojeda <ojeda@kernel.org> # Rust
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c9c0b889e2d33d49b06bb716b95a192ed3449173
Author: Borislav Petkov (AMD) <bp@alien8.de>
Date:   Sat Aug 5 00:06:43 2023 +0200

    x86/CPU/AMD: Do not leak quotient data after a division by 0

    commit 77245f1c3c6495521f6a3af082696ee2f8ce3921 upstream.

    Under certain circumstances, an integer division by 0 which faults, can
    leave stale quotient data from a previous division operation on Zen1
    microarchitectures.

    Do a dummy division 0/1 before returning from the #DE exception handler
    in order to avoid any leaks of potentially sensitive data.

    Signed-off-by: Borislav Petkov (AMD) <bp@alien8.de>
    Cc: <stable@kernel.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 493c80a4f20b741fded847b51ff73ccd7887cc74
Author: Andi Shyti <andi.shyti@linux.intel.com>
Date:   Tue Jul 25 02:19:48 2023 +0200

    drm/i915/gt: Enable the CCS_FLUSH bit in the pipe control and in the CS

    [ Upstream commit 824df77ab2107d8d4740b834b276681a41ae1ac8 ]

    Enable the CCS_FLUSH bit 13 in the control pipe for render and
    compute engines in platforms starting from Meteor Lake (BSPEC
    43904 and 47112).

    For the copy engine add MI_FLUSH_DW_CCS (bit 16) in the command
    streamer.

    Fixes: 972282c4cf24 ("drm/i915/gen12: Add aux table invalidate for all engines")
    Requires: 8da173db894a ("drm/i915/gt: Rename flags with bit_group_X according to the
datasheet")
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: Jonathan Cavitt <jonathan.cavitt@intel.com>
    Cc: Nirmoy Das <nirmoy.das@intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Matt Roper <matthew.d.roper@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
```

```
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-6-
andi.shyti@linux.intel.com
    (cherry picked from commit b70df82b428774875c7c56d3808102165891547c)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 43f5167d2cdcbd78665456ddea5b1c4f1af1c496
Author: Andi Shyti <andi.shyti@linux.intel.com>
Date:   Tue Jul 25 02:19:50 2023 +0200

    drm/i915/gt: Support aux invalidation on all engines

    [ Upstream commit 6a35f22d222528e1b157c6978c9424d2f8cbe0a1 ]

    Perform some refactoring with the purpose of keeping in one
    single place all the operations around the aux table
    invalidation.

    With this refactoring add more engines where the invalidation
    should be performed.

    Fixes: 972282c4cf24 ("drm/i915/gen12: Add aux table invalidate for all engines")
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: Jonathan Cavitt <jonathan.cavitt@intel.com>
    Cc: Matt Roper <matthew.d.roper@intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-8-
andi.shyti@linux.intel.com
    (cherry picked from commit 76ff7789d6e63d1a10b3b58f5c70b2e640c7a880)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e66e37671c18e8f2965f38cac99753e1ab0772f7
Author: Jonathan Cavitt <jonathan.cavitt@intel.com>
Date:   Tue Jul 25 02:19:49 2023 +0200

    drm/i915/gt: Poll aux invalidation register bit on invalidation

    [ Upstream commit 0fde2f23516a00fd90dfb980b66b4665fcbfa659 ]

    For platforms that use Aux CCS, wait for aux invalidation to
    complete by checking the aux invalidation register bit is
    cleared.

    Fixes: 972282c4cf24 ("drm/i915/gen12: Add aux table invalidate for all engines")
    Signed-off-by: Jonathan Cavitt <jonathan.cavitt@intel.com>
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Reviewed-by: Matt Roper <matthew.d.roper@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-7-
andi.shyti@linux.intel.com
    (cherry picked from commit d459c86f00aa98028d155a012c65dc42f7c37e76)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 10be2cb87f0a19bc93ea483ba48cfcc1f8e0f5bc
Author: Andi Shyti <andi.shyti@linux.intel.com>
Date:   Tue Jul 25 02:19:47 2023 +0200

    drm/i915/gt: Rename flags with bit_group_X according to the datasheet
```

```
    [ Upstream commit 592b228f12e15867a63e3a6eeeb54c5c12662a62 ]

    In preparation of the next patch align with the datasheet (BSPEC
    47112) with the naming of the pipe control set of flag values.
    The variable "flags" in gen12_emit_flush_rcs() is applied as a
    set of flags called Bit Group 1.

    Define also the Bit Group 0 as bit_group_0 where currently only
    PIPE_CONTROL0_HDC_PIPELINE_FLUSH bit is set.

    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Matt Roper <matthew.d.roper@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-5-
andi.shyti@linux.intel.com
    (cherry picked from commit f2dcd21d5a22e13f2fbfe7ab65149038b93cf2ff)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 73400908a97b305d21e993ad76d2a23a44302b1e
Author: Tejas Upadhyay <tejas.upadhyay@intel.com>
Date:   Thu Jun 1 16:39:59 2023 +0530

    drm/i915/gt: Add workaround 14016712196

    [ Upstream commit d922b80b1010cd6164fa7d3c197b4fbf94b47beb ]

    For mtl, workaround suggests that, SW insert a
    dummy PIPE_CONTROL prior to PIPE_CONTROL which
    contains a post sync: Timestamp or Write Immediate.

    Bspec: 72197

    V5:
      - Remove ret variable - Andi
    V4:
      - Update commit message, avoid returing cs - Andi/Matt
    V3:
      - Wrap dummy pipe control stuff in API - Andi
    V2:
      - Fix  kernel test robot warnings

    Closes: https://lore.kernel.org/oe-kbuild-all/202305121525.3EWdGoBY-lkp@intel.com/
    Signed-off-by: Tejas Upadhyay <tejas.upadhyay@intel.com>
    Reviewed-by: Andi Shyti <andi.shyti@linux.intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230601110959.1715927-1-
tejas.upadhyay@intel.com
    Stable-dep-of: 592b228f12e1 ("drm/i915/gt: Rename flags with bit_group_X according to
the datasheet")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 17b66e10b134879deb9cbe8d6c0ac40d59f647a1
Author: Jonathan Cavitt <jonathan.cavitt@intel.com>
Date:   Tue Jul 25 02:19:46 2023 +0200

    drm/i915/gt: Ensure memory quiesced before invalidation

    [ Upstream commit 78a6ccd65fa3a7cc697810db079cc4b84dff03d5 ]

    All memory traffic must be quiesced before requesting
    an aux invalidation on platforms that use Aux CCS.
```

```
    Fixes: 972282c4cf24 ("drm/i915/gen12: Add aux table invalidate for all engines")
    Requires: a2a4aa0eef3b ("drm/i915: Add the gen12_needs_ccs_aux_inv helper")
    Signed-off-by: Jonathan Cavitt <jonathan.cavitt@intel.com>
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-4-
andi.shyti@linux.intel.com
    (cherry picked from commit ad8ebf12217e451cd19804b1c3e97ad56491c74a)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c0660d36ecd8d5751a94a796b716a6e0b113cb22
Author: Andi Shyti <andi.shyti@linux.intel.com>
Date:   Tue Jul 25 02:19:45 2023 +0200

    drm/i915: Add the gen12_needs_ccs_aux_inv helper

    [ Upstream commit b2f59e9026038a5bbcbc0019fa58f963138211ee ]

    We always assumed that a device might either have AUX or FLAT
    CCS, but this is an approximation that is not always true, e.g.
    PVC represents an exception.

    Set the basis for future finer selection by implementing a
    boolean gen12_needs_ccs_aux_inv() function that tells whether aux
    invalidation is needed or not.

    Currently PVC is the only exception to the above mentioned rule.

    Requires: 059ae7ae2a1c ("drm/i915/gt: Cleanup aux invalidation registers")
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: Matt Roper <matthew.d.roper@intel.com>
    Cc: Jonathan Cavitt <jonathan.cavitt@intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Matt Roper <matthew.d.roper@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-3-
andi.shyti@linux.intel.com
    (cherry picked from commit c827655b87ad201ebe36f2e28d16b5491c8f7801)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7f87080d2e9744f67da17121363729c21cba4e53
Author: Xu Yang <xu.yang_2@nxp.com>
Date:   Mon Jul 17 10:28:33 2023 +0800

    ARM: dts: nxp/imx6sll: fix wrong property name in usbphy node

    [ Upstream commit ee70b908f77a9d8f689dea986f09e6d7dc481934 ]

    Property name "phy-3p0-supply" is used instead of "phy-reg_3p0-supply".

    Fixes: 9f30b6b1a957 ("ARM: dts: imx: Add basic dtsi file for imx6sll")
    cc: <stable@vger.kernel.org>
    Signed-off-by: Xu Yang <xu.yang_2@nxp.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 544fdf64d6984b3eccddd58195c0d74df50a9f23
Author: Sean Christopherson <seanjc@google.com>
Date:   Fri Jul 21 15:33:52 2023 -0700
```

```
    selftests/rseq: Play nice with binaries statically linked against glibc 2.35+

    [ Upstream commit 3bcbc20942db5d738221cca31a928efc09827069 ]

    To allow running rseq and KVM's rseq selftests as statically linked
    binaries, initialize the various "trampoline" pointers to point directly
    at the expect glibc symbols, and skip the dlysm() lookups if the rseq
    size is non-zero, i.e. the binary is statically linked *and* the libc
    registered its own rseq.

    Define weak versions of the symbols so as not to break linking against
    libc versions that don't support rseq in any capacity.

    The KVM selftests in particular are often statically linked so that they
    can be run on targets with very limited runtime environments, i.e. test
    machines.

    Fixes: 233e667e1ae3 ("selftests/rseq: Uplift rseq selftests for compatibility with
glibc-2.35")
    Cc: Aaron Lewis <aaronlewis@google.com>
    Cc: kvm@vger.kernel.org
    Cc: stable@vger.kernel.org
    Signed-off-by: Sean Christopherson <seanjc@google.com>
    Message-Id: <20230721223352.2333911-1-seanjc@google.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 33f735ef8dfecdd8bf3c8544d1db15079353e920
Author: Lijo Lazar <lijo.lazar@amd.com>
Date:   Fri Feb 24 18:01:38 2023 +0530

    drm/amdgpu: Use apt name for FW reserved region

    commit db3b5cb64a9ca301d14ed027e470834316720e42 upstream.

    Use the generic term fw_reserved_memory for FW reserve region. This
    region may also hold discovery TMR in addition to other reserve
    regions. This region size could be larger than discovery tmr size, hence
    don't change the discovery tmr size based on this.

    Signed-off-by: Lijo Lazar <lijo.lazar@amd.com>
    Reviewed-by: Le Ma <le.ma@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    This change fixes reading IP discovery from debugfs.
    It needed to be hand modified because GC 9.4.3 support isn't
    introduced in older kernels until 228ce176434b ("drm/amdgpu: Handle
    VRAM dependencies on GFXIP9.4.3")
    Link: https://gitlab.freedesktop.org/drm/amd/-/issues/2748
    Signed-off-by: Mario Limonciello <mario.limonciello@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 61977b1526bae8b8643ae586e63033c2bb1f3c04
Author: Alexander Stein <alexander.stein@ew.tq-group.com>
Date:   Mon May 15 09:21:37 2023 +0200

    drm/imx/ipuv3: Fix front porch adjustment upon hactive aligning

    [ Upstream commit ee31742bf17636da1304af77b2cb1c29b5dda642 ]

    When hactive is not aligned to 8 pixels, it is aligned accordingly and
    hfront porch needs to be reduced the same amount. Unfortunately the front
    porch is set to the difference rather than reducing it. There are some
    Samsung TVs which can't cope with a front porch of instead of 70.
```

```
    Fixes: 94dfec48fca7 ("drm/imx: Add 8 pixel alignment fix")
    Signed-off-by: Alexander Stein <alexander.stein@ew.tq-group.com>
    Reviewed-by: Philipp Zabel <p.zabel@pengutronix.de>
    Link: https://lore.kernel.org/r/20230515072137.116211-1-alexander.stein@ew.tq-group.com
    [p.zabel@pengutronix.de: Fixed subject]
    Signed-off-by: Philipp Zabel <p.zabel@pengutronix.de>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230515072137.116211-1-
alexander.stein@ew.tq-group.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bbfa34c97d8b8e2e12647cd4dc2f141ae241d90a
Author: Aneesh Kumar K.V <aneesh.kumar@linux.ibm.com>
Date:   Mon Jul 24 23:43:20 2023 +0530

    powerpc/mm/altmap: Fix altmap boundary check

    [ Upstream commit 6722b25712054c0f903b839b8f5088438dd04df3 ]

    altmap->free includes the entire free space from which altmap blocks
    can be allocated. So when checking whether the kernel is doing altmap
    block free, compute the boundary correctly, otherwise memory hotunplug
    can fail.

    Fixes: 9ef34630a461 ("powerpc/mm: Fallback to RAM if the altmap is unusable")
    Signed-off-by: "Aneesh Kumar K.V" <aneesh.kumar@linux.ibm.com>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://msgid.link/20230724181320.471386-1-aneesh.kumar@linux.ibm.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 49e57caf967a969f6b955c88805f2d160910aa12
Author: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
Date:   Wed Jul 19 23:55:01 2023 +0200

    mtd: rawnand: fsl_upm: Fix an off-by one test in fun_exec_op()

    [ Upstream commit c6abce60338aa2080973cd95be0aedad528bb41f ]

    'op->cs' is copied in 'fun->mchip_number' which is used to access the
    'mchip_offsets' and the 'rnb_gpio' arrays.
    These arrays have NAND_MAX_CHIPS elements, so the index must be below this
    limit.

    Fix the sanity check in order to avoid the NAND_MAX_CHIPS value. This
    would lead to out-of-bound accesses.

    Fixes: 54309d657767 ("mtd: rawnand: fsl_upm: Implement exec_op()")
    Signed-off-by: Christophe JAILLET <christophe.jaillet@wanadoo.fr>
    Reviewed-by: Dan Carpenter <dan.carpenter@linaro.org>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-
mtd/cd01cba1c7eda58bdabaae174c78c067325803d2.1689803636.git.christophe.jaillet@wanadoo.fr
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 65df5d2ecc2c091e577030f455060d01fa7fdc88
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Wed Jul 19 21:00:25 2023 +0200

    mtd: spi-nor: avoid holes in struct spi_mem_op

    [ Upstream commit 71c8f9cf2623d0db79665f876b95afcdd8214aec ]

    gcc gets confused when -ftrivial-auto-var-init=pattern is used on sparse
    bit fields such as 'struct spi_mem_op', which caused the previous false
    positive warning about an uninitialized variable:
```

```
    drivers/mtd/spi-nor/spansion.c: error: 'op' is used uninitialized [-
Werror=uninitialized]

    In fact, the variable is fully initialized and gcc does not see it being
    used, so the warning is entirely bogus. The problem appears to be
    a misoptimization in the initialization of single bit fields when the
    rest of the bytes are not initialized.

    A previous workaround added another initialization, which ended up
    shutting up the warning in spansion.c, though it apparently still happens
    in other files as reported by Peter Foley in the gcc bugzilla. The
    workaround of adding a fake initialization seems particularly bad
    because it would set values that can never be correct but prevent the
    compiler from warning about actually missing initializations.

    Revert the broken workaround and instead pad the structure to only
    have bitfields that add up to full bytes, which should avoid this
    behavior in all drivers.

    I also filed a new bug against gcc with what I found, so this can
    hopefully be addressed in future gcc releases. At the moment, only
    gcc-12 and gcc-13 are affected.

    Cc: Peter Foley <pefoley2@pefoley.com>
    Cc: Pedro Falcato <pedro.falcato@gmail.com>
    Link: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=110743
    Link: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=108402
    Link: https://godbolt.org/z/efMMsG1Kx
    Fixes: 420c4495b5e56 ("mtd: spi-nor: spansion: make sure local struct does not contain
garbage")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Acked-by: Mark Brown <broonie@kernel.org>
    Acked-by: Tudor Ambarus <tudor.ambarus@linaro.org>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230719190045.4007391-1-arnd@kernel.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 45d69917a4af6c869193f95932dc6d6f15d5ef86
Author: Chen-Yu Tsai <wenst@chromium.org>
Date:   Wed Jul 19 15:42:50 2023 +0800

    clk: mediatek: mt8183: Add back SSPM related clocks

    [ Upstream commit 1eb8d61ac5c9c7ec56bb96d433532807509b9288 ]

    This reverts commit 860690a93ef23b567f781c1b631623e27190f101.

    On the MT8183, the SSPM related clocks were removed claiming a lack of
    usage. This however causes some issues when the driver was converted to
    the new simple-probe mechanism. This mechanism allocates enough space
    for all the clocks defined in the clock driver, not the highest index
    in the DT binding. This leads to out-of-bound writes if their are holes
    in the DT binding or the driver (due to deprecated or unimplemented
    clocks). These errors can go unnoticed and cause memory corruption,
    leading to crashes in unrelated areas, or nothing at all. KASAN will
    detect them.

    Add the SSPM related clocks back to the MT8183 clock driver to fully
    implement the DT binding. The SSPM clocks are for the power management
    co-processor, and should never be turned off. They are marked as such.

    Fixes: 3f37ba7cc385 ("clk: mediatek: mt8183: Convert all remaining clocks to common
probe")
    Signed-off-by: Chen-Yu Tsai <wenst@chromium.org>
```

```
    Link: https://lore.kernel.org/r/20230719074251.1219089-1-wenst@chromium.org
    Reviewed-by: AngeloGioacchino Del Regno <angelogioacchino.delregno@collabora.com>
    Signed-off-by: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 182ac84852a833fff8df048f74d993d4bf8ed192
Author: Johan Jonker <jbx6244@gmail.com>
Date:   Fri Jul 14 17:21:21 2023 +0200

    mtd: rawnand: rockchip: Align hwecc vs. raw page helper layouts

    [ Upstream commit ea690ad78dd611e3906df5b948a516000b05c1cb ]

    Currently, read/write_page_hwecc() and read/write_page_raw() are not
    aligned: there is a mismatch in the OOB bytes which are not
    read/written at the same offset in both cases (raw vs. hwecc).

    This is a real problem when relying on the presence of the Page
    Addresses (PA) when using the NAND chip as a boot device, as the
    BootROM expects additional data in the OOB area at specific locations.

    Rockchip boot blocks are written per 4 x 512 byte sectors per page.
    Each page with boot blocks must have a page address (PA) pointer in OOB
    to the next page. Pages are written in a pattern depending on the NAND chip ID.

    Generate boot block page address and pattern for hwecc in user space
    and copy PA data to/from the already reserved last 4 bytes before ECC
    in the chip->oob_poi data layout.

    Align the different helpers. This change breaks existing jffs2 users.

    Fixes: 058e0e847d54 ("mtd: rawnand: rockchip: NFC driver for RK3308, RK2928 and others")
    Signed-off-by: Johan Jonker <jbx6244@gmail.com>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/5e782c08-862b-51ae-47ff-3299940928ca@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 550322382c5c8c323a0f221fa09bf4b3c7c2b68e
Author: Johan Jonker <jbx6244@gmail.com>
Date:   Fri Jul 14 17:21:01 2023 +0200

    mtd: rawnand: rockchip: fix oobfree offset and description

    [ Upstream commit d0ca3b92b7a6f42841ea9da8492aaf649db79780 ]

    Rockchip boot blocks are written per 4 x 512 byte sectors per page.
    Each page with boot blocks must have a page address (PA) pointer in OOB
    to the next page.

    The currently advertised free OOB area starts at offset 6, like
    if 4 PA bytes were located right after the BBM. This is wrong as the
    PA bytes are located right before the ECC bytes.

    Fix the layout by allowing access to all bytes between the BBM and the
    PA bytes instead of reserving 4 bytes right after the BBM.

    This change breaks existing jffs2 users.

    Fixes: 058e0e847d54 ("mtd: rawnand: rockchip: NFC driver for RK3308, RK2928 and others")
    Signed-off-by: Johan Jonker <jbx6244@gmail.com>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/d202f12d-188c-20e8-f2c2-9cc874ad4d22@gmail.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b3e2e796be07c685db351c1c43f2d44cc8a6a404
```

```
Author: Roger Quadros <rogerq@kernel.org>
Date:   Sun Jun 25 00:10:21 2023 +0530

    mtd: rawnand: omap_elm: Fix incorrect type in assignment

    [ Upstream commit d8403b9eeee66d5dd81ecb9445800b108c267ce3 ]

    Once the ECC word endianness is converted to BE32, we force cast it
    to u32 so we can use elm_write_reg() which in turn uses writel().

    Fixes below sparse warnings:

       drivers/mtd/nand/raw/omap_elm.c:180:37: sparse:     expected unsigned int [usertype]
val
       drivers/mtd/nand/raw/omap_elm.c:180:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:185:37: sparse:     expected unsigned int [usertype]
val
       drivers/mtd/nand/raw/omap_elm.c:185:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:190:37: sparse:     expected unsigned int [usertype]
val
       drivers/mtd/nand/raw/omap_elm.c:190:37: sparse:     got restricted __be32 [usertype]
    >> drivers/mtd/nand/raw/omap_elm.c:200:40: sparse: sparse: restricted __be32 degrades to
integer
       drivers/mtd/nand/raw/omap_elm.c:206:39: sparse: sparse: restricted __be32 degrades to
integer
       drivers/mtd/nand/raw/omap_elm.c:210:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:210:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:213:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:213:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:216:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:216:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:219:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:219:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:222:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:222:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:225:37: sparse:     expected unsigned int [assigned]
[usertype] val
       drivers/mtd/nand/raw/omap_elm.c:225:37: sparse:     got restricted __be32 [usertype]
       drivers/mtd/nand/raw/omap_elm.c:228:39: sparse: sparse: restricted __be32 degrades to
integer

    Fixes: bf22433575ef ("mtd: devices: elm: Add support for ELM error correction")
    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/oe-kbuild-all/202306212211.WDXokuWh-lkp@intel.com/
    Signed-off-by: Roger Quadros <rogerq@kernel.org>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230624184021.7740-1-rogerq@kernel.org
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a126124c86c56aeef5da0c088c89cc9b9184bf85
Author: Pavel Begunkov <asml.silence@gmail.com>
Date:   Fri May 19 15:21:16 2023 +0100

    io_uring: annotate offset timeout races

    commit 5498bf28d8f2bd63a46ad40f4427518615fb793f upstream.

    It's racy to read ->cached_cq_tail without taking proper measures
    (usually grabbing ->completion_lock) as timeout requests with CQE
    offsets do, however they have never had a good semantics for from
```

```
    when they start counting. Annotate racy reads with data_race().

    Reported-by: syzbot+cb265db2f3f3468ef436@syzkaller.appspotmail.com
    Signed-off-by: Pavel Begunkov <asml.silence@gmail.com>
    Link:
https://lore.kernel.org/r/4de3685e185832a92a572df2be2c735d2e21a83d.1684506056.git.asml.silen
ce@gmail.com
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit af0f716ad3b039cab9d426da63a5ee6c88751185
Author: Chao Yu <chao@kernel.org>
Date:   Thu Jun 29 19:11:44 2023 +0800

    f2fs: fix to do sanity check on direct node in truncate_dnode()

    commit a6ec83786ab9f13f25fb18166dee908845713a95 upstream.

    syzbot reports below bug:

    BUG: KASAN: slab-use-after-free in f2fs_truncate_data_blocks_range+0x122a/0x14c0
fs/f2fs/file.c:574
    Read of size 4 at addr ffff88802a25c000 by task syz-executor148/5000

    CPU: 1 PID: 5000 Comm: syz-executor148 Not tainted 6.4.0-rc7-syzkaller-00041-
ge660abd551f1 #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
05/27/2023
    Call Trace:
     <TASK>
     __dump_stack lib/dump_stack.c:88 [inline]
     dump_stack_lvl+0xd9/0x150 lib/dump_stack.c:106
     print_address_description.constprop.0+0x2c/0x3c0 mm/kasan/report.c:351
     print_report mm/kasan/report.c:462 [inline]
     kasan_report+0x11c/0x130 mm/kasan/report.c:572
     f2fs_truncate_data_blocks_range+0x122a/0x14c0 fs/f2fs/file.c:574
     truncate_dnode+0x229/0x2e0 fs/f2fs/node.c:944
     f2fs_truncate_inode_blocks+0x64b/0xde0 fs/f2fs/node.c:1154
     f2fs_do_truncate_blocks+0x4ac/0xf30 fs/f2fs/file.c:721
     f2fs_truncate_blocks+0x7b/0x300 fs/f2fs/file.c:749
     f2fs_truncate.part.0+0x4a5/0x630 fs/f2fs/file.c:799
     f2fs_truncate include/linux/fs.h:825 [inline]
     f2fs_setattr+0x1738/0x2090 fs/f2fs/file.c:1006
     notify_change+0xb2c/0x1180 fs/attr.c:483
     do_truncate+0x143/0x200 fs/open.c:66
     handle_truncate fs/namei.c:3295 [inline]
     do_open fs/namei.c:3640 [inline]
     path_openat+0x2083/0x2750 fs/namei.c:3791
     do_filp_open+0x1ba/0x410 fs/namei.c:3818
     do_sys_openat2+0x16d/0x4c0 fs/open.c:1356
     do_sys_open fs/open.c:1372 [inline]
     __do_sys_creat fs/open.c:1448 [inline]
     __se_sys_creat fs/open.c:1442 [inline]
     __x64_sys_creat+0xcd/0x120 fs/open.c:1442
     do_syscall_x64 arch/x86/entry/common.c:50 [inline]
     do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80
     entry_SYSCALL_64_after_hwframe+0x63/0xcd

    The root cause is, inodeA references inodeB via inodeB's ino, once inodeA
    is truncated, it calls truncate_dnode() to truncate data blocks in inodeB's
    node page, it traverse mapping data from node->i.i_addr[0] to
    node->i.i_addr[ADDRS_PER_BLOCK() - 1], result in out-of-boundary access.

    This patch fixes to add sanity check on dnode page in truncate_dnode(),
    so that, it can help to avoid triggering such issue, and once it encounters
```

```
    such issue, it will record newly introduced ERROR_INVALID_NODE_REFERENCE
    error into superblock, later fsck can detect such issue and try repairing.

    Also, it removes f2fs_truncate_data_blocks() for cleanup due to the
    function has only one caller, and uses f2fs_truncate_data_blocks_range()
    instead.

    Reported-and-tested-by: syzbot+12cb4425b22169b52036@syzkaller.appspotmail.com
    Closes: https://lore.kernel.org/linux-f2fs-devel/000000000000f3038a05fef867f8@google.com
    Signed-off-by: Chao Yu <chao@kernel.org>
    Signed-off-by: Jaegeuk Kim <jaegeuk@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f775ceb0cb530e4a469b718fb2a24843071087f5
Author: Filipe Manana <fdmanana@suse.com>
Date:   Fri Jun 30 16:03:44 2023 +0100

    btrfs: remove BUG_ON()'s in add_new_free_space()

    commit d8ccbd21918fd7fa6ce3226cffc22c444228e8ad upstream.

    At add_new_free_space() we have these BUG_ON()'s that are there to deal
    with any failure to add free space to the in memory free space cache.
    Such failures are mostly -ENOMEM that should be very rare. However there's
    no need to have these BUG_ON()'s, we can just return any error to the
    caller and all callers and their upper call chain are already dealing with
    errors.

    So just make add_new_free_space() return any errors, while removing the
    BUG_ON()'s, and returning the total amount of added free space to an
    optional u64 pointer argument.

    Reported-by: syzbot+3ba856e07b7127889d8c@syzkaller.appspotmail.com
    Link: https://lore.kernel.org/linux-btrfs/000000000000e9cb8305ff4e8327@google.com/
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 27f92aad4f17eaeca6fd3190a7048e3752f72917
Author: Jan Kara <jack@suse.cz>
Date:   Tue Jun 13 12:25:52 2023 +0200

    ext2: Drop fragment support

    commit 404615d7f1dcd4cca200e9a7a9df3a1dcae1dd62 upstream.

    Ext2 has fields in superblock reserved for subblock allocation support.
    However that never landed. Drop the many years dead code.

    Reported-by: syzbot+af5e10f73dbff48f70af@syzkaller.appspotmail.com
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d160941e1537c1b53f8c1de247bc8420b8619094
Author: Jason Gunthorpe <jgg@ziepe.ca>
Date:   Mon Jun 19 15:27:25 2023 -0300

    mm/gup: do not return 0 from pin_user_pages_fast() for bad args

    commit 9883c7f84053cec2826ca3c56254601b5ce9cdbe upstream.

    These routines are not intended to return zero, the callers cannot do
    anything sane with a 0 return.  They should return an error which means
    future calls to GUP will not succeed, or they should return some non-zero
```

```
    number of pinned pages which means GUP should be called again.

    If start + nr_pages overflows it should return -EOVERFLOW to signal the
    arguments are invalid.

    Syzkaller keeps tripping on this when fuzzing GUP arguments.

    Link: https://lkml.kernel.org/r/0-v1-3d5ed1f20d50+104-gup_overflow_jgg@nvidia.com
    Signed-off-by: Jason Gunthorpe <jgg@nvidia.com>
    Reported-by: syzbot+353c7be4964c6253f24a@syzkaller.appspotmail.com
    Closes: https://lore.kernel.org/all/000000000000094fdd05faa4d3a4@google.com
    Reviewed-by: John Hubbard <jhubbard@nvidia.com>
    Reviewed-by: Lorenzo Stoakes <lstoakes@gmail.com>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4abda85197ba5d695e6040d580b4b409ce0d3733
Author: Jan Kara <jack@suse.cz>
Date:   Thu Jun 15 13:38:48 2023 +0200

    fs: Protect reconfiguration of sb read-write from racing writes

    commit c541dce86c537714b6761a79a969c1623dfa222b upstream.

    The reconfigure / remount code takes a lot of effort to protect
    filesystem's reconfiguration code from racing writes on remounting
    read-only. However during remounting read-only filesystem to read-write
    mode userspace writes can start immediately once we clear SB_RDONLY
    flag. This is inconvenient for example for ext4 because we need to do
    some writes to the filesystem (such as preparation of quota files)
    before we can take userspace writes so we are clearing SB_RDONLY flag
    before we are fully ready to accept userpace writes and syzbot has found
    a way to exploit this [1]. Also as far as I'm reading the code
    the filesystem remount code was protected from racing writes in the
    legacy mount path by the mount's MNT_READONLY flag so this is relatively
    new problem. It is actually fairly easy to protect remount read-write
    from racing writes using sb->s_readonly_remount flag so let's just do
    that instead of having to workaround these races in the filesystem code.

    [1] https://lore.kernel.org/all/00000000000006a0df05f6667499@google.com/T/

    Signed-off-by: Jan Kara <jack@suse.cz>
    Message-Id: <20230615113848.8439-1-jack@suse.cz>
    Signed-off-by: Christian Brauner <brauner@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0dd3e0c31bf3e933fb85faf1443833aef90b8e46
Author: Alan Stern <stern@rowland.harvard.edu>
Date:   Wed Jul 12 10:15:10 2023 -0400

    net: usbnet: Fix WARNING in usbnet_start_xmit/usb_submit_urb

    commit 5e1627cb43ddf1b24b92eb26f8d958a3f5676ccb upstream.

    The syzbot fuzzer identified a problem in the usbnet driver:

    usb 1-1: BOGUS urb xfer, pipe 3 != type 1
    WARNING: CPU: 0 PID: 754 at drivers/usb/core/urb.c:504 usb_submit_urb+0xed6/0x1880
drivers/usb/core/urb.c:504
    Modules linked in:
    CPU: 0 PID: 754 Comm: kworker/0:2 Not tainted 6.4.0-rc7-syzkaller-00014-g692b7dc87ca6 #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
05/27/2023
    Workqueue: mld mld_ifc_work
```

```
    RIP: 0010:usb_submit_urb+0xed6/0x1880 drivers/usb/core/urb.c:504
    Code: 7c 24 18 e8 2c b4 5b fb 48 8b 7c 24 18 e8 42 07 f0 fe 41 89 d8 44 89 e1 4c 89 ea
 48 89 c6 48 c7 c7 a0 c9 fc 8a e8 5a 6f 23 fb <0f> 0b e9 58 f8 ff ff e8 fe b3 5b fb 48 81 c5
 c0 05 00 00 e9 84 f7
    RSP: 0018:ffffc9000463f568 EFLAGS: 00010086
    RAX: 0000000000000000 RBX: 0000000000000001 RCX: 0000000000000000
    RDX: ffff88801eb28000 RSI: ffffffff814c03b7 RDI: 0000000000000001
    RBP: ffff8881443b7190 R08: 0000000000000001 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000001 R12: 0000000000000003
    R13: ffff88802a77cb18 R14: 0000000000000003 R15: ffff888018262500
    FS:  0000000000000000(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 0000556a99c15a18 CR3: 0000000028c71000 CR4: 0000000000350ef0
    Call Trace:
     <TASK>
     usbnet_start_xmit+0xfe5/0x2190 drivers/net/usb/usbnet.c:1453
     __netdev_start_xmit include/linux/netdevice.h:4918 [inline]
     netdev_start_xmit include/linux/netdevice.h:4932 [inline]
     xmit_one net/core/dev.c:3578 [inline]
     dev_hard_start_xmit+0x187/0x700 net/core/dev.c:3594
    ...

    This bug is caused by the fact that usbnet trusts the bulk endpoint
    addresses its probe routine receives in the driver_info structure, and
    it does not check to see that these endpoints actually exist and have
    the expected type and directions.

    The fix is simply to add such a check.

    Reported-and-tested-by: syzbot+63ee658b9a100ffadbe2@syzkaller.appspotmail.com
    Closes: https://lore.kernel.org/linux-usb/000000000000a56e9105d0cec021@google.com/
    Signed-off-by: Alan Stern <stern@rowland.harvard.edu>
    CC: Oliver Neukum <oneukum@suse.com>
    Link: https://lore.kernel.org/r/ea152b6d-44df-4f8a-95c6-4db51143dcc1@rowland.harvard.edu
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6da717fc5d900758c60987c52d95a2b8bee3abf3
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Wed Jun 7 19:19:02 2023 +0900

    debugobjects: Recheck debug_objects_enabled before reporting

    commit 8b64d420fe2450f82848178506d3e3a0bd195539 upstream.

    syzbot is reporting false a positive ODEBUG message immediately after
    ODEBUG was disabled due to OOM.

      [ 1062.309646][T22911] ODEBUG: Out of memory. ODEBUG disabled
      [ 1062.886755][ T5171] ------------[ cut here ]------------
      [ 1062.892770][ T5171] ODEBUG: assert_init not available (active state 0) object:
ffffc900056afb20 object type: timer_list hint: process_timeout+0x0/0x40

      CPU 0 [ T5171]                CPU 1 [T22911]
      --------------                --------------
      debug_object_assert_init() {
        if (!debug_objects_enabled)
          return;
        db = get_bucket(addr);
                                    lookup_object_or_alloc() {
                                      debug_objects_enabled = 0;
                                      return NULL;
                                    }
                                    debug_objects_oom() {
                                      pr_warn("Out of memory. ODEBUG disabled\n");
```

```
                                      // all buckets get emptied here, and
                                    }
        lookup_object_or_alloc(addr, db, descr, false, true) {
          // this bucket is already empty.
          return ERR_PTR(-ENOENT);
        }
        // Emits false positive warning.
        debug_print_object(&o, "assert_init");
      }
```

    Recheck debug_object_enabled in debug_print_object() to avoid that.

    Reported-by: syzbot <syzbot+7937ba6a50bdd00fffdf@syzkaller.appspotmail.com>
    Suggested-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lore.kernel.org/r/492fe2ae-5141-d548-ebd5-62f5fe2e57f7@I-love.SAKURA.ne.jp
    Closes: https://syzkaller.appspot.com/bug?extid=7937ba6a50bdd00fffdf
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 10426afe65c8bf7b24dd0c7be4dcc65f86fc99f9
Author: Sungwoo Kim <iam@sung-woo.kim>
Date: Wed May 31 01:39:56 2023 -0400

    Bluetooth: L2CAP: Fix use-after-free in l2cap_sock_ready_cb

    commit 1728137b33c00d5a2b5110ed7aafb42e7c32e4a1 upstream.

    l2cap_sock_release(sk) frees sk. However, sk's children are still alive
    and point to the already free'd sk's address.
    To fix this, l2cap_sock_release(sk) also cleans sk's children.

    ==
    BUG: KASAN: use-after-free in l2cap_sock_ready_cb+0xb7/0x100
net/bluetooth/l2cap_sock.c:1650
    Read of size 8 at addr ffff888104617aa8 by task kworker/u3:0/276

    CPU: 0 PID: 276 Comm: kworker/u3:0 Not tainted 6.2.0-00001-gef397bd4d5fb-dirty #59
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014
    Workqueue: hci2 hci_rx_work
    Call Trace:
     <TASK>
     __dump_stack lib/dump_stack.c:88 [inline]
     dump_stack_lvl+0x72/0x95 lib/dump_stack.c:106
     print_address_description mm/kasan/report.c:306 [inline]
     print_report+0x175/0x478 mm/kasan/report.c:417
     kasan_report+0xb1/0x130 mm/kasan/report.c:517
     l2cap_sock_ready_cb+0xb7/0x100 net/bluetooth/l2cap_sock.c:1650
     l2cap_chan_ready+0x10e/0x1e0 net/bluetooth/l2cap_core.c:1386
     l2cap_config_req+0x753/0x9f0 net/bluetooth/l2cap_core.c:4480
     l2cap_bredr_sig_cmd net/bluetooth/l2cap_core.c:5739 [inline]
     l2cap_sig_channel net/bluetooth/l2cap_core.c:6509 [inline]
     l2cap_recv_frame+0xe2e/0x43c0 net/bluetooth/l2cap_core.c:7788
     l2cap_recv_acldata+0x6ed/0x7e0 net/bluetooth/l2cap_core.c:8506
     hci_acldata_packet net/bluetooth/hci_core.c:3813 [inline]
     hci_rx_work+0x66e/0xbc0 net/bluetooth/hci_core.c:4048
     process_one_work+0x4ea/0x8e0 kernel/workqueue.c:2289
     worker_thread+0x364/0x8e0 kernel/workqueue.c:2436
     kthread+0x1b9/0x200 kernel/kthread.c:376
     ret_from_fork+0x2c/0x50 arch/x86/entry/entry_64.S:308
     </TASK>

    Allocated by task 288:
     kasan_save_stack+0x22/0x50 mm/kasan/common.c:45
     kasan_set_track+0x25/0x30 mm/kasan/common.c:52

```
       ____kasan_kmalloc mm/kasan/common.c:374 [inline]
       __kasan_kmalloc+0x82/0x90 mm/kasan/common.c:383
       kasan_kmalloc include/linux/kasan.h:211 [inline]
       __do_kmalloc_node mm/slab_common.c:968 [inline]
       __kmalloc+0x5a/0x140 mm/slab_common.c:981
       kmalloc include/linux/slab.h:584 [inline]
       sk_prot_alloc+0x113/0x1f0 net/core/sock.c:2040
       sk_alloc+0x36/0x3c0 net/core/sock.c:2093
       l2cap_sock_alloc.constprop.0+0x39/0x1c0 net/bluetooth/l2cap_sock.c:1852
       l2cap_sock_create+0x10d/0x220 net/bluetooth/l2cap_sock.c:1898
       bt_sock_create+0x183/0x290 net/bluetooth/af_bluetooth.c:132
       __sock_create+0x226/0x380 net/socket.c:1518
       sock_create net/socket.c:1569 [inline]
       __sys_socket_create net/socket.c:1606 [inline]
       __sys_socket_create net/socket.c:1591 [inline]
       __sys_socket+0x112/0x200 net/socket.c:1639
       __do_sys_socket net/socket.c:1652 [inline]
       __se_sys_socket net/socket.c:1650 [inline]
       __x64_sys_socket+0x40/0x50 net/socket.c:1650
       do_syscall_x64 arch/x86/entry/common.c:50 [inline]
       do_syscall_64+0x3f/0x90 arch/x86/entry/common.c:80
       entry_SYSCALL_64_after_hwframe+0x72/0xdc

     Freed by task 288:
       kasan_save_stack+0x22/0x50 mm/kasan/common.c:45
       kasan_set_track+0x25/0x30 mm/kasan/common.c:52
       kasan_save_free_info+0x2e/0x50 mm/kasan/generic.c:523
       ____kasan_slab_free mm/kasan/common.c:236 [inline]
       ____kasan_slab_free mm/kasan/common.c:200 [inline]
       __kasan_slab_free+0x10a/0x190 mm/kasan/common.c:244
       kasan_slab_free include/linux/kasan.h:177 [inline]
       slab_free_hook mm/slub.c:1781 [inline]
       slab_free_freelist_hook mm/slub.c:1807 [inline]
       slab_free mm/slub.c:3787 [inline]
       __kmem_cache_free+0x88/0x1f0 mm/slub.c:3800
       sk_prot_free net/core/sock.c:2076 [inline]
       __sk_destruct+0x347/0x430 net/core/sock.c:2168
       sk_destruct+0x9c/0xb0 net/core/sock.c:2183
       __sk_free+0x82/0x220 net/core/sock.c:2194
       sk_free+0x7c/0xa0 net/core/sock.c:2205
       sock_put include/net/sock.h:1991 [inline]
       l2cap_sock_kill+0x256/0x2b0 net/bluetooth/l2cap_sock.c:1257
       l2cap_sock_release+0x1a7/0x220 net/bluetooth/l2cap_sock.c:1428
       __sock_release+0x80/0x150 net/socket.c:650
       sock_close+0x19/0x30 net/socket.c:1368
       __fput+0x17a/0x5c0 fs/file_table.c:320
       task_work_run+0x132/0x1c0 kernel/task_work.c:179
       resume_user_mode_work include/linux/resume_user_mode.h:49 [inline]
       exit_to_user_mode_loop kernel/entry/common.c:171 [inline]
       exit_to_user_mode_prepare+0x113/0x120 kernel/entry/common.c:203
       __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline]
       syscall_exit_to_user_mode+0x21/0x50 kernel/entry/common.c:296
       do_syscall_64+0x4c/0x90 arch/x86/entry/common.c:86
       entry_SYSCALL_64_after_hwframe+0x72/0xdc

     The buggy address belongs to the object at ffff888104617800
       which belongs to the cache kmalloc-1k of size 1024
     The buggy address is located 680 bytes inside of
       1024-byte region [ffff888104617800, ffff888104617c00)

     The buggy address belongs to the physical page:
     page:00000000dbca6a80 refcount:1 mapcount:0 mapping:0000000000000000
  index:0xffff888104614000 pfn:0x104614
     head:00000000dbca6a80 order:2 compound_mapcount:0 subpages_mapcount:0
  compound_pincount:0
```

```
    flags: 0x200000000010200(slab|head|node=0|zone=2)
    raw: 0200000000010200 ffff888100041dc0 ffffea0004212c10 ffffea0004234b10
    raw: ffff888104614000 0000000000080002 00000001ffffffff 0000000000000000
    page dumped because: kasan: bad access detected

    Memory state around the buggy address:
     ffff888104617980: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     ffff888104617a00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
    >ffff888104617a80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
                                     ^
     ffff888104617b00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
     ffff888104617b80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
    ==================================================================

    Ack: This bug is found by FuzzBT with a modified Syzkaller. Other
    contributors are Ruoyu Wu and Hui Peng.
    Signed-off-by: Sungwoo Kim <iam@sung-woo.kim>
    Signed-off-by: Luiz Augusto von Dentz <luiz.von.dentz@intel.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e28f376dd8dfcc4e880ac101184132bc08703f6e
Author: Prince Kumar Maurya <princekumarmaurya06@gmail.com>
Date:   Tue May 30 18:31:41 2023 -0700

    fs/sysv: Null check to prevent null-ptr-deref bug

    commit ea2b62f305893992156a798f665847e0663c9f41 upstream.

    sb_getblk(inode->i_sb, parent) return a null ptr and taking lock on
    that leads to the null-ptr-deref bug.

    Reported-by: syzbot+aad58150cbc64ba41bdc@syzkaller.appspotmail.com
    Closes: https://syzkaller.appspot.com/bug?extid=aad58150cbc64ba41bdc
    Signed-off-by: Prince Kumar Maurya <princekumarmaurya06@gmail.com>
    Message-Id: <20230531013141.19487-1-princekumarmaurya06@gmail.com>
    Signed-off-by: Christian Brauner <brauner@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e0a30f9d1a5ecacbbb3aa5bf8d1e039998278ac6
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Sun May 28 00:25:31 2023 +0900

    kasan,kmsan: remove __GFP_KSWAPD_RECLAIM usage from kasan/kmsan

    commit 726ccdba1521007fab4b2b7565d255fa0f2b770c upstream.

    syzbot is reporting lockdep warning in __stack_depot_save(), for
    the caller of __stack_depot_save() (i.e. __kasan_record_aux_stack() in
    this report) is responsible for masking __GFP_KSWAPD_RECLAIM flag in
    order not to wake kswapd which in turn wakes kcompactd.

    Since kasan/kmsan functions might be called with arbitrary locks held,
    mask __GFP_KSWAPD_RECLAIM flag from all GFP_NOWAIT/GFP_ATOMIC allocations
    in kasan/kmsan.

    Note that kmsan_save_stack_with_flags() is changed to mask both
    __GFP_DIRECT_RECLAIM flag and __GFP_KSWAPD_RECLAIM flag, for
    wakeup_kswapd() from wake_all_kswapds() from __alloc_pages_slowpath()
    calls wakeup_kcompactd() if __GFP_KSWAPD_RECLAIM flag is set and
    __GFP_DIRECT_RECLAIM flag is not set.

    Link: https://lkml.kernel.org/r/656cb4f5-998b-c8d7-3c61-c2d37aa90f9a@I-love.SAKURA.ne.jp
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Reported-by: syzbot <syzbot+ece2915262061d6e0ac1@syzkaller.appspotmail.com>
```

```
    Closes: https://syzkaller.appspot.com/bug?extid=ece2915262061d6e0ac1
    Reviewed-by: "Huang, Ying" <ying.huang@intel.com>
    Reviewed-by: Alexander Potapenko <glider@google.com>
    Cc: Andrey Konovalov <andreyknvl@gmail.com>
    Cc: Andrey Ryabinin <ryabinin.a.a@gmail.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Cc: Marco Elver <elver@google.com>
    Cc: Mel Gorman <mgorman@techsingularity.net>
    Cc: Vincenzo Frascino <vincenzo.frascino@arm.com>
    Cc: Vlastimil Babka <vbabka@suse.cz>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 664dbb356f671439337d87cc5582638eeb3c47cc
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Tue Mar 28 20:05:16 2023 +0900

    fs/ntfs3: Use __GFP_NOWARN allocation at ntfs_load_attr_list()

    commit ea303f72d70ce2f0b0aa94ab127085289768c5a6 upstream.

    syzbot is reporting too large allocation at ntfs_load_attr_list(), for
    a crafted filesystem can have huge data_size.

    Reported-by: syzbot <syzbot+89dbb3a789a5b9711793@syzkaller.appspotmail.com>
    Link: https://syzkaller.appspot.com/bug?extid=89dbb3a789a5b9711793
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Signed-off-by: Konstantin Komarov <almaz.alexandrovich@paragon-software.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 33391c7e1a2ad612bf3922cc168cb09a46bbe236
Author: Roman Gushchin <roman.gushchin@linux.dev>
Date:   Tue May 2 09:08:38 2023 -0700

    mm: kmem: fix a NULL pointer dereference in obj_stock_flush_required()

    commit 3b8abb3239530c423c0b97e42af7f7e856e1ee96 upstream.

    KCSAN found an issue in obj_stock_flush_required():
    stock->cached_objcg can be reset between the check and dereference:

    ==================================================================
    BUG: KCSAN: data-race in drain_all_stock / drain_obj_stock

    write to 0xffff888237c2a2f8 of 8 bytes by task 19625 on cpu 0:
     drain_obj_stock+0x408/0x4e0 mm/memcontrol.c:3306
     refill_obj_stock+0x9c/0x1e0 mm/memcontrol.c:3340
     obj_cgroup_uncharge+0xe/0x10 mm/memcontrol.c:3408
     memcg_slab_free_hook mm/slab.h:587 [inline]
     __cache_free mm/slab.c:3373 [inline]
     __do_kmem_cache_free mm/slab.c:3577 [inline]
     kmem_cache_free+0x105/0x280 mm/slab.c:3602
     __d_free fs/dcache.c:298 [inline]
     dentry_free fs/dcache.c:375 [inline]
     __dentry_kill+0x422/0x4a0 fs/dcache.c:621
     dentry_kill+0x8d/0x1e0
     dput+0x118/0x1f0 fs/dcache.c:913
     __fput+0x3bf/0x570 fs/file_table.c:329
     ____fput+0x15/0x20 fs/file_table.c:349
     task_work_run+0x123/0x160 kernel/task_work.c:179
     resume_user_mode_work include/linux/resume_user_mode.h:49 [inline]
     exit_to_user_mode_loop+0xcf/0xe0 kernel/entry/common.c:171
     exit_to_user_mode_prepare+0x6a/0xa0 kernel/entry/common.c:203
     __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline]
     syscall_exit_to_user_mode+0x26/0x140 kernel/entry/common.c:296
```

```
       do_syscall_64+0x4d/0xc0 arch/x86/entry/common.c:86
       entry_SYSCALL_64_after_hwframe+0x63/0xcd

      read to 0xffff888237c2a2f8 of 8 bytes by task 19632 on cpu 1:
       obj_stock_flush_required mm/memcontrol.c:3319 [inline]
       drain_all_stock+0x174/0x2a0 mm/memcontrol.c:2361
       try_charge_memcg+0x6d0/0xd10 mm/memcontrol.c:2703
       try_charge mm/memcontrol.c:2837 [inline]
       mem_cgroup_charge_skmem+0x51/0x140 mm/memcontrol.c:7290
       sock_reserve_memory+0xb1/0x390 net/core/sock.c:1025
       sk_setsockopt+0x800/0x1e70 net/core/sock.c:1525
       udp_lib_setsockopt+0x99/0x6c0 net/ipv4/udp.c:2692
       udp_setsockopt+0x73/0xa0 net/ipv4/udp.c:2817
       sock_common_setsockopt+0x61/0x70 net/core/sock.c:3668
       __sys_setsockopt+0x1c3/0x230 net/socket.c:2271
       __do_sys_setsockopt net/socket.c:2282 [inline]
       __se_sys_setsockopt net/socket.c:2279 [inline]
       __x64_sys_setsockopt+0x66/0x80 net/socket.c:2279
       do_syscall_x64 arch/x86/entry/common.c:50 [inline]
       do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80
       entry_SYSCALL_64_after_hwframe+0x63/0xcd

      value changed: 0xffff8881382d52c0 -> 0xffff888138893740

      Reported by Kernel Concurrency Sanitizer on:
      CPU: 1 PID: 19632 Comm: syz-executor.0 Not tainted 6.3.0-rc2-syzkaller-00387-
  g534293368afa #0
      Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
  03/02/2023

      Fix it by using READ_ONCE()/WRITE_ONCE() for all accesses to
      stock->cached_objcg.

      Link: https://lkml.kernel.org/r/20230502160839.361544-1-roman.gushchin@linux.dev
      Fixes: bf4f059954dc ("mm: memcg/slab: obj_cgroup API")
      Signed-off-by: Roman Gushchin <roman.gushchin@linux.dev>
      Reported-by: syzbot+774c29891415ab0fd29d@syzkaller.appspotmail.com
      Reported-by: Dmitry Vyukov <dvyukov@google.com>
        Link: https://lore.kernel.org/linux-mm/CACT4Y+ZfucZhM60YPphWiCLJr6+SGFhT+jjm8k1P-
  a_8Kkxsjg@mail.gmail.com/T/#t
      Reviewed-by: Yosry Ahmed <yosryahmed@google.com>
      Acked-by: Shakeel Butt <shakeelb@google.com>
      Reviewed-by: Dmitry Vyukov <dvyukov@google.com>
      Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 11e760b5e7f25c54392aafb54c4bbe489f1aa443
  Author: Linus Torvalds <torvalds@linux-foundation.org>
  Date:   Thu Aug 3 11:35:53 2023 -0700

      file: reinstate f_pos locking optimization for regular files

      commit 797964253d358cf8d705614dda394dbe30120223 upstream.

      In commit 20ea1e7d13c1 ("file: always lock position for
      FMODE_ATOMIC_POS") we ended up always taking the file pos lock, because
      pidfd_getfd() could get a reference to the file even when it didn't have
      an elevated file count due to threading of other sharing cases.

      But Mateusz Guzik reports that the extra locking is actually measurable,
      so let's re-introduce the optimization, and only force the locking for
      directory traversal.

      Directories need the lock for correctness reasons, while regular files
      only need it for "POSIX semantics".  Since pidfd_getfd() is about
```

```
    debuggers etc special things that are _way_ outside of POSIX, we can
    relax the rules for that case.

    Reported-by: Mateusz Guzik <mjguzik@gmail.com>
    Cc: Christian Brauner <brauner@kernel.org>
    Link: https://lore.kernel.org/linux-fsdevel/20230803095311.ijpvhx3fyrbkasul@f/
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 158b3678c86b38ff0d89cc4c3ea19ae9e555f774
Author: Geert Uytterhoeven <geert+renesas@glider.be>
Date:   Tue Jul 11 17:08:12 2023 +0200

    clk: imx93: Propagate correct error in imx93_clocks_probe()

    commit a29b2fccf5f2689a9637be85ff1f51c834c6fb33 upstream.

    smatch reports:

        drivers/clk/imx/clk-imx93.c:294 imx93_clocks_probe() error: uninitialized symbol
'base'.

    Indeed, in case of an error, the wrong (yet uninitialized) variable is
    converted to an error code and returned.
    Fix this by propagating the error code in the correct variable.

    Fixes: e02ba11b45764705 ("clk: imx93: fix memory leak and missing unwind goto in
imx93_clocks_probe")
    Reported-by: Dan Carpenter <dan.carpenter@linaro.org>
    Closes: https://lore.kernel.org/all/9c2acd81-3ad8-485d-819e-9e4201277831@kadam.mountain
    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/all/202306161533.4YDmL22b-lkp@intel.com/
    Signed-off-by: Geert Uytterhoeven <geert+renesas@glider.be>
    Link: https://lore.kernel.org/r/20230711150812.3562221-1-geert+renesas@glider.be
    Reviewed-by: Peng Fan <peng.fan@nxp.com>
    Signed-off-by: Stephen Boyd <sboyd@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ee9968d72af899dc4d192761e807aa7051f637b6
Author: Stephen Rothwell <sfr@canb.auug.org.au>
Date:   Tue Jun 13 16:46:39 2023 +1000

    sunvnet: fix sparc64 build error after gso code split

    commit d9ffa069e006fa2873b94fbf2387546942d4f85b upstream.

    After merging the net-next tree, today's linux-next build (sparc64
    defconfig) failed like this:

    drivers/net/ethernet/sun/sunvnet_common.c: In function 'vnet_handle_offloads':
    drivers/net/ethernet/sun/sunvnet_common.c:1277:16: error: implicit declaration of
function 'skb_gso_segment'; did you mean 'skb_gso_reset'? [-Werror=implicit-function-
declaration]
     1277 |         segs = skb_gso_segment(skb, dev->features & ~NETIF_F_TSO);
          |                ^~~~~~~~~~~~~~~
          |                skb_gso_reset
    drivers/net/ethernet/sun/sunvnet_common.c:1277:14: warning: assignment to 'struct
sk_buff *' from 'int' makes pointer from integer without a cast [-Wint-conversion]
     1277 |         segs = skb_gso_segment(skb, dev->features & ~NETIF_F_TSO);
          |              ^

    Fixes: d457a0e329b0 ("net: move gso declarations and functions to their own files")
    Signed-off-by: Stephen Rothwell <sfr@canb.auug.org.au>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Link: https://lore.kernel.org/r/20230613164639.164b2991@canb.auug.org.au
```

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

commit 5415dde3e76a9e6386f6f03e9014c0c5c4046a7b
Author: Mike Kravetz <mike.kravetz@oracle.com>
Date:   Wed Jun 21 14:24:02 2023 -0700

    Revert "page cache: fix page_cache_next/prev_miss off by one"

    commit 16f8eb3eea9eb2a1568279d64ca4dc977e7aa538 upstream.

    This reverts commit 9425c591e06a9ab27a145ba655fb50532cf0bcc9

    The reverted commit fixed up routines primarily used by readahead code
    such that they could also be used by hugetlb.  Unfortunately, this
    caused a performance regression as pointed out by the Closes: tag.

    The hugetlb code which uses page_cache_next_miss will be addressed in
    a subsequent patch.

    Link: https://lkml.kernel.org/r/20230621212403.174710-1-mike.kravetz@oracle.com
    Fixes: 9425c591e06a ("page cache: fix page_cache_next/prev_miss off by one")
    Signed-off-by: Mike Kravetz <mike.kravetz@oracle.com>
    Reported-by: kernel test robot <oliver.sang@intel.com>
    Closes: https://lore.kernel.org/oe-lkp/202306211346.1e9ff03e-oliver.sang@intel.com
    Reviewed-by: Sidhartha Kumar <sidhartha.kumar@oracle.com>
    Cc: Ackerley Tng <ackerleytng@google.com>
    Cc: Erdem Aktas <erdemaktas@google.com>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Muchun Song <songmuchun@bytedance.com>
    Cc: Vishal Annapurve <vannapurve@google.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6df7dd9270c580e9aa2ab4872168c61fd7279bb3
Author: Andi Shyti <andi.shyti@linux.intel.com>
Date:   Tue Jul 25 02:19:44 2023 +0200

    drm/i915/gt: Cleanup aux invalidation registers

    commit d14560ac1b595aa2e792365e91fea6aeaee66c2b upstream.

    Fix the 'NV' definition postfix that is supposed to be INV.

    Take the chance to also order properly the registers based on
    their address and call the GEN12_GFX_CCS_AUX_INV address as
    GEN12_CCS_AUX_INV like all the other similar registers.

    Remove also VD1, VD3 and VE1 registers that don't exist and add
    BCS0 and CCS0.

    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Cc: <stable@vger.kernel.org> # v5.8+
    Reviewed-by: Nirmoy Das <nirmoy.das@intel.com>
    Reviewed-by: Andrzej Hajda <andrzej.hajda@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230725001950.1014671-2-
andi.shyti@linux.intel.com
    (cherry picked from commit 2f0b927d3ca3440445975ebde27f3df1c3ed6f76)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ed7ac41cfcefcefd0f278eb17ca9a0f69e81cb9e
Author: Janusz Krzysztofik <janusz.krzysztofik@linux.intel.com>
Date:   Thu Jul 20 11:35:44 2023 +0200
```

drm/i915: Fix premature release of request's reusable memory

commit a337b64f0d5717248a0c894e2618e658e6a9de9f upstream.

Infinite waits for completion of GPU activity have been observed in CI,
mostly inside __i915_active_wait(), triggered by igt@gem_barrier_race or
igt@perf@stress-open-close. Root cause analysis, based of ftrace dumps
generated with a lot of extra trace_printk() calls added to the code,
revealed loops of request dependencies being accidentally built,
preventing the requests from being processed, each waiting for completion
of another one's activity.

After we substitute a new request for a last active one tracked on a
timeline, we set up a dependency of our new request to wait on completion
of current activity of that previous one. While doing that, we must take
care of keeping the old request still in memory until we use its
attributes for setting up that await dependency, or we can happen to set
up the await dependency on an unrelated request that already reuses the
memory previously allocated to the old one, already released. Combined
with perf adding consecutive kernel context remote requests to different
user context timelines, unresolvable loops of await dependencies can be
built, leading do infinite waits.

We obtain a pointer to the previous request to wait upon when we
substitute it with a pointer to our new request in an active tracker,
e.g. in intel_timeline.last_request. In some processing paths we protect
that old request from being freed before we use it by getting a reference
to it under RCU protection, but in others, e.g. __i915_request_commit()
-> __i915_request_add_to_timeline() -> __i915_request_ensure_ordering(),
we don't. But anyway, since the requests' memory is SLAB_FAILSAFE_BY_RCU,
that RCU protection is not sufficient against reuse of memory.

We could protect i915_request's memory from being prematurely reused by
calling its release function via call_rcu() and using rcu_read_lock()
consequently, as proposed in v1. However, that approach leads to
significant (up to 10 times) increase of SLAB utilization by i915_request
SLAB cache. Another potential approach is to take a reference to the
previous active fence.

When updating an active fence tracker, we first lock the new fence,
substitute a pointer of the current active fence with the new one, then we
lock the substituted fence. With this approach, there is a time window
after the substitution and before the lock when the request can be
concurrently released by an interrupt handler and its memory reused, then
we may happen to lock and return a new, unrelated request.

Always get a reference to the current active fence first, before
replacing it with a new one. Having it protected from premature release
and reuse, lock it and then replace with the new one but only if not
yet signalled via a potential concurrent interrupt nor replaced with
another one by a potential concurrent thread, otherwise retry, starting
from getting a reference to the new current one. Adjust users to not
get a reference to the previous active fence themselves and always put the
reference got by __i915_active_fence_set() when no longer needed.

v3: Fix lockdep splat reports and other issues caused by incorrect use of
    try_cmpxchg() (use (cmpxchg() != prev) instead)
v2: Protect request's memory by getting a reference to it in favor of
    delegating its release to call_rcu() (Chris)

Closes: https://gitlab.freedesktop.org/drm/intel/-/issues/8211
Fixes: df9f85d8582e ("drm/i915: Serialise i915_active_fence_set() with itself")
Suggested-by: Chris Wilson <chris@chris-wilson.co.uk>
Signed-off-by: Janusz Krzysztofik <janusz.krzysztofik@linux.intel.com>

```
    Cc: <stable@vger.kernel.org> # v5.6+
    Reviewed-by: Andi Shyti <andi.shyti@linux.intel.com>
    Signed-off-by: Andi Shyti <andi.shyti@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20230720093543.832147-2-
janusz.krzysztofik@linux.intel.com
    (cherry picked from commit 946e047a3d88d46d15b5c5af0414098e12b243f7)
    Signed-off-by: Tvrtko Ursulin <tvrtko.ursulin@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 49b3b979e79faef129605018ad82aa0f2258f2f7
Author: Guchun Chen <guchun.chen@amd.com>
Date:   Mon Jul 24 10:42:29 2023 +0800

    drm/ttm: check null pointer before accessing when swapping

    commit 2dedcf414bb01b8d966eb445db1d181d92304fb2 upstream.

    Add a check to avoid null pointer dereference as below:

    [   90.002283] general protection fault, probably for non-canonical
    address 0xdfffc0000000000: 0000 [#1] PREEMPT SMP KASAN NOPTI
    [   90.002292] KASAN: null-ptr-deref in range
    [0x0000000000000000-0x0000000000000007]
    [   90.002346]  ? exc_general_protection+0x159/0x240
    [   90.002352]  ? asm_exc_general_protection+0x26/0x30
    [   90.002357]  ? ttm_bo_evict_swapout_allowable+0x322/0x5e0 [ttm]
    [   90.002365]  ? ttm_bo_evict_swapout_allowable+0x42e/0x5e0 [ttm]
    [   90.002373]  ttm_bo_swapout+0x134/0x7f0 [ttm]
    [   90.002383]  ? __pfx_ttm_bo_swapout+0x10/0x10 [ttm]
    [   90.002391]  ? lock_acquire+0x44d/0x4f0
    [   90.002398]  ? ttm_device_swapout+0xa5/0x260 [ttm]
    [   90.002412]  ? lock_acquired+0x355/0xa00
    [   90.002416]  ? do_raw_spin_trylock+0xb6/0x190
    [   90.002421]  ? __pfx_lock_acquired+0x10/0x10
    [   90.002426]  ? ttm_global_swapout+0x25/0x210 [ttm]
    [   90.002442]  ttm_device_swapout+0x198/0x260 [ttm]
    [   90.002456]  ? __pfx_ttm_device_swapout+0x10/0x10 [ttm]
    [   90.002472]  ttm_global_swapout+0x75/0x210 [ttm]
    [   90.002486]  ttm_tt_populate+0x187/0x3f0 [ttm]
    [   90.002501]  ttm_bo_handle_move_mem+0x437/0x590 [ttm]
    [   90.002517]  ttm_bo_validate+0x275/0x430 [ttm]
    [   90.002530]  ? __pfx_ttm_bo_validate+0x10/0x10 [ttm]
    [   90.002544]  ? kasan_save_stack+0x33/0x60
    [   90.002550]  ? kasan_set_track+0x25/0x30
    [   90.002554]  ? __kasan_kmalloc+0x8f/0xa0
    [   90.002558]  ? amdgpu_gtt_mgr_new+0x81/0x420 [amdgpu]
    [   90.003023]  ? ttm_resource_alloc+0xf6/0x220 [ttm]
    [   90.003038]  amdgpu_bo_pin_restricted+0x2dd/0x8b0 [amdgpu]
    [   90.003210]  ? __x64_sys_ioctl+0x131/0x1a0
    [   90.003210]  ? do_syscall_64+0x60/0x90

    Fixes: a2848d08742c ("drm/ttm: never consider pinned BOs for eviction&swap")
    Tested-by: Mikhail Gavrilov <mikhail.v.gavrilov@gmail.com>
    Signed-off-by: Guchun Chen <guchun.chen@amd.com>
    Reviewed-by: Alex Deucher <alexander.deucher@amd.com>
    Reviewed-by: Christian König <christian.koenig@amd.com>
    Cc: stable@vger.kernel.org
    Link: https://patchwork.freedesktop.org/patch/msgid/20230724024229.1118444-1-
guchun.chen@amd.com
    Signed-off-by: Christian König <christian.koenig@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e3c24712441ec156e64f41776d810c7af756e75a
Author: Aleksa Sarai <cyphar@cyphar.com>
Date:   Sun Aug 6 02:11:58 2023 +1000
```

```
    open: make RESOLVE_CACHED correctly test for O_TMPFILE

    commit a0fc452a5d7fed986205539259df1d60546f536c upstream.

    O_TMPFILE is actually __O_TMPFILE|O_DIRECTORY. This means that the old
    fast-path check for RESOLVE_CACHED would reject all users passing
    O_DIRECTORY with -EAGAIN, when in fact the intended test was to check
    for __O_TMPFILE.

    Cc: stable@vger.kernel.org # v5.12+
    Fixes: 99668f618062 ("fs: expose LOOKUP_CACHED through openat2() RESOLVE_CACHED")
    Signed-off-by: Aleksa Sarai <cyphar@cyphar.com>
    Message-Id: <20230806-resolve_cached-o_tmpfile-v1-1-7ba16308465e@cyphar.com>
    Signed-off-by: Christian Brauner <brauner@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 64b7ae70678bb074f1b7e606483ff24ca36764e2
Author: Mark Brown <broonie@kernel.org>
Date:   Thu Aug 3 19:33:21 2023 +0100

    arm64/ptrace: Don't enable SVE when setting streaming SVE

    commit 045aecdfcb2e060db142d83a0f4082380c465d2c upstream.

    Systems which implement SME without also implementing SVE are
    architecturally valid but were not initially supported by the kernel,
    unfortunately we missed one issue in the ptrace code.

    The SVE register setting code is shared between SVE and streaming mode
    SVE. When we set full SVE register state we currently enable TIF_SVE
    unconditionally, in the case where streaming SVE is being configured on a
    system that supports vanilla SVE this is not an issue since we always
    initialise enough state for both vector lengths but on a system which only
    support SME it will result in us attempting to restore the SVE vector
    length after having set streaming SVE registers.

    Fix this by making the enabling of SVE conditional on setting SVE vector
    state. If we set streaming SVE state and SVE was not already enabled this
    will result in a SVE access trap on next use of normal SVE, this will cause
    us to flush our register state but this is fine since the only way to
    trigger a SVE access trap would be to exit streaming mode which will cause
    the in register state to be flushed anyway.

    Fixes: e12310a0d30f ("arm64/sme: Implement ptrace support for streaming mode SVE
registers")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230803-arm64-fix-ptrace-ssve-no-sve-v1-1-
49df214bfb3e@kernel.org
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d22576f19038a01572e51b9ebde30dbf1f52b53
Author: Mark Brown <broonie@kernel.org>
Date:   Thu Aug 3 01:19:06 2023 +0100

    arm64/ptrace: Flush FP state when setting ZT0

    commit 89a65c3f170e5c3b05a626046c68354e2afd7912 upstream.

    When setting ZT0 via ptrace we do not currently force a reload of the
    floating point register state from memory, do that to ensure that the newly
    set value gets loaded into the registers on next task execution.
```

```
    The function was templated off the function for FPSIMD which due to our
    providing the option of embedding a FPSIMD regset within the SVE regset
    does not directly include the flush.

    Fixes: f90b529bcbe5 ("arm64/sme: Implement ZT0 ptrace support")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230803-arm64-fix-ptrace-zt0-flush-v1-1-
72e854eaf96e@kernel.org
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cfb2b39db5383d1cd269dd140dc2239cb291c726
Author: Mark Brown <broonie@kernel.org>
Date:   Thu Aug 3 19:33:22 2023 +0100

    arm64/fpsimd: Sync FPSIMD state with SVE for SME only systems

    commit 507ea5dd92d23fcf10e4d1a68a443c86a49753ed upstream.

    Currently we guard FPSIMD/SVE state conversions with a check for the system
    supporting SVE but SME only systems may need to sync streaming mode SVE
    state so add a check for SME support too.  These functions are only used
    by the ptrace code.

    Fixes: e12310a0d30f ("arm64/sme: Implement ptrace support for streaming mode SVE
registers")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230803-arm64-fix-ptrace-ssve-no-sve-v1-2-
49df214bfb3e@kernel.org
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 789c015d9a18e882993009ac5e1e5f1499895dba
Author: Mark Brown <broonie@kernel.org>
Date:   Thu Aug 3 00:46:39 2023 +0100

    arm64/fpsimd: Clear SME state in the target task when setting the VL

    commit c9bb40b7f786662e33d71afe236442b0b61f0446 upstream.

    When setting SME vector lengths we clear TIF_SME to reenable SME traps,
    doing a reallocation of the backing storage on next use. We do this using
    clear_thread_flag() which operates on the current thread, meaning that when
    setting the vector length via ptrace we may both not force traps for the
    target task and force a spurious flush of any SME state that the tracing
    task may have.

    Clear the flag in the target task.

    Fixes: e12310a0d30f ("arm64/sme: Implement ptrace support for streaming mode SVE
registers")
    Reported-by: David Spickett <David.Spickett@arm.com>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230803-arm64-fix-ptrace-tif-sme-v1-1-
88312fd6fbfd@kernel.org
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0693012274fd01f35981e943e13ef712a09f4abc
Author: Mark Brown <broonie@kernel.org>
Date:   Thu Aug 3 19:33:23 2023 +0100
```

```
    arm64/fpsimd: Sync and zero pad FPSIMD state for streaming SVE

    commit 69af56ae56a48a2522aad906c4461c6c7c092737 upstream.

    We have a function sve_sync_from_fpsimd_zeropad() which is used by the
    ptrace code to update the SVE state when the user writes to the the
    FPSIMD register set.  Currently this checks that the task has SVE
    enabled but this will miss updates for tasks which have streaming SVE
    enabled if SVE has not been enabled for the thread, also do the
    conversion if the task has streaming SVE enabled.

    Fixes: e12310a0d30f ("arm64/sme: Implement ptrace support for streaming mode SVE
registers")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20230803-arm64-fix-ptrace-ssve-no-sve-v1-3-
49df214bfb3e@kernel.org
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 434ebb3960b7f4a013f02422b2677b042f58ed52
Author: Mike Rapoport (IBM) <rppt@kernel.org>
Date:   Thu Aug 3 09:24:04 2023 +0300

    parisc/mm: preallocate fixmap page tables at init

    commit c2ff2b736c41cc63bb0aaec85cccfead9fbcfe92 upstream.

    Christoph Biedl reported early OOM on recent kernels:

        swapper: page allocation failure: order:0, mode:0x100(__GFP_ZERO),
    nodemask=(null)
        CPU: 0 PID: 0 Comm: swapper Not tainted 6.3.0-rc4+ #16
        Hardware name: 9000/785/C3600
        Backtrace:
         [<10408594>] show_stack+0x48/0x5c
         [<10e152d8>] dump_stack_lvl+0x48/0x64
         [<10e15318>] dump_stack+0x24/0x34
         [<105cf7f8>] warn_alloc+0x10c/0x1c8
         [<105d068c>] __alloc_pages+0xbbc/0xcf8
         [<105d0e4c>] __get_free_pages+0x28/0x78
         [<105ad10c>] __pte_alloc_kernel+0x30/0x98
         [<10406934>] set_fixmap+0xec/0xf4
         [<10411ad4>] patch_map.constprop.0+0xa8/0xdc
         [<10411bb0>] __patch_text_multiple+0xa8/0x208
         [<10411d78>] patch_text+0x30/0x48
         [<1041246c>] arch_jump_label_transform+0x90/0xcc
         [<1056f734>] jump_label_update+0xd4/0x184
         [<1056fc9c>] static_key_enable_cpuslocked+0xc0/0x110
         [<1056fd08>] static_key_enable+0x1c/0x2c
         [<1011362c>] init_mem_debugging_and_hardening+0xdc/0xf8
         [<1010141c>] start_kernel+0x5f0/0xa98
         [<10105da8>] start_parisc+0xb8/0xe4

        Mem-Info:
        active_anon:0 inactive_anon:0 isolated_anon:0
         active_file:0 inactive_file:0 isolated_file:0
         unevictable:0 dirty:0 writeback:0
         slab_reclaimable:0 slab_unreclaimable:0
         mapped:0 shmem:0 pagetables:0
         sec_pagetables:0 bounce:0
         kernel_misc_reclaimable:0
         free:0 free_pcp:0 free_cma:0
        Node 0 active_anon:0kB inactive_anon:0kB active_file:0kB
    inactive_file:0kB unevictable:0kB isolated(anon):0kB isolated(file):0kB
```

```
    mapped:0kB dirty:0kB writeback:0kB shmem:0kB
    +writeback_tmp:0kB kernel_stack:0kB pagetables:0kB sec_pagetables:0kB
    all_unreclaimable? no
        Normal free:0kB boost:0kB min:0kB low:0kB high:0kB
    reserved_highatomic:0KB active_anon:0kB inactive_anon:0kB active_file:0kB
    inactive_file:0kB unevictable:0kB writepending:0kB
    +present:1048576kB managed:1039360kB mlocked:0kB bounce:0kB free_pcp:0kB
    local_pcp:0kB free_cma:0kB
        lowmem_reserve[]: 0 0
        Normal: 0*4kB 0*8kB 0*16kB 0*32kB 0*64kB 0*128kB 0*256kB 0*512kB
    0*1024kB 0*2048kB 0*4096kB = 0kB
        0 total pagecache pages
        0 pages in swap cache
        Free swap  = 0kB
        Total swap = 0kB
        262144 pages RAM
        0 pages HighMem/MovableOnly
        2304 pages reserved
        Backtrace:
         [<10411d78>] patch_text+0x30/0x48
         [<1041246c>] arch_jump_label_transform+0x90/0xcc
         [<1056f734>] jump_label_update+0xd4/0x184
         [<1056fc9c>] static_key_enable_cpuslocked+0xc0/0x110
         [<1056fd08>] static_key_enable+0x1c/0x2c
         [<1011362c>] init_mem_debugging_and_hardening+0xdc/0xf8
         [<1010141c>] start_kernel+0x5f0/0xa98
         [<10105da8>] start_parisc+0xb8/0xe4

        Kernel Fault: Code=15 (Data TLB miss fault) at addr 0f7fe3c0
        CPU: 0 PID: 0 Comm: swapper Not tainted 6.3.0-rc4+ #16
        Hardware name: 9000/785/C3600

    This happens because patching static key code temporarily maps it via
    fixmap and if it happens before page allocator is initialized set_fixmap()
    cannot allocate memory using pte_alloc_kernel().

    Make sure that fixmap page tables are preallocated early so that
    pte_offset_kernel() in set_fixmap() never resorts to pte allocation.

    Signed-off-by: Mike Rapoport (IBM) <rppt@kernel.org>
    Acked-by: Vlastimil Babka <vbabka@suse.cz>
    Signed-off-by: Helge Deller <deller@gmx.de>
    Tested-by: Christoph Biedl <linux-kernel.bfrz@manchmal.in-ulm.de>
    Tested-by: John David Anglin <dave.anglin@bell.net>
    Cc: <stable@vger.kernel.org> # v6.4+
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 410b0d2d54e69d7d23c2fead6d3f45c96e57be8e
Author: Naveen N Rao <naveen@kernel.org>
Date:   Wed Jun 21 10:43:49 2023 +0530

    powerpc/ftrace: Create a dummy stackframe to fix stack unwind

    commit 41a506ef71eb38d94fe133f565c87c3e06ccc072 upstream.

    With ppc64 -mprofile-kernel and ppc32 -pg, profiling instructions to
    call into ftrace are emitted right at function entry. The instruction
    sequence used is minimal to reduce overhead. Crucially, a stackframe is
    not created for the function being traced. This breaks stack unwinding
    since the function being traced does not have a stackframe for itself.
    As such, it never shows up in the backtrace:

    /sys/kernel/debug/tracing # echo 1 > /proc/sys/kernel/stack_tracer_enabled
    /sys/kernel/debug/tracing # cat stack_trace
            Depth    Size   Location    (17 entries)
```

```
            -----    ----   --------
       0)     4144      32   ftrace_call+0x4/0x44
       1)     4112     432   get_page_from_freelist+0x26c/0x1ad0
       2)     3680     496   __alloc_pages+0x290/0x1280
       3)     3184     336   __folio_alloc+0x34/0x90
       4)     2848     176   vma_alloc_folio+0xd8/0x540
       5)     2672     272   __handle_mm_fault+0x700/0x1cc0
       6)     2400     208   handle_mm_fault+0xf0/0x3f0
       7)     2192      80   ___do_page_fault+0x3e4/0xbe0
       8)     2112     160   do_page_fault+0x30/0xc0
       9)     1952     256   data_access_common_virt+0x210/0x220
      10)     1696     400   0xc00000000f16b100
      11)     1296     384   load_elf_binary+0x804/0x1b80
      12)      912     208   bprm_execve+0x2d8/0x7e0
      13)      704      64   do_execveat_common+0x1d0/0x2f0
      14)      640     160   sys_execve+0x54/0x70
      15)      480      64   system_call_exception+0x138/0x350
      16)      416     416   system_call_common+0x160/0x2c4

    Fix this by having ftrace create a dummy stackframe for the function
    being traced. With this, backtraces now capture the function being
    traced:

    /sys/kernel/debug/tracing # cat stack_trace
            Depth    Size   Location    (17 entries)
            -----    ----   --------
      0)     3888      32   _raw_spin_trylock+0x8/0x70
      1)     3856     576   get_page_from_freelist+0x26c/0x1ad0
      2)     3280      64   __alloc_pages+0x290/0x1280
      3)     3216     336   __folio_alloc+0x34/0x90
      4)     2880     176   vma_alloc_folio+0xd8/0x540
      5)     2704     416   __handle_mm_fault+0x700/0x1cc0
      6)     2288      96   handle_mm_fault+0xf0/0x3f0
      7)     2192      48   ___do_page_fault+0x3e4/0xbe0
      8)     2144     192   do_page_fault+0x30/0xc0
      9)     1952     608   data_access_common_virt+0x210/0x220
     10)     1344      16   0xc0000000334bbb50
     11)     1328     416   load_elf_binary+0x804/0x1b80
     12)      912      64   bprm_execve+0x2d8/0x7e0
     13)      848     176   do_execveat_common+0x1d0/0x2f0
     14)      672     192   sys_execve+0x54/0x70
     15)      480      64   system_call_exception+0x138/0x350
     16)      416     416   system_call_common+0x160/0x2c4

    This results in two additional stores in the ftrace entry code, but
    produces reliable backtraces.

    Fixes: 153086644fd1 ("powerpc/ftrace: Add support for -mprofile-kernel ftrace ABI")
    Cc: stable@vger.kernel.org
    Signed-off-by: Naveen N Rao <naveen@kernel.org>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://msgid.link/20230621051349.759567-1-naveen@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f2556c93c388230197992c867e3fb39ebaa7a742
Author: Paulo Alcantara <pc@manguebit.com>
Date:   Wed Aug 2 13:43:03 2023 -0300

    smb: client: fix dfs link mount against w2k8

    commit 11260c3d608b59231f4c228147a795ab21a10b33 upstream.

    Customer reported that they couldn't mount their DFS link that was
    seen by the client as a DFS interlink -- special form of DFS link
    where its single target may point to a different DFS namespace -- and
```

```
    it turned out that it was just a regular DFS link where its referral
    header flags missed the StorageServers bit thus making the client
    think it couldn't tree connect to target directly without requiring
    further referrals.

    When the DFS link referral header flags misses the StoraServers bit
    and its target doesn't respond to any referrals, then tree connect to
    it.

    Fixes: a1c0d00572fc ("cifs: share dfs connections and supers")
    Cc: stable@vger.kernel.org
    Signed-off-by: Paulo Alcantara (SUSE) <pc@manguebit.com>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 063c9ce8e74e07bf94f99cd13146f42867875e8b
Author: Jiri Olsa <jolsa@kernel.org>
Date:   Tue Jul 25 10:42:06 2023 +0200

    bpf: Disable preemption in bpf_event_output

    commit d62cc390c2e99ae267ffe4b8d7e2e08b6c758c32 upstream.

    We received report [1] of kernel crash, which is caused by
    using nesting protection without disabled preemption.

    The bpf_event_output can be called by programs executed by
    bpf_prog_run_array_cg function that disabled migration but
    keeps preemption enabled.

    This can cause task to be preempted by another one inside the
    nesting protection and lead eventually to two tasks using same
    perf_sample_data buffer and cause crashes like:

      BUG: kernel NULL pointer dereference, address: 0000000000000001
      #PF: supervisor instruction fetch in kernel mode
      #PF: error_code(0x0010) - not-present page
      ...
      ? perf_output_sample+0x12a/0x9a0
      ? finish_task_switch.isra.0+0x81/0x280
      ? perf_event_output+0x66/0xa0
      ? bpf_event_output+0x13a/0x190
      ? bpf_event_output_data+0x22/0x40
      ? bpf_prog_dfc84bbde731b257_cil_sock4_connect+0x40a/0xacb
      ? xa_load+0x87/0xe0
      ? __cgroup_bpf_run_filter_sock_addr+0xc1/0x1a0
      ? release_sock+0x3e/0x90
      ? sk_setsockopt+0x1a1/0x12f0
      ? udp_pre_connect+0x36/0x50
      ? inet_dgram_connect+0x93/0xa0
      ? __sys_connect+0xb4/0xe0
      ? udp_setsockopt+0x27/0x40
      ? __pfx_udp_push_pending_frames+0x10/0x10
      ? __sys_setsockopt+0xdf/0x1a0
      ? __x64_sys_connect+0xf/0x20
      ? do_syscall_64+0x3a/0x90
      ? entry_SYSCALL_64_after_hwframe+0x72/0xdc

    Fixing this by disabling preemption in bpf_event_output.

    [1] https://github.com/cilium/cilium/issues/26756
    Cc: stable@vger.kernel.org
    Reported-by: Oleg "livelace" Popov <o.popov@livelace.ru>
    Closes: https://github.com/cilium/cilium/issues/26756
    Fixes: 2a916f2f546c ("bpf: Use migrate_disable/enable in array macros and cgroup/lirc
```

```
code.")
    Acked-by: Hou Tao <houtao1@huawei.com>
    Signed-off-by: Jiri Olsa <jolsa@kernel.org>
    Link: https://lore.kernel.org/r/20230725084206.580930-3-jolsa@kernel.org
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7bb7b479d1117cdd9ea366e5bd2f08c94f7ef0b8
Author: Ilya Dryomov <idryomov@gmail.com>
Date:   Tue Aug 1 19:14:24 2023 +0200

    rbd: prevent busy loop when requesting exclusive lock

    commit 9d01e07fd1bfb4daae156ab528aa196f5ac2b2bc upstream.

    Due to rbd_try_acquire_lock() effectively swallowing all but
    EBLOCKLISTED error from rbd_try_lock() ("request lock anyway") and
    rbd_request_lock() returning ETIMEDOUT error not only for an actual
    notify timeout but also when the lock owner doesn't respond, a busy
    loop inside of rbd_acquire_lock() between rbd_try_acquire_lock() and
    rbd_request_lock() is possible.

    Requesting the lock on EBUSY error (returned by get_lock_owner_info()
    if an incompatible lock or invalid lock owner is detected) makes very
    little sense.  The same goes for ETIMEDOUT error (might pop up pretty
    much anywhere if osd_request_timeout option is set) and many others.

    Just fail I/O requests on rbd_dev->acquiring_list immediately on any
    error from rbd_try_lock().

    Cc: stable@vger.kernel.org # 588159009d5b: rbd: retrieve and check lock owner twice
before blocklisting
    Cc: stable@vger.kernel.org
    Signed-off-by: Ilya Dryomov <idryomov@gmail.com>
    Reviewed-by: Dongsheng Yang <dongsheng.yang@easystack.cn>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 73626b70b361ddda7c380e52c236aa4f2487c402
Author: Michael Kelley <mikelley@microsoft.com>
Date:   Fri Jul 21 21:51:16 2023 -0700

    x86/hyperv: Disable IBT when hypercall page lacks ENDBR instruction

    commit d5ace2a776442d80674eff9ed42e737f7dd95056 upstream.

    On hardware that supports Indirect Branch Tracking (IBT), Hyper-V VMs
    with ConfigVersion 9.3 or later support IBT in the guest. However,
    current versions of Hyper-V have a bug in that there's not an ENDBR64
    instruction at the beginning of the hypercall page. Since hypercalls are
    made with an indirect call to the hypercall page, all hypercall attempts
    fail with an exception and Linux panics.

    A Hyper-V fix is in progress to add ENDBR64. But guard against the Linux
    panic by clearing X86_FEATURE_IBT if the hypercall page doesn't start
    with ENDBR. The VM will boot and run without IBT.

    If future Linux 32-bit kernels were to support IBT, additional hypercall
    page hackery would be needed to make IBT work for such kernels in a
    Hyper-V VM.

    Cc: stable@vger.kernel.org
    Signed-off-by: Michael Kelley <mikelley@microsoft.com>
    Link: https://lore.kernel.org/r/1690001476-98594-1-git-send-email-mikelley@microsoft.com
    Signed-off-by: Wei Liu <wei.liu@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 01c3a34f5ed7a396dd8e771d9e656dd519e8aae1
Author: Paul Fertser <fercerpav@gmail.com>
Date:   Mon Jun 5 10:34:07 2023 +0300

    wifi: mt76: mt7615: do not advertise 5 GHz on first phy of MT7615D (DBDC)

    commit 421033deb91521aa6a9255e495cb106741a52275 upstream.

    On DBDC devices the first (internal) phy is only capable of using
    2.4 GHz band, and the 5 GHz band is exposed via a separate phy object,
    so avoid the false advertising.

    Reported-by: Rani Hod <rani.hod@gmail.com>
    Closes: https://github.com/openwrt/openwrt/pull/12361
    Fixes: 7660a1bd0c22 ("mt76: mt7615: register ext_phy if DBDC is detected")
    Cc: stable@vger.kernel.org
    Signed-off-by: Paul Fertser <fercerpav@gmail.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Acked-by: Felix Fietkau <nbd@nbd.name>
    Signed-off-by: Kalle Valo <kvalo@kernel.org>
    Link: https://lore.kernel.org/r/20230605073408.8699-1-fercerpav@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ea6cce8d689930ba480f7b02af8d9fc686534ab0
Author: Laszlo Ersek <lersek@redhat.com>
Date:   Mon Jul 31 18:42:37 2023 +0200

    net: tap_open(): set sk_uid from current_fsuid()

    commit 5c9241f3ceab3257abe2923a59950db0dc8bb737 upstream.

    Commit 66b2c338adce initializes the "sk_uid" field in the protocol socket
    (struct sock) from the "/dev/tapX" device node's owner UID. Per original
    commit 86741ec25462 ("net: core: Add a UID field to struct sock.",
    2016-11-04), that's wrong: the idea is to cache the UID of the userspace
    process that creates the socket. Commit 86741ec25462 mentions socket() and
    accept(); with "tap", the action that creates the socket is
    open("/dev/tapX").

    Therefore the device node's owner UID is irrelevant. In most cases,
    "/dev/tapX" will be owned by root, so in practice, commit 66b2c338adce has
    no observable effect:

    - before, "sk_uid" would be zero, due to undefined behavior
      (CVE-2023-1076),

    - after, "sk_uid" would be zero, due to "/dev/tapX" being owned by root.

    What matters is the (fs)UID of the process performing the open(), so cache
    that in "sk_uid".

    Cc: Eric Dumazet <edumazet@google.com>
    Cc: Lorenzo Colitti <lorenzo@google.com>
    Cc: Paolo Abeni <pabeni@redhat.com>
    Cc: Pietro Borrello <borrello@diag.uniroma1.it>
    Cc: netdev@vger.kernel.org
    Cc: stable@vger.kernel.org
    Fixes: 66b2c338adce ("tap: tap_open(): correctly initialize socket uid")
    Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=2173435
    Signed-off-by: Laszlo Ersek <lersek@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 36161e7d40e7293d7f213e16d881042d15c8a53a
```

```
Author: Laszlo Ersek <lersek@redhat.com>
Date:   Mon Jul 31 18:42:36 2023 +0200

    net: tun_chr_open(): set sk_uid from current_fsuid()

    commit 9bc3047374d5bec163e83e743709e23753376f0c upstream.

    Commit a096ccca6e50 initializes the "sk_uid" field in the protocol socket
    (struct sock) from the "/dev/net/tun" device node's owner UID. Per
    original commit 86741ec25462 ("net: core: Add a UID field to struct
    sock.", 2016-11-04), that's wrong: the idea is to cache the UID of the
    userspace process that creates the socket. Commit 86741ec25462 mentions
    socket() and accept(); with "tun", the action that creates the socket is
    open("/dev/net/tun").

    Therefore the device node's owner UID is irrelevant. In most cases,
    "/dev/net/tun" will be owned by root, so in practice, commit a096ccca6e50
    has no observable effect:

    - before, "sk_uid" would be zero, due to undefined behavior
      (CVE-2023-1076),

    - after, "sk_uid" would be zero, due to "/dev/net/tun" being owned by root.

    What matters is the (fs)UID of the process performing the open(), so cache
    that in "sk_uid".

    Cc: Eric Dumazet <edumazet@google.com>
    Cc: Lorenzo Colitti <lorenzo@google.com>
    Cc: Paolo Abeni <pabeni@redhat.com>
    Cc: Pietro Borrello <borrello@diag.uniroma1.it>
    Cc: netdev@vger.kernel.org
    Cc: stable@vger.kernel.org
    Fixes: a096ccca6e50 ("tun: tun_chr_open(): correctly initialize socket uid")
    Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=2173435
    Signed-off-by: Laszlo Ersek <lersek@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 040d7f19f2066449d292b8394ff483a09bd86412
Author: Dinh Nguyen <dinguyen@kernel.org>
Date:   Tue Jul 11 15:44:30 2023 -0500

    arm64: dts: stratix10: fix incorrect I2C property for SCL signal

    commit db66795f61354c373ecdadbdae1ed253a96c47cb upstream.

    The correct dts property for the SCL falling time is
    "i2c-scl-falling-time-ns".

    Fixes: c8da1d15b8a4 ("arm64: dts: stratix10: i2c clock running out of spec")
    Cc: stable@vger.kernel.org
    Signed-off-by: Dinh Nguyen <dinguyen@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a0ac32cf61e5a76e2429e486925a52ee41dd75e3
Author: Jiri Olsa <jolsa@kernel.org>
Date:   Tue Jul 25 10:42:05 2023 +0200

    bpf: Disable preemption in bpf_perf_event_output

    commit f2c67a3e60d1071b65848efaa8c3b66c363dd025 upstream.

    The nesting protection in bpf_perf_event_output relies on disabled
    preemption, which is guaranteed for kprobes and tracepoints.
```

```
    However bpf_perf_event_output can be also called from uprobes context
    through bpf_prog_run_array_sleepable function which disables migration,
    but keeps preemption enabled.

    This can cause task to be preempted by another one inside the nesting
    protection and lead eventually to two tasks using same perf_sample_data
    buffer and cause crashes like:

      kernel tried to execute NX-protected page - exploit attempt? (uid: 0)
      BUG: unable to handle page fault for address: ffffffff82be3eea
      ...
      Call Trace:
       ? __die+0x1f/0x70
       ? page_fault_oops+0x176/0x4d0
       ? exc_page_fault+0x132/0x230
       ? asm_exc_page_fault+0x22/0x30
       ? perf_output_sample+0x12b/0x910
       ? perf_event_output+0xd0/0x1d0
       ? bpf_perf_event_output+0x162/0x1d0
       ? bpf_prog_c6271286d9a4c938_krava1+0x76/0x87
       ? __uprobe_perf_func+0x12b/0x540
       ? uprobe_dispatcher+0x2c4/0x430
       ? uprobe_notify_resume+0x2da/0xce0
       ? atomic_notifier_call_chain+0x7b/0x110
       ? exit_to_user_mode_prepare+0x13e/0x290
       ? irqentry_exit_to_user_mode+0x5/0x30
       ? asm_exc_int3+0x35/0x40

    Fixing this by disabling preemption in bpf_perf_event_output.

    Cc: stable@vger.kernel.org
    Fixes: 8c7dcb84e3b7 ("bpf: implement sleepable uprobes by chaining gps")
    Acked-by: Hou Tao <houtao1@huawei.com>
    Signed-off-by: Jiri Olsa <jolsa@kernel.org>
    Link: https://lore.kernel.org/r/20230725084206.580930-2-jolsa@kernel.org
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d92b04b2eae095697062ee11e75b652840dc0546
Author: Song Shuai <suagrfillet@gmail.com>
Date:   Mon Jul 24 18:09:16 2023 +0800

    riscv: Export va_kernel_pa_offset in vmcoreinfo

    commit fbe7d19d2b7fcbd38905ba9f691be8f245c6faa6 upstream.

    Since RISC-V Linux v6.4, the commit 3335068f8721 ("riscv: Use
    PUD/P4D/PGD pages for the linear mapping") changes phys_ram_base
    from the physical start of the kernel to the actual start of the DRAM.

    The Crash-utility's VTOP() still uses phys_ram_base and kernel_map.virt_addr
    to translate kernel virtual address, that failed the Crash with Linux v6.4 [1].

    Export kernel_map.va_kernel_pa_offset in vmcoreinfo to help Crash translate
    the kernel virtual address correctly.

    Fixes: 3335068f8721 ("riscv: Use PUD/P4D/PGD pages for the linear mapping")
    Link: https://lore.kernel.org/linux-riscv/20230724040649.220279-1-suagrfillet@gmail.com/
[1]
    Signed-off-by: Song Shuai <suagrfillet@gmail.com>
    Reviewed-by: Xianting Tian  <xianting.tian@linux.alibaba.com>
    Reviewed-by: Alexandre Ghiti <alexghiti@rivosinc.com>
    Link: https://lore.kernel.org/r/20230724100917.309061-1-suagrfillet@gmail.com
    Cc: stable@vger.kernel.org
```

```
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 501a38b801f48afd0d86cda79a2196801a9d8b55
Author: Arseniy Krasnov <AVKrasnov@sberdevices.ru>
Date:   Wed Jul 5 09:52:10 2023 +0300

    mtd: rawnand: meson: fix OOB available bytes for ECC

    commit 7e6b04f9238eab0f684fafd158c1f32ea65b9eaa upstream.

    It is incorrect to calculate number of OOB bytes for ECC engine using
    some "already known" ECC step size (1024 bytes here). Number of such
    bytes for ECC engine must be whole OOB except 2 bytes for bad block
    marker, while proper ECC step size and strength will be selected by
    ECC logic.

    Fixes: 8fae856c5350 ("mtd: rawnand: meson: add support for Amlogic NAND flash
controller")
    Cc: <Stable@vger.kernel.org>
    Signed-off-by: Arseniy Krasnov <AVKrasnov@sberdevices.ru>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-mtd/20230705065211.293500-1-AVKrasnov@sberdevices.ru
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6c26c42e076e49904cfb5a980050378b3014fd9e
Author: Olivier Maignial <olivier.maignial@hotmail.fr>
Date:   Fri Jun 23 17:33:37 2023 +0200

    mtd: spinand: winbond: Fix ecc_get_status

    commit f5a05060670a4d8d6523afc7963eb559c2e3615f upstream.

    Reading ECC status is failing.

    w25n02kv_ecc_get_status() is using on-stack buffer for
    SPINAND_GET_FEATURE_OP() output. It is not suitable for
    DMA needs of spi-mem.

    Fix this by using the spi-mem operations dedicated buffer
    spinand->scratchbuf.

    See
    spinand->scratchbuf:

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/include/linux/mtd/sp
inand.h?h=v6.3#n418
    spi_mem_check_op():
    https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/spi/spi-
mem.c?h=v6.3#n199

    Fixes: 6154c7a58348 ("mtd: spinand: winbond: add Winbond W25N02KV flash support")
    Cc: stable@vger.kernel.org
    Signed-off-by: Olivier Maignial <olivier.maignial@hotmail.fr>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-
mtd/DB4P250MB1032EDB9E36B764A33769039FE23A@DB4P250MB1032.EURP250.PROD.OUTLOOK.COM
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fa4cfb7d204cafac504de81f3b8e9d94d0c200a8
Author: Olivier Maignial <olivier.maignial@hotmail.fr>
Date:   Fri Jun 23 17:33:36 2023 +0200

    mtd: spinand: toshiba: Fix ecc_get_status
```

```
    commit 8544cda94dae6be3f1359539079c68bb731428b1 upstream.

    Reading ECC status is failing.

    tx58cxgxsxraix_ecc_get_status() is using on-stack buffer
    for SPINAND_GET_FEATURE_OP() output. It is not suitable
    for DMA needs of spi-mem.

    Fix this by using the spi-mem operations dedicated buffer
    spinand->scratchbuf.

    See
    spinand->scratchbuf:

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/include/linux/mtd/sp
inand.h?h=v6.3#n418
    spi_mem_check_op():
    https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/spi/spi-
mem.c?h=v6.3#n199

    Fixes: 10949af1681d ("mtd: spinand: Add initial support for Toshiba TC58CVG2S0H")
    Cc: stable@vger.kernel.org
    Signed-off-by: Olivier Maignial <olivier.maignial@hotmail.fr>
    Signed-off-by: Miquel Raynal <miquel.raynal@bootlin.com>
    Link: https://lore.kernel.org/linux-
mtd/DB4P250MB1032553D05FBE36DEE0D311EFE23A@DB4P250MB1032.EURP250.PROD.OUTLOOK.COM
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e705b1680001e216d973c7de870ea17f6dd2a7d1
Author: Sungjong Seo <sj1557.seo@samsung.com>
Date:   Fri Jul 14 17:43:54 2023 +0900

    exfat: release s_lock before calling dir_emit()

    commit ff84772fd45d486e4fc78c82e2f70ce5333543e6 upstream.

    There is a potential deadlock reported by syzbot as below:

    ======================================================
    WARNING: possible circular locking dependency detected
    6.4.0-next-20230707-syzkaller #0 Not tainted
    ------------------------------------------------------
    syz-executor330/5073 is trying to acquire lock:
    ffff8880218527a0 (&mm->mmap_lock){++++}-{3:3}, at: mmap_read_lock_killable
include/linux/mmap_lock.h:151 [inline]
    ffff8880218527a0 (&mm->mmap_lock){++++}-{3:3}, at: get_mmap_lock_carefully
mm/memory.c:5293 [inline]
    ffff8880218527a0 (&mm->mmap_lock){++++}-{3:3}, at: lock_mm_and_find_vma+0x369/0x510
mm/memory.c:5344
    but task is already holding lock:
    ffff888019f760e0 (&sbi->s_lock){+.+.}-{3:3}, at: exfat_iterate+0x117/0xb50
fs/exfat/dir.c:232

    which lock already depends on the new lock.

    Chain exists of:
      &mm->mmap_lock --> mapping.invalidate_lock#3 --> &sbi->s_lock

     Possible unsafe locking scenario:

           CPU0                    CPU1
           ----                    ----
      lock(&sbi->s_lock);
                                   lock(mapping.invalidate_lock#3);
                                   lock(&sbi->s_lock);
```

```
        rlock(&mm->mmap_lock);

    Let's try to avoid above potential deadlock condition by moving dir_emit*()
    out of sbi->s_lock coverage.

    Fixes: ca06197382bd ("exfat: add directory operations")
    Cc: stable@vger.kernel.org #v5.7+
    Reported-by: syzbot+1741a5d9b79989c10bdc@syzkaller.appspotmail.com
    Link: https://lore.kernel.org/lkml/00000000000078ee7e060066270b@google.com/T/#u
    Tested-by: syzbot+1741a5d9b79989c10bdc@syzkaller.appspotmail.com
    Signed-off-by: Sungjong Seo <sj1557.seo@samsung.com>
    Signed-off-by: Namjae Jeon <linkinjeon@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e1a73ba43cf883cb37f6331aca5a4c5be6350982
Author: Namjae Jeon <linkinjeon@kernel.org>
Date:   Thu Jul 13 21:59:37 2023 +0900

    exfat: check if filename entries exceeds max filename length

    commit d42334578eba1390859012ebb91e1e556d51db49 upstream.

    exfat_extract_uni_name copies characters from a given file name entry into
    the 'uniname' variable. This variable is actually defined on the stack of
    the exfat_readdir() function. According to the definition of
    the 'exfat_uni_name' type, the file name should be limited 255 characters
    (+ null teminator space), but the exfat_get_uniname_from_ext_entry()
    function can write more characters because there is no check if filename
    entries exceeds max filename length. This patch add the check not to copy
    filename characters when exceeding max filename length.

    Cc: stable@vger.kernel.org
    Cc: Yuezhang Mo <Yuezhang.Mo@sony.com>
    Reported-by: Maxim Suhanov <dfirblog@gmail.com>
    Reviewed-by: Sungjong Seo <sj1557.seo@samsung.com>
    Signed-off-by: Namjae Jeon <linkinjeon@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0c5c3e8a2550b6b2a304b45f260296db9c09df96
Author: gaoming <gaoming20@hihonor.com>
Date:   Wed Jul 5 15:15:15 2023 +0800

    exfat: use kvmalloc_array/kvfree instead of kmalloc_array/kfree

    commit daf60d6cca26e50d65dac374db92e58de745ad26 upstream.

    The call stack shown below is a scenario in the Linux 4.19 kernel.
    Allocating memory failed where exfat fs use kmalloc_array due to
    system memory fragmentation, while the u-disk was inserted without
    recognition.
    Devices such as u-disk using the exfat file system are pluggable and
    may be insert into the system at any time.
    However, long-term running systems cannot guarantee the continuity of
    physical memory. Therefore, it's necessary to address this issue.

    Binder:2632_6: page allocation failure: order:4,
     mode:0x6040c0(GFP_KERNEL|__GFP_COMP), nodemask=(null)
    Call trace:
    [242178.097582]  dump_backtrace+0x0/0x4
    [242178.097589]  dump_stack+0xf4/0x134
    [242178.097598]  warn_alloc+0xd8/0x144
    [242178.097603]  __alloc_pages_nodemask+0x1364/0x1384
    [242178.097608]  kmalloc_order+0x2c/0x510
    [242178.097612]  kmalloc_order_trace+0x40/0x16c
    [242178.097618]  __kmalloc+0x360/0x408
```

```
      [242178.097624]  load_alloc_bitmap+0x160/0x284
      [242178.097628]  exfat_fill_super+0xa3c/0xe7c
      [242178.097635]  mount_bdev+0x2e8/0x3a0
      [242178.097638]  exfat_fs_mount+0x40/0x50
      [242178.097643]  mount_fs+0x138/0x2e8
      [242178.097649]  vfs_kern_mount+0x90/0x270
      [242178.097655]  do_mount+0x798/0x173c
      [242178.097659]  ksys_mount+0x114/0x1ac
      [242178.097665]  __arm64_sys_mount+0x24/0x34
      [242178.097671]  el0_svc_common+0xb8/0x1b8
      [242178.097676]  el0_svc_handler+0x74/0x90
      [242178.097681]  el0_svc+0x8/0x340

      By analyzing the exfat code,we found that continuous physical memory
      is not required here,so kvmalloc_array is used can solve this problem.

      Cc: stable@vger.kernel.org
      Signed-off-by: gaoming <gaoming20@hihonor.com>
      Signed-off-by: Namjae Jeon <linkinjeon@kernel.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 3d4d2e55b31cc1cbc9dc29d88599f7736900c9ec
  Author: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
  Date:   Wed Jul 19 08:16:52 2023 +0200

      firmware: arm_scmi: Drop OF node reference in the transport channel setup

      commit da042eb4f061a0b54aedadcaa15391490c48e1ad upstream.

      The OF node reference obtained from of_parse_phandle() should be dropped
      if node is not compatible with arm,scmi-shmem.

      Fixes: 507cd4d2c5eb ("firmware: arm_scmi: Add compatibility checks for shmem node")
      Cc: <stable@vger.kernel.org>
      Signed-off-by: Krzysztof Kozlowski <krzysztof.kozlowski@linaro.org>
      Reviewed-by: Cristian Marussi <cristian.marussi@arm.com>
      Link: https://lore.kernel.org/r/20230719061652.8850-1-krzysztof.kozlowski@linaro.org
      Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 0b39dfaf255beb6913ab5f9ecf83af6736b416de
  Author: Xiubo Li <xiubli@redhat.com>
  Date:   Tue Jul 25 12:03:59 2023 +0800

      ceph: defer stopping mdsc delayed_work

      commit e7e607bd00481745550389a29ecabe33e13d67cf upstream.

      Flushing the dirty buffer may take a long time if the cluster is
      overloaded or if there is network issue. So we should ping the
      MDSs periodically to keep alive, else the MDS will blocklist
      the kclient.

      Cc: stable@vger.kernel.org
      Link: https://tracker.ceph.com/issues/61843
      Signed-off-by: Xiubo Li <xiubli@redhat.com>
      Reviewed-by: Milind Changire <mchangir@redhat.com>
      Signed-off-by: Ilya Dryomov <idryomov@gmail.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit c29cc7eef96b25441d8e2173e58374af5d935834
  Author: Ross Maynard <bids.7405@bigpond.com>
  Date:   Mon Jul 31 15:42:04 2023 +1000

      USB: zaurus: Add ID for A-300/B-500/C-700
```

```
    commit b99225b4fe297d07400f9e2332ecd7347b224f8d upstream.

    The SL-A300, B500/5600, and C700 devices no longer auto-load because of
    "usbnet: Remove over-broad module alias from zaurus."
    This patch adds IDs for those 3 devices.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=217632
    Fixes: 16adf5d07987 ("usbnet: Remove over-broad module alias from zaurus.")
    Signed-off-by: Ross Maynard <bids.7405@bigpond.com>
    Cc: stable@vger.kernel.org
    Acked-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/69b5423b-2013-9fc9-9569-58e707d9bafb@bigpond.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7d4b6fd65815c811c389742b6b2547033cf090a4
Author: Ilya Dryomov <idryomov@gmail.com>
Date:   Tue Aug 1 19:14:24 2023 +0200

    libceph: fix potential hang in ceph_osdc_notify()

    commit e6e2843230799230fc5deb8279728a7218b0d63c upstream.

    If the cluster becomes unavailable, ceph_osdc_notify() may hang even
    with osd_request_timeout option set because linger_notify_finish_wait()
    waits for MWatchNotify NOTIFY_COMPLETE message with no associated OSD
    request in flight -- it's completely asynchronous.

    Introduce an additional timeout, derived from the specified notify
    timeout.  While at it, switch both waits to killable which is more
    correct.

    Cc: stable@vger.kernel.org
    Signed-off-by: Ilya Dryomov <idryomov@gmail.com>
    Reviewed-by: Dongsheng Yang <dongsheng.yang@easystack.cn>
    Reviewed-by: Xiubo Li <xiubli@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b31ef844d43040a0faae33cb465665be620db3a6
Author: Song Shuai <suagrfillet@gmail.com>
Date:   Mon Jul 24 18:09:17 2023 +0800

    Documentation: kdump: Add va_kernel_pa_offset for RISCV64

    commit 640c503d7dbd7d34a62099c933f4db0ed77ccbec upstream.

    RISC-V Linux exports "va_kernel_pa_offset" in vmcoreinfo to help
    Crash-utility translate the kernel virtual address correctly.

    Here adds the definition of "va_kernel_pa_offset".

    Fixes: 3335068f8721 ("riscv: Use PUD/P4D/PGD pages for the linear mapping")
    Link: https://lore.kernel.org/linux-riscv/20230724040649.220279-1-suagrfillet@gmail.com/
    Signed-off-by: Song Shuai <suagrfillet@gmail.com>
    Reviewed-by: Alexandre Ghiti <alexghiti@rivosinc.com>
    Link: https://lore.kernel.org/r/20230724100917.309061-2-suagrfillet@gmail.com
    Cc: stable@vger.kernel.org
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d678df66b1af29cfe3240baeafc01bbdde75d256
Author: Michael Kelley <mikelley@microsoft.com>
Date:   Thu Jul 20 14:05:02 2023 -0700
```

```
    scsi: storvsc: Limit max_sectors for virtual Fibre Channel devices

    commit 010c1e1c5741365dbbf44a5a5bb9f30192875c4c upstream.

    The Hyper-V host is queried to get the max transfer size that it supports,
    and this value is used to set max_sectors for the synthetic SCSI
    controller.  However, this max transfer size may be too large for virtual
    Fibre Channel devices, which are limited to 512 Kbytes.  If a larger
    transfer size is used with a vFC device, Hyper-V always returns an error,
    and storvsc logs a message like this where the SRB status and SCSI status
    are both zero:

    hv_storvsc <GUID>: tag#197 cmd 0x8a status: scsi 0x0 srb 0x0 hv 0xc0000001

    Add logic to limit the max transfer size to 512 Kbytes for vFC devices.

    Fixes: 1d3e0980782f ("scsi: storvsc: Correct reporting of Hyper-V I/O size limits")
    Cc: stable@vger.kernel.org
    Signed-off-by: Michael Kelley <mikelley@microsoft.com>
    Link: https://lore.kernel.org/r/1689887102-32806-1-git-send-email-mikelley@microsoft.com
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2c2aa8d0e2a54fb3ced7c6ec8a3780071d686538
Author: Steffen Maier <maier@linux.ibm.com>
Date:   Mon Jul 24 16:51:56 2023 +0200

    scsi: zfcp: Defer fc_rport blocking until after ADISC response

    commit e65851989001c0c9ba9177564b13b38201c0854c upstream.

    Storage devices are free to send RSCNs, e.g. for internal state changes. If
    this happens on all connected paths, zfcp risks temporarily losing all
    paths at the same time. This has strong requirements on multipath
    configuration such as "no_path_retry queue".

    Avoid such situations by deferring fc_rport blocking until after the ADISC
    response, when any actual state change of the remote port became clear.
    The already existing port recovery triggers explicitly block the fc_rport.
    The triggers are: on ADISC reject or timeout (typical cable pull case), and
    on ADISC indicating that the remote port has changed its WWPN or
    the port is meanwhile no longer open.

    As a side effect, this also removes a confusing direct function call to
    another work item function zfcp_scsi_rport_work() instead of scheduling
    that other work item. It was probably done that way to have the rport block
    side effect immediate and synchronous to the caller.

    Fixes: a2fa0aede07c ("[SCSI] zfcp: Block FC transport rports early on errors")
    Cc: stable@vger.kernel.org #v2.6.30+
    Reviewed-by: Benjamin Block <bblock@linux.ibm.com>
    Reviewed-by: Fedor Loshakov <loshakov@linux.ibm.com>
    Signed-off-by: Steffen Maier <maier@linux.ibm.com>
    Link: https://lore.kernel.org/r/20230724145156.3920244-1-maier@linux.ibm.com
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0c8515842b824a21b981141247bfb7bfc27db455
Author: Boqun Feng <boqun.feng@gmail.com>
Date:   Sat Jul 29 18:29:02 2023 -0700

    rust: allocator: Prevent mis-aligned allocation

    commit b3d8aa84bbfe9b58ccc5332cacf8ea17200af310 upstream.
```

```
    Currently the rust allocator simply passes the size of the type Layout
    to krealloc(), and in theory the alignment requirement from the type
    Layout may be larger than the guarantee provided by SLAB, which means
    the allocated object is mis-aligned.

    Fix this by adjusting the allocation size to the nearest power of two,
    which SLAB always guarantees a size-aligned allocation. And because Rust
    guarantees that the original size must be a multiple of alignment and
    the alignment must be a power of two, then the alignment requirement is
    satisfied.

    Suggested-by: Vlastimil Babka <vbabka@suse.cz>
    Co-developed-by: "Andreas Hindborg (Samsung)" <nmi@metaspace.dk>
    Signed-off-by: "Andreas Hindborg (Samsung)" <nmi@metaspace.dk>
    Signed-off-by: Boqun Feng <boqun.feng@gmail.com>
    Cc: stable@vger.kernel.org # v6.1+
    Acked-by: Vlastimil Babka <vbabka@suse.cz>
    Fixes: 247b365dc8dc ("rust: add `kernel` crate")
    Link: https://github.com/Rust-for-Linux/linux/issues/974
    Link: https://lore.kernel.org/r/20230730012905.643822-2-boqun.feng@gmail.com
    [ Applied rewording of comment as discussed in the mailing list. ]
    Signed-off-by: Miguel Ojeda <ojeda@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a2543be8a7e5b99a8203bd556bb234edd5497c1
Author: Stefano Garzarella <sgarzare@redhat.com>
Date:   Thu Aug 3 10:54:54 2023 +0200

    test/vsock: remove vsock_perf executable on `make clean`

    [ Upstream commit 3c50c8b240390907c9a33c86d25d850520db6dfa ]

    We forgot to add vsock_perf to the rm command in the `clean`
    target, so now we have a left over after `make clean` in
    tools/testing/vsock.

    Fixes: 8abbffd27ced ("test/vsock: vsock_perf utility")
    Cc: AVKrasnov@sberdevices.ru
    Signed-off-by: Stefano Garzarella <sgarzare@redhat.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Tested-by: Simon Horman <horms@kernel.org> # build-tested
    Link: https://lore.kernel.org/r/20230803085454.30897-1-sgarzare@redhat.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d6d195f2e7b181d021acb9a81c680f14da30e461
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:15:00 2023 +0000

    tcp_metrics: fix data-race in tcpm_suck_dst() vs fastopen

    [ Upstream commit ddf251fa2bc1d3699eec0bae6ed0bc373b8fda79 ]

    Whenever tcpm_new() reclaims an old entry, tcpm_suck_dst()
    would overwrite data that could be read from tcp_fastopen_cache_get()
    or tcp_metrics_fill_info().

    We need to acquire fastopen_seqlock to maintain consistency.

    For newly allocated objects, tcpm_new() can switch to kzalloc()
    to avoid an extra fastopen_seqlock acquisition.

    Fixes: 1fe4c481ba63 ("net-tcp: Fast Open client - cookie cache")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
```

```
    Cc: Yuchung Cheng <ycheng@google.com>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230802131500.1478140-7-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit fd147efc9610372e69a53c171e801b0b15111f50
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:14:59 2023 +0000

    tcp_metrics: annotate data-races around tm->tcpm_net

    [ Upstream commit d5d986ce42c71a7562d32c4e21e026b0f87befec ]

    tm->tcpm_net can be read or written locklessly.

    Instead of changing write_pnet() and read_pnet() and potentially
    hurt performance, add the needed READ_ONCE()/WRITE_ONCE()
    in tm_net() and tcpm_new().

    Fixes: 849e8a0ca8d5 ("tcp_metrics: Add a field tcpm_net and verify it matches on
lookup")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230802131500.1478140-6-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6f27b67fc329b9d582133161efca8f86c733310d
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:14:58 2023 +0000

    tcp_metrics: annotate data-races around tm->tcpm_vals[]

    [ Upstream commit 8c4d04f6b443869d25e59822f7cec88d647028a9 ]

    tm->tcpm_vals[] values can be read or written locklessly.

    Add needed READ_ONCE()/WRITE_ONCE() to document this,
    and force use of tcp_metric_get() and tcp_metric_set()

    Fixes: 51c5d0c4b169 ("tcp: Maintain dynamic metrics in local cache.")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d8064e2859ae1858660892d0a9c8585c262908bc
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:14:57 2023 +0000

    tcp_metrics: annotate data-races around tm->tcpm_lock

    [ Upstream commit 285ce119a3c6c4502585936650143e54c8692788 ]

    tm->tcpm_lock can be read or written locklessly.

    Add needed READ_ONCE()/WRITE_ONCE() to document this.

    Fixes: 51c5d0c4b169 ("tcp: Maintain dynamic metrics in local cache.")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
```

```
    Link: https://lore.kernel.org/r/20230802131500.1478140-4-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cbe9433586be80b0d346d5a61beeceb1f2aef96b
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:14:56 2023 +0000

    tcp_metrics: annotate data-races around tm->tcpm_stamp

    [ Upstream commit 949ad62a5d5311d36fce2e14fe5fed3f936da51c ]

    tm->tcpm_stamp can be read or written locklessly.

    Add needed READ_ONCE()/WRITE_ONCE() to document this.

    Also constify tcpm_check_stamp() dst argument.

    Fixes: 51c5d0c4b169 ("tcp: Maintain dynamic metrics in local cache.")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230802131500.1478140-3-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f47cff05d3b3b189173c9f0aba707723a62d20de
Author: Eric Dumazet <edumazet@google.com>
Date:   Wed Aug 2 13:14:55 2023 +0000

    tcp_metrics: fix addr_same() helper

    [ Upstream commit e6638094d7af6c7b9dcca05ad009e79e31b4f670 ]

    Because v4 and v6 families use separate inetpeer trees (respectively
    net->ipv4.peers and net->ipv6.peers), inetpeer_addr_cmp(a, b) assumes
    a & b share the same family.

    tcp_metrics use a common hash table, where entries can have different
    families.

    We must therefore make sure to not call inetpeer_addr_cmp()
    if the families do not match.

    Fixes: d39d14ffa24c ("net: Add helper function to compare inetpeer addresses")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230802131500.1478140-2-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9defb2b3843c8972891c1c7170504810e5a9feb1
Author: Jonas Gorski <jonas.gorski@bisdn.de>
Date:   Wed Aug 2 11:23:56 2023 +0200

    prestera: fix fallback to previous version on same major version

    [ Upstream commit b755c25fbcd568821a3bb0e0d5c2daa5fcb00bba ]

    When both supported and previous version have the same major version,
    and the firmwares are missing, the driver ends in a loop requesting the
    same (previous) version over and over again:

        [   76.327413] Prestera DX 0000:01:00.0: missing latest
```

```
 mrvl/prestera/mvsw_prestera_fw-v4.1.img firmware, fall-back to previous 4.0 version
         [   76.339802] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.352162] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.364502] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.376848] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.389183] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.401522] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.413860] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         [   76.426199] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.0.img firmware, fall-back to previous 4.0 version
         ...

     Fix this by inverting the check to that we aren't yet at the previous
     version, and also check the minor version.

     This also catches the case where both versions are the same, as it was
     after commit bb5dbf2cc64d ("net: marvell: prestera: add firmware v4.0
     support").

     With this fix applied:

         [   88.499622] Prestera DX 0000:01:00.0: missing latest
 mrvl/prestera/mvsw_prestera_fw-v4.1.img firmware, fall-back to previous 4.0 version
         [   88.511995] Prestera DX 0000:01:00.0: failed to request previous firmware:
 mrvl/prestera/mvsw_prestera_fw-v4.0.img
         [   88.522403] Prestera DX: probe of 0000:01:00.0 failed with error -2

     Fixes: 47f26018a414 ("net: marvell: prestera: try to load previous fw version")
     Signed-off-by: Jonas Gorski <jonas.gorski@bisdn.de>
     Acked-by: Elad Nachman <enachman@marvell.com>
     Reviewed-by: Jesse Brandeburg <jesse.brandeburg@intel.com>
     Acked-by: Taras Chornyi <taras.chornyi@plvision.eu>
     Link: https://lore.kernel.org/r/20230802092357.163944-1-jonas.gorski@bisdn.de
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit ac2b423db9b100740c5c992ded375edaaeeefa9b
 Author: Leon Romanovsky <leon@kernel.org>
 Date:   Mon Jul 31 14:58:42 2023 +0300

     net/mlx5e: Set proper IPsec source port in L4 selector

     [ Upstream commit 62da08331f1a2bef9d0148613133ce8e640a2f8d ]

     Fix typo in setup_fte_upper_proto_match() where destination UDP port
     was used instead of source port.

     Fixes: a7385187a386 ("net/mlx5e: IPsec, support upper protocol selector field offload")
     Signed-off-by: Leon Romanovsky <leonro@nvidia.com>
     Link:
 https://lore.kernel.org/r/ffc024a4d192113103f392b0502688366ca88c1f.1690803944.git.leonro@nvi
 dia.com
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit a1e071efa766fa1ae7e57e569e2ea1eb25c050d0
 Author: Jianbo Liu <jianbol@nvidia.com>
 Date:   Mon Jul 31 14:58:41 2023 +0300
```

```
    net/mlx5: fs_core: Skip the FTs in the same FS_TYPE_PRIO_CHAINS fs_prio

    [ Upstream commit c635ca45a7a2023904a1f851e99319af7b87017d ]

    In the cited commit, new type of FS_TYPE_PRIO_CHAINS fs_prio was added
    to support multiple parallel namespaces for multi-chains. And we skip
    all the flow tables under the fs_node of this type unconditionally,
    when searching for the next or previous flow table to connect for a
    new table.

    As this search function is also used for find new root table when the
    old one is being deleted, it will skip the entire FS_TYPE_PRIO_CHAINS
    fs_node next to the old root. However, new root table should be chosen
    from it if there is any table in it. Fix it by skipping only the flow
    tables in the same FS_TYPE_PRIO_CHAINS fs_node when finding the
    closest FT for a fs_node.

    Besides, complete the connecting from FTs of previous priority of prio
    because there should be multiple prevs after this fs_prio type is
    introduced. And also the next FT should be chosen from the first flow
    table next to the prio in the same FS_TYPE_PRIO_CHAINS fs_prio, if
    this prio is the first child.

    Fixes: 328edb499f99 ("net/mlx5: Split FDB fast path prio to multiple namespaces")
    Signed-off-by: Jianbo Liu <jianbol@nvidia.com>
    Reviewed-by: Paul Blakey <paulb@nvidia.com>
    Signed-off-by: Leon Romanovsky <leonro@nvidia.com>
    Link:
https://lore.kernel.org/r/7a95754df479e722038996c97c97b062b372591f.1690803944.git.leonro@nvi
dia.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b1aa1b267b9167ab560117e42bce997bdada05af
Author: Jianbo Liu <jianbol@nvidia.com>
Date:   Mon Jul 31 14:58:40 2023 +0300

    net/mlx5: fs_core: Make find_closest_ft more generic

    [ Upstream commit 618d28a535a0582617465d14e05f3881736a2962 ]

    As find_closest_ft_recursive is called to find the closest FT, the
    first parameter of find_closest_ft can be changed from fs_prio to
    fs_node. Thus this function is extended to find the closest FT for the
    nodes of any type, not only prios, but also the sub namespaces.

    Signed-off-by: Jianbo Liu <jianbol@nvidia.com>
    Signed-off-by: Leon Romanovsky <leonro@nvidia.com>
    Link:
https://lore.kernel.org/r/d3962c2b443ec8dde7a740dc742a1f052d5e256c.1690803944.git.leonro@nvi
dia.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Stable-dep-of: c635ca45a7a2 ("net/mlx5: fs_core: Skip the FTs in the same
FS_TYPE_PRIO_CHAINS fs_prio")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 23c195ce6f4aec86e1c9e1ea1c800381c4b465c7
Author: Benjamin Poirier <bpoirier@nvidia.com>
Date:   Mon Jul 31 16:02:08 2023 -0400

    vxlan: Fix nexthop hash size

    [ Upstream commit 0756384fb1bd38adb2ebcfd1307422f433a1d772 ]
```

```
    The nexthop code expects a 31 bit hash, such as what is returned by
    fib_multipath_hash() and rt6_multipath_hash(). Passing the 32 bit hash
    returned by skb_get_hash() can lead to problems related to the fact that
    'int hash' is a negative number when the MSB is set.

    In the case of hash threshold nexthop groups, nexthop_select_path_hthr()
    will disproportionately select the first nexthop group entry. In the case
    of resilient nexthop groups, nexthop_select_path_res() may do an out of
    bounds access in nh_buckets[], for example:
        hash = -912054133
        num_nh_buckets = 2
        bucket_index = 65535

    which leads to the following panic:

    BUG: unable to handle page fault for address: ffffc900025910c8
    PGD 100000067 P4D 100000067 PUD 10026b067 PMD 0
    Oops: 0002 [#1] PREEMPT SMP KASAN NOPTI
    CPU: 4 PID: 856 Comm: kworker/4:3 Not tainted 6.5.0-rc2+ #34
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.2-debian-1.16.2-1
04/01/2014
    Workqueue: ipv6_addrconf addrconf_dad_work
    RIP: 0010:nexthop_select_path+0x197/0xbf0
    Code: c1 e4 05 be 08 00 00 00 4c 8b 35 a4 14 7e 01 4e 8d 6c 25 00 4a 8d 7c 25 08 48 01
dd e8 c2 25 15 ff 49 8d 7d 08 e8 39 13 15 ff <4d> 89 75 08 48 89 ef e8 7d 12 15 ff 48 8b 5d
00 e8 14 55 2f 00 85
    RSP: 0018:ffff88810c36f260 EFLAGS: 00010246
    RAX: 0000000000000000 RBX: 00000000002000c0 RCX: ffffffffaf02dd77
    RDX: dfffffc0000000000 RSI: 0000000000000008 RDI: ffffc900025910c8
    RBP: ffffc900025910c0 R08: 0000000000000001 R09: fffff520004b2219
    R10: ffffc900025910cf R11: 31392d2068736168 R12: 00000000002000c0
    R13: ffffc900025910c0 R14: 00000000fffef608 R15: ffff88811840e900
    FS:  0000000000000000(0000) GS:ffff8881f7000000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: ffffc900025910c8 CR3: 0000000129d00000 CR4: 0000000000750ee0
    PKRU: 55555554
    Call Trace:
     <TASK>
     ? __die+0x23/0x70
     ? page_fault_oops+0x1ee/0x5c0
     ? __pfx_is_prefetch.constprop.0+0x10/0x10
     ? __pfx_page_fault_oops+0x10/0x10
     ? search_bpf_extables+0xfe/0x1c0
     ? fixup_exception+0x3b/0x470
     ? exc_page_fault+0xf6/0x110
     ? asm_exc_page_fault+0x26/0x30
     ? nexthop_select_path+0x197/0xbf0
     ? nexthop_select_path+0x197/0xbf0
     ? lock_is_held_type+0xe7/0x140
     vxlan_xmit+0x5b2/0x2340
     ? __lock_acquire+0x92b/0x3370
     ? __pfx_vxlan_xmit+0x10/0x10
     ? __pfx___lock_acquire+0x10/0x10
     ? __pfx_register_lock_class+0x10/0x10
     ? skb_network_protocol+0xce/0x2d0
     ? dev_hard_start_xmit+0xca/0x350
     ? __pfx_vxlan_xmit+0x10/0x10
     dev_hard_start_xmit+0xca/0x350
     __dev_queue_xmit+0x513/0x1e20
     ? __pfx___dev_queue_xmit+0x10/0x10
     ? __pfx_lock_release+0x10/0x10
     ? mark_held_locks+0x44/0x90
     ? skb_push+0x4c/0x80
     ? eth_header+0x81/0xe0
     ? __pfx_eth_header+0x10/0x10
```

```
      ? neigh_resolve_output+0x215/0x310
      ? ip6_finish_output2+0x2ba/0xc90
      ip6_finish_output2+0x2ba/0xc90
      ? lock_release+0x236/0x3e0
      ? ip6_mtu+0xbb/0x240
      ? __pfx_ip6_finish_output2+0x10/0x10
      ? find_held_lock+0x83/0xa0
      ? lock_is_held_type+0xe7/0x140
      ip6_finish_output+0x1ee/0x780
      ip6_output+0x138/0x460
      ? __pfx_ip6_output+0x10/0x10
      ? __pfx___lock_acquire+0x10/0x10
      ? __pfx_ip6_finish_output+0x10/0x10
      NF_HOOK.constprop.0+0xc0/0x420
      ? __pfx_NF_HOOK.constprop.0+0x10/0x10
      ? ndisc_send_skb+0x2c0/0x960
      ? __pfx_lock_release+0x10/0x10
      ? __local_bh_enable_ip+0x93/0x110
      ? lock_is_held_type+0xe7/0x140
      ndisc_send_skb+0x4be/0x960
      ? __pfx_ndisc_send_skb+0x10/0x10
      ? mark_held_locks+0x65/0x90
      ? find_held_lock+0x83/0xa0
      ndisc_send_ns+0xb0/0x110
      ? __pfx_ndisc_send_ns+0x10/0x10
      addrconf_dad_work+0x631/0x8e0
      ? lock_acquire+0x180/0x3f0
      ? __pfx_addrconf_dad_work+0x10/0x10
      ? mark_held_locks+0x24/0x90
      process_one_work+0x582/0x9c0
      ? __pfx_process_one_work+0x10/0x10
      ? __pfx_do_raw_spin_lock+0x10/0x10
      ? mark_held_locks+0x24/0x90
      worker_thread+0x93/0x630
      ? __kthread_parkme+0xdc/0x100
      ? __pfx_worker_thread+0x10/0x10
      kthread+0x1a5/0x1e0
      ? __pfx_kthread+0x10/0x10
      ret_from_fork+0x34/0x60
      ? __pfx_kthread+0x10/0x10
      ret_from_fork_asm+0x1b/0x30
     RIP: 0000:0x0
     Code: Unable to access opcode bytes at 0xffffffffffffffd6.
     RSP: 0000:0000000000000000 EFLAGS: 00000000 ORIG_RAX: 0000000000000000
     RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000
     RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000
     RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000
     R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000
     R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
      </TASK>
     Modules linked in:
     CR2: ffffc900025910c8
     ---[ end trace 0000000000000000 ]---
     RIP: 0010:nexthop_select_path+0x197/0xbf0
     Code: c1 e4 05 be 08 00 00 00 4c 8b 35 a4 14 7e 01 4e 8d 6c 25 00 4a 8d 7c 25 08 48 01
  dd e8 c2 25 15 ff 49 8d 7d 08 e8 39 13 15 ff <4d> 89 75 08 48 89 ef e8 7d 12 15 ff 48 8b 5d
  00 e8 14 55 2f 00 85
     RSP: 0018:ffff88810c36f260 EFLAGS: 00010246
     RAX: 0000000000000000 RBX: 00000000002000c0 RCX: ffffffffaf02dd77
     RDX: dfffc0000000000 RSI: 0000000000000008 RDI: ffffc900025910c8
     RBP: ffffc900025910c0 R08: 0000000000000001 R09: fffff520004b2219
     R10: ffffc900025910cf R11: 31392d2068736168 R12: 00000000002000c0
     R13: ffffc900025910c0 R14: 00000000fffef608 R15: ffff88811840e900
     FS:  0000000000000000(0000) GS:ffff8881f7000000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
```

```
    CR2: ffffffffffffffd6 CR3: 0000000129d00000 CR4: 0000000000750ee0
    PKRU: 55555554
    Kernel panic - not syncing: Fatal exception in interrupt
    Kernel Offset: 0x2ca00000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-
0xffffffffbfffffff)
    ---[ end Kernel panic - not syncing: Fatal exception in interrupt ]---

    Fix this problem by ensuring the MSB of hash is 0 using a right shift - the
    same approach used in fib_multipath_hash() and rt6_multipath_hash().

    Fixes: 1274e1cc4226 ("vxlan: ecmp support for mac fdb entries")
    Signed-off-by: Benjamin Poirier <bpoirier@nvidia.com>
    Reviewed-by: Ido Schimmel <idosch@nvidia.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3326c711f18d18fe6e1f5d83d3a7eab07e5a1560
Author: Yue Haibing <yuehaibing@huawei.com>
Date:   Tue Aug 1 14:43:18 2023 +0800

    ip6mr: Fix skb_under_panic in ip6mr_cache_report()

    [ Upstream commit 30e0191b16e8a58e4620fa3e2839ddc7b9d4281c ]

    skbuff: skb_under_panic: text:ffffffff88771f69 len:56 put:-4
     head:ffff88805f86a800 data:ffff887f5f86a850 tail:0x88 end:0x2c0 dev:pim6reg
     ------------[ cut here ]------------
     kernel BUG at net/core/skbuff.c:192!
     invalid opcode: 0000 [#1] PREEMPT SMP KASAN
     CPU: 2 PID: 22968 Comm: kworker/2:11 Not tainted 6.5.0-rc3-00044-g0a8db05b571a #236
     Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014
     Workqueue: ipv6_addrconf addrconf_dad_work
     RIP: 0010:skb_panic+0x152/0x1d0
     Call Trace:
      <TASK>
      skb_push+0xc4/0xe0
      ip6mr_cache_report+0xd69/0x19b0
      reg_vif_xmit+0x406/0x690
      dev_hard_start_xmit+0x17e/0x6e0
      __dev_queue_xmit+0x2d6a/0x3d20
      vlan_dev_hard_start_xmit+0x3ab/0x5c0
      dev_hard_start_xmit+0x17e/0x6e0
      __dev_queue_xmit+0x2d6a/0x3d20
      neigh_connected_output+0x3ed/0x570
      ip6_finish_output2+0x5b5/0x1950
      ip6_finish_output+0x693/0x11c0
      ip6_output+0x24b/0x880
      NF_HOOK.constprop.0+0xfd/0x530
      ndisc_send_skb+0x9db/0x1400
      ndisc_send_rs+0x12a/0x6c0
      addrconf_dad_completed+0x3c9/0xea0
      addrconf_dad_work+0x849/0x1420
      process_one_work+0xa22/0x16e0
      worker_thread+0x679/0x10c0
      ret_from_fork+0x28/0x60
      ret_from_fork_asm+0x11/0x20

    When setup a vlan device on dev pim6reg, DAD ns packet may sent on reg_vif_xmit().
    reg_vif_xmit()
        ip6mr_cache_report()
            skb_push(skb, -skb_network_offset(pkt));//skb_network_offset(pkt) is 4
    And skb_push declared as:
            void *skb_push(struct sk_buff *skb, unsigned int len);
                    skb->data -= len;
```

```
                    //0xffff88805f86a84c - 0xfffffffc = 0xffff887f5f86a850
    skb->data is set to 0xffff887f5f86a850, which is invalid mem addr, lead to skb_push()
fails.

    Fixes: 14fb64e1f449 ("[IPV6] MROUTE: Support PIM-SM (SSM).")
    Signed-off-by: Yue Haibing <yuehaibing@huawei.com>
    Reviewed-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 29d6fe395087710280f8e11d4ae79569c4cb14b7
Author: Alexandra Winter <wintera@linux.ibm.com>
Date:   Tue Aug 1 10:00:16 2023 +0200

    s390/qeth: Don't call dev_close/dev_open (DOWN/UP)

    [ Upstream commit 1cfef80d4c2b2c599189f36f36320b205d9447d9 ]

    dev_close() and dev_open() are issued to change the interface state to DOWN
    or UP (dev->flags IFF_UP). When the netdev is set DOWN it loses e.g its
    Ipv6 addresses and routes. We don't want this in cases of device recovery
    (triggered by hardware or software) or when the qeth device is set
    offline.

    Setting a qeth device offline or online and device recovery actions call
    netif_device_detach() and/or netif_device_attach(). That will reset or
    set the LOWER_UP indication i.e. change the dev->state Bit
    __LINK_STATE_PRESENT. That is enough to e.g. cause bond failovers, and
    still preserves the interface settings that are handled by the network
    stack.

    Don't call dev_open() nor dev_close() from the qeth device driver. Let the
    network stack handle this.

    Fixes: d4560150cb47 ("s390/qeth: call dev_close() during recovery")
    Signed-off-by: Alexandra Winter <wintera@linux.ibm.com>
    Reviewed-by: Wenjia Zhang <wenjia@linux.ibm.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 199fde04bd875d28b3a5ca525eaaa004eec6e947
Author: Lin Ma <linma@zju.edu.cn>
Date:   Tue Aug 1 09:32:48 2023 +0800

    net: dcb: choose correct policy to parse DCB_ATTR_BCN

    [ Upstream commit 31d49ba033095f6e8158c60f69714a500922e0c3 ]

    The dcbnl_bcn_setcfg uses erroneous policy to parse tb[DCB_ATTR_BCN],
    which is introduced in commit 859ee3c43812 ("DCB: Add support for DCB
    BCN"). Please see the comment in below code

    static int dcbnl_bcn_setcfg(...)
    {
      ...
      ret = nla_parse_nested_deprecated(..., dcbnl_pfc_up_nest, .. )
      // !!! dcbnl_pfc_up_nest for attributes
      //  DCB_PFC_UP_ATTR_0 to DCB_PFC_UP_ATTR_ALL in enum dcbnl_pfc_up_attrs
      ...
      for (i = DCB_BCN_ATTR_RP_0; i <= DCB_BCN_ATTR_RP_7; i++) {
      // !!! DCB_BCN_ATTR_RP_0 to DCB_BCN_ATTR_RP_7 in enum dcbnl_bcn_attrs
        ...
        value_byte = nla_get_u8(data[i]);
        ...
      }
```

```
      ...
      for (i = DCB_BCN_ATTR_BCNA_0; i <= DCB_BCN_ATTR_RI; i++) {
      // !!! DCB_BCN_ATTR_BCNA_0 to DCB_BCN_ATTR_RI in enum dcbnl_bcn_attrs
      ...
        value_int = nla_get_u32(data[i]);
      ...
      }
      ...
    }

    That is, the nla_parse_nested_deprecated uses dcbnl_pfc_up_nest
    attributes to parse nlattr defined in dcbnl_pfc_up_attrs. But the
    following access code fetch each nlattr as dcbnl_bcn_attrs attributes.
    By looking up the associated nla_policy for dcbnl_bcn_attrs. We can find
    the beginning part of these two policies are "same".

    static const struct nla_policy dcbnl_pfc_up_nest[...] = {
            [DCB_PFC_UP_ATTR_0]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_1]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_2]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_3]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_4]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_5]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_6]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_7]   = {.type = NLA_U8},
            [DCB_PFC_UP_ATTR_ALL] = {.type = NLA_FLAG},
    };

    static const struct nla_policy dcbnl_bcn_nest[...] = {
            [DCB_BCN_ATTR_RP_0]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_1]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_2]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_3]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_4]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_5]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_6]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_7]         = {.type = NLA_U8},
            [DCB_BCN_ATTR_RP_ALL]       = {.type = NLA_FLAG},
            // from here is somewhat different
            [DCB_BCN_ATTR_BCNA_0]       = {.type = NLA_U32},
            ...
            [DCB_BCN_ATTR_ALL]          = {.type = NLA_FLAG},
    };

    Therefore, the current code is buggy and this
    nla_parse_nested_deprecated could overflow the dcbnl_pfc_up_nest and use
    the adjacent nla_policy to parse attributes from DCB_BCN_ATTR_BCNA_0.

    Hence use the correct policy dcbnl_bcn_nest to parse the nested
    tb[DCB_ATTR_BCN] TLV.

    Fixes: 859ee3c43812 ("DCB: Add support for DCB BCN")
    Signed-off-by: Lin Ma <linma@zju.edu.cn>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Link: https://lore.kernel.org/r/20230801013248.87240-1-linma@zju.edu.cn
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 893096a7e5fd61cb666b4ead2fa69324e1f2aade
Author: Michael Chan <michael.chan@broadcom.com>
Date:   Mon Jul 31 07:20:43 2023 -0700

    bnxt_en: Fix max_mtu setting for multi-buf XDP

    [ Upstream commit 08450ea98ae98d5a35145b675b76db616046ea11 ]
```

```
    The existing code does not allow the MTU to be set to the maximum even
    after an XDP program supporting multiple buffers is attached.  Fix it
    to set the netdev->max_mtu to the maximum value if the attached XDP
    program supports mutiple buffers, regardless of the current MTU value.

    Also use a local variable dev instead of repeatedly using bp->dev.

    Fixes: 1dc4c557bfed ("bnxt: adding bnxt_xdp_build_skb to build skb from multibuffer
xdp_buff")
    Reviewed-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Reviewed-by: Ajit Khaparde <ajit.khaparde@broadcom.com>
    Reviewed-by: Andy Gospodarek <andrew.gospodarek@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Link: https://lore.kernel.org/r/20230731142043.58855-3-michael.chan@broadcom.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ae0e135dc900827687ecc684c2bbb57aae48d318
Author: Somnath Kotur <somnath.kotur@broadcom.com>
Date:   Mon Jul 31 07:20:42 2023 -0700

    bnxt_en: Fix page pool logic for page size >= 64K

    [ Upstream commit f6974b4c2d8e1062b5a52228ee47293c15b4ee1e ]

    The RXBD length field on all bnxt chips is 16-bit and so we cannot
    support a full page when the native page size is 64K or greater.
    The non-XDP (non page pool) code path has logic to handle this but
    the XDP page pool code path does not handle this.  Add the missing
    logic to use page_pool_dev_alloc_frag() to allocate 32K chunks if
    the page size is 64K or greater.

    Fixes: 9f4b28301ce6 ("bnxt: XDP multibuffer enablement")
    Link: https://lore.kernel.org/netdev/20230728231829.235716-2-michael.chan@broadcom.com/
    Reviewed-by: Andy Gospodarek <andrew.gospodarek@broadcom.com>
    Signed-off-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Link: https://lore.kernel.org/r/20230731142043.58855-2-michael.chan@broadcom.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5d015db7e60d1b2266bbf4febed8aed1ee9de596
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Mon Jul 31 11:15:53 2023 -0700

    selftest: net: Assert on a proper value in so_incoming_cpu.c.

    [ Upstream commit 3ff1617450eceb290ac17120fc172815e09a93cf ]

    Dan Carpenter reported an error spotted by Smatch.

      ./tools/testing/selftests/net/so_incoming_cpu.c:163 create_clients()
      error: uninitialized symbol 'ret'.

    The returned value of sched_setaffinity() should be checked with
    ASSERT_EQ(), but the value was not saved in a proper variable,
    resulting in an error above.

    Let's save the returned value of with sched_setaffinity().

    Fixes: 6df96146b202 ("selftest: Add test for SO_INCOMING_CPU.")
    Reported-by: Dan Carpenter <dan.carpenter@linaro.org>
    Closes: https://lore.kernel.org/linux-kselftest/fe376760-33b6-4fc9-88e8-
178e809af1ac@moroto.mountain/
```

```
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Link: https://lore.kernel.org/r/20230731181553.5392-1-kuniyu@amazon.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0c1763e20c665c63149d9ccae85110c2b9ae7273
Author: Mark Brown <broonie@kernel.org>
Date:   Mon Jul 31 11:48:32 2023 +0100

    net: netsec: Ignore 'phy-mode' on SynQuacer in DT mode

    [ Upstream commit f3bb7759a924713bc54d15f6d0d70733b5935fad ]

    As documented in acd7aaf51b20 ("netsec: ignore 'phy-mode' device
    property on ACPI systems") the SocioNext SynQuacer platform ships with
    firmware defining the PHY mode as RGMII even though the physical
    configuration of the PHY is for TX and RX delays.  Since bbc4d71d63549bc
    ("net: phy: realtek: fix rtl8211e rx/tx delay config") this has caused
    misconfiguration of the PHY, rendering the network unusable.

    This was worked around for ACPI by ignoring the phy-mode property but
    the system is also used with DT.  For DT instead if we're running on a
    SynQuacer force a working PHY mode, as well as the standard EDK2
    firmware with DT there are also some of these systems that use u-boot
    and might not initialise the PHY if not netbooting.  Newer firmware
    imagaes for at least EDK2 are available from Linaro so print a warning
    when doing this.

    Fixes: 533dd11a12f6 ("net: socionext: Add Synquacer NetSec driver")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Acked-by: Ard Biesheuvel <ardb@kernel.org>
    Acked-by: Ilias Apalodimas <ilias.apalodimas@linaro.org>
    Reviewed-by: Andrew Lunn <andrew@lunn.ch>
    Link: https://lore.kernel.org/r/20230731-synquacer-net-v3-1-944be5f06428@kernel.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d948a2ab44eabcb6de1b68ef10deb9119df8c4fe
Author: Yuanjun Gong <ruc_gongyuanjun@163.com>
Date:   Mon Jul 31 17:05:35 2023 +0800

    net: korina: handle clk prepare error in korina_probe()

    [ Upstream commit 0b6291ad1940c403734312d0e453e8dac9148f69 ]

    in korina_probe(), the return value of clk_prepare_enable()
    should be checked since it might fail. we can use
    devm_clk_get_optional_enabled() instead of devm_clk_get_optional()
    and clk_prepare_enable() to automatically handle the error.

    Fixes: e4cd854ec487 ("net: korina: Get mdio input clock via common clock framework")
    Signed-off-by: Yuanjun Gong <ruc_gongyuanjun@163.com>
    Link: https://lore.kernel.org/r/20230731090535.21416-1-ruc_gongyuanjun@163.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 18e647c0174957bc726305f713d8db4c3d0fdd0d
Author: Dan Carpenter <dan.carpenter@linaro.org>
Date:   Mon Jul 31 10:42:32 2023 +0300

    net: ll_temac: fix error checking of irq_of_parse_and_map()

    [ Upstream commit ef45e8400f5bb66b03cc949f76c80e2a118447de ]

    Most kernel functions return negative error codes but some irq functions
```

```
    return zero on error.  In this code irq_of_parse_and_map(), returns zero
    and platform_get_irq() returns negative error codes.  We need to handle
    both cases appropriately.

    Fixes: 8425c41d1ef7 ("net: ll_temac: Extend support to non-device-tree platforms")
    Signed-off-by: Dan Carpenter <dan.carpenter@linaro.org>
    Acked-by: Esben Haabendal <esben@geanix.com>
    Reviewed-by: Yang Yingliang <yangyingliang@huawei.com>
    Reviewed-by: Harini Katakam <harini.katakam@amd.com>
    Link: https://lore.kernel.org/r/3d0aef75-06e0-45a5-a2a6-2cc4738d4143@moroto.mountain
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1b7c8cbbd6f66d1105ad77446de89c2168324982
Author: Tomas Glozar <tglozar@redhat.com>
Date:   Fri Jul 28 08:44:11 2023 +0200

    bpf: sockmap: Remove preempt_disable in sock_map_sk_acquire

    [ Upstream commit 13d2618b48f15966d1adfe1ff6a1985f5eef40ba ]

    Disabling preemption in sock_map_sk_acquire conflicts with GFP_ATOMIC
    allocation later in sk_psock_init_link on PREEMPT_RT kernels, since
    GFP_ATOMIC might sleep on RT (see bpf: Make BPF and PREEMPT_RT co-exist
    patchset notes for details).

    This causes calling bpf_map_update_elem on BPF_MAP_TYPE_SOCKMAP maps to
    BUG (sleeping function called from invalid context) on RT kernels.

    preempt_disable was introduced together with lock_sk and rcu_read_lock
    in commit 99ba2b5aba24e ("bpf: sockhash, disallow bpf_tcp_close and update
    in parallel"), probably to match disabled migration of BPF programs, and
    is no longer necessary.

    Remove preempt_disable to fix BUG in sock_map_update_common on RT.

    Signed-off-by: Tomas Glozar <tglozar@redhat.com>
    Reviewed-by: Jakub Sitnicki <jakub@cloudflare.com>
    Link: https://lore.kernel.org/all/20200224140131.461979697@linutronix.de/
    Fixes: 99ba2b5aba24 ("bpf: sockhash, disallow bpf_tcp_close and update in parallel")
    Reviewed-by: John Fastabend <john.fastabend@gmail.com>
    Link: https://lore.kernel.org/r/20230728064411.305576-1-tglozar@redhat.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a836184b670f59e24d3a0f7c07115ec6e6ce6900
Author: valis <sec@valis.email>
Date:   Sat Jul 29 08:32:02 2023 -0400

    net/sched: cls_route: No longer copy tcf_result on update to avoid use-after-free

    [ Upstream commit b80b829e9e2c1b3f7aae34855e04d8f6ecaf13c8 ]

    When route4_change() is called on an existing filter, the whole
    tcf_result struct is always copied into the new instance of the filter.

    This causes a problem when updating a filter bound to a class,
    as tcf_unbind_filter() is always called on the old instance in the
    success path, decreasing filter_cnt of the still referenced class
    and allowing it to be deleted, leading to a use-after-free.

    Fix this by no longer copying the tcf_result struct from the old filter.

    Fixes: 1109c00547fc ("net: sched: RCU cls_route")
    Reported-by: valis <sec@valis.email>
```

```
    Reported-by: Bing-Jhong Billy Jheng <billy@starlabs.sg>
    Signed-off-by: valis <sec@valis.email>
    Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Reviewed-by: Victor Nogueira <victor@mojatatu.com>
    Reviewed-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: M A Ramdhan <ramdhan@starlabs.sg>
    Link: https://lore.kernel.org/r/20230729123202.72406-4-jhs@mojatatu.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 7d848d718aeb3b482e177b682dd04e76dd413afb
Author: valis <sec@valis.email>
Date:   Sat Jul 29 08:32:01 2023 -0400

    net/sched: cls_fw: No longer copy tcf_result on update to avoid use-after-free

    [ Upstream commit 76e42ae831991c828cffa8c37736ebfb831ad5ec ]

    When fw_change() is called on an existing filter, the whole
    tcf_result struct is always copied into the new instance of the filter.

    This causes a problem when updating a filter bound to a class,
    as tcf_unbind_filter() is always called on the old instance in the
    success path, decreasing filter_cnt of the still referenced class
    and allowing it to be deleted, leading to a use-after-free.

    Fix this by no longer copying the tcf_result struct from the old filter.

    Fixes: e35a8ee5993b ("net: sched: fw use RCU")
    Reported-by: valis <sec@valis.email>
    Reported-by: Bing-Jhong Billy Jheng <billy@starlabs.sg>
    Signed-off-by: valis <sec@valis.email>
    Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Reviewed-by: Victor Nogueira <victor@mojatatu.com>
    Reviewed-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: M A Ramdhan <ramdhan@starlabs.sg>
    Link: https://lore.kernel.org/r/20230729123202.72406-3-jhs@mojatatu.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4b717802428fa02cbcbb61209f638f65f9cd4710
Author: valis <sec@valis.email>
Date:   Sat Jul 29 08:32:00 2023 -0400

    net/sched: cls_u32: No longer copy tcf_result on update to avoid use-after-free

    [ Upstream commit 3044b16e7c6fe5d24b1cdbcf1bd0a9d92d1ebd81 ]

    When u32_change() is called on an existing filter, the whole
    tcf_result struct is always copied into the new instance of the filter.

    This causes a problem when updating a filter bound to a class,
    as tcf_unbind_filter() is always called on the old instance in the
    success path, decreasing filter_cnt of the still referenced class
    and allowing it to be deleted, leading to a use-after-free.

    Fix this by no longer copying the tcf_result struct from the old filter.

    Fixes: de5df63228fc ("net: sched: cls_u32 changes to knode must appear atomic to
readers")
    Reported-by: valis <sec@valis.email>
    Reported-by: M A Ramdhan <ramdhan@starlabs.sg>
    Signed-off-by: valis <sec@valis.email>
    Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Reviewed-by: Victor Nogueira <victor@mojatatu.com>
```

```
    Reviewed-by: Pedro Tammela <pctammela@mojatatu.com>
    Reviewed-by: M A Ramdhan <ramdhan@starlabs.sg>
    Link: https://lore.kernel.org/r/20230729123202.72406-2-jhs@mojatatu.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 937345720d18f1ad006ba3d5dcb3fa121037b8a2
Author: Hou Tao <houtao1@huawei.com>
Date:   Sat Jul 29 17:51:07 2023 +0800

    bpf, cpumap: Handle skb as well when clean up ptr_ring

    [ Upstream commit 7c62b75cd1a792e14b037fa4f61f9b18914e7de1 ]

    The following warning was reported when running xdp_redirect_cpu with
    both skb-mode and stress-mode enabled:

      ------------[ cut here ]------------
      Incorrect XDP memory type (-2128176192) usage
      WARNING: CPU: 7 PID: 1442 at net/core/xdp.c:405
      Modules linked in:
      CPU: 7 PID: 1442 Comm: kworker/7:0 Tainted: G  6.5.0-rc2+ #1
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996)
      Workqueue: events __cpu_map_entry_free
      RIP: 0010:__xdp_return+0x1e4/0x4a0
      ......
      Call Trace:
       <TASK>
       ? show_regs+0x65/0x70
       ? __warn+0xa5/0x240
       ? __xdp_return+0x1e4/0x4a0
       ......
       xdp_return_frame+0x4d/0x150
       __cpu_map_entry_free+0xf9/0x230
       process_one_work+0x6b0/0xb80
       worker_thread+0x96/0x720
       kthread+0x1a5/0x1f0
       ret_from_fork+0x3a/0x70
       ret_from_fork_asm+0x1b/0x30
       </TASK>

    The reason for the warning is twofold. One is due to the kthread
    cpu_map_kthread_run() is stopped prematurely. Another one is
    __cpu_map_ring_cleanup() doesn't handle skb mode and treats skbs in
    ptr_ring as XDP frames.

    Prematurely-stopped kthread will be fixed by the preceding patch and
    ptr_ring will be empty when __cpu_map_ring_cleanup() is called. But
    as the comments in __cpu_map_ring_cleanup() said, handling and freeing
    skbs in ptr_ring as well to "catch any broken behaviour gracefully".

    Fixes: 11941f8a8536 ("bpf: cpumap: Implement generic cpumap")
    Signed-off-by: Hou Tao <houtao1@huawei.com>
    Acked-by: Jesper Dangaard Brouer <hawk@kernel.org>
    Link: https://lore.kernel.org/r/20230729095107.1722450-3-houtao@huaweicloud.com
    Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ecb45b852af5e88257020b88bea5ff0798d72aca
Author: Hou Tao <houtao1@huawei.com>
Date:   Sat Jul 29 17:51:06 2023 +0800

    bpf, cpumap: Make sure kthread is running before map update returns

    [ Upstream commit 640a604585aa30f93e39b17d4d6ba69fcb1e66c9 ]
```

```
    The following warning was reported when running stress-mode enabled
    xdp_redirect_cpu with some RT threads:

      ------------[ cut here ]------------
      WARNING: CPU: 4 PID: 65 at kernel/bpf/cpumap.c:135
      CPU: 4 PID: 65 Comm: kworker/4:1 Not tainted 6.5.0-rc2+ #1
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996)
      Workqueue: events cpu_map_kthread_stop
      RIP: 0010:put_cpu_map_entry+0xda/0x220
      ......
      Call Trace:
       <TASK>
       ? show_regs+0x65/0x70
       ? __warn+0xa5/0x240
       ......
       ? put_cpu_map_entry+0xda/0x220
       cpu_map_kthread_stop+0x41/0x60
       process_one_work+0x6b0/0xb80
       worker_thread+0x96/0x720
       kthread+0x1a5/0x1f0
       ret_from_fork+0x3a/0x70
       ret_from_fork_asm+0x1b/0x30
       </TASK>

    The root cause is the same as commit 436901649731 ("bpf: cpumap: Fix memory
    leak in cpu_map_update_elem"). The kthread is stopped prematurely by
    kthread_stop() in cpu_map_kthread_stop(), and kthread() doesn't call
    cpu_map_kthread_run() at all but XDP program has already queued some
    frames or skbs into ptr_ring. So when __cpu_map_ring_cleanup() checks
    the ptr_ring, it will find it was not emptied and report a warning.

    An alternative fix is to use __cpu_map_ring_cleanup() to drop these
    pending frames or skbs when kthread_stop() returns -EINTR, but it may
    confuse the user, because these frames or skbs have been handled
    correctly by XDP program. So instead of dropping these frames or skbs,
    just make sure the per-cpu kthread is running before
    __cpu_map_entry_alloc() returns.

    After apply the fix, the error handle for kthread_stop() will be
    unnecessary because it will always return 0, so just remove it.

    Fixes: 6710e1126934 ("bpf: introduce new bpf cpu map type BPF_MAP_TYPE_CPUMAP")
    Signed-off-by: Hou Tao <houtao1@huawei.com>
    Reviewed-by: Pu Lehui <pulehui@huawei.com>
    Acked-by: Jesper Dangaard Brouer <hawk@kernel.org>
    Link: https://lore.kernel.org/r/20230729095107.1722450-2-houtao@huaweicloud.com
    Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 382535c063deff43ab194dad779bed7bccaa37fe
Author: Andrii Nakryiko <andrii@kernel.org>
Date:   Tue Jun 13 15:35:32 2023 -0700

    bpf: Centralize permissions checks for all BPF map types

    [ Upstream commit 6c3eba1c5e283fd2bb1c076dbfcb47f569c3bfde ]

    This allows to do more centralized decisions later on, and generally
    makes it very explicit which maps are privileged and which are not
    (e.g., LRU_HASH and LRU_PERCPU_HASH, which are privileged HASH variants,
    as opposed to unprivileged HASH and HASH_PERCPU; now this is explicit
    and easy to verify).

    Signed-off-by: Andrii Nakryiko <andrii@kernel.org>
```

```
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Stanislav Fomichev <sdf@google.com>
    Link: https://lore.kernel.org/bpf/20230613223533.3689589-4-andrii@kernel.org
    Stable-dep-of: 640a604585aa ("bpf, cpumap: Make sure kthread is running before map
update returns")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 514b9c25ffe61ce4f14779677c804b207b64578d
Author: Andrii Nakryiko <andrii@kernel.org>
Date:   Tue Jun 13 15:35:31 2023 -0700

    bpf: Inline map creation logic in map_create() function

    [ Upstream commit 22db41226b679768df8f0a4ff5de8e58f625f45b ]

    Currently find_and_alloc_map() performs two separate functions: some
    argument sanity checking and partial map creation workflow hanling.
    Neither of those functions are self-sufficient and are augmented by
    further checks and initialization logic in the caller (map_create()
    function). So unify all the sanity checks, permission checks, and
    creation and initialization logic in one linear piece of code in
    map_create() instead. This also make it easier to further enhance
    permission checks and keep them located in one place.

    Signed-off-by: Andrii Nakryiko <andrii@kernel.org>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Stanislav Fomichev <sdf@google.com>
    Link: https://lore.kernel.org/bpf/20230613223533.3689589-3-andrii@kernel.org
    Stable-dep-of: 640a604585aa ("bpf, cpumap: Make sure kthread is running before map
update returns")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3cf214f2755bb22bf056e6a155f4314bb6378d50
Author: Andrii Nakryiko <andrii@kernel.org>
Date:   Tue Jun 13 15:35:30 2023 -0700

    bpf: Move unprivileged checks into map_create() and bpf_prog_load()

    [ Upstream commit 1d28635abcf1914425d6516e641978011984c58a ]

    Make each bpf() syscall command a bit more self-contained, making it
    easier to further enhance it. We move sysctl_unprivileged_bpf_disabled
    handling down to map_create() and bpf_prog_load(), two special commands
    in this regard.

    Also swap the order of checks, calling bpf_capable() only if
    sysctl_unprivileged_bpf_disabled is true, avoiding unnecessary audit
    messages.

    Signed-off-by: Andrii Nakryiko <andrii@kernel.org>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Stanislav Fomichev <sdf@google.com>
    Link: https://lore.kernel.org/bpf/20230613223533.3689589-2-andrii@kernel.org
    Stable-dep-of: 640a604585aa ("bpf, cpumap: Make sure kthread is running before map
update returns")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0acc483f9b4d81d0013c39ddf2f0076268a5e9d8
Author: Michal Schmidt <mschmidt@redhat.com>
Date:   Sat Jul 29 17:15:16 2023 +0200

    octeon_ep: initialize mbox mutexes

    [ Upstream commit 611e1b016c7beceec5ae82ac62d4a7ca224c8f9d ]
```

```
    The two mbox-related mutexes are destroyed in octep_ctrl_mbox_uninit(),
    but the corresponding mutex_init calls were missing.
    A "DEBUG_LOCKS_WARN_ON(lock->magic != lock)" warning was emitted with
    CONFIG_DEBUG_MUTEXES on.

    Initialize the two mutexes in octep_ctrl_mbox_init().

    Fixes: 577f0d1b1c5f ("octeon_ep: add separate mailbox command and response queues")
    Signed-off-by: Michal Schmidt <mschmidt@redhat.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Link: https://lore.kernel.org/r/20230729151516.24153-1-mschmidt@redhat.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 34115ad1e5b5a9199cdf313c569ea761eb97132e
Author: Jakub Kicinski <kuba@kernel.org>
Date:   Fri Jul 28 13:50:20 2023 -0700

    bnxt: don't handle XDP in netpoll

    [ Upstream commit 37b61cda9c1606cd8b6445d900ca9dc03185e8b6 ]

    Similarly to other recently fixed drivers make sure we don't
    try to access XDP or page pool APIs when NAPI budget is 0.
    NAPI budget of 0 may mean that we are in netpoll.

    This may result in running software IRQs in hard IRQ context,
    leading to deadlocks or crashes.

    To make sure bnapi->tx_pkts don't get wiped without handling
    the events, move clearing the field into the handler itself.
    Remember to clear tx_pkts after reset (bnxt_enable_napi())
    as it's technically possible that netpoll will accumulate
    some tx_pkts and then a reset will happen, leaving tx_pkts
    out of sync with reality.

    Fixes: 322b87ca55f2 ("bnxt_en: add page_pool support")
    Reviewed-by: Andy Gospodarek <gospo@broadcom.com>
    Reviewed-by: Michael Chan <michael.chan@broadcom.com>
    Link: https://lore.kernel.org/r/20230728205020.2784844-1-kuba@kernel.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f7cdae9fbda20f975ebc232153c1f6c8fe56e750
Author: Rafal Rogalski <rafalx.rogalski@intel.com>
Date:   Fri Jul 28 10:12:43 2023 -0700

    ice: Fix RDMA VSI removal during queue rebuild

    [ Upstream commit 4b31fd4d77ffa430d0b74ba1885ea0a41594f202 ]

    During qdisc create/delete, it is necessary to rebuild the queue
    of VSIs. An error occurred because the VSIs created by RDMA were
    still active.

    Added check if RDMA is active. If yes, it disallows qdisc changes
    and writes a message in the system logs.

    Fixes: 348048e724a0 ("ice: Implement iidc operations")
    Signed-off-by: Rafal Rogalski <rafalx.rogalski@intel.com>
    Signed-off-by: Mateusz Palczewski <mateusz.palczewski@intel.com>
    Signed-off-by: Kamil Maziarz <kamil.maziarz@intel.com>
    Tested-by: Bharathi Sreenivas <bharathi.sreenivas@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
```

```
    Link: https://lore.kernel.org/r/20230728171243.2446101-1-anthony.l.nguyen@intel.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a54bf862ccad27c8b91795b69a9ca105f8e6295e
Author: Duoming Zhou <duoming@zju.edu.cn>
Date:   Wed Jul 26 16:14:07 2023 +0800

    net: usb: lan78xx: reorder cleanup operations to avoid UAF bugs

    [ Upstream commit 1e7417c188d0a83fb385ba2dbe35fd2563f2b6f3 ]

    The timer dev->stat_monitor can schedule the delayed work dev->wq and
    the delayed work dev->wq can also arm the dev->stat_monitor timer.

    When the device is detaching, the net_device will be deallocated. but
    the net_device private data could still be dereferenced in delayed work
    or timer handler. As a result, the UAF bugs will happen.

    One racy situation is shown below:

          (Thread 1)                 |      (Thread 2)
    lan78xx_stat_monitor()           |
     ...                             |  lan78xx_disconnect()
     lan78xx_defer_kevent()          |    ...
      ...                            |    cancel_delayed_work_sync(&dev->wq);
      schedule_delayed_work()        |    ...
      (wait some time)               |    free_netdev(net); //free net_device
      lan78xx_delayedwork()          |
      //use net_device private data  |
      dev-> //use                    |

    Although we use cancel_delayed_work_sync() to cancel the delayed work
    in lan78xx_disconnect(), it could still be scheduled in timer handler
    lan78xx_stat_monitor().

    Another racy situation is shown below:

          (Thread 1)                 |      (Thread 2)
    lan78xx_delayedwork              |
     mod_timer()                     |  lan78xx_disconnect()
                                     |   cancel_delayed_work_sync()
     (wait some time)                |   if (timer_pending(&dev->stat_monitor))
                                     |       del_timer_sync(&dev->stat_monitor);
     lan78xx_stat_monitor()          |   ...
      lan78xx_defer_kevent()         |   free_netdev(net); //free
       //use net_device private data|
       dev-> //use                   |

    Although we use del_timer_sync() to delete the timer, the function
    timer_pending() returns 0 when the timer is activated. As a result,
    the del_timer_sync() will not be executed and the timer could be
    re-armed.

    In order to mitigate this bug, We use timer_shutdown_sync() to shutdown
    the timer and then use cancel_delayed_work_sync() to cancel the delayed
    work. As a result, the net_device could be deallocated safely.

    What's more, the dev->flags is set to EVENT_DEV_DISCONNECT in
    lan78xx_disconnect(). But it could still be set to EVENT_STAT_UPDATE
    in lan78xx_stat_monitor(). So this patch put the set_bit() behind
    timer_shutdown_sync().

    Fixes: 77dfff5bb7e2 ("lan78xx: Fix race condition in disconnect handling")
    Signed-off-by: Duoming Zhou <duoming@zju.edu.cn>
```

```
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 57b3fe08ae06ef11af007b4a182629b12a961e30
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Jul 28 17:07:05 2023 -0700

    net/sched: taprio: Limit TCA_TAPRIO_ATTR_SCHED_CYCLE_TIME to INT_MAX.

    [ Upstream commit e739718444f7bf2fa3d70d101761ad83056ca628 ]

    syzkaller found zero division error [0] in div_s64_rem() called from
    get_cycle_time_elapsed(), where sched->cycle_time is the divisor.

    We have tests in parse_taprio_schedule() so that cycle_time will never
    be 0, and actually cycle_time is not 0 in get_cycle_time_elapsed().

    The problem is that the types of divisor are different; cycle_time is
    s64, but the argument of div_s64_rem() is s32.

    syzkaller fed this input and 0x100000000 is cast to s32 to be 0.

      @TCA_TAPRIO_ATTR_SCHED_CYCLE_TIME={0xc, 0x8, 0x100000000}

    We use s64 for cycle_time to cast it to ktime_t, so let's keep it and
    set max for cycle_time.

    While at it, we prevent overflow in setup_txtime() and add another
    test in parse_taprio_schedule() to check if cycle_time overflows.

    Also, we add a new tdc test case for this issue.

    [0]:
    divide error: 0000 [#1] PREEMPT SMP KASAN NOPTI
    CPU: 1 PID: 103 Comm: kworker/1:3 Not tainted 6.5.0-rc1-00330-g60cc1f7d0605 #3
    Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-
prebuilt.qemu.org 04/01/2014
    Workqueue: ipv6_addrconf addrconf_dad_work
    RIP: 0010:div_s64_rem include/linux/math64.h:42 [inline]
    RIP: 0010:get_cycle_time_elapsed net/sched/sch_taprio.c:223 [inline]
    RIP: 0010:find_entry_to_transmit+0x252/0x7e0 net/sched/sch_taprio.c:344
    Code: 3c 02 00 0f 85 5e 05 00 00 48 8b 4c 24 08 4d 8b bd 40 01 00 00 48 8b 7c 24 48 48
89 c8 4c 29 f8 48 63 f7 48 99 48 89 74 24 70 <48> f7 fe 48 29 d1 48 8d 04 0f 49 89 cc 48 89
44 24 20 49 8d 85 10
    RSP: 0018:ffffc90000acf260 EFLAGS: 00010206
    RAX: 177450e0347560cf RBX: 0000000000000000 RCX: 177450e0347560cf
    RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000100000000
    RBP: 0000000000000056 R08: 0000000000000000 R09: ffffed10020a0934
    R10: ffff8880105049a7 R11: ffff88806cf3a520 R12: ffff888010504800
    R13: ffff88800c00d800 R14: ffff8880105049a0 R15: 0000000000000000
    FS:  0000000000000000(0000) GS:ffff88806cf00000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007f0edf84f0e8 CR3: 000000000d73c002 CR4: 0000000000770ee0
    PKRU: 55555554
    Call Trace:
     <TASK>
     get_packet_txtime net/sched/sch_taprio.c:508 [inline]
     taprio_enqueue_one+0x900/0xff0 net/sched/sch_taprio.c:577
     taprio_enqueue+0x378/0xae0 net/sched/sch_taprio.c:658
     dev_qdisc_enqueue+0x46/0x170 net/core/dev.c:3732
     __dev_xmit_skb net/core/dev.c:3821 [inline]
     __dev_queue_xmit+0x1b2f/0x3000 net/core/dev.c:4169
     dev_queue_xmit include/linux/netdevice.h:3088 [inline]
     neigh_resolve_output net/core/neighbour.c:1552 [inline]
     neigh_resolve_output+0x4a7/0x780 net/core/neighbour.c:1532
```

```
       neigh_output include/net/neighbour.h:544 [inline]
       ip6_finish_output2+0x924/0x17d0 net/ipv6/ip6_output.c:135
       __ip6_finish_output+0x620/0xaa0 net/ipv6/ip6_output.c:196
       ip6_finish_output net/ipv6/ip6_output.c:207 [inline]
       NF_HOOK_COND include/linux/netfilter.h:292 [inline]
       ip6_output+0x206/0x410 net/ipv6/ip6_output.c:228
       dst_output include/net/dst.h:458 [inline]
       NF_HOOK.constprop.0+0xea/0x260 include/linux/netfilter.h:303
       ndisc_send_skb+0x872/0xe80 net/ipv6/ndisc.c:508
       ndisc_send_ns+0xb5/0x130 net/ipv6/ndisc.c:666
       addrconf_dad_work+0xc14/0x13f0 net/ipv6/addrconf.c:4175
       process_one_work+0x92c/0x13a0 kernel/workqueue.c:2597
       worker_thread+0x60f/0x1240 kernel/workqueue.c:2748
       kthread+0x2fe/0x3f0 kernel/kthread.c:389
       ret_from_fork+0x2c/0x50 arch/x86/entry/entry_64.S:308
       </TASK>
      Modules linked in:

      Fixes: 4cfd5779bd6e ("taprio: Add support for txtime-assist mode")
      Reported-by: syzkaller <syzkaller@googlegroups.com>
      Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
      Co-developed-by: Eric Dumazet <edumazet@google.com>
      Co-developed-by: Pedro Tammela <pctammela@mojatatu.com>
      Acked-by: Vinicius Costa Gomes <vinicius.gomes@intel.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 7e7c4fdee5c755544423cf748edc063759761287
  Author: Eric Dumazet <edumazet@google.com>
  Date:   Fri Jul 28 15:03:18 2023 +0000

      net: annotate data-races around sk->sk_priority

      [ Upstream commit 8bf43be799d4b242ea552a14db10456446be843e ]

      sk_getsockopt() runs locklessly. This means sk->sk_priority
      can be read while other threads are changing its value.

      Other reads also happen without socket lock being held.

      Add missing annotations where needed.

      Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
      Signed-off-by: Eric Dumazet <edumazet@google.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 9ceaff1556dc7ed71614f6b7677090fe104fcdbb
  Author: Eric Dumazet <edumazet@google.com>
  Date:   Fri Jul 28 15:03:17 2023 +0000

      net: add missing data-race annotation for sk_ll_usec

      [ Upstream commit e5f0d2dd3c2faa671711dac6d3ff3cef307bcfe3 ]

      In a prior commit I forgot that sk_getsockopt() reads
      sk->sk_ll_usec without holding a lock.

      Fixes: 0dbffbb5335a ("net: annotate data race around sk_ll_usec")
      Signed-off-by: Eric Dumazet <edumazet@google.com>
      Signed-off-by: David S. Miller <davem@davemloft.net>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit eb2604f073eefe57c524291fa18ce9949303e52e
  Author: Eric Dumazet <edumazet@google.com>
```

```
Date:   Fri Jul 28 15:03:16 2023 +0000

    net: add missing data-race annotations around sk->sk_peek_off

    [ Upstream commit 11695c6e966b0ec7ed1d16777d294cef865a5c91 ]

    sk_getsockopt() runs locklessly, thus we need to annotate the read
    of sk->sk_peek_off.

    While we are at it, add corresponding annotations to sk_set_peek_off()
    and unix_set_peek_off().

    Fixes: b9bb53f3836f ("sock: convert sk_peek_offset functions to WRITE_ONCE")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Willem de Bruijn <willemb@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b76d2fa662b7292fec3d095eb7864fa9eb14a480
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:15 2023 +0000

    net: annotate data-races around sk->sk_mark

    [ Upstream commit 3c5b4d69c358a9275a8de98f87caf6eda644b086 ]

    sk->sk_mark is often read while another thread could change the value.

    Fixes: 4a19ec5800fc ("[NET]: Introducing socket mark socket option.")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea47de09819b1bec22a5eab95bac5ebb38bbae52
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:14 2023 +0000

    net: add missing READ_ONCE(sk->sk_rcvbuf) annotation

    [ Upstream commit b4b553253091cafe9ec38994acf42795e073bef5 ]

    In a prior commit, I forgot to change sk_getsockopt()
    when reading sk->sk_rcvbuf locklessly.

    Fixes: ebb3b78db7bf ("tcp: annotate sk->sk_rcvbuf lockless reads")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4b5bda4502dd1999b0a8786dc43863ee59e21292
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:13 2023 +0000

    net: add missing READ_ONCE(sk->sk_sndbuf) annotation

    [ Upstream commit 74bc084327c643499474ba75df485607da37dd6e ]

    In a prior commit, I forgot to change sk_getsockopt()
    when reading sk->sk_sndbuf locklessly.

    Fixes: e292f05e0df7 ("tcp: annotate sk->sk_sndbuf lockless reads")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 4685a86b9f25993954056546cd90c32091d11f62
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:11 2023 +0000

    net: add missing READ_ONCE(sk->sk_rcvlowat) annotation

    [ Upstream commit e6d12bdb435d23ff6c1890c852d85408a2f496ee ]

    In a prior commit, I forgot to change sk_getsockopt()
    when reading sk->sk_rcvlowat locklessly.

    Fixes: eac66402d1c3 ("net: annotate sk->sk_rcvlowat lockless reads")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 98ee7a0fe181196387c86366583a1e6ab37e5f7f
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:10 2023 +0000

    net: annotate data-races around sk->sk_max_pacing_rate

    [ Upstream commit ea7f45ef77b39e72244d282e47f6cb1ef4135cd2 ]

    sk_getsockopt() runs locklessly. This means sk->sk_max_pacing_rate
    can be read while other threads are changing its value.

    Fixes: 62748f32d501 ("net: introduce SO_MAX_PACING_RATE")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d0e273bc559bdcea8f4757c37b238b751de732ed
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:09 2023 +0000

    net: annotate data-race around sk->sk_txrehash

    [ Upstream commit c76a0328899bbe226f8adeb88b8da9e4167bd316 ]

    sk_getsockopt() runs locklessly. This means sk->sk_txrehash
    can be read while other threads are changing its value.

    Other locations were handled in commit cb6cd2cec799
    ("tcp: Change SYN ACK retransmit behaviour to account for rehash")

    Fixes: 26859240e4ee ("txhash: Add socket option to control TX hash rethink behavior")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Akhmat Karakotov <hmukos@yandex-team.ru>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6269d3eaf443b2656b03f88acd89fea504278b5c
Author: Eric Dumazet <edumazet@google.com>
Date:   Fri Jul 28 15:03:08 2023 +0000

    net: annotate data-races around sk->sk_reserved_mem

    [ Upstream commit fe11fdcb4207907d80cda2e73777465d68131e66 ]

    sk_getsockopt() runs locklessly. This means sk->sk_reserved_mem
    can be read while other threads are changing its value.

    Add missing annotations where they are needed.
```

```
    Fixes: 2bb2f5fb21b0 ("net: add new socket option SO_RESERVE_MEM")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Wei Wang <weiwan@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5ac34598be7bddea5556f1145fb9332ed0b9a6b2
Author: Richard Gobert <richardbgobert@gmail.com>
Date:   Thu Jul 27 17:33:56 2023 +0200

    net: gro: fix misuse of CB in udp socket lookup

    [ Upstream commit 7938cd15436873f649f31cb867bac2d88ca564d0 ]

    This patch fixes a misuse of IP{6}CB(skb) in GRO, while calling to
    `udp6_lib_lookup2` when handling udp tunnels. `udp6_lib_lookup2` fetch the
    device from CB. The fix changes it to fetch the device from `skb->dev`.
    l3mdev case requires special attention since it has a master and a slave
    device.

    Fixes: a6024562ffd7 ("udp: Add GRO functions to UDP socket")
    Reported-by: Gal Pressman <gal@nvidia.com>
    Signed-off-by: Richard Gobert <richardbgobert@gmail.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bbe07adbaf39c2c5a95c3ca7eb52b2119d50af7d
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu Jun 8 19:17:37 2023 +0000

    net: move gso declarations and functions to their own files

    [ Upstream commit d457a0e329b0bfd3a1450e0b1a18cd2b47a25a08 ]

    Move declarations into include/net/gso.h and code into net/core/gso.c

    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: Stanislav Fomichev <sdf@google.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Reviewed-by: David Ahern <dsahern@kernel.org>
    Link: https://lore.kernel.org/r/20230608191738.3947077-1-edumazet@google.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Stable-dep-of: 7938cd154368 ("net: gro: fix misuse of CB in udp socket lookup")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3e0d2545f94a30aa62156904fabbdb553c2a481f
Author: Konstantin Khorenko <khorenko@virtuozzo.com>
Date:   Thu Jul 27 18:26:09 2023 +0300

    qed: Fix scheduling in a tasklet while getting stats

    [ Upstream commit e346e231b42bcae6822a6326acfb7b741e9e6026 ]

    Here we've got to a situation when tasklet called usleep_range() in PTT
    acquire logic, thus welcome to the "scheduling while atomic" BUG().

      BUG: scheduling while atomic: swapper/24/0/0x00000100

       [<ffffffffb41c6199>] schedule+0x29/0x70
       [<ffffffffb41c5512>] schedule_hrtimeout_range_clock+0xb2/0x150
       [<ffffffffb41c55c3>] schedule_hrtimeout_range+0x13/0x20
       [<ffffffffb41c3bcf>] usleep_range+0x4f/0x70
       [<ffffffffc08d3e58>] qed_ptt_acquire+0x38/0x100 [qed]
       [<ffffffffc08eac48>] _qed_get_vport_stats+0x458/0x580 [qed]
```

```
        [<ffffffffc08ead8c>] qed_get_vport_stats+0x1c/0xd0 [qed]
        [<ffffffffc08dffd3>] qed_get_protocol_stats+0x93/0x100 [qed]
                             qed_mcp_send_protocol_stats
                 case MFW_DRV_MSG_GET_LAN_STATS:
                 case MFW_DRV_MSG_GET_FCOE_STATS:
                 case MFW_DRV_MSG_GET_ISCSI_STATS:
                 case MFW_DRV_MSG_GET_RDMA_STATS:
        [<ffffffffc08e36d8>] qed_mcp_handle_events+0x2d8/0x890 [qed]
                             qed_int_assertion
                             qed_int_attentions
        [<ffffffffc08d9490>] qed_int_sp_dpc+0xa50/0xdc0 [qed]
        [<ffffffffb3aa7623>] tasklet_action+0x83/0x140
        [<ffffffffb41d9125>] __do_softirq+0x125/0x2bb
        [<ffffffffb41d560c>] call_softirq+0x1c/0x30
        [<ffffffffb3a30645>] do_softirq+0x65/0xa0
        [<ffffffffb3aa78d5>] irq_exit+0x105/0x110
        [<ffffffffb41d8996>] do_IRQ+0x56/0xf0
```

    Fix this by making caller to provide the context whether it could be in
    atomic context flow or not when getting stats from QED driver.
    QED driver based on the context provided decide to schedule out or not
    when acquiring the PTT BAR window.

    We faced the BUG_ON() while getting vport stats, but according to the
    code same issue could happen for fcoe and iscsi statistics as well, so
    fixing them too.

    Fixes: 6c75424612a7 ("qed: Add support for NCSI statistics.")
    Fixes: 1e128c81290a ("qed: Add support for hardware offloaded FCoE.")
    Fixes: 2f2b2614e893 ("qed: Provide iSCSI statistics to management")
    Cc: Sudarsana Kalluru <skalluru@marvell.com>
    Cc: David Miller <davem@davemloft.net>
    Cc: Manish Chopra <manishc@marvell.com>

    Signed-off-by: Konstantin Khorenko <khorenko@virtuozzo.com>
    Reviewed-by: Simon Horman <horms@kernel.org>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3a234a4867914b4a8848fd8076f98f6f0060b1da
Author: Thierry Reding <treding@nvidia.com>
Date: Wed Jul 26 18:32:00 2023 +0200

    net: stmmac: tegra: Properly allocate clock bulk data

    [Upstream commit a0b1b2055be34c0ec1371764d040164cde1ead79]

    The clock data is an array of struct clk_bulk_data, so make sure to
    allocate enough memory.

    Fixes: d8ca113724e7 ("net: stmmac: tegra: Add MGBE support")
    Signed-off-by: Thierry Reding <treding@nvidia.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ea496e48d656b23e51a8ee7d0e714e80c4df072d
Author: Chengfeng Ye <dg573847474@gmail.com>
Date: Thu Jul 27 08:56:19 2023 +0000

    mISDN: hfcpci: Fix potential deadlock on &hc->lock

    [Upstream commit 56c6be35fcbed54279df0a2c9e60480a61841d6f]

    As &hc->lock is acquired by both timer _hfcpci_softirq() and hardirq

```
    hfcpci_int(), the timer should disable irq before lock acquisition
    otherwise deadlock could happen if the timmer is preemtped by the hadr irq.

    Possible deadlock scenario:
    hfcpci_softirq() (timer)
        -> _hfcpci_softirq()
        -> spin_lock(&hc->lock);
            <irq interruption>
            -> hfcpci_int()
            -> spin_lock(&hc->lock); (deadlock here)

    This flaw was found by an experimental static analysis tool I am developing
    for irq-related deadlock.

    The tentative patch fixes the potential deadlock by spin_lock_irq()
    in timer.

    Fixes: b36b654a7e82 ("mISDN: Create /sys/class/mISDN")
    Signed-off-by: Chengfeng Ye <dg573847474@gmail.com>
    Link: https://lore.kernel.org/r/20230727085619.7419-1-dg573847474@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit de14cff7164d9333230ed3d9e59f8b8f66ef8cc7
Author: Jamal Hadi Salim <jhs@mojatatu.com>
Date:   Wed Jul 26 09:51:51 2023 -0400

    net: sched: cls_u32: Fix match key mis-addressing

    [ Upstream commit e68409db995380d1badacba41ff24996bd396171 ]

    A match entry is uniquely identified with an "address" or "path" in the
    form of: hashtable ID(12b):bucketid(8b):nodeid(12b).

    When creating table match entries all of hash table id, bucket id and
    node (match entry id) are needed to be either specified by the user or
    reasonable in-kernel defaults are used. The in-kernel default for a table id is
    0x800(omnipresent root table); for bucketid it is 0x0. Prior to this fix there
    was none for a nodeid i.e. the code assumed that the user passed the correct
    nodeid and if the user passes a nodeid of 0 (as Mingi Cho did) then that is what
    was used. But nodeid of 0 is reserved for identifying the table. This is not
    a problem until we dump. The dump code notices that the nodeid is zero and
    assumes it is referencing a table and therefore references table struct
    tc_u_hnode instead of what was created i.e match entry struct tc_u_knode.

    Ming does an equivalent of:
    tc filter add dev dummy0 parent 10: prio 1 handle 0x1000 \
    protocol ip u32 match ip src 10.0.0.1/32 classid 10:1 action ok

    Essentially specifying a table id 0, bucketid 1 and nodeid of zero
    Tableid 0 is remapped to the default of 0x800.
    Bucketid 1 is ignored and defaults to 0x00.
    Nodeid was assumed to be what Ming passed - 0x000

    dumping before fix shows:
    ~$ tc filter ls dev dummy0 parent 10:
    filter protocol ip pref 1 u32 chain 0
    filter protocol ip pref 1 u32 chain 0 fh 800: ht divisor 1
    filter protocol ip pref 1 u32 chain 0 fh 800: ht divisor -30591

    Note that the last line reports a table instead of a match entry
    (you can tell this because it says "ht divisor...").
    As a result of reporting the wrong data type (misinterpretting of struct
    tc_u_knode as being struct tc_u_hnode) the divisor is reported with value
    of -30591. Ming identified this as part of the heap address
```

```
    (physmap_base is 0xffff8880 (-30591 - 1)).

    The fix is to ensure that when table entry matches are added and no
    nodeid is specified (i.e nodeid == 0) then we get the next available
    nodeid from the table's pool.

    After the fix, this is what the dump shows:
    $ tc filter ls dev dummy0 parent 10:
    filter protocol ip pref 1 u32 chain 0
    filter protocol ip pref 1 u32 chain 0 fh 800: ht divisor 1
    filter protocol ip pref 1 u32 chain 0 fh 800::800 order 2048 key ht 800 bkt 0 flowid
10:1 not_in_hw
      match 0a000001/ffffffff at 12
            action order 1: gact action pass
             random type none pass val 0
             index 1 ref 1 bind 1

    Reported-by: Mingi Cho <mgcho.minic@gmail.com>
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Jamal Hadi Salim <jhs@mojatatu.com>
    Link: https://lore.kernel.org/r/20230726135151.416917-1-jhs@mojatatu.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4034838ab0d54242c9d749bafb8404ccd01f2c91
Author: Georg Müller <georgmueller@gmx.net>
Date:   Fri Jul 28 17:18:12 2023 +0200

    perf test uprobe_from_different_cu: Skip if there is no gcc

    [ Upstream commit 98ce8e4a9dcfb448b30a2d7a16190f4a00382377 ]

    Without gcc, the test will fail.

    On cleanup, ignore probe removal errors. Otherwise, in case of an error
    adding the probe, the temporary directory is not removed.

    Fixes: 56cbeacf14353057 ("perf probe: Add test for regression introduced by switch to
die_get_decl_file()")
    Signed-off-by: Georg Müller <georgmueller@gmx.net>
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Georg Müller <georgmueller@gmx.net>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Masami Hiramatsu <mhiramat@kernel.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link: https://lore.kernel.org/r/20230728151812.454806-2-georgmueller@gmx.net
    Link: https://lore.kernel.org/r/CAP-
5=fUP6UuLgRty3t2=fQsQi3k4hDMz415vWdp1x88QMvZ8ug@mail.gmail.com/
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 76d0f82f68a2a1120fdd359575cc7946637dd72a
Author: Yuanjun Gong <ruc_gongyuanjun@163.com>
Date:   Thu Jul 27 01:05:06 2023 +0800

    net: dsa: fix value check in bcm_sf2_sw_probe()

    [ Upstream commit dadc5b86cc9459581f37fe755b431adc399ea393 ]

    in bcm_sf2_sw_probe(), check the return value of clk_prepare_enable()
```

```
    and return the error code if clk_prepare_enable() returns an
    unexpected value.

    Fixes: e9ec5c3bd238 ("net: dsa: bcm_sf2: request and handle clocks")
    Signed-off-by: Yuanjun Gong <ruc_gongyuanjun@163.com>
    Reviewed-by: Florian Fainelli <florian.fainelli@broadcom.com>
    Link: https://lore.kernel.org/r/20230726170506.16547-1-ruc_gongyuanjun@163.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 00757f58e37b2d9a6f99e15be484712390cd2bab
Author: Lin Ma <linma@zju.edu.cn>
Date:   Wed Jul 26 15:53:14 2023 +0800

    rtnetlink: let rtnl_bridge_setlink checks IFLA_BRIDGE_MODE length

    [ Upstream commit d73ef2d69c0dba5f5a1cb9600045c873bab1fb7f ]

    There are totally 9 ndo_bridge_setlink handlers in the current kernel,
    which are 1) bnxt_bridge_setlink, 2) be_ndo_bridge_setlink 3)
    i40e_ndo_bridge_setlink 4) ice_bridge_setlink 5)
    ixgbe_ndo_bridge_setlink 6) mlx5e_bridge_setlink 7)
    nfp_net_bridge_setlink 8) qeth_l2_bridge_setlink 9) br_setlink.

    By investigating the code, we find that 1-7 parse and use nlattr
    IFLA_BRIDGE_MODE but 3 and 4 forget to do the nla_len check. This can
    lead to an out-of-attribute read and allow a malformed nlattr (e.g.,
    length 0) to be viewed as a 2 byte integer.

    To avoid such issues, also for other ndo_bridge_setlink handlers in the
    future. This patch adds the nla_len check in rtnl_bridge_setlink and
    does an early error return if length mismatches. To make it works, the
    break is removed from the parsing for IFLA_BRIDGE_FLAGS to make sure
    this nla_for_each_nested iterates every attribute.

    Fixes: b1edc14a3fbf ("ice: Implement ice_bridge_getlink and ice_bridge_setlink")
    Fixes: 51616018dd1b ("i40e: Add support for getlink, setlink ndo ops")
    Suggested-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Lin Ma <linma@zju.edu.cn>
    Acked-by: Nikolay Aleksandrov <razor@blackwall.org>
    Reviewed-by: Hangbin Liu <liuhangbin@gmail.com>
    Link: https://lore.kernel.org/r/20230726075314.1059224-1-linma@zju.edu.cn
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 95b2e27b11398e7d716864cdc8f61e61b3d9dbce
Author: Lin Ma <linma@zju.edu.cn>
Date:   Tue Jul 25 10:33:30 2023 +0800

    bpf: Add length check for SK_DIAG_BPF_STORAGE_REQ_MAP_FD parsing

    [ Upstream commit bcc29b7f5af6797702c2306a7aacb831fc5ce9cb ]

    The nla_for_each_nested parsing in function bpf_sk_storage_diag_alloc
    does not check the length of the nested attribute. This can lead to an
    out-of-attribute read and allow a malformed nlattr (e.g., length 0) to
    be viewed as a 4 byte integer.

    This patch adds an additional check when the nlattr is getting counted.
    This makes sure the latter nla_get_u32 can access the attributes with
    the correct length.

    Fixes: 1ed4d92458a9 ("bpf: INET_DIAG support in bpf_sk_storage")
    Suggested-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Lin Ma <linma@zju.edu.cn>
```

```
    Reviewed-by: Jakub Kicinski <kuba@kernel.org>
    Link: https://lore.kernel.org/r/20230725023330.422856-1-linma@zju.edu.cn
    Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 471f59b3455314f0cafacf3096453727876355a9
Author: Shay Drory <shayd@nvidia.com>
Date:   Sun Jun 25 11:07:38 2023 +0300

    net/mlx5: Unregister devlink params in case interface is down

    [ Upstream commit 53d737dfd3d7b023fa9fa445ea3f3db0ac9da402 ]

    Currently, in case an interface is down, mlx5 driver doesn't
    unregister its devlink params, which leads to this WARN[1].
    Fix it by unregistering devlink params in that case as well.

    [1]
    [  295.244769 ] WARNING: CPU: 15 PID: 1 at net/core/devlink.c:9042
devlink_free+0x174/0x1fc
    [  295.488379 ] CPU: 15 PID: 1 Comm: shutdown Tainted: G S         OE 5.15.0-
1017.19.3.g0677e61-bluefield #g0677e61
    [  295.509330 ] Hardware name: https://www.mellanox.com BlueField SoC/BlueField SoC,
BIOS 4.2.0.12761 Jun  6 2023
    [  295.543096 ] pc : devlink_free+0x174/0x1fc
    [  295.551104 ] lr : mlx5_devlink_free+0x18/0x2c [mlx5_core]
    [  295.561816 ] sp : ffff80000809b850
    [  295.711155 ] Call trace:
    [  295.716030 ]  devlink_free+0x174/0x1fc
    [  295.723346 ]  mlx5_devlink_free+0x18/0x2c [mlx5_core]
    [  295.733351 ]  mlx5_sf_dev_remove+0x98/0xb0 [mlx5_core]
    [  295.743534 ]  auxiliary_bus_remove+0x2c/0x50
    [  295.751893 ]  __device_release_driver+0x19c/0x280
    [  295.761120 ]  device_release_driver+0x34/0x50
    [  295.769649 ]  bus_remove_device+0xdc/0x170
    [  295.777656 ]  device_del+0x17c/0x3a4
    [  295.784620 ]  mlx5_sf_dev_remove+0x28/0xf0 [mlx5_core]
    [  295.794800 ]  mlx5_sf_dev_table_destroy+0x98/0x110 [mlx5_core]
    [  295.806375 ]  mlx5_unload+0x34/0xd0 [mlx5_core]
    [  295.815339 ]  mlx5_unload_one+0x70/0xe4 [mlx5_core]
    [  295.824998 ]  shutdown+0xb0/0xd8 [mlx5_core]
    [  295.833439 ]  pci_device_shutdown+0x3c/0xa0
    [  295.841651 ]  device_shutdown+0x170/0x340
    [  295.849486 ]  __do_sys_reboot+0x1f4/0x2a0
    [  295.857322 ]  __arm64_sys_reboot+0x2c/0x40
    [  295.865329 ]  invoke_syscall+0x78/0x100
    [  295.872817 ]  el0_svc_common.constprop.0+0x54/0x184
    [  295.882392 ]  do_el0_svc+0x30/0xac
    [  295.889008 ]  el0_svc+0x48/0x160
    [  295.895278 ]  el0t_64_sync_handler+0xa4/0x130
    [  295.903807 ]  el0t_64_sync+0x1a4/0x1a8
    [  295.911120 ] ---[ end trace 4f1d2381d00d9dce  ]---

    Fixes: fe578cbb2f05 ("net/mlx5: Move devlink registration before mlx5_load")
    Signed-off-by: Shay Drory <shayd@nvidia.com>
    Reviewed-by: Maher Sanalla <msanalla@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3280f8a40190ffe7aeb185973daf1c17375ba19f
Author: Chris Mi <cmi@nvidia.com>
Date:   Mon Jul 17 08:32:51 2023 +0300

    net/mlx5: fs_chains: Fix ft prio if ignore_flow_level is not supported
```

```
    [ Upstream commit 61eab651f6e96791cfad6db45f1107c398699b2d ]

    The cited commit sets ft prio to fs_base_prio. But if
    ignore_flow_level it not supported, ft prio must be set based on
    tc filter prio. Otherwise, all the ft prio are the same on the same
    chain. It is invalid if ignore_flow_level is not supported.

    Fix it by setting ft prio based on tc filter prio and setting
    fs_base_prio to 0 for fdb.

    Fixes: 8e80e5648092 ("net/mlx5: fs_chains: Refactor to detach chains from tc usage")
    Signed-off-by: Chris Mi <cmi@nvidia.com>
    Reviewed-by: Paul Blakey <paulb@nvidia.com>
    Reviewed-by: Roi Dayan <roid@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bd964343338e70709fb47f57693646c8d5187ba6
Author: Jianbo Liu <jianbol@nvidia.com>
Date:   Mon May 8 03:36:10 2023 +0000

    net/mlx5e: kTLS, Fix protection domain in use syndrome when devlink reload

    [ Upstream commit 3e4cf1dd2ce413f4be3e2c9062fb470e2ad2be88 ]

    There are DEK objects cached in DEK pool after kTLS is used, and they
    are freed only in mlx5e_ktls_cleanup().

    mlx5e_destroy_mdev_resources() is called in mlx5e_suspend() to
    free mdev resources, including protection domain (PD). However, PD is
    still referenced by the cached DEK objects in this case, because
    profile->cleanup() (and therefore mlx5e_ktls_cleanup()) is called
    after mlx5e_suspend() during devlink reload. So the following FW
    syndrome is generated:

     mlx5_cmd_out_err:803:(pid 12948): DEALLOC_PD(0x801) op_mod(0x0) failed,
        status bad resource state(0x9), syndrome (0xef0c8a), err(-22)

    To avoid this syndrome, move DEK pool destruction to
    mlx5e_ktls_cleanup_tx(), which is called by profile->cleanup_tx(). And
    move pool creation to mlx5e_ktls_init_tx() for symmetry.

    Fixes: f741db1a5171 ("net/mlx5e: kTLS, Improve connection rate by using fast update
encryption key")
    Signed-off-by: Jianbo Liu <jianbol@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 02a84eb2af6bea7871cd34264fb27f141f005fd9
Author: Dragos Tatulea <dtatulea@nvidia.com>
Date:   Mon Apr 24 18:19:00 2023 +0300

    net/mlx5e: xsk: Fix crash on regular rq reactivation

    [ Upstream commit 39646d9bcd1a65d2396328026626859a1dab59d7 ]

    When the regular rq is reactivated after the XSK socket is closed
    it could be reading stale cqes which eventually corrupts the rq.
    This leads to no more traffic being received on the regular rq and a
    crash on the next close or deactivation of the rq.

    Kal Cuttler Conely reported this issue as a crash on the release
    path when the xdpsock sample program is stopped (killed) and restarted
    in sequence while traffic is running.
```

```
    This patch flushes all cqes when during the rq flush. The cqe flushing
    is done in the reset state of the rq. mlx5e_rq_to_ready code is moved
    into the flush function to allow for this.

    Fixes: 082a9edf12fe ("net/mlx5e: xsk: Flush RQ on XSK activation to save memory")
    Reported-by: Kal Cutter Conley <kal.conley@dectris.com>
    Closes: https://lore.kernel.org/xdp-newbies/CAHApi-nUAs4TeFWUDV915CZJo07XVg2Vp63-
no7UDfj6wur9nQ@mail.gmail.com
    Signed-off-by: Dragos Tatulea <dtatulea@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 58a113a35846d9a5bd759beb332e551e28451f09
Author: Dragos Tatulea <dtatulea@nvidia.com>
Date:   Tue Jul 18 11:13:33 2023 +0300

    net/mlx5e: xsk: Fix invalid buffer access for legacy rq

    [ Upstream commit e0f52298fee449fec37e3e3c32df60008b509b16 ]

    The below crash can be encountered when using xdpsock in rx mode for
    legacy rq: the buffer gets released in the XDP_REDIRECT path, and then
    once again in the driver. This fix sets the flag to avoid releasing on
    the driver side.

    XSK handling of buffers for legacy rq was relying on the caller to set
    the skip release flag. But the referenced fix started using fragment
    counts for pages instead of the skip flag.

    Crash log:
     general protection fault, probably for non-canonical address 0xffff8881217e3a: 0000
[#1] SMP
     CPU: 0 PID: 14 Comm: ksoftirqd/0 Not tainted 6.5.0-rc1+ #31
     Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-
prebuilt.qemu.org 04/01/2014
     RIP: 0010:bpf_prog_03b13f331978c78c+0xf/0x28
     Code:  ...
     RSP: 0018:ffff88810082fc98 EFLAGS: 00010246
     RAX: 0000000000000000 RBX: ffff888138404901 RCX: c0ffffc900027cbc
     RDX: ffffffffa000b514 RSI: 00ffff8881217e32 RDI: ffff888138404901
     RBP: ffff88810082fc98 R08: 0000000000091100 R09: 0000000000000006
     R10: 0000000000000800 R11: 0000000000000800 R12: ffffc9000027a000
     R13: ffff8881217e2dc0 R14: ffff8881217e2910 R15: ffff8881217e2f00
     FS:  0000000000000000(0000) GS:ffff88852c800000(0000) knlGS:0000000000000000
     CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     CR2: 0000564cb2e2cde0 CR3: 000000010e603004 CR4: 0000000000370eb0
     DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     Call Trace:
      <TASK>
      ? die_addr+0x32/0x80
      ? exc_general_protection+0x192/0x390
      ? asm_exc_general_protection+0x22/0x30
      ? 0xffffffffa000b514
      ? bpf_prog_03b13f331978c78c+0xf/0x28
      mlx5e_xdp_handle+0x48/0x670 [mlx5_core]
      ? dev_gro_receive+0x3b5/0x6e0
      mlx5e_xsk_skb_from_cqe_linear+0x6e/0x90 [mlx5_core]
      mlx5e_handle_rx_cqe+0x55/0x100 [mlx5_core]
      mlx5e_poll_rx_cq+0x87/0x6e0 [mlx5_core]
      mlx5e_napi_poll+0x45e/0x6b0 [mlx5_core]
      __napi_poll+0x25/0x1a0
      net_rx_action+0x28a/0x300
```

```
       __do_softirq+0xcd/0x279
       ? sort_range+0x20/0x20
       run_ksoftirqd+0x1a/0x20
       smpboot_thread_fn+0xa2/0x130
       kthread+0xc9/0xf0
       ? kthread_complete_and_exit+0x20/0x20
       ret_from_fork+0x1f/0x30
       </TASK>
      Modules linked in: mlx5_ib mlx5_core rpcrdma rdma_ucm ib_iser libiscsi
  scsi_transport_iscsi ib_umad rdma_cm ib_ipoib iw_cm ib_cm ib_uverbs ib_core xt_conntrack
  xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter
  overlay zram zsmalloc fuse [last unloaded: mlx5_core]
      ---[ end trace 0000000000000000 ]---

     Fixes: 7abd955a58fb ("net/mlx5e: RX, Fix page_pool page fragment tracking for XDP")
     Signed-off-by: Dragos Tatulea <dtatulea@nvidia.com>
     Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
     Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 36697c592cd0809e626df01b3644c23ac522a4d0
  Author: Jianbo Liu <jianbol@nvidia.com>
  Date:   Mon Jul 3 08:28:16 2023 +0000

     net/mlx5e: Move representor neigh cleanup to profile cleanup_tx

     [ Upstream commit d03b6e6f31820b84f7449cca022047f36c42bc3f ]

     For IP tunnel encapsulation in ECMP (Equal-Cost Multipath) mode, as
     the flow is duplicated to the peer eswitch, the related neighbour
     information on the peer uplink representor is created as well.

     In the cited commit, eswitch devcom unpair is moved to uplink unload
     API, specifically the profile->cleanup_tx. If there is a encap rule
     offloaded in ECMP mode, when one eswitch does unpair (because of
     unloading the driver, for instance), and the peer rule from the peer
     eswitch is going to be deleted, the use-after-free error is triggered
     while accessing neigh info, as it is already cleaned up in uplink's
     profile->disable, which is before its profile->cleanup_tx.

     To fix this issue, move the neigh cleanup to profile's cleanup_tx
     callback, and after mlx5e_cleanup_uplink_rep_tx is called. The neigh
     init is moved to init_tx for symmeter.

     [ 2453.376299] BUG: KASAN: slab-use-after-free in
  mlx5e_rep_neigh_entry_release+0x109/0x3a0 [mlx5_core]
     [ 2453.379125] Read of size 4 at addr ffff888127af9008 by task modprobe/2496

     [ 2453.381542] CPU: 7 PID: 2496 Comm: modprobe Tainted: G    B              6.4.0-rc7+
  #15
     [ 2453.383386] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-
  gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014
     [ 2453.384335] Call Trace:
     [ 2453.384625]  <TASK>
     [ 2453.384891]  dump_stack_lvl+0x33/0x50
     [ 2453.385285]  print_report+0xc2/0x610
     [ 2453.385667]  ? __virt_addr_valid+0xb1/0x130
     [ 2453.386091]  ? mlx5e_rep_neigh_entry_release+0x109/0x3a0 [mlx5_core]
     [ 2453.386757]  kasan_report+0xae/0xe0
     [ 2453.387123]  ? mlx5e_rep_neigh_entry_release+0x109/0x3a0 [mlx5_core]
     [ 2453.387798]  mlx5e_rep_neigh_entry_release+0x109/0x3a0 [mlx5_core]
     [ 2453.388465]  mlx5e_rep_encap_entry_detach+0xa6/0xe0 [mlx5_core]
     [ 2453.389111]  mlx5e_encap_dealloc+0xa7/0x100 [mlx5_core]
     [ 2453.389706]  mlx5e_tc_tun_encap_dests_unset+0x61/0xb0 [mlx5_core]
     [ 2453.390361]  mlx5_free_flow_attr_actions+0x11e/0x340 [mlx5_core]
```

```
[ 2453.391015]  ? complete_all+0x43/0xd0
[ 2453.391398]  ? free_flow_post_acts+0x38/0x120 [mlx5_core]
[ 2453.392004]  mlx5e_tc_del_fdb_flow+0x4ae/0x690 [mlx5_core]
[ 2453.392618]  mlx5e_tc_del_fdb_peers_flow+0x308/0x370 [mlx5_core]
[ 2453.393276]  mlx5e_tc_clean_fdb_peer_flows+0xf5/0x140 [mlx5_core]
[ 2453.393925]  mlx5_esw_offloads_unpair+0x86/0x540 [mlx5_core]
[ 2453.394546]  ? mlx5_esw_offloads_set_ns_peer.isra.0+0x180/0x180 [mlx5_core]
[ 2453.395268]  ? down_write+0xaa/0x100
[ 2453.395652]  mlx5_esw_offloads_devcom_event+0x203/0x530 [mlx5_core]
[ 2453.396317]  mlx5_devcom_send_event+0xbb/0x190 [mlx5_core]
[ 2453.396917]  mlx5_esw_offloads_devcom_cleanup+0xb0/0xd0 [mlx5_core]
[ 2453.397582]  mlx5e_tc_esw_cleanup+0x42/0x120 [mlx5_core]
[ 2453.398182]  mlx5e_rep_tc_cleanup+0x15/0x30 [mlx5_core]
[ 2453.398768]  mlx5e_cleanup_rep_tx+0x6c/0x80 [mlx5_core]
[ 2453.399367]  mlx5e_detach_netdev+0xee/0x120 [mlx5_core]
[ 2453.399957]  mlx5e_netdev_change_profile+0x84/0x170 [mlx5_core]
[ 2453.400598]  mlx5e_vport_rep_unload+0xe0/0xf0 [mlx5_core]
[ 2453.403781]  mlx5_eswitch_unregister_vport_reps+0x15e/0x190 [mlx5_core]
[ 2453.404479]  ? mlx5_eswitch_register_vport_reps+0x200/0x200 [mlx5_core]
[ 2453.405170]  ? up_write+0x39/0x60
[ 2453.405529]  ? kernfs_remove_by_name_ns+0xb7/0xe0
[ 2453.405985]  auxiliary_bus_remove+0x2e/0x40
[ 2453.406405]  device_release_driver_internal+0x243/0x2d0
[ 2453.406900]  ? kobject_put+0x42/0x2d0
[ 2453.407284]  bus_remove_device+0x128/0x1d0
[ 2453.407687]  device_del+0x240/0x550
[ 2453.408053]  ? waiting_for_supplier_show+0xe0/0xe0
[ 2453.408511]  ? kobject_put+0xfa/0x2d0
[ 2453.408889]  ? __kmem_cache_free+0x14d/0x280
[ 2453.409310]  mlx5_rescan_drivers_locked.part.0+0xcd/0x2b0 [mlx5_core]
[ 2453.409973]  mlx5_unregister_device+0x40/0x50 [mlx5_core]
[ 2453.410561]  mlx5_uninit_one+0x3d/0x110 [mlx5_core]
[ 2453.411111]  remove_one+0x89/0x130 [mlx5_core]
[ 2453.411628]  pci_device_remove+0x59/0xf0
[ 2453.412026]  device_release_driver_internal+0x243/0x2d0
[ 2453.412511]  ? parse_option_str+0x14/0x90
[ 2453.412915]  driver_detach+0x7b/0xf0
[ 2453.413289]  bus_remove_driver+0xb5/0x160
[ 2453.413685]  pci_unregister_driver+0x3f/0xf0
[ 2453.414104]  mlx5_cleanup+0xc/0x20 [mlx5_core]
```

Fixes: 2be5bd42a5bb ("net/mlx5: Handle pairing of E-switch via uplink un/load APIs")
Signed-off-by: Jianbo Liu <jianbol@nvidia.com>
Reviewed-by: Vlad Buslov <vladbu@nvidia.com>
Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 90c226e467858924c02be8dcef91cc385741efaf
Author: Amir Tzin <amirtz@nvidia.com>
Date: Tue May 30 20:11:14 2023 +0300

net/mlx5e: Fix crash moving to switchdev mode when ntuple offload is set

[Upstream commit 3ec43c1b082a8804472430e1253544d75f4b540e]

Moving to switchdev mode with ntuple offload on causes the kernel to
crash since fs->arfs is freed during nic profile cleanup flow.

Ntuple offload is not supported in switchdev mode and it is already
unset by mlx5 fix feature ndo in switchdev mode. Verify fs->arfs is
valid before disabling it.

trace:
[] RIP: 0010:_raw_spin_lock_bh+0x17/0x30
[] arfs_del_rules+0x44/0x1a0 [mlx5_core]

```
      [] mlx5e_arfs_disable+0xe/0x20 [mlx5_core]
      [] mlx5e_handle_feature+0x3d/0xb0 [mlx5_core]
      [] ? __rtnl_unlock+0x25/0x50
      [] mlx5e_set_features+0xfe/0x160 [mlx5_core]
      [] __netdev_update_features+0x278/0xa50
      [] ? netdev_run_todo+0x5e/0x2a0
      [] netdev_update_features+0x22/0x70
      [] ? _cond_resched+0x15/0x30
      [] mlx5e_attach_netdev+0x12a/0x1e0 [mlx5_core]
      [] mlx5e_netdev_attach_profile+0xa1/0xc0 [mlx5_core]
      [] mlx5e_netdev_change_profile+0x77/0xe0 [mlx5_core]
      [] mlx5e_vport_rep_load+0x1ed/0x290 [mlx5_core]
      [] mlx5_esw_offloads_rep_load+0x88/0xd0 [mlx5_core]
      [] esw_offloads_load_rep.part.38+0x31/0x50 [mlx5_core]
      [] esw_offloads_enable+0x6c5/0x710 [mlx5_core]
      [] mlx5_eswitch_enable_locked+0x1bb/0x290 [mlx5_core]
      [] mlx5_devlink_eswitch_mode_set+0x14f/0x320 [mlx5_core]
      [] devlink_nl_cmd_eswitch_set_doit+0x94/0x120
      [] genl_family_rcv_msg_doit.isra.17+0x113/0x150
      [] genl_family_rcv_msg+0xb7/0x170
      [] ? devlink_nl_cmd_port_split_doit+0x100/0x100
      [] genl_rcv_msg+0x47/0xa0
      [] ? genl_family_rcv_msg+0x170/0x170
      [] netlink_rcv_skb+0x4c/0x130
      [] genl_rcv+0x24/0x40
      [] netlink_unicast+0x19a/0x230
      [] netlink_sendmsg+0x204/0x3d0
      [] sock_sendmsg+0x50/0x60

      Fixes: 90b22b9bcd24 ("net/mlx5e: Disable Rx ntuple offload for uplink representor")
      Signed-off-by: Amir Tzin <amirtz@nvidia.com>
      Reviewed-by: Aya Levin <ayal@nvidia.com>
      Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 2e76da7bda602def8449d3e7fbe19b77fd99f5ee
  Author: Chris Mi <cmi@nvidia.com>
  Date:   Thu Jun 29 11:32:03 2023 +0300

      net/mlx5e: Don't hold encap tbl lock if there is no encap action

      [ Upstream commit 93a331939d1d1c6c3422bc09ec43cac658594b34 ]

      The cited commit holds encap tbl lock unconditionally when setting
      up dests. But it may cause the following deadlock:

       PID: 1063722  TASK: ffffa062ca5d0000  CPU: 13   COMMAND: "handler8"
        #0 [ffffb14de05b7368] __schedule at ffffffffa1d5aa91
        #1 [ffffb14de05b7410] schedule at ffffffffa1d5afdb
        #2 [ffffb14de05b7430] schedule_preempt_disabled at ffffffffa1d5b528
        #3 [ffffb14de05b7440] __mutex_lock at ffffffffa1d5d6cb
        #4 [ffffb14de05b74e8] mutex_lock_nested at ffffffffa1d5ddeb
        #5 [ffffb14de05b74f8] mlx5e_tc_tun_encap_dests_set at ffffffffc12f2096 [mlx5_core]
        #6 [ffffb14de05b7568] post_process_attr at ffffffffc12d9fc5 [mlx5_core]
        #7 [ffffb14de05b75a0] mlx5e_tc_add_fdb_flow at ffffffffc12de877 [mlx5_core]
        #8 [ffffb14de05b75f0] __mlx5e_add_fdb_flow at ffffffffc12e0eef [mlx5_core]
        #9 [ffffb14de05b7660] mlx5e_tc_add_flow at ffffffffc12e12f7 [mlx5_core]
       #10 [ffffb14de05b76b8] mlx5e_configure_flower at ffffffffc12e1686 [mlx5_core]
       #11 [ffffb14de05b7720] mlx5e_rep_indr_offload at ffffffffc12e3817 [mlx5_core]
       #12 [ffffb14de05b7730] mlx5e_rep_indr_setup_tc_cb at ffffffffc12e388a [mlx5_core]
       #13 [ffffb14de05b7740] tc_setup_cb_add at ffffffffa1ab2ba8
       #14 [ffffb14de05b77a0] fl_hw_replace_filter at ffffffffc0bdec2f [cls_flower]
       #15 [ffffb14de05b7868] fl_change at ffffffffc0be6caa [cls_flower]
       #16 [ffffb14de05b7908] tc_new_tfilter at ffffffffa1ab71f0
```

```
      [1031218.028143]  wait_for_completion+0x24/0x30
      [1031218.028589]  mlx5e_update_route_decap_flows+0x9a/0x1e0 [mlx5_core]
      [1031218.029256]  mlx5e_tc_fib_event_work+0x1ad/0x300 [mlx5_core]
      [1031218.029885]  process_one_work+0x24e/0x510

      Actually no need to hold encap tbl lock if there is no encap action.
      Fix it by checking if encap action exists or not before holding
      encap tbl lock.

      Fixes: 37c3b9fa7ccf ("net/mlx5e: Prevent encap offload when neigh update is running")
      Signed-off-by: Chris Mi <cmi@nvidia.com>
      Reviewed-by: Vlad Buslov <vladbu@nvidia.com>
      Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 0302414ca593b3a0a5e1527caf8761a21388a4d1
  Author: Shay Drory <shayd@nvidia.com>
  Date:   Mon Jul 3 17:34:44 2023 +0300

      net/mlx5: Honor user input for migratable port fn attr

      [ Upstream commit 0507f2c8be0d345fe7014147c027cea6dc1c00a4 ]

      Currently, whenever a user is setting migratable port fn attr, the
      driver is always turn migratable capability on.
      Fix it by honor the user input

      Fixes: e5b9642a33be ("net/mlx5: E-Switch, Implement devlink port function cmds to
  control migratable")
      Signed-off-by: Shay Drory <shayd@nvidia.com>
      Reviewed-by: Roi Dayan <roid@nvidia.com>
      Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit cc94d516c3a9d6899e5dac121ab1a87bbf9f0217
  Author: Yuanjun Gong <ruc_gongyuanjun@163.com>
  Date:   Tue Jul 25 14:56:55 2023 +0800

      net/mlx5e: fix return value check in mlx5e_ipsec_remove_trailer()

      [ Upstream commit e5bcb7564d3bd0c88613c76963c5349be9c511c5 ]

      mlx5e_ipsec_remove_trailer() should return an error code if function
      pskb_trim() returns an unexpected value.

      Fixes: 2ac9cfe78223 ("net/mlx5e: IPSec, Add Innova IPSec offload TX data path")
      Signed-off-by: Yuanjun Gong <ruc_gongyuanjun@163.com>
      Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
      Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit c265d8c2e25546a6b7ee16d36f2bb79b6160c2c3
  Author: Zhengchao Shao <shaozhengchao@huawei.com>
  Date:   Sat Jul 8 15:13:07 2023 +0800

      net/mlx5: fix potential memory leak in mlx5e_init_rep_rx

      [ Upstream commit c6cf0b6097bf1bf1b2a89b521e9ecd26b581a93a ]

      The memory pointed to by the priv->rx_res pointer is not freed in the error
      path of mlx5e_init_rep_rx, which can lead to a memory leak. Fix by freeing
      the memory in the error path, thereby making the error path identical to
      mlx5e_cleanup_rep_rx().

      Fixes: af8bbf730068 ("net/mlx5e: Convert mlx5e_flow_steering member of mlx5e_priv to
```

```
pointer")
    Signed-off-by: Zhengchao Shao <shaozhengchao@huawei.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 622d71d99124e69f7bf2e2b7a89f5f444a24d235
Author: Zhengchao Shao <shaozhengchao@huawei.com>
Date:   Wed Jul 5 20:15:27 2023 +0800

    net/mlx5: DR, fix memory leak in mlx5dr_cmd_create_reformat_ctx

    [ Upstream commit 5dd77585dd9d0e03dd1bceb95f0269a7eaf6b936 ]

    when mlx5_cmd_exec failed in mlx5dr_cmd_create_reformat_ctx, the memory
    pointed by 'in' is not released, which will cause memory leak. Move memory
    release after mlx5_cmd_exec.

    Fixes: 1d9186476e12 ("net/mlx5: DR, Add direct rule command utilities")
    Signed-off-by: Zhengchao Shao <shaozhengchao@huawei.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 957702c389a83ccc93f2de3d6d066c823dbf641d
Author: Zhengchao Shao <shaozhengchao@huawei.com>
Date:   Tue Jul 4 15:06:40 2023 +0800

    net/mlx5e: fix double free in macsec_fs_tx_create_crypto_table_groups

    [ Upstream commit aeb660171b0663847fa04806a96302ac6112ad26 ]

    In function macsec_fs_tx_create_crypto_table_groups(), when the ft->g
    memory is successfully allocated but the 'in' memory fails to be
    allocated, the memory pointed to by ft->g is released once. And in function
    macsec_fs_tx_create(), macsec_fs_tx_destroy() is called to release the
    memory pointed to by ft->g again. This will cause double free problem.

    Fixes: e467b283ffd5 ("net/mlx5e: Add MACsec TX steering rules")
    Signed-off-by: Zhengchao Shao <shaozhengchao@huawei.com>
    Reviewed-by: Simon Horman <simon.horman@corigine.com>
    Reviewed-by: Leon Romanovsky <leonro@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1d23e51cc6a67b310ac7f74dd99423d0985ddb27
Author: Ilan Peer <ilan.peer@intel.com>
Date:   Sun Jul 23 23:10:43 2023 +0300

    wifi: cfg80211: Fix return value in scan logic

    [ Upstream commit fd7f08d92fcd7cc3eca0dd6c853f722a4c6176df ]

    The reporter noticed a warning when running iwlwifi:

    WARNING: CPU: 8 PID: 659 at mm/page_alloc.c:4453 __alloc_pages+0x329/0x340

    As cfg80211_parse_colocated_ap() is not expected to return a negative
    value return 0 and not a negative value if cfg80211_calc_short_ssid()
    fails.

    Fixes: c8cb5b854b40f ("nl80211/cfg80211: support 6 GHz scanning")
    Closes: https://bugzilla.kernel.org/show_bug.cgi?id=217675
    Signed-off-by: Ilan Peer <ilan.peer@intel.com>
```

```
    Signed-off-by: Kalle Valo <kvalo@kernel.org>
    Link: https://lore.kernel.org/r/20230723201043.3007430-1-ilan.peer@intel.com
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0ab6fac370329339863448eccddf858fccd6dce3
Author: Haixin Yu <yuhaixin.yhx@linux.alibaba.com>
Date:   Mon Jul 24 13:06:54 2023 +0800

    perf pmu arm64: Fix reading the PMU cpu slots in sysfs

    [ Upstream commit 9754353d0ab123d71bf572a483ecc8b330ef36a3 ]

    Commit f8ad6018ce3c065a ("perf pmu: Remove duplication around
    EVENT_SOURCE_DEVICE_PATH") uses sysfs__read_ull() to read a full sysfs
    path, which will never succeeds as it already comes with the sysfs mount
    point in it, which sysfs__read_ull() will add again.

    Fix it by reading the file using filename__read_ull(), that will not add
    the sysfs mount point.

    Fixes: f8ad6018ce3c065a ("perf pmu: Remove duplication around EVENT_SOURCE_DEVICE_PATH")
    Signed-off-by: Haixin Yu <yuhaixin.yhx@linux.alibaba.com>
    Tested-by: Jing Zhang <renyu.zj@linux.alibaba.com>
    Cc: Adrian Hunter <adrian.hunter@intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Ian Rogers <irogers@google.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: James Clark <james.clark@arm.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: John Garry <john.g.garry@oracle.com>
    Cc: Leo Yan <leo.yan@linaro.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Mike Leach <mike.leach@linaro.org>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Will Deacon <will@kernel.org>
    Cc: linux-arm-kernel@lists.infradead.org
    Link: https://lore.kernel.org/r/ZL4G7rWXkfv-Ectq@B-Q60VQ05P-2326.local
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b845249a4e5e38ec4502fe24035d55c2fec5def1
Author: Gao Xiang <xiang@kernel.org>
Date:   Wed Jul 19 14:54:59 2023 +0800

    erofs: fix wrong primary bvec selection on deduplicated extents

    [ Upstream commit 94c43de73521d8ed7ebcfc6191d9dace1cbf7caa ]

    When handling deduplicated compressed data, there can be multiple
    decompressed extents pointing to the same compressed data in one shot.

    In such cases, the bvecs which belong to the longest extent will be
    selected as the primary bvecs for real decompressors to decode and the
    other duplicated bvecs will be directly copied from the primary bvecs.

    Previously, only relative offsets of the longest extent were checked to
    decompress the primary bvecs.  On rare occasions, it can be incorrect
    if there are several extents with the same start relative offset.
    As a result, some short bvecs could be selected for decompression and
    then cause data corruption.

    For example, as Shijie Sun reported off-list, considering the following
    extents of a file:
     117:   903345..  915250 |   11905 :     385024..    389120 |    4096
```

```
     ...
      119:   919729..  930323 |   10594 :     385024..    389120 |    4096
     ...
      124:   968881..  980786 |   11905 :     385024..    389120 |    4096

     The start relative offset is the same: 2225, but extent 119 (919729..
     930323) is shorter than the others.

     Let's restrict the bvec length in addition to the start offset if bvecs
     are not full.

     Reported-by: Shijie Sun <sunshijie@xiaomi.com>
     Fixes: 5c2a64252c5d ("erofs: introduce partial-referenced pclusters")
     Tested-by Shijie Sun <sunshijie@xiaomi.com>
     Reviewed-by: Yue Hu <huyue2@coolpad.com>
     Reviewed-by: Chao Yu <chao@kernel.org>
     Signed-off-by: Gao Xiang <hsiangkao@linux.alibaba.com>
     Link: https://lore.kernel.org/r/20230719065459.60083-1-hsiangkao@linux.alibaba.com
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 53980121e1a601cd1ec135b25fe6c6748170e379
Author: Heiko Carstens <hca@linux.ibm.com>
Date:   Thu Jul 27 20:29:39 2023 +0200

     KVM: s390: fix sthyi error handling

     [ Upstream commit 0c02cc576eac161601927b41634f80bfd55bfa9e ]

     Commit 9fb6c9b3fea1 ("s390/sthyi: add cache to store hypervisor info")
     added cache handling for store hypervisor info. This also changed the
     possible return code for sthyi_fill().

     Instead of only returning a condition code like the sthyi instruction would
     do, it can now also return a negative error value (-ENOMEM). handle_styhi()
     was not changed accordingly. In case of an error, the negative error value
     would incorrectly injected into the guest PSW.

     Add proper error handling to prevent this, and update the comment which
     describes the possible return values of sthyi_fill().

     Fixes: 9fb6c9b3fea1 ("s390/sthyi: add cache to store hypervisor info")
     Reviewed-by: Christian Borntraeger <borntraeger@linux.ibm.com>
     Link: https://lore.kernel.org/r/20230727182939.2050744-1-hca@linux.ibm.com
     Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 601e467e29a960f7ab7ec4075afc6a68c3532a65
Author: Sven Schnelle <svens@linux.ibm.com>
Date:   Wed Jul 26 11:10:19 2023 +0200

     s390/vmem: split pages when debug pagealloc is enabled

     [ Upstream commit edc1e4b6e26536868ef819a735e04a5b32c10589 ]

     Since commit bb1520d581a3 ("s390/mm: start kernel with DAT enabled")
     the kernel crashes early during boot when debug pagealloc is enabled:

     mem auto-init: stack:off, heap alloc:off, heap free:off
     addressing exception: 0005 ilc:2 [#1] SMP DEBUG_PAGEALLOC
     Modules linked in:
     CPU: 0 PID: 0 Comm: swapper Not tainted 6.5.0-rc3-09759-gc5666c912155 #630
     [..]
     Krnl Code: 00000000001325f6: ec5600248064 cgrj %r5,%r6,8,000000000013263e
                00000000001325fc: eb880002000c srlg %r8,%r8,2
               #0000000000132602: b2210051     ipte %r5,%r1,%r0,0
```

```
               >0000000000132606: b90400d1     lgr %r13,%r1
                0000000000132606: 41605008     la %r6,8(%r5)
                000000000013260e: a7db1000     aghi %r13,4096
                0000000000132612: b221006d     ipte %r6,%r13,%r0,0
                0000000000132616: e3d0d0000171 lay %r13,4096(%r13)

     Call Trace:
      __kernel_map_pages+0x14e/0x320
      __free_pages_ok+0x23a/0x5a8)
      free_low_memory_core_early+0x214/0x2c8
      memblock_free_all+0x28/0x58
      mem_init+0xb6/0x228
      mm_core_init+0xb6/0x3b0
      start_kernel+0x1d2/0x5a8
      startup_continue+0x36/0x40
     Kernel panic - not syncing: Fatal exception: panic_on_oops

     This is caused by using large mappings on machines with EDAT1/EDAT2. Add
     the code to split the mappings into 4k pages if debug pagealloc is enabled
     by CONFIG_DEBUG_PAGEALLOC_ENABLE_DEFAULT or the debug_pagealloc kernel
     command line option.

     Fixes: bb1520d581a3 ("s390/mm: start kernel with DAT enabled")
     Signed-off-by: Sven Schnelle <svens@linux.ibm.com>
     Reviewed-by: Heiko Carstens <hca@linux.ibm.com>
     Signed-off-by: Heiko Carstens <hca@linux.ibm.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cfa54fb51357450a3037b0a9833d07976c4e6195
Author: ndesaulniers@google.com <ndesaulniers@google.com>
Date:   Tue Aug 1 15:22:17 2023 -0700

     word-at-a-time: use the same return type for has_zero regardless of endianness

     [ Upstream commit 79e8328e5acbe691bbde029a52c89d70dcbc22f3 ]

     Compiling big-endian targets with Clang produces the diagnostic:

       fs/namei.c:2173:13: warning: use of bitwise '|' with boolean operands [-Wbitwise-
instead-of-logical]
             } while (!(has_zero(a, &adata, &constants) | has_zero(b, &bdata, &constants)));
                        ~^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                                    ||
       fs/namei.c:2173:13: note: cast one or both operands to int to silence this warning

     It appears that when has_zero was introduced, two definitions were
     produced with different signatures (in particular different return
     types).

     Looking at the usage in hash_name() in fs/namei.c, I suspect that
     has_zero() is meant to be invoked twice per while loop iteration; using
     logical-or would not update `bdata` when `a` did not have zeros.  So I
     think it's preferred to always return an unsigned long rather than a
     bool than update the while loop in hash_name() to use a logical-or
     rather than bitwise-or.

     [ Also changed powerpc version to do the same  - Linus ]

     Link: https://github.com/ClangBuiltLinux/linux/issues/1832
     Link: https://lore.kernel.org/lkml/20230801-bitwise-v1-1-799bec468dc4@google.com/
     Fixes: 36126f8f2ed8 ("word-at-a-time: make the interfaces truly generic")
     Debugged-by: Nathan Chancellor <nathan@kernel.org>
     Signed-off-by: Nick Desaulniers <ndesaulniers@google.com>
     Acked-by: Heiko Carstens <hca@linux.ibm.com>
     Cc: Arnd Bergmann <arnd@arndb.de>
```

```
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0326a5bd6f58eb68722e2c434849d71d82293a79
Author: Durai Manickam KR <durai.manickamkr@microchip.com>
Date:   Wed Jul 12 15:30:42 2023 +0530

    ARM: dts: at91: sam9x60: fix the SOC detection

    [ Upstream commit f6ad3c13f1b8c4e785cb7bd423887197142f47b0 ]

    Remove the dbgu compatible strings in the UART submodule of the
    flexcom for the proper SOC detection.

    Fixes: 99c808335877 (ARM: dts: at91: sam9x60: Add missing flexcom definitions)
    Signed-off-by: Durai Manickam KR <durai.manickamkr@microchip.com>
    Link: https://lore.kernel.org/r/20230712100042.317856-1-durai.manickamkr@microchip.com
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 08491e1cadd64663dce70f02eb9a8a37c21d0141
Author: Claudiu Beznea <claudiu.beznea@microchip.com>
Date:   Fri Jun 16 13:16:43 2023 +0300

    ARM: dts: at91: use generic name for shutdown controller

    [ Upstream commit 327ca228e58be498446244eb7cf39b892adda5d7 ]

    Use poweroff generic name for shdwc node to cope with device tree
    specifications.

    Signed-off-by: Claudiu Beznea <claudiu.beznea@microchip.com>
    Acked-by: Nicolas Ferre <nicolas.ferre@microchip.com>
    Link: https://lore.kernel.org/r/20230616101646.879480-2-claudiu.beznea@microchip.com
    Stable-dep-of: f6ad3c13f1b8 ("ARM: dts: at91: sam9x60: fix the SOC detection")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit ccec3e7f2b793e14f484a323adc3e33d714ca52f
Author: Claudiu Beznea <claudiu.beznea@microchip.com>
Date:   Wed May 17 12:41:18 2023 +0300

    ARM: dts: at91: use clock-controller name for sckc nodes

    [ Upstream commit 3ecb546333089195b6a1508cb58627b0797a26ca ]

    Use clock-controller generic name for slow clock controller nodes.

    Signed-off-by: Claudiu Beznea <claudiu.beznea@microchip.com>
    Link: https://lore.kernel.org/r/20230517094119.2894220-5-claudiu.beznea@microchip.com
    Stable-dep-of: f6ad3c13f1b8 ("ARM: dts: at91: sam9x60: fix the SOC detection")
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 51a34cb21fd61eaaadfce4e09816675b672ef2b7
Author: Claudiu Beznea <claudiu.beznea@microchip.com>
Date:   Wed May 17 12:41:15 2023 +0300

    ARM: dts: at91: use clock-controller name for PMC nodes

    [ Upstream commit d08f92bdfb2dc4a2a14237cfd8a22c568781797c ]

    Use clock-controller generic name for PMC nodes.

    Signed-off-by: Claudiu Beznea <claudiu.beznea@microchip.com>
    Link: https://lore.kernel.org/r/20230517094119.2894220-2-claudiu.beznea@microchip.com
    Stable-dep-of: f6ad3c13f1b8 ("ARM: dts: at91: sam9x60: fix the SOC detection")
```

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

commit a5bdeb37a0536704d03c830720148f0bf8caf8f4
Author: Cristian Marussi <cristian.marussi@arm.com>
Date:   Wed Jul 19 18:35:33 2023 +0100

    firmware: arm_scmi: Fix chan_free cleanup on SMC

    [ Upstream commit d1ff11d7ad8704f8d615f6446041c221b2d2ec4d ]

    SCMI transport based on SMC can optionally use an additional IRQ to
    signal message completion. The associated interrupt handler is currently
    allocated using devres but on shutdown the core SCMI stack will call
    .chan_free() well before any managed cleanup is invoked by devres.
    As a consequence, the arrival of a late reply to an in-flight pending
    transaction could still trigger the interrupt handler well after the
    SCMI core has cleaned up the channels, with unpleasant results.

    Inhibit further message processing on the IRQ path by explicitly freeing
    the IRQ inside .chan_free() callback itself.

    Fixes: dd820ee21d5e ("firmware: arm_scmi: Augment SMC/HVC to allow optional interrupt")
    Reported-by: Bjorn Andersson <andersson@kernel.org>
    Signed-off-by: Cristian Marussi <cristian.marussi@arm.com>
    Link: https://lore.kernel.org/r/20230719173533.2739319-1-cristian.marussi@arm.com
    Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5dcc40b28c3f4e202a9d8ce41e142949d2973a20
Author: Lucas Stach <l.stach@pengutronix.de>
Date:   Mon Jul 17 16:54:09 2023 +0200

    soc: imx: imx8mp-blk-ctrl: register HSIO PLL clock as bus_power_dev child

    [ Upstream commit 53cab4d871690c49fac87c657cbf459e39c5b93b ]

    The blk-ctrl device is deliberately placed outside of the GPC power
    domain as it needs to control the power sequencing of the blk-ctrl
    domains together with the GPC domains.

    Clock runtime PM works by operating on the clock parent device, which
    doesn't translate into the neccessary GPC power domain action if the
    clk parent is not part of the GPC power domain. Use the bus_power_device
    as the parent for the clock to trigger the proper GPC domain actions on
    clock runtime power management.

    Fixes: 2cbee26e5d59 ("soc: imx: imx8mp-blk-ctrl: expose high performance PLL clock")
    Reported-by: Yannic Moog <Y.Moog@phytec.de>
    Signed-off-by: Lucas Stach <l.stach@pengutronix.de>
    Tested-by: Yannic Moog <y.moog@phytec.de>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2b41891b9cc1ebd82a148b2e974c4fb3c5760c1d
Author: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
Date:   Sun Jul 9 23:30:19 2023 +0300

    ARM: dts: nxp/imx: limit sk-imx53 supported frequencies

    [ Upstream commit c486762fb17c99fd642beea3e1e4744d093c262a ]

    The SK-IMX53 board, bearing i.MX536A CPU, is not stable when running at
    1.2 GHz (default iMX53 maximum). The SoC is only rated up to 800 MHz.
    Disable 1.2 GHz and 1 GHz frequencies.
```

```
    Fixes: 0b8576d8440a ("ARM: dts: imx: Add support for SK-iMX53 board")
    Signed-off-by: Dmitry Baryshkov <dmitry.baryshkov@linaro.org>
    Reviewed-by: Fabio Estevam <festevam@gmail.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8626358813be9f5ebaac0eb916830554144be315
Author: Yury Norov <yury.norov@gmail.com>
Date:   Mon Jul 17 12:17:03 2023 -0700

    lib/bitmap: workaround const_eval test build failure

    [ Upstream commit 2356d198d2b4ddec24efea98271cb3be230bc787 ]

    When building with Clang, and when KASAN and GCOV_PROFILE_ALL are both
    enabled, the test fails to build [1]:

    >> lib/test_bitmap.c:920:2: error: call to '__compiletime_assert_239' declared with
'error' attribute: BUILD_BUG_ON failed: !__builtin_constant_p(res)
               BUILD_BUG_ON(!__builtin_constant_p(res));
               ^
       include/linux/build_bug.h:50:2: note: expanded from macro 'BUILD_BUG_ON'
               BUILD_BUG_ON_MSG(condition, "BUILD_BUG_ON failed: " #condition)
               ^
       include/linux/build_bug.h:39:37: note: expanded from macro 'BUILD_BUG_ON_MSG'
       #define BUILD_BUG_ON_MSG(cond, msg) compiletime_assert(!(cond), msg)
                                           ^
       include/linux/compiler_types.h:352:2: note: expanded from macro 'compiletime_assert'
               _compiletime_assert(condition, msg, __compiletime_assert_, __COUNTER__)
               ^
       include/linux/compiler_types.h:340:2: note: expanded from macro '_compiletime_assert'
               __compiletime_assert(condition, msg, prefix, suffix)
               ^
       include/linux/compiler_types.h:333:4: note: expanded from macro
'__compiletime_assert'
                               prefix ## suffix();                             \
                               ^
       <scratch space>:185:1: note: expanded from here
       __compiletime_assert_239

    Originally it was attributed to s390, which now looks seemingly wrong. The
    issue is not related to bitmap code itself, but it breaks build for a given
    configuration.

    Disabling the const_eval test under that config may potentially hide other
    bugs. Instead, workaround it by disabling GCOV for the test_bitmap unless
    the compiler will get fixed.

    [1] https://github.com/ClangBuiltLinux/linux/issues/1874

    Reported-by: kernel test robot <lkp@intel.com>
    Closes: https://lore.kernel.org/oe-kbuild-all/202307171254.yFcH97ej-lkp@intel.com/
    Fixes: dc34d5036692 ("lib: test_bitmap: add compile-time optimization/evaluations
assertions")
    Co-developed-by: Nathan Chancellor <nathan@kernel.org>
    Signed-off-by: Nathan Chancellor <nathan@kernel.org>
    Signed-off-by: Yury Norov <yury.norov@gmail.com>
    Reviewed-by: Nick Desaulniers <ndesaulniers@google.com>
    Reviewed-by: Alexander Lobakin <aleksander.lobakin@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9851630bbd45905a5ef226912957600c6e184982
Author: Sukrut Bellary <sukrut.bellary@linux.com>
Date:   Tue Jul 18 01:55:29 2023 -0700
```

```
    firmware: arm_scmi: Fix signed error return values handling

    [ Upstream commit 81b233b8dd72f2d1df3da8bd4bd4f8c5e84937b9 ]

    Handle signed error return values returned by simple_write_to_buffer().
    In case of an error, return the error code.

    Fixes: 3c3d818a9317 ("firmware: arm_scmi: Add core raw transmission support")
    Reported-by: Dan Carpenter <dan.carpenter@linaro.org>
    Signed-off-by: Sukrut Bellary <sukrut.bellary@linux.com>
    Reviewed-by: Cristian Marussi <cristian.marussi@arm.com>
    Tested-by: Cristian Marussi <cristian.marussi@arm.com>
    Reviewed-by: Dan Carpenter <dan.carpenter@linaro.org>
    Link: https://lore.kernel.org/r/20230718085529.258899-1-sukrut.bellary@linux.com
    Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 32e44b7d7acc34f74c2f06f00607d04d23d710b6
Author: Punit Agrawal <punit.agrawal@bytedance.com>
Date:   Mon Jul 17 18:17:02 2023 +0100

    firmware: smccc: Fix use of uninitialised results structure

    [ Upstream commit d05799d7b4a39fa71c65aa277128ac7c843ffcdc ]

    Commit 35727af2b15d ("irqchip/gicv3: Workaround for NVIDIA erratum
    T241-FABRIC-4") moved the initialisation of the SoC version to
    arm_smccc_version_init() but forgot to update the results structure
    and it's usage.

    Fix the use of the uninitialised results structure and update the
    error strings.

    Fixes: 35727af2b15d ("irqchip/gicv3: Workaround for NVIDIA erratum T241-FABRIC-4")
    Signed-off-by: Punit Agrawal <punit.agrawal@bytedance.com>
    Cc: Sudeep Holla <sudeep.holla@arm.com>
    Cc: Marc Zyngier <maz@kernel.org>
    Cc: Vikram Sethi <vsethi@nvidia.com>
    Cc: Shanker Donthineni <sdonthineni@nvidia.com>
    Acked-by: Marc Zyngier <maz@kernel.org>
    Link: https://lore.kernel.org/r/20230717171702.424253-1-punit.agrawal@bytedance.com
    Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4cfbe33a8897e6ef9ba102ddd7f616d68797c28b
Author: Benjamin Gaignard <benjamin.gaignard@collabora.com>
Date:   Fri Jul 7 11:42:00 2023 +0200

    arm64: dts: freescale: Fix VPU G2 clock

    [ Upstream commit b27bfc5103c72f84859bd32731b6a09eafdeda05 ]

    Set VPU G2 clock to 300MHz like described in documentation.
    This fixes pixels error occurring with large resolution ( >= 2560x1600)
    HEVC test stream when using the postprocessor to produce NV12.

    Fixes: 4ac7e4a81272 ("arm64: dts: imx8mq: Enable both G1 and G2 VPU's with vpu-blk-
ctrl")
    Signed-off-by: Benjamin Gaignard <benjamin.gaignard@collabora.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 25ee70d51d0a7d34fbac71bab8abcc77d777d8f2
Author: Hugo Villeneuve <hvilleneuve@dimonoff.com>
Date:   Tue Jul 4 09:48:00 2023 -0400
```

```
    arm64: dts: imx8mn-var-som: add missing pull-up for onboard PHY reset pinmux

    [ Upstream commit 253be5b53c2792fb4384f8005b05421e6f040ee3 ]

    For SOMs with an onboard PHY, the RESET_N pull-up resistor is
    currently deactivated in the pinmux configuration. When the pinmux
    code selects the GPIO function for this pin, with a default direction
    of input, this prevents the RESET_N pin from being taken to the proper
    3.3V level (deasserted), and this results in the PHY being not
    detected since it is held in reset.

    Taken from RESET_N pin description in ADIN13000 datasheet:
        This pin requires a 1K pull-up resistor to AVDD_3P3.

    Activate the pull-up resistor to fix the issue.

    Fixes: ade0176dd8a0 ("arm64: dts: imx8mn-var-som: Add Variscite VAR-SOM-MX8MN System on
Module")
    Signed-off-by: Hugo Villeneuve <hvilleneuve@dimonoff.com>
    Reviewed-by: Fabio Estevam <festevam@gmail.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d5d5cbb0d05194bd3f3abb0eca02d51212eb12bc
Author: Yashwanth Varakala <y.varakala@phytec.de>
Date:   Fri Jun 16 11:50:09 2023 +0200

    arm64: dts: phycore-imx8mm: Correction in gpio-line-names

    [ Upstream commit 1ef0aa137a96c5f0564f2db0c556a4f0f60ce8f5 ]

    Remove unused nINT_ETHPHY entry from gpio-line-names in gpio1 nodes of
    phyCORE-i.MX8MM and phyBOARD-Polis-i.MX8MM devicetrees.

    Fixes: ae6847f26ac9 ("arm64: dts: freescale: Add phyBOARD-Polis-i.MX8MM support")
    Signed-off-by: Yashwanth Varakala <y.varakala@phytec.de>
    Signed-off-by: Cem Tenruh <c.tenruh@phytec.de>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3fe4b022bbb596f6123edff7a0a169c15059a0fb
Author: Yashwanth Varakala <y.varakala@phytec.de>
Date:   Fri Jun 16 11:50:07 2023 +0200

    arm64: dts: phycore-imx8mm: Label typo-fix of VPU

    [ Upstream commit cddeefc1663294fb74b31ff5029a83c0e819ff3a ]

    Corrected the label of the VPU regulator node (buck 3)
    from reg_vdd_gpu to reg_vdd_vpu.

    Fixes: ae6847f26ac9 ("arm64: dts: freescale: Add phyBOARD-Polis-i.MX8MM support")
    Signed-off-by: Yashwanth Varakala <y.varakala@phytec.de>
    Signed-off-by: Cem Tenruh <c.tenruh@phytec.de>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 599c6d1ce1b68a3a92f23b69a658d7042119f53d
Author: Tim Harvey <tharvey@gateworks.com>
Date:   Tue Jun 6 08:40:30 2023 -0700

    arm64: dts: imx8mm-venice-gw7904: disable disp_blk_ctrl

    [ Upstream commit f7a0b57524cf811ac06257a5099f1b7c19ee7310 ]
```

```
    The GW7904 does not connect the VDD_MIPI power rails thus MIPI is
    disabled. However we must also disable disp_blk_ctrl as it uses the
    pgc_mipi power domain and without it being disabled imx8m-blk-ctrl will
    fail to probe:
    imx8m-blk-ctrl 32e28000.blk-ctrl: error -ETIMEDOUT: failed to attach
    power domain "mipi-dsi"
    imx8m-blk-ctrl: probe of 32e28000.blk-ctrl failed with error -110

    Fixes: b999bdaf0597 ("arm64: dts: imx: Add i.mx8mm Gateworks gw7904 dts support")
    Signed-off-by: Tim Harvey <tharvey@gateworks.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 132d600e52a70808367f5960e0870a4c9f465d97
Author: Tim Harvey <tharvey@gateworks.com>
Date:   Tue Jun 6 08:39:45 2023 -0700

    arm64: dts: imx8mm-venice-gw7903: disable disp_blk_ctrl

    [ Upstream commit 3e7d3c5e13b05dda9db92d98803a626378e75438 ]

    The GW7903 does not connect the VDD_MIPI power rails thus MIPI is
    disabled. However we must also disable disp_blk_ctrl as it uses the
    pgc_mipi power domain and without it being disabled imx8m-blk-ctrl will
    fail to probe:
    imx8m-blk-ctrl 32e28000.blk-ctrl: error -ETIMEDOUT: failed to attach power domain "mipi-
dsi"
    imx8m-blk-ctrl: probe of 32e28000.blk-ctrl failed with error -110

    Fixes: a72ba91e5bc7 ("arm64: dts: imx: Add i.mx8mm Gateworks gw7903 dts support")
    Signed-off-by: Tim Harvey <tharvey@gateworks.com>
    Signed-off-by: Shawn Guo <shawnguo@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 5ed4b021a3ba543f2cdf20a406e64994c94e5812
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Wed Aug 2 17:26:20 2023 +0000

    iommu/arm-smmu-v3: Document nesting-related errata

    commit 0bfbfc526c70606bf0fad302e4821087cbecfaf4 upstream

    Both MMU-600 and MMU-700 have similar errata around TLB invalidation
    while both stages of translation are active, which will need some
    consideration once nesting support is implemented. For now, though,
    it's very easy to make our implicit lack of nesting support explicit
    for those cases, so they're less likely to be missed in future.

    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Nicolin Chen <nicolinc@nvidia.com>
    Link:
https://lore.kernel.org/r/696da78d32bb4491f898f11b0bb4d850a8aa7c6a.1683731256.git.robin.murp
hy@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Easwar Hariharan <eahariha@linux.microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c7af8840f215d4ae4c33b16fcaf874bd6f3a75c6
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Wed Aug 2 17:26:19 2023 +0000

    iommu/arm-smmu-v3: Add explicit feature for nesting

    commit 1d9777b9f3d55b4b6faf186ba4f1d6fb560c0523 upstream
```

```
    In certain cases we may want to refuse to allow nested translation even
    when both stages are implemented, so let's add an explicit feature for
    nesting support which we can control in its own right. For now this
    merely serves as documentation, but it means a nice convenient check
    will be ready and waiting for the future nesting code.

    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Nicolin Chen <nicolinc@nvidia.com>
    Link:
https://lore.kernel.org/r/136c3f4a3a84cc14a5a1978ace57dfd3ed67b688.1683731256.git.robin.murp
hy@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Easwar Hariharan <eahariha@linux.microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8f2c8d8a8dec1d15611434f84ea2437e6e352de6
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Wed Aug 2 17:26:18 2023 +0000

    iommu/arm-smmu-v3: Document MMU-700 erratum 2812531

    commit 309a15cb16bb075da1c99d46fb457db6a1a2669e upstream

    To work around MMU-700 erratum 2812531 we need to ensure that certain
    sequences of commands cannot be issued without an intervening sync. In
    practice this falls out of our current command-batching machinery
    anyway - each batch only contains a single type of invalidation command,
    and ends with a sync. The only exception is when a batch is sufficiently
    large to need issuing across multiple command queue slots, wherein the
    earlier slots will not contain a sync and thus may in theory interleave
    with another batch being issued in parallel to create an affected
    sequence across the slot boundary.

    Since MMU-700 supports range invalidate commands and thus we will prefer
    to use them (which also happens to avoid conditions for other errata),
    I'm not entirely sure it's even possible for a single high-level
    invalidate call to generate a batch of more than 63 commands, but for
    the sake of robustness and documentation, wire up an option to enforce
    that a sync is always inserted for every slot issued.

    The other aspect is that the relative order of DVM commands cannot be
    controlled, so DVM cannot be used. Again that is already the status quo,
    but since we have at least defined ARM_SMMU_FEAT_BTM, we can explicitly
    disable it for documentation purposes even if it's not wired up anywhere
    yet.

    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Nicolin Chen <nicolinc@nvidia.com>
    Link:
https://lore.kernel.org/r/330221cdfd0003cd51b6c04e7ff3566741ad8374.1683731256.git.robin.murp
hy@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Easwar Hariharan <eahariha@linux.microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f6a58dbe36d7d61fb624631097f6fc371656f3d6
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Wed Aug 2 17:26:17 2023 +0000

    iommu/arm-smmu-v3: Work around MMU-600 erratum 1076982

    commit f322e8af35c7f23a8c08b595c38d6c855b2d836f upstream

    MMU-600 versions prior to r1p0 fail to correctly generate a WFE wakeup
```

```
    event when the command queue transitions fom full to non-full. We can
    easily work around this by simply hiding the SEV capability such that we
    fall back to polling for space in the queue - since MMU-600 implements
    MSIs we wouldn't expect to need SEV for sync completion either, so this
    should have little to no impact.

    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Reviewed-by: Nicolin Chen <nicolinc@nvidia.com>
    Tested-by: Nicolin Chen <nicolinc@nvidia.com>
    Link:
https://lore.kernel.org/r/08adbe3d01024d8382a478325f73b56851f76e49.1683731256.git.robin.murp
hy@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Easwar Hariharan <eahariha@linux.microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 93a0b7d43de51b38c4c91d4121e447018132119a
Author: Jann Horn <jannh@google.com>
Date:   Wed Jul 26 23:41:03 2023 +0200

    mm: lock_vma_under_rcu() must check vma->anon_vma under vma lock

    commit 657b5146955eba331e01b9a6ae89ce2e716ba306 upstream.

    lock_vma_under_rcu() tries to guarantee that __anon_vma_prepare() can't
    be called in the VMA-locked page fault path by ensuring that
    vma->anon_vma is set.

    However, this check happens before the VMA is locked, which means a
    concurrent move_vma() can concurrently call unlink_anon_vmas(), which
    disassociates the VMA's anon_vma.

    This means we can get UAF in the following scenario:

      THREAD 1                  THREAD 2
      ========                  ========
      <page fault>
        lock_vma_under_rcu()
          rcu_read_lock()
          mas_walk()
          check vma->anon_vma

                                mremap() syscall
                                  move_vma()
                                   vma_start_write()
                                    unlink_anon_vmas()
                                <syscall end>

        handle_mm_fault()
          __handle_mm_fault()
            handle_pte_fault()
              do_pte_missing()
                do_anonymous_page()
                  anon_vma_prepare()
                    __anon_vma_prepare()
                      find_mergeable_anon_vma()
                        mas_walk() [looks up VMA X]

                                munmap() syscall (deletes VMA X)

                        reusable_anon_vma() [called on freed VMA X]

    This is a security bug if you can hit it, although an attacker would
    have to win two races at once where the first race window is only a few
    instructions wide.
```

```
    This patch is based on some previous discussion with Linus Torvalds on
    the security list.

    Cc: stable@vger.kernel.org
    Fixes: 5e31275cc997 ("mm: add per-VMA lock and helper functions to control it")
    Signed-off-by: Jann Horn <jannh@google.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```