```
commit 3c730ee65d574cbf2d05559cda2cb07d8f3f8b7a
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Aug 31 17:18:21 2022 +0200

    Linux 5.19.6

    Link: https://lore.kernel.org/r/20220829105808.828227973@linuxfoundation.org
    Tested-by: Florian Fainelli <f.fainelli@gmail.com>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Zan Aziz <zanaziz313@gmail.com>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Sudip Mukherjee <sudip.mukherjee@codethink.co.uk>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Fenil Jain <fkjainco@gmail.com>
    Tested-by: Rudi Heitbaum <rudi@heitbaum.com>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Tested-by: Jiri Slaby <jirislaby@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a36df92c7ff7ecde2fb362241d0ab024dddd0597
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Thu Aug 25 23:26:47 2022 +0200

    bpf: Don't use tnum_range on array range checking for poke descriptors

    commit a657182a5c5150cdfacb6640aad1d2712571a409 upstream.

    Hsin-Wei reported a KASAN splat triggered by their BPF runtime fuzzer which
    is based on a customized syzkaller:

      BUG: KASAN: slab-out-of-bounds in bpf_int_jit_compile+0x1257/0x13f0
      Read of size 8 at addr ffff888004e90b58 by task syz-executor.0/1489
      CPU: 1 PID: 1489 Comm: syz-executor.0 Not tainted 5.19.0 #1
      Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
      1.13.0-1ubuntu1.1 04/01/2014
      Call Trace:
       <TASK>
       dump_stack_lvl+0x9c/0xc9
       print_address_description.constprop.0+0x1f/0x1f0
       ? bpf_int_jit_compile+0x1257/0x13f0
       kasan_report.cold+0xeb/0x197
       ? kvmalloc_node+0x170/0x200
       ? bpf_int_jit_compile+0x1257/0x13f0
       bpf_int_jit_compile+0x1257/0x13f0
       ? arch_prepare_bpf_dispatcher+0xd0/0xd0
       ? rcu_read_lock_sched_held+0x43/0x70
       bpf_prog_select_runtime+0x3e8/0x640
       ? bpf_obj_name_cpy+0x149/0x1b0
       bpf_prog_load+0x102f/0x2220
       ? __bpf_prog_put.constprop.0+0x220/0x220
       ? find_held_lock+0x2c/0x110
       ? __might_fault+0xd6/0x180
       ? lock_downgrade+0x6e0/0x6e0
       ? lock_is_held_type+0xa6/0x120
       ? __might_fault+0x147/0x180
       __sys_bpf+0x137b/0x6070
       ? bpf_perf_link_attach+0x530/0x530
       ? new_sync_read+0x600/0x600
       ? __fget_files+0x255/0x450
       ? lock_downgrade+0x6e0/0x6e0
       ? fput+0x30/0x1a0
```

```
        ? ksys_write+0x1a8/0x260
        __x64_sys_bpf+0x7a/0xc0
        ? syscall_enter_from_user_mode+0x21/0x70
        do_syscall_64+0x3b/0x90
        entry_SYSCALL_64_after_hwframe+0x63/0xcd
       RIP: 0033:0x7f917c4e2c2d

    The problem here is that a range of tnum_range(0, map->max_entries - 1) has
    limited ability to represent the concrete tight range with the tnum as the
    set of resulting states from value + mask can result in a superset of the
    actual intended range, and as such a tnum_in(range, reg->var_off) check may
    yield true when it shouldn't, for example tnum_range(0, 2) would result in
    00XX -> v = 0000, m = 0011 such that the intended set of {0, 1, 2} is here
    represented by a less precise superset of {0, 1, 2, 3}. As the register is
    known const scalar, really just use the concrete reg->var_off.value for the
    upper index check.

    Fixes: d2e4c1e6c294 ("bpf: Constant map key tracking for prog array pokes")
    Reported-by: Hsin-Wei Hung <hsinweih@uci.edu>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Cc: Shung-Hsi Yu <shung-hsi.yu@suse.com>
    Acked-by: John Fastabend <john.fastabend@gmail.com>
    Link:
https://lore.kernel.org/r/984b37f9fdf7ac36831d2137415a4a915744c1b6.1661462653.git.daniel@iog
earbox.net
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f0e5ce88e1cf2734afbb5ad6377c7bd7ad0992d3
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sat Aug 20 00:14:16 2022 +0100

    riscv: dts: microchip: mpfs: remove pci axi address translation property

    commit e4009c5fa77b4356aa37ce002e9f9952dfd7a615 upstream.

    An AXI master address translation table property was inadvertently
    added to the device tree & this was not caught by dtbs_check at the
    time. Remove the property - it should not be in mpfs.dtsi anyway as
    it would be more suitable in -fabric.dtsi nor does it actually apply
    to the version of the reference design we are using for upstream.

    Link: https://www.microsemi.com/document-portal/doc_download/1245812-polarfire-fpga-and-
polarfire-soc-fpga-pci-express-user-guide # Section 1.3.3
    Fixes: 528a5b1f2556 ("riscv: dts: microchip: add new peripherals to icicle kit device
tree")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 14f158b9770fd55bacb5087588d8038aa9b80f67
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sat Aug 20 00:14:15 2022 +0100

    riscv: dts: microchip: mpfs: remove bogus card-detect-delay

    commit 2b55915d27dcaa35f54bad7925af0a76001079bc upstream.

    Recent versions of dt-schema warn about a previously undetected
    undocumented property:
    arch/riscv/boot/dts/microchip/mpfs-icicle-kit.dtb: mmc@20008000: Unevaluated properties
are not allowed ('card-detect-delay' was unexpected)
            From schema: Documentation/devicetree/bindings/mmc/cdns,sdhci.yaml

    There are no GPIOs connected to MSSIO6B4 pin K3 so adding the common
    cd-debounce-delay-ms property makes no sense. The Cadence IP has a
```

```
    register that sets the card detect delay as "DP * tclk". On MPFS, this
    clock frequency is not configurable (it must be 200 MHz) & the FPGA
    comes out of reset with this register already set.

    Fixes: bc47b2217f24 ("riscv: dts: microchip: add the sundance polarberry")
    Fixes: 0fa6107eca41 ("RISC-V: Initial DTS for Microchip ICICLE board")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a8604d23a8122df7ff929ce4c5d2be1b4be9bb6e
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sat Aug 20 00:14:14 2022 +0100

    riscv: dts: microchip: mpfs: remove ti,fifo-depth property

    commit 72a05748cbd285567d69f173f8694e3471b79f20 upstream.

    Recent versions of dt-schema warn about a previously undetected
    undocument property on the icicle & polarberry devicetrees:

    arch/riscv/boot/dts/microchip/mpfs-icicle-kit.dtb: ethernet@20112000: ethernet-phy@8:
Unevaluated properties are not allowed ('ti,fifo-depth' was unexpected)
            From schema: Documentation/devicetree/bindings/net/cdns,macb.yaml

    I know what you're thinking, the binding doesn't look to be the problem
    and I agree. I am not sure why a TI vendor property was ever actually
    added since it has no meaning... just get rid of it.

    Fixes: bc47b2217f24 ("riscv: dts: microchip: add the sundance polarberry")
    Fixes: 0fa6107eca41 ("RISC-V: Initial DTS for Microchip ICICLE board")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5977375a7dba19bc882faeeabac9bd271e78b4f6
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sat Aug 20 00:14:13 2022 +0100

    riscv: dts: microchip: mpfs: fix incorrect pcie child node name

    commit 3f67e69976035352db110443916bcce32c7f64ac upstream.

    Recent versions of dt-schema complain about the PCIe controller's child
    node name:
    arch/riscv/boot/dts/microchip/mpfs-icicle-kit.dtb: pcie@2000000000: Unevaluated
properties are not allowed ('clock-names', 'clocks', 'legacy-interrupt-controller',
'microchip,axi-m-atr0' were unexpected)
                From schema: Documentation/devicetree/bindings/pci/microchip,pcie-host.yaml
    Make the dts match the correct property name in the dts.

    Fixes: 528a5b1f2556 ("riscv: dts: microchip: add new peripherals to icicle kit device
tree")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f24ee7391a75b9577fdf40b16039e0b6a97abae3
Author: Mike Christie <michael.christie@oracle.com>
Date:   Thu Aug 11 20:12:06 2022 -0500

    scsi: core: Fix passthrough retry counter handling

    commit fac8e558da9485e13a0ae0488aa0b8a8c307cd34 upstream.

    Passthrough users will set the scsi_cmnd->allowed value and were expecting
    up to $allowed retries. The problem is that before:
```

```
    commit 6aded12b10e0 ("scsi: core: Remove struct scsi_request")

    we used to set the retries on the scsi_request then copy them over to
    scsi_cmnd->allowed in scsi_setup_scsi_cmnd. With that patch we now set
    scsi_cmnd->allowed to 0 in scsi_prepare_cmd and overwrite what the
    passthrough user set.

    This moves the allowed initialization to after the blk_rq_is_passthrough()
    check so it's only done for the non-passthrough path where the ULD
    init_command will normally set an allowed value it prefers.

    Link: https://lore.kernel.org/r/20220812011206.9157-1-michael.christie@oracle.com
    Fixes: 6aded12b10e0 ("scsi: core: Remove struct scsi_request")
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Mike Christie <michael.christie@oracle.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 828f57ac75eaccd6607ee4d1468d34e983e32c68
Author: Saurabh Sengar <ssengar@linux.microsoft.com>
Date:   Thu Aug 4 08:55:34 2022 -0700

    scsi: storvsc: Remove WQ_MEM_RECLAIM from storvsc_error_wq

    commit d957e7ffb2c72410bcc1a514153a46719255a5da upstream.

    storvsc_error_wq workqueue should not be marked as WQ_MEM_RECLAIM as it
    doesn't need to make forward progress under memory pressure.  Marking this
    workqueue as WQ_MEM_RECLAIM may cause deadlock while flushing a
    non-WQ_MEM_RECLAIM workqueue.  In the current state it causes the following
    warning:

    [   14.506347] ------------[ cut here ]------------
    [   14.506354] workqueue: WQ_MEM_RECLAIM storvsc_error_wq_0:storvsc_remove_lun is
flushing !WQ_MEM_RECLAIM events_freezable_power_:disk_events_workfn
    [   14.506360] WARNING: CPU: 0 PID: 8 at <-snip->kernel/workqueue.c:2623
check_flush_dependency+0xb5/0x130
    [   14.506390] CPU: 0 PID: 8 Comm: kworker/u4:0 Not tainted 5.4.0-1086-azure
#91~18.04.1-Ubuntu
    [   14.506391] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine,
BIOS Hyper-V UEFI Release v4.1 05/09/2022
    [   14.506393] Workqueue: storvsc_error_wq_0 storvsc_remove_lun
    [   14.506395] RIP: 0010:check_flush_dependency+0xb5/0x130
                    <-snip->
    [   14.506408] Call Trace:
    [   14.506412]  __flush_work+0xf1/0x1c0
    [   14.506414]  __cancel_work_timer+0x12f/0x1b0
    [   14.506417]  ? kernfs_put+0xf0/0x190
    [   14.506418]  cancel_delayed_work_sync+0x13/0x20
    [   14.506420]  disk_block_events+0x78/0x80
    [   14.506421]  del_gendisk+0x3d/0x2f0
    [   14.506423]  sr_remove+0x28/0x70
    [   14.506427]  device_release_driver_internal+0xef/0x1c0
    [   14.506428]  device_release_driver+0x12/0x20
    [   14.506429]  bus_remove_device+0xe1/0x150
    [   14.506431]  device_del+0x167/0x380
    [   14.506432]  __scsi_remove_device+0x11d/0x150
    [   14.506433]  scsi_remove_device+0x26/0x40
    [   14.506434]  storvsc_remove_lun+0x40/0x60
    [   14.506436]  process_one_work+0x209/0x400
    [   14.506437]  worker_thread+0x34/0x400
    [   14.506439]  kthread+0x121/0x140
    [   14.506440]  ? process_one_work+0x400/0x400
    [   14.506441]  ? kthread_park+0x90/0x90
    [   14.506443]  ret_from_fork+0x35/0x40
```

```
    [   14.506445] ---[ end trace 2d9633159fdc6ee7 ]---

    Link: https://lore.kernel.org/r/1659628534-17539-1-git-send-email-
ssengar@linux.microsoft.com
    Fixes: 436ad9413353 ("scsi: storvsc: Allow only one remove lun work item to be issued
per lun")
    Reviewed-by: Michael Kelley <mikelley@microsoft.com>
    Signed-off-by: Saurabh Sengar <ssengar@linux.microsoft.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a292244e5bfa8800bd2f9d42c1878b30cb728181
Author: Kiwoong Kim <kwmad.kim@samsung.com>
Date:   Tue Aug 2 10:42:31 2022 +0900

    scsi: ufs: core: Enable link lost interrupt

    commit 6d17a112e9a63ff6a5edffd1676b99e0ffbcd269 upstream.

    Link lost is treated as fatal error with commit c99b9b230149 ("scsi: ufs:
    Treat link loss as fatal error"), but the event isn't registered as
    interrupt source. Enable it.

    Link: https://lore.kernel.org/r/1659404551-160958-1-git-send-email-kwmad.kim@samsung.com
    Fixes: c99b9b230149 ("scsi: ufs: Treat link loss as fatal error")
    Reviewed-by: Bart Van Assche <bvanassche@acm.org>
    Signed-off-by: Kiwoong Kim <kwmad.kim@samsung.com>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0761b0e818c7b41c0a2c61477a944314150c0ccc
Author: Mark Brown <broonie@kernel.org>
Date:   Wed Aug 17 19:23:24 2022 +0100

    arm64/sme: Don't flush SVE register state when handling SME traps

    commit 714f3cbd70a4db9f9b7fe5b8a032896ed33fb824 upstream.

    Currently as part of handling a SME access trap we flush the SVE register
    state. This is not needed and would corrupt register state if the task has
    access to the SVE registers already. For non-streaming mode accesses the
    required flushing will be done in the SVE access trap. For streaming
    mode SVE register accesses the architecture guarantees that the register
    state will be flushed when streaming mode is entered or exited so there is
    no need for us to do so. Simply remove the register initialisation.

    Fixes: 8bd7f91c03d8 ("arm64/sme: Implement traps and syscall handling for SME")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Reviewed-by: Catalin Marinas <catalin.marinas@arm.com>
    Link: https://lore.kernel.org/r/20220817182324.638214-5-broonie@kernel.org
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a8d79f9d1a4d90b7b4eb8bf7aa61995359aeb02e
Author: Mark Brown <broonie@kernel.org>
Date:   Wed Aug 17 19:23:23 2022 +0100

    arm64/sme: Don't flush SVE register state when allocating SME storage

    commit 826a4fdd2ada9e5923c58bdd168f31a42e958ffc upstream.

    Currently when taking a SME access trap we allocate storage for the SVE
    register state in order to be able to handle storage of streaming mode SVE.
    Due to the original usage in a purely SVE context the SVE register state
    allocation this also flushes the register state for SVE if storage was
```

```
    already allocated but in the SME context this is not desirable. For a SME
    access trap to be taken the task must not be in streaming mode so either
    there already is SVE register state present for regular SVE mode which would
    be corrupted or the task does not have TIF_SVE and the flush is redundant.

    Fix this by adding a flag to sve_alloc() indicating if we are in a SVE
    context and need to flush the state. Freshly allocated storage is always
    zeroed either way.

    Fixes: 8bd7f91c03d8 ("arm64/sme: Implement traps and syscall handling for SME")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Reviewed-by: Catalin Marinas <catalin.marinas@arm.com>
    Link: https://lore.kernel.org/r/20220817182324.638214-4-broonie@kernel.org
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 913fe86ae9038cb450c573ea991499c4f32d1264
Author: Mark Brown <broonie@kernel.org>
Date:   Wed Aug 17 19:23:22 2022 +0100

    arm64/signal: Flush FPSIMD register state when disabling streaming mode

    commit ea64baacbc36a0d552aec0d87107182f40211131 upstream.

    When handling a signal delivered to a context with streaming mode enabled
    we will disable streaming mode for the signal handler, when doing so we
    should also flush the saved FPSIMD register state like exiting streaming
    mode in the hardware would do so that if that state is reloaded we get the
    same behaviour. Without this we will reload whatever the last FPSIMD state
    that was saved for the task was.

    Fixes: 40a8e87bb328 ("arm64/sme: Disable ZA and streaming mode when handling signals")
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Reviewed-by: Catalin Marinas <catalin.marinas@arm.com>
    Link: https://lore.kernel.org/r/20220817182324.638214-3-broonie@kernel.org
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f83cbd14c79459b03f1d0235c76533c5628b7263
Author: Mark Rutland <mark.rutland@arm.com>
Date:   Wed Aug 17 16:40:22 2022 +0100

    arm64: fix rodata=full

    commit 2e8cff0a0eee87b27f0cf87ad8310eb41b5886ab upstream.

    On arm64, "rodata=full" has been suppored (but not documented) since
    commit:

      c55191e96caa9d78 ("arm64: mm: apply r/o permissions of VM areas to its linear alias as
well")

    As it's necessary to determine the rodata configuration early during
    boot, arm64 has an early_param() handler for this, whereas init/main.c
    has a __setup() handler which is run later.

    Unfortunately, this split meant that since commit:

      f9a40b0890658330 ("init/main.c: return 1 from handled __setup() functions")

    ... passing "rodata=full" would result in a spurious warning from the
    __setup() handler (though RO permissions would be configured
    appropriately).

    Further, "rodata=full" has been broken since commit:
```

```
      0d6ea3ac94ca77c5 ("lib/kstrtox.c: add "false"/"true" support to kstrtobool()")

    ... which caused strtobool() to parse "full" as false (in addition to
    many other values not documented for the "rodata=" kernel parameter.

    This patch fixes this breakage by:

    * Moving the core parameter parser to an __early_param(), such that it
      is available early.

    * Adding an (optional) arch hook which arm64 can use to parse "full".

    * Updating the documentation to mention that "full" is valid for arm64.

    * Having the core parameter parser handle "on" and "off" explicitly,
      such that any undocumented values (e.g. typos such as "ful") are
      reported as errors rather than being silently accepted.

    Note that __setup() and early_param() have opposite conventions for
    their return values, where __setup() uses 1 to indicate a parameter was
    handled and early_param() uses 0 to indicate a parameter was handled.

    Fixes: f9a40b089065 ("init/main.c: return 1 from handled __setup() functions")
    Fixes: 0d6ea3ac94ca ("lib/kstrtox.c: add "false"/"true" support to kstrtobool()")
    Signed-off-by: Mark Rutland <mark.rutland@arm.com>
    Cc: Andy Shevchenko <andy.shevchenko@gmail.com>
    Cc: Ard Biesheuvel <ardb@kernel.org>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Jagdish Gediya <jvgediya@linux.ibm.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Randy Dunlap <rdunlap@infradead.org>
    Cc: Will Deacon <will@kernel.org>
    Reviewed-by: Ard Biesheuvel <ardb@kernel.org>
    Link: https://lore.kernel.org/r/20220817154022.3974645-1-mark.rutland@arm.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ec76a1de1d65cdca53918f7b3258b1938a147ed1
Author: Ian Rogers <irogers@google.com>
Date:   Mon Aug 22 14:33:51 2022 -0700

    perf stat: Clear evsel->reset_group for each stat run

    commit bf515f024e4c0ca46a1b08c4f31860c01781d8a5 upstream.

    If a weak group is broken then the reset_group flag remains set for
    the next run. Having reset_group set means the counter isn't created
    and ultimately a segfault.

    A simple reproduction of this is:

      # perf stat -r2 -e
'{cycles,cycles,cycles,cycles,cycles,cycles,cycles,cycles,cycles,cycles}:W

    which will be added as a test in the next patch.

    Fixes: 4804e0111662d7d8 ("perf stat: Use affinity for opening events")
    Reviewed-by: Andi Kleen <ak@linux.intel.com>
    Signed-off-by: Ian Rogers <irogers@google.com>
    Tested-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Tested-by: Xing Zhengjun <zhengjun.xing@linux.intel.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Andi Kleen <ak@linux.intel.com>
    Cc: Ingo Molnar <mingo@redhat.com>
```

```
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Kan Liang <kan.liang@linux.intel.com>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: Stephane Eranian <eranian@google.com>
    Link: https://lore.kernel.org/r/20220822213352.75721-1-irogers@google.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6d7a4a140cfcea05278217dd21e86835e2dc6087
Author: Stephane Eranian <eranian@google.com>
Date:   Wed Aug 17 22:46:13 2022 -0700

    perf/x86/intel/ds: Fix precise store latency handling

    commit d4bdb0bebc5ba3299d74f123c782d99cd4e25c49 upstream.

    With the existing code in store_latency_data(), the memory operation (mem_op)
    returned to the user is always OP_LOAD where in fact, it should be OP_STORE.
    This comes from the fact that the function is simply grabbing the information
    from a data source map which covers only load accesses. Intel 12th gen CPU
    offers precise store sampling that captures both the data source and latency.
    Therefore it can use the data source mapping table but must override the
    memory operation to reflect stores instead of loads.

    Fixes: 61b985e3e775 ("perf/x86/intel: Add perf core PMU support for Sapphire Rapids")
    Signed-off-by: Stephane Eranian <eranian@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Link: https://lkml.kernel.org/r/20220818054613.1548130-1-eranian@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 291f8baead174e17654465dcccc47e87530f8896
Author: Stephane Eranian <eranian@google.com>
Date:   Wed Aug 3 09:00:31 2022 -0700

    perf/x86/intel/uncore: Fix broken read_counter() for SNB IMC PMU

    commit 11745ecfe8fea4b4a4c322967a7605d2ecbd5080 upstream.

    Existing code was generating bogus counts for the SNB IMC bandwidth counters:

    $ perf stat -a -I 1000 -e uncore_imc/data_reads/,uncore_imc/data_writes/
         1.000327813           1,024.03 MiB  uncore_imc/data_reads/
         1.000327813              20.73 MiB  uncore_imc/data_writes/
         2.000580153         261,120.00 MiB  uncore_imc/data_reads/
         2.000580153              23.28 MiB  uncore_imc/data_writes/

    The problem was introduced by commit:
      07ce734dd8ad ("perf/x86/intel/uncore: Clean up client IMC")

    Where the read_counter callback was replace to point to the generic
    uncore_mmio_read_counter() function.

    The SNB IMC counters are freerunnig 32-bit counters laid out contiguously in
    MMIO. But uncore_mmio_read_counter() is using a readq() call to read from
    MMIO therefore reading 64-bit from MMIO. Although this is okay for the
    uncore_perf_event_update() function because it is shifting the value based
    on the actual counter width to compute a delta, it is not okay for the
    uncore_pmu_event_start() which is simply reading the counter  and therefore
    priming the event->prev_count with a bogus value which is responsible for
    causing bogus deltas in the perf stat command above.

    The fix is to reintroduce the custom callback for read_counter for the SNB
    IMC PMU and use readl() instead of readq(). With the change the output of
```

```
    perf stat is back to normal:
    $ perf stat -a -I 1000 -e uncore_imc/data_reads/,uncore_imc/data_writes/
         1.000120987            296.94 MiB  uncore_imc/data_reads/
         1.000120987            138.42 MiB  uncore_imc/data_writes/
         2.000403144            175.91 MiB  uncore_imc/data_reads/
         2.000403144             68.50 MiB  uncore_imc/data_writes/

    Fixes: 07ce734dd8ad ("perf/x86/intel/uncore: Clean up client IMC")
    Signed-off-by: Stephane Eranian <eranian@google.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Reviewed-by: Kan Liang <kan.liang@linux.intel.com>
    Link: https://lore.kernel.org/r/20220803160031.1379788-1-eranian@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a9271d39d6dc8a9b2fba6ed9312f8d77ba9f5379
Author: James Clark <james.clark@arm.com>
Date:   Thu Jul 28 10:39:46 2022 +0100

    perf python: Fix build when PYTHON_CONFIG is user supplied

    commit bc9e7fe313d5e56d4d5f34bcc04d1165f94f86fb upstream.

    The previous change to Python autodetection had a small mistake where
    the auto value was used to determine the Python binary, rather than the
    user supplied value. The Python binary is only used for one part of the
    build process, rather than the final linking, so it was producing
    correct builds in most scenarios, especially when the auto detected
    value matched what the user wanted, or the system only had a valid set
    of Pythons.

    Change it so that the Python binary path is derived from either the
    PYTHON_CONFIG value or PYTHON value, depending on what is specified by
    the user. This was the original intention.

    This error was spotted in a build failure an odd cross compilation
    environment after commit 4c41cb46a732fe82 ("perf python: Prefer
    python3") was merged.

    Fixes: 630af16eee495f58 ("perf tools: Use Python devtools for version autodetection
rather than runtime")
    Signed-off-by: James Clark <james.clark@arm.com>
    Acked-by: Ian Rogers <irogers@google.com>
    Cc: Alexander Shishkin <alexander.shishkin@linux.intel.com>
    Cc: Ingo Molnar <mingo@redhat.com>
    Cc: James Clark <james.clark@arm.com>
    Cc: Jiri Olsa <jolsa@kernel.org>
    Cc: Mark Rutland <mark.rutland@arm.com>
    Cc: Namhyung Kim <namhyung@kernel.org>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Link: https://lore.kernel.org/r/20220728093946.1337642-1-james.clark@arm.com
    Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b2f10baf4d67e1a8c0ec52643c20d1895b0f749a
Author: Yu Kuai <yukuai3@huawei.com>
Date:   Tue Jul 26 20:22:24 2022 +0800

    blk-mq: fix io hung due to missing commit_rqs

    commit 65fac0d54f374625b43a9d6ad1f2c212bd41f518 upstream.

    Currently, in virtio_scsi, if 'bd->last' is not set to true while
    dispatching request, such io will stay in driver's queue, and driver
    will wait for block layer to dispatch more rqs. However, if block
    layer failed to dispatch more rq, it should trigger commit_rqs to
```

```
    inform driver.

    There is a problem in blk_mq_try_issue_list_directly() that commit_rqs
    won't be called:

    // assume that queue_depth is set to 1, list contains two rq
    blk_mq_try_issue_list_directly
     blk_mq_request_issue_directly
     // dispatch first rq
     // last is false
      __blk_mq_try_issue_directly
       blk_mq_get_dispatch_budget
       // succeed to get first budget
       __blk_mq_issue_directly
        scsi_queue_rq
         cmd->flags |= SCMD_LAST
          virtscsi_queuecommand
           kick = (sc->flags & SCMD_LAST) != 0
           // kick is false, first rq won't issue to disk
     queued++

     blk_mq_request_issue_directly
     // dispatch second rq
      __blk_mq_try_issue_directly
       blk_mq_get_dispatch_budget
       // failed to get second budget
     ret == BLK_STS_RESOURCE
      blk_mq_request_bypass_insert
     // errors is still 0

     if (!list_empty(list) || errors && ...)
      // won't pass, commit_rqs won't be called

    In this situation, first rq relied on second rq to dispatch, while
    second rq relied on first rq to complete, thus they will both hung.

    Fix the problem by also treat 'BLK_STS_*RESOURCE' as 'errors' since
    it means that request is not queued successfully.

    Same problem exists in blk_mq_dispatch_rq_list(), 'BLK_STS_*RESOURCE'
    can't be treated as 'errors' here, fix the problem by calling
    commit_rqs if queue_rq return 'BLK_STS_*RESOURCE'.

    Fixes: d666ba98f849 ("blk-mq: add mq_ops->commit_rqs()")
    Signed-off-by: Yu Kuai <yukuai3@huawei.com>
    Reviewed-by: Ming Lei <ming.lei@redhat.com>
    Link: https://lore.kernel.org/r/20220726122224.1790882-1-yukuai1@huaweicloud.com
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ca949183c3407a790100ac1d9fc10821a5fd887f
Author: Salvatore Bonaccorso <carnil@debian.org>
Date:   Mon Aug 1 11:15:30 2022 +0200

    Documentation/ABI: Mention retbleed vulnerability info file for sysfs

    commit 00da0cb385d05a89226e150a102eb49d8abb0359 upstream.

    While reporting for the AMD retbleed vulnerability was added in

      6b80b59b3555 ("x86/bugs: Report AMD retbleed vulnerability")

    the new sysfs file was not mentioned so far in the ABI documentation for
    sysfs-devices-system-cpu. Fix that.
```

```
    Fixes: 6b80b59b3555 ("x86/bugs: Report AMD retbleed vulnerability")
    Signed-off-by: Salvatore Bonaccorso <carnil@debian.org>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Link: https://lore.kernel.org/r/20220801091529.325327-1-carnil@debian.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 43365c8fbb3ca6d60ecb32b5c0f91e1563dd0ac1
Author: Prike Liang <Prike.Liang@amd.com>
Date:   Wed Aug 24 11:16:51 2022 +0800

    drm/amdkfd: Fix isa version for the GC 10.3.7

    commit ee8086dbc1585d9f4020a19447388246a5cff5c8 upstream.

    Correct the isa version for handling KFD test.

    Fixes: 7c4f4f197e0c ("drm/amdkfd: Add GC 10.3.6 and 10.3.7 KFD definitions")
    Signed-off-by: Prike Liang <Prike.Liang@amd.com>
    Reviewed-by: Aaron Liu <aaron.liu@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b864bc2ad49f413d670888abd737b2b5da3e5310
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Fri Aug 19 13:01:35 2022 +0200

    x86/nospec: Fix i386 RSB stuffing

    commit 332924973725e8cdcc783c175f68cf7e162cb9e5 upstream.

    Turns out that i386 doesn't unconditionally have LFENCE, as such the
    loop in __FILL_RETURN_BUFFER isn't actually speculation safe on such
    chips.

    Fixes: ba6e31af2be9 ("x86/speculation: Add LFENCE to RSB fill sequence")
    Reported-by: Ben Hutchings <ben@decadent.org.uk>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Link: https://lkml.kernel.org/r/Yv9tj9vbQ9nNlXoY@worktop.programming.kicks-ass.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7b0163c1b07b7ff1717aa975821c40df98786ddc
Author: Liam Howlett <liam.howlett@oracle.com>
Date:   Wed Aug 10 16:02:25 2022 +0000

    binder_alloc: add missing mmap_lock calls when using the VMA

    commit 44e602b4e52f70f04620bbbf4fe46ecb40170bde upstream.

    Take the mmap_read_lock() when using the VMA in binder_alloc_print_pages()
    and when checking for a VMA in binder_alloc_new_buf_locked().

    It is worth noting binder_alloc_new_buf_locked() drops the VMA read lock
    after it verifies a VMA exists, but may be taken again deeper in the call
    stack, if necessary.

    Link: https://lkml.kernel.org/r/20220810160209.1630707-1-Liam.Howlett@oracle.com
    Fixes: a43cfc87caaf (android: binder: stop saving a pointer to the VMA)
    Signed-off-by: Liam R. Howlett <Liam.Howlett@oracle.com>
    Reported-by: Ondrej Mosnacek <omosnace@redhat.com>
    Reported-by: <syzbot+a7b60a176ec13cafb793@syzkaller.appspotmail.com>
    Acked-by: Carlos Llamas <cmllamas@google.com>
    Tested-by: Ondrej Mosnacek <omosnace@redhat.com>
    Cc: Minchan Kim <minchan@kernel.org>
    Cc: Christian Brauner (Microsoft) <brauner@kernel.org>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
    Cc: Hridya Valsaraju <hridya@google.com>
    Cc: Joel Fernandes <joel@joelfernandes.org>
    Cc: Martijn Coenen <maco@android.com>
    Cc: Suren Baghdasaryan <surenb@google.com>
    Cc: Todd Kjos <tkjos@android.com>
    Cc: Matthew Wilcox (Oracle) <willy@infradead.org>
    Cc: "Arve Hjønnevåg" <arve@android.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b887868c4e6b9e8094909f3874444048345fce8a
Author: Zenghui Yu <yuzenghui@huawei.com>
Date:   Tue Aug 9 12:38:48 2022 +0800

    arm64: Fix match_list for erratum 1286807 on Arm Cortex-A76

    commit 5e1e087457c94ad7fafbe1cf6f774c6999ee29d4 upstream.

    Since commit 51f559d66527 ("arm64: Enable repeat tlbi workaround on KRYO4XX
    gold CPUs"), we failed to detect erratum 1286807 on Cortex-A76 because its
    entry in arm64_repeat_tlbi_list[] was accidently corrupted by this commit.

    Fix this issue by creating a separate entry for Kryo4xx Gold.

    Fixes: 51f559d66527 ("arm64: Enable repeat tlbi workaround on KRYO4XX gold CPUs")
    Cc: Shreyas K K <quic_shrekk@quicinc.com>
    Signed-off-by: Zenghui Yu <yuzenghui@huawei.com>
    Acked-by: Marc Zyngier <maz@kernel.org>
    Link: https://lore.kernel.org/r/20220809043848.969-1-yuzenghui@huawei.com
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f42a9819ba84bed2e609a4dff56af37063dcabdc
Author: Guoqing Jiang <guoqing.jiang@linux.dev>
Date:   Wed Aug 17 20:05:14 2022 +0800

    md: call __md_stop_writes in md_stop

    commit 0dd84b319352bb8ba64752d4e45396d8b13e6018 upstream.

    From the link [1], we can see raid1d was running even after the path
    raid_dtr -> md_stop -> __md_stop.

    Let's stop write first in destructor to align with normal md-raid to
    fix the KASAN issue.

    [1]. https://lore.kernel.org/linux-
raid/CAPhsuW5gc4AakdGNdF8ubpezAuDLFOYUO_sfMZcec6hQFm8nhg@mail.gmail.com/T/#m7f12bf90481c02c6
d2da68c64aeed4779b7df74a

    Fixes: 48df498daf62 ("md: move bitmap_destroy to the beginning of __md_stop")
    Reported-by: Mikulas Patocka <mpatocka@redhat.com>
    Signed-off-by: Guoqing Jiang <guoqing.jiang@linux.dev>
    Signed-off-by: Song Liu <song@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4d83d9b7d5ddbbabfd62af393a02c40ddd2a03db
Author: Guoqing Jiang <guoqing.jiang@linux.dev>
Date:   Wed Aug 17 20:05:13 2022 +0800

    Revert "md-raid: destroy the bitmap after destroying the thread"

    commit 1d258758cf06a0734482989911d184dd5837ed4e upstream.

    This reverts commit e151db8ecfb019b7da31d076130a794574c89f6f. Because it
```

```
    obviously breaks clustered raid as noticed by Neil though it fixed KASAN
    issue for dm-raid, let's revert it and fix KASAN issue in next commit.

    [1]. https://lore.kernel.org/linux-raid/a6657e08-b6a7-358b-2d2a-
0ac37d49d23a@linux.dev/T/#m95ac225cab7409f66c295772483d091084a6d470

    Fixes: e151db8ecfb0 ("md-raid: destroy the bitmap after destroying the thread")
    Signed-off-by: Guoqing Jiang <guoqing.jiang@linux.dev>
    Signed-off-by: Song Liu <song@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ba8da1806c4f24be1a0c5ab645b5c92864eab919
Author: David Hildenbrand <david@redhat.com>
Date:   Thu Aug 11 12:34:34 2022 +0200

    mm/hugetlb: fix hugetlb not supporting softdirty tracking

    commit f96f7a40874d7c746680c0b9f57cef2262ae551f upstream.

    Patch series "mm/hugetlb: fix write-fault handling for shared mappings", v2.

    I observed that hugetlb does not support/expect write-faults in shared
    mappings that would have to map the R/O-mapped page writable -- and I
    found two case where we could currently get such faults and would
    erroneously map an anon page into a shared mapping.

    Reproducers part of the patches.

    I propose to backport both fixes to stable trees.  The first fix needs a
    small adjustment.


    This patch (of 2):

    Staring at hugetlb_wp(), one might wonder where all the logic for shared
    mappings is when stumbling over a write-protected page in a shared
    mapping.  In fact, there is none, and so far we thought we could get away
    with that because e.g., mprotect() should always do the right thing and
    map all pages directly writable.

    Looks like we were wrong:

    ---------------------------------------------------------------------------
     #include <stdio.h>
     #include <stdlib.h>
     #include <string.h>
     #include <fcntl.h>
     #include <unistd.h>
     #include <errno.h>
     #include <sys/mman.h>

     #define HUGETLB_SIZE (2 * 1024 * 1024u)

     static void clear_softdirty(void)
     {
             int fd = open("/proc/self/clear_refs", O_WRONLY);
             const char *ctrl = "4";
             int ret;

             if (fd < 0) {
                     fprintf(stderr, "open(clear_refs) failed\n");
                     exit(1);
             }
             ret = write(fd, ctrl, strlen(ctrl));
             if (ret != strlen(ctrl)) {
```

```
                fprintf(stderr, "write(clear_refs) failed\n");
                exit(1);
        }
        close(fd);
 }

 int main(int argc, char **argv)
 {
        char *map;
        int fd;

        fd = open("/dev/hugepages/tmp", O_RDWR | O_CREAT);
        if (!fd) {
                fprintf(stderr, "open() failed\n");
                return -errno;
        }
        if (ftruncate(fd, HUGETLB_SIZE)) {
                fprintf(stderr, "ftruncate() failed\n");
                return -errno;
        }

        map = mmap(NULL, HUGETLB_SIZE, PROT_READ|PROT_WRITE, MAP_SHARED, fd, 0);
        if (map == MAP_FAILED) {
                fprintf(stderr, "mmap() failed\n");
                return -errno;
        }

        *map = 0;

        if (mprotect(map, HUGETLB_SIZE, PROT_READ)) {
                fprintf(stderr, "mmprotect() failed\n");
                return -errno;
        }

        clear_softdirty();

        if (mprotect(map, HUGETLB_SIZE, PROT_READ|PROT_WRITE)) {
                fprintf(stderr, "mmprotect() failed\n");
                return -errno;
        }

        *map = 0;

        return 0;
 }
-------------------------------------------------------------------------

Above test fails with SIGBUS when there is only a single free hugetlb page.
 # echo 1 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
 # ./test
 Bus error (core dumped)

And worse, with sufficient free hugetlb pages it will map an anonymous page
into a shared mapping, for example, messing up accounting during unmap
and breaking MAP_SHARED semantics:
 # echo 2 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
 # ./test
 # cat /proc/meminfo | grep HugePages_
 HugePages_Total:       2
 HugePages_Free:        1
 HugePages_Rsvd:    18446744073709551615
 HugePages_Surp:        0

Reason in this particular case is that vma_wants_writenotify() will
return "true", removing VM_SHARED in vma_set_page_prot() to map pages
```

```
    write-protected. Let's teach vma_wants_writenotify() that hugetlb does not
    support softdirty tracking.

    Link: https://lkml.kernel.org/r/20220811103435.188481-1-david@redhat.com
    Link: https://lkml.kernel.org/r/20220811103435.188481-2-david@redhat.com
    Fixes: 64e455079e1b ("mm: softdirty: enable write notifications on VMAs after
VM_SOFTDIRTY cleared")
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Peter Feiner <pfeiner@google.com>
    Cc: Kirill A. Shutemov <kirill.shutemov@linux.intel.com>
    Cc: Cyrill Gorcunov <gorcunov@openvz.org>
    Cc: Pavel Emelyanov <xemul@parallels.com>
    Cc: Jamie Liu <jamieliu@google.com>
    Cc: Hugh Dickins <hughd@google.com>
    Cc: Naoya Horiguchi <n-horiguchi@ah.jp.nec.com>
    Cc: Bjorn Helgaas <bhelgaas@google.com>
    Cc: Muchun Song <songmuchun@bytedance.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: <stable@vger.kernel.org>    [3.18+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5192d4ae17a563039876faae8a66e99a04bc1c34
Author: Jens Axboe <axboe@kernel.dk>
Date:   Thu Aug 25 10:17:25 2022 -0600

    io_uring: fix issue with io_write() not always undoing sb_start_write()

    commit e053aaf4da56cbf0afb33a0fda4a62188e2c0637 upstream.

    This is actually an older issue, but we never used to hit the -EAGAIN
    path before having done sb_start_write(). Make sure that we always call
    kiocb_end_write() if we need to retry the write, so that we keep the
    calls to sb_start_write() etc balanced.

    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e8f1d2fd811384b2f91043580b9b3c1c6eaef73d
Author: Jiri Slaby <jirislaby@kernel.org>
Date:   Wed Aug 10 09:06:09 2022 +0200

    Revert "zram: remove double compression logic"

    commit 37887783b3fef877bf34b8992c9199864da4afcb upstream.

    This reverts commit e7be8d1dd983156b ("zram: remove double compression
    logic") as it causes zram failures.  It does not revert cleanly, PTR_ERR
    handling was introduced in the meantime.  This is handled by appropriate
    IS_ERR.

    When under memory pressure, zs_malloc() can fail.  Before the above
    commit, the allocation was retried with direct reclaim enabled (GFP_NOIO).
    After the commit, it is not -- only __GFP_KSWAPD_RECLAIM is tried.

    So when the failure occurs under memory pressure, the overlaying
    filesystem such as ext2 (mounted by ext4 module in this case) can emit
    failures, making the (file)system unusable:
      EXT4-fs warning (device zram0): ext4_end_bio:343: I/O error 10 writing to inode 16386
starting block 159744)
      Buffer I/O error on device zram0, logical block 159744

    With direct reclaim, memory is really reclaimed and allocation succeeds,
```

```
    eventually.  In the worst case, the oom killer is invoked, which is proper
    outcome if user sets up zram too large (in comparison to available RAM).

    This very diff doesn't apply to 5.19 (stable) cleanly (see PTR_ERR note
    above). Use revert of e7be8d1dd983 directly.

    Link: https://bugzilla.suse.com/show_bug.cgi?id=1202203
    Link: https://lkml.kernel.org/r/20220810070609.14402-1-jslaby@suse.cz
    Fixes: e7be8d1dd983 ("zram: remove double compression logic")
    Signed-off-by: Jiri Slaby <jslaby@suse.cz>
    Reviewed-by: Sergey Senozhatsky <senozhatsky@chromium.org>
    Cc: Minchan Kim <minchan@kernel.org>
    Cc: Nitin Gupta <ngupta@vflare.org>
    Cc: Alexey Romanov <avromanov@sberdevices.ru>
    Cc: Dmitry Rokosov <ddrokosov@sberdevices.ru>
    Cc: Lukas Czerner <lczerner@redhat.com>
    Cc: <stable@vger.kernel.org>    [5.19]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c4ce7913dfd2a9cf567dbf70246e6b71934c3e5e
Author: Heinrich Schuchardt <heinrich.schuchardt@canonical.com>
Date:   Wed Aug 17 15:25:21 2022 +0200

    riscv: dts: microchip: correct L2 cache interrupts

    commit 34fc9cc3aebe8b9e27d3bc821543dd482dc686ca upstream.

    The "PolarFire SoC MSS Technical Reference Manual" documents the
    following PLIC interrupts:

    1 - L2 Cache Controller Signals when a metadata correction event occurs
    2 - L2 Cache Controller Signals when an uncorrectable metadata event occurs
    3 - L2 Cache Controller Signals when a data correction event occurs
    4 - L2 Cache Controller Signals when an uncorrectable data event occurs

    This differs from the SiFive FU540 which only has three L2 cache related
    interrupts.

    The sequence in the device tree is defined by an enum:

        enum {
                DIR_CORR = 0,
                DATA_CORR,
                DATA_UNCORR,
                DIR_UNCORR,
        };

    So the correct sequence of the L2 cache interrupts is

        interrupts = <1>, <3>, <4>, <2>;

    [Conor]
    This manifests as an unusable system if the l2-cache driver is enabled,
    as the wrong interrupt gets cleared & the handler prints errors to the
    console ad infinitum.

    Fixes: 0fa6107eca41 ("RISC-V: Initial DTS for Microchip ICICLE board")
    CC: stable@vger.kernel.org # 5.15: e35b07a7df9b: riscv: dts: microchip: mpfs: Group
tuples in interrupt properties
    Signed-off-by: Heinrich Schuchardt <heinrich.schuchardt@canonical.com>
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8e86aef0a601bc9731c38d4a5b3f0ee5aa99b2d
```

```
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sun Aug 14 15:12:38 2022 +0100

    riscv: traps: add missing prototype

    commit d951b20b9def73dcc39a5379831525d0d2a537e9 upstream.

    Sparse complains:
    arch/riscv/kernel/traps.c:213:6: warning: symbol 'shadow_stack' was not declared. Should
it be static?

    The variable is used in entry.S, so declare shadow_stack there
    alongside SHADOW_OVERFLOW_STACK_SIZE.

    Fixes: 31da94c25aea ("riscv: add VMAP_STACK overflow detection")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20220814141237.493457-5-mail@conchuod.ie
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f80d72069ede35765d4eb738c855d2cfed734f9a
Author: Conor Dooley <conor.dooley@microchip.com>
Date:   Sun Aug 14 15:12:37 2022 +0100

    riscv: signal: fix missing prototype warning

    commit b5c3aca86d2698c4850b6ee8b341938025d2780c upstream.

    Fix the warning:
    arch/riscv/kernel/signal.c:316:27: warning: no previous prototype for function
'do_notify_resume' [-Wmissing-prototypes]
    asmlinkage __visible void do_notify_resume(struct pt_regs *regs,

    All other functions in the file are static & none of the existing
    headers stood out as an obvious location. Create signal.h to hold the
    declaration.

    Fixes: e2c0cdfba7f6 ("RISC-V: User-facing API")
    Signed-off-by: Conor Dooley <conor.dooley@microchip.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20220814141237.493457-4-mail@conchuod.ie
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 45d47bd9b96e7874b98dbcc7602fe2826c5d62a6
Author: Juergen Gross <jgross@suse.com>
Date:   Thu Aug 25 16:19:18 2022 +0200

    xen/privcmd: fix error exit of privcmd_ioctl_dm_op()

    commit c5deb27895e017a0267de0a20d140ad5fcc55a54 upstream.

    The error exit of privcmd_ioctl_dm_op() is calling unlock_pages()
    potentially with pages being NULL, leading to a NULL dereference.

    Additionally lock_pages() doesn't check for pin_user_pages_fast()
    having been completely successful, resulting in potentially not
    locking all pages into memory. This could result in sporadic failures
    when using the related memory in user mode.

    Fix all of that by calling unlock_pages() always with the real number
    of pinned pages, which will be zero in case pages being NULL, and by
    checking the number of pages pinned by pin_user_pages_fast() matching
    the expected number of pages.
```

```
    Cc: <stable@vger.kernel.org>
    Fixes: ab520be8cd5d ("xen/privcmd: Add IOCTL_PRIVCMD_DM_OP")
    Reported-by: Rustam Subkhankulov <subkhankulov@ispras.ru>
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Reviewed-by: Jan Beulich <jbeulich@suse.com>
    Reviewed-by: Oleksandr Tyshchenko <oleksandr_tyshchenko@epam.com>
    Link: https://lore.kernel.org/r/20220825141918.3581-1-jgross@suse.com
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f377ac7597ba6a631ed98888e8027f9a7b2dbe7e
Author: Heming Zhao <ocfs2-devel@oss.oracle.com>
Date:   Mon Aug 15 16:57:54 2022 +0800

    ocfs2: fix freeing uninitialized resource on ocfs2_dlm_shutdown

    commit 550842cc60987b269e31b222283ade3e1b6c7fc8 upstream.

    After commit 0737e01de9c4 ("ocfs2: ocfs2_mount_volume does cleanup job
    before return error"), any procedure after ocfs2_dlm_init() fails will
    trigger crash when calling ocfs2_dlm_shutdown().

    ie: On local mount mode, no dlm resource is initialized.  If
    ocfs2_mount_volume() fails in ocfs2_find_slot(), error handling will call
    ocfs2_dlm_shutdown(), then does dlm resource cleanup job, which will
    trigger kernel crash.

    This solution should bypass uninitialized resources in
    ocfs2_dlm_shutdown().

    Link: https://lkml.kernel.org/r/20220815085754.20417-1-heming.zhao@suse.com
    Fixes: 0737e01de9c4 ("ocfs2: ocfs2_mount_volume does cleanup job before return error")
    Signed-off-by: Heming Zhao <heming.zhao@suse.com>
    Reviewed-by: Joseph Qi <joseph.qi@linux.alibaba.com>
    Cc: Mark Fasheh <mark@fasheh.com>
    Cc: Joel Becker <jlbec@evilplan.org>
    Cc: Junxiao Bi <junxiao.bi@oracle.com>
    Cc: Changwei Ge <gechangwei@live.cn>
    Cc: Gang He <ghe@suse.com>
    Cc: Jun Piao <piaojun@huawei.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a25f09216071fe49cf453746f04785be538d1234
Author: David Howells <dhowells@redhat.com>
Date:   Tue Aug 23 02:10:56 2022 -0500

    smb3: missing inode locks in punch hole

    commit ba0803050d610d5072666be727bca5e03e55b242 upstream.

    smb3 fallocate punch hole was not grabbing the inode or filemap_invalidate
    locks so could have race with pagemap reinstantiating the page.

    Cc: stable@vger.kernel.org
    Signed-off-by: David Howells <dhowells@redhat.com>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8e3ba23a67de984f4156f0663f1f603ff6c15815
Author: Karol Herbst <kherbst@redhat.com>
Date:   Fri Aug 19 22:09:28 2022 +0200
```

```
    nouveau: explicitly wait on the fence in nouveau_bo_move_m2mf

    commit 6b04ce966a738ecdd9294c9593e48513c0dc90aa upstream.

    It is a bit unlcear to us why that's helping, but it does and unbreaks
    suspend/resume on a lot of GPUs without any known drawbacks.

    Cc: stable@vger.kernel.org # v5.15+
    Closes: https://gitlab.freedesktop.org/drm/nouveau/-/issues/156
    Signed-off-by: Karol Herbst <kherbst@redhat.com>
    Reviewed-by: Lyude Paul <lyude@redhat.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20220819200928.401416-1-
kherbst@redhat.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f1a7466258b7fbb171728e0efabaef038ed1e1e6
Author: Riwen Lu <luriwen@kylinos.cn>
Date:   Tue Aug 23 15:43:42 2022 +0800

    ACPI: processor: Remove freq Qos request for all CPUs

    commit 36527b9d882362567ceb4eea8666813280f30e6f upstream.

    The freq Qos request would be removed repeatedly if the cpufreq policy
    relates to more than one CPU. Then, it would cause the "called for unknown
    object" warning.

    Remove the freq Qos request for each CPU relates to the cpufreq policy,
    instead of removing repeatedly for the last CPU of it.

    Fixes: a1bb46c36ce3 ("ACPI: processor: Add QoS requests for all CPUs")
    Reported-by: Jeremy Linton <Jeremy.Linton@arm.com>
    Tested-by: Jeremy Linton <jeremy.linton@arm.com>
    Signed-off-by: Riwen Lu <luriwen@kylinos.cn>
    Cc: 5.4+ <stable@vger.kernel.org> # 5.4+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c061d697a304cc652a21eae4c252299de7e28cc5
Author: Matthew Wilcox (Oracle) <willy@infradead.org>
Date:   Sat Jul 30 05:25:18 2022 +0100

    shmem: update folio if shmem_replace_page() updates the page

    commit 9dfb3b8d655022760ca68af11821f1c63aa547c3 upstream.

    If we allocate a new page, we need to make sure that our folio matches
    that new page.

    If we do end up in this code path, we store the wrong page in the shmem
    inode's page cache, and I would rather imagine that data corruption
    ensues.

    This will be solved by changing shmem_replace_page() to
    shmem_replace_folio(), but this is the minimal fix.

    Link: https://lkml.kernel.org/r/20220730042518.1264767-1-willy@infradead.org
    Fixes: da08e9b79323 ("mm/shmem: convert shmem_swapin_page() to shmem_swapin_folio()")
    Signed-off-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Reviewed-by: William Kucharski <william.kucharski@oracle.com>
    Cc: Hugh Dickins <hughd@google.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 5f4d2b0caf2063e8b2560bf39be9c39443b3e91e
Author: Shakeel Butt <shakeelb@google.com>
Date:   Wed Aug 17 17:21:39 2022 +0000

    Revert "memcg: cleanup racy sum avoidance code"

    commit dbb16df6443c59e8a1ef21c2272fcf387d600ddf upstream.

    This reverts commit 96e51ccf1af33e82f429a0d6baebba29c6448d0f.

    Recently we started running the kernel with rstat infrastructure on
    production traffic and begin to see negative memcg stats values.
    Particularly the 'sock' stat is the one which we observed having negative
    value.

    $ grep "sock " /mnt/memory/job/memory.stat
    sock 253952
    total_sock 18446744073708724224

    Re-run after couple of seconds

    $ grep "sock " /mnt/memory/job/memory.stat
    sock 253952
    total_sock 53248

    For now we are only seeing this issue on large machines (256 CPUs) and
    only with 'sock' stat.  I think the networking stack increase the stat on
    one cpu and decrease it on another cpu much more often.  So, this negative
    sock is due to rstat flusher flushing the stats on the CPU that has seen
    the decrement of sock but missed the CPU that has increments.  A typical
    race condition.

    For easy stable backport, revert is the most simple solution.  For long
    term solution, I am thinking of two directions.  First is just reduce the
    race window by optimizing the rstat flusher.  Second is if the reader sees
    a negative stat value, force flush and restart the stat collection.
    Basically retry but limited.

    Link: https://lkml.kernel.org/r/20220817172139.3141101-1-shakeelb@google.com
    Fixes: 96e51ccf1af33e8 ("memcg: cleanup racy sum avoidance code")
    Signed-off-by: Shakeel Butt <shakeelb@google.com>
    Cc: "Michal Koutný" <mkoutny@suse.com>
    Cc: Johannes Weiner <hannes@cmpxchg.org>
    Cc: Michal Hocko <mhocko@kernel.org>
    Cc: Roman Gushchin <roman.gushchin@linux.dev>
    Cc: Muchun Song <songmuchun@bytedance.com>
    Cc: David Hildenbrand <david@redhat.com>
    Cc: Yosry Ahmed <yosryahmed@google.com>
    Cc: Greg Thelen <gthelen@google.com>
    Cc: <stable@vger.kernel.org>    [5.15]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f08ccb792d3eaf1dc62d8cbf6a30d6522329f660
Author: Shigeru Yoshida <syoshida@redhat.com>
Date:   Fri Aug 19 03:13:36 2022 +0900

    fbdev: fbcon: Properly revert changes when vc_resize() failed

    commit a5a923038d70d2d4a86cb4e3f32625a5ee6e7e24 upstream.

    fbcon_do_set_font() calls vc_resize() when font size is changed.
    However, if if vc_resize() failed, current implementation doesn't
    revert changes for font size, and this causes inconsistent state.
```

```
    syzbot reported unable to handle page fault due to this issue [1].
    syzbot's repro uses fault injection which cause failure for memory
    allocation, so vc_resize() failed.

    This patch fixes this issue by properly revert changes for font
    related date when vc_resize() failed.

    Link: https://syzkaller.appspot.com/bug?id=3443d3a1fa6d964dd7310a0cb1696d165a3e07c4 [1]
    Reported-by: syzbot+a168dbeaaa7778273c1b@syzkaller.appspotmail.com
    Signed-off-by: Shigeru Yoshida <syoshida@redhat.com>
    Signed-off-by: Helge Deller <deller@gmx.de>
    CC: stable@vger.kernel.org # 5.15+
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fbdc482d43eda40a70de4b0155843d5472f6de62
Author: Brian Foster <bfoster@redhat.com>
Date:   Tue Aug 16 11:54:07 2022 -0400

    s390: fix double free of GS and RI CBs on fork() failure

    commit 13cccafe0edcd03bf1c841de8ab8a1c8e34f77d9 upstream.

    The pointers for guarded storage and runtime instrumentation control
    blocks are stored in the thread_struct of the associated task. These
    pointers are initially copied on fork() via arch_dup_task_struct()
    and then cleared via copy_thread() before fork() returns. If fork()
    happens to fail after the initial task dup and before copy_thread(),
    the newly allocated task and associated thread_struct memory are
    freed via free_task() -> arch_release_task_struct(). This results in
    a double free of the guarded storage and runtime info structs
    because the fields in the failed task still refer to memory
    associated with the source task.

    This problem can manifest as a BUG_ON() in set_freepointer() (with
    CONFIG_SLAB_FREELIST_HARDENED enabled) or KASAN splat (if enabled)
    when running trinity syscall fuzz tests on s390x. To avoid this
    problem, clear the associated pointer fields in
    arch_dup_task_struct() immediately after the new task is copied.
    Note that the RI flag is still cleared in copy_thread() because it
    resides in thread stack memory and that is where stack info is
    copied.

    Signed-off-by: Brian Foster <bfoster@redhat.com>
    Fixes: 8d9047f8b967c ("s390/runtime instrumentation: simplify task exit handling")
    Fixes: 7b83c6297d2fc ("s390/guarded storage: simplify task exit handling")
    Cc: <stable@vger.kernel.org> # 4.15
    Reviewed-by: Gerald Schaefer <gerald.schaefer@linux.ibm.com>
    Reviewed-by: Heiko Carstens <hca@linux.ibm.com>
    Link: https://lore.kernel.org/r/20220816155407.537372-1-bfoster@redhat.com
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bb125123f60ea05211d4b3e5ff9dfa7e9ddd43ab
Author: Paulo Alcantara <pc@cjr.nz>
Date:   Fri Aug 19 17:00:19 2022 -0300

    cifs: skip extra NULL byte in filenames

    commit a1d2eb51f0a33c28f5399a1610e66b3fbd24e884 upstream.

    Since commit:
     cifs: alloc_path_with_tree_prefix: do not append sep. if the path is empty
    alloc_path_with_tree_prefix() function was no longer including the
    trailing separator when @path is empty, although @out_len was still
    assuming a path separator thus adding an extra byte to the final
```

```
    filename.

    This has caused mount issues in some Synology servers due to the extra
    NULL byte in filenames when sending SMB2_CREATE requests with
    SMB2_FLAGS_DFS_OPERATIONS set.

    Fix this by checking if @path is not empty and then add extra byte for
    separator.  Also, do not include any trailing NULL bytes in filename
    as MS-SMB2 requires it to be 8-byte aligned and not NULL terminated.

    Cc: stable@vger.kernel.org
    Fixes: 7eacba3b00a3 ("cifs: alloc_path_with_tree_prefix: do not append sep. if the path
is empty")
    Signed-off-by: Paulo Alcantara (SUSE) <pc@cjr.nz>
    Signed-off-by: Steve French <stfrench@microsoft.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5fcf81e308d1f4ae95f31690d2a80b7061385ff9
Author: Peter Xu <peterx@redhat.com>
Date:   Tue Aug 23 18:11:38 2022 -0400

    mm/mprotect: only reference swap pfn page if type match

    commit 3d2f78f08cd8388035ac375e731ec1ac1b79b09d upstream.

    Yu Zhao reported a bug after the commit "mm/swap: Add swp_offset_pfn() to
    fetch PFN from swap entry" added a check in swp_offset_pfn() for swap type [1]:

      kernel BUG at include/linux/swapops.h:117!
      CPU: 46 PID: 5245 Comm: EventManager_De Tainted: G S         O L 6.0.0-dbg-DEV #2
      RIP: 0010:pfn_swap_entry_to_page+0x72/0xf0
      Code: c6 48 8b 36 48 83 fe ff 74 53 48 01 d1 48 83 c1 08 48 8b 09 f6
      c1 01 75 7b 66 90 48 89 c1 48 8b 09 f6 c1 01 74 74 5d c3 eb 9e <0f> 0b
      48 ba ff ff ff ff 03 00 00 00 eb ae a9 ff 0f 00 00 75 13 48
      RSP: 0018:ffffa59e73fabb80 EFLAGS: 00010282
      RAX: 00000000ffffffe8 RBX: 0c00000000000000 RCX: ffffcd5440000000
      RDX: 1fffffffff7a80a RSI: 0000000000000000 RDI: 0c0000000000042b
      RBP: ffffa59e73fabb80 R08: ffff9965ca6e8bb8 R09: 0000000000000000
      R10: ffffffffa5a2f62d R11: 0000030b372e9fff R12: ffff997b79db5738
      R13: 000000000000042b R14: 0c0000000000042b R15: 1fffffffff7a80a
      FS:  00007f549d1bb700(0000) GS:ffff99d3cf680000(0000) knlGS:0000000000000000
      CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
      CR2: 0000440d035b3180 CR3: 0000002243176004 CR4: 00000000003706e0
      DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
      DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
      Call Trace:
       <TASK>
       change_pte_range+0x36e/0x880
       change_p4d_range+0x2e8/0x670
       change_protection_range+0x14e/0x2c0
       mprotect_fixup+0x1ee/0x330
       do_mprotect_pkey+0x34c/0x440
       __x64_sys_mprotect+0x1d/0x30

    It triggers because pfn_swap_entry_to_page() could be called upon e.g. a
    genuine swap entry.

    Fix it by only calling it when it's a write migration entry where the page*
    is used.

    [1] https://lore.kernel.org/lkml/CAOUHufaVC2Za-p8m0aiHw6YkheDcrO-
C3wRGixwDS32VTS+k1w@mail.gmail.com/

    Link: https://lkml.kernel.org/r/20220823221138.45602-1-peterx@redhat.com
    Fixes: 6c287605fd56 ("mm: remember exclusively mapped anonymous pages with
```

```
PG_anon_exclusive")
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Reported-by: Yu Zhao <yuzhao@google.com>
    Tested-by: Yu Zhao <yuzhao@google.com>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Cc: "Huang, Ying" <ying.huang@intel.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3ada1b3e58db255a14ec73a59d7913e84dc5a8a4
Author: Miaohe Lin <linmiaohe@huawei.com>
Date:   Tue Jul 12 21:05:42 2022 +0800

    mm/hugetlb: avoid corrupting page->mapping in hugetlb_mcopy_atomic_pte

    commit ab74ef708dc51df7cf2b8a890b9c6990fac5c0c6 upstream.

    In MCOPY_ATOMIC_CONTINUE case with a non-shared VMA, pages in the page
    cache are installed in the ptes.  But hugepage_add_new_anon_rmap is called
    for them mistakenly because they're not vm_shared.  This will corrupt the
    page->mapping used by page cache code.

    Link: https://lkml.kernel.org/r/20220712130542.18836-1-linmiaohe@huawei.com
    Fixes: f619147104c8 ("userfaultfd: add UFFDIO_CONTINUE ioctl")
    Signed-off-by: Miaohe Lin <linmiaohe@huawei.com>
    Reviewed-by: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9ae15c4ba2be1e5a62503b6d873e84beb5fcbb5a
Author: Liu Shixin <liushixin2@huawei.com>
Date:   Fri Aug 19 17:40:05 2022 +0800

    bootmem: remove the vmemmap pages from kmemleak in put_page_bootmem

    commit dd0ff4d12dd284c334f7e9b07f8f335af856ac78 upstream.

    The vmemmap pages is marked by kmemleak when allocated from memblock.
    Remove it from kmemleak when freeing the page.  Otherwise, when we reuse
    the page, kmemleak may report such an error and then stop working.

     kmemleak: Cannot insert 0xffff98fb6eab3d40 into the object search tree (overlaps
existing)
     kmemleak: Kernel memory leak detector disabled
     kmemleak: Object 0xffff98fb6be00000 (size 335544320):
     kmemleak:   comm "swapper", pid 0, jiffies 4294892296
     kmemleak:   min_count = 0
     kmemleak:   count = 0
     kmemleak:   flags = 0x1
     kmemleak:   checksum = 0
     kmemleak:   backtrace:

    Link: https://lkml.kernel.org/r/20220819094005.2928241-1-liushixin2@huawei.com
    Fixes: f41f2ed43ca5 (mm: hugetlb: free the vmemmap pages associated with each HugeTLB
page)
    Signed-off-by: Liu Shixin <liushixin2@huawei.com>
    Reviewed-by: Muchun Song <songmuchun@bytedance.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Oscar Salvador <osalvador@suse.de>
    Cc: <stable@vger.kernel.org>
```

```
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8eaa24d57ab6a3f95be50c947a885f983869e8cb
Author: Gerald Schaefer <gerald.schaefer@linux.ibm.com>
Date:   Wed Aug 17 15:26:03 2022 +0200

    s390/mm: do not trigger write fault when vma does not allow VM_WRITE

    commit 41ac42f137080bc230b5882e3c88c392ab7f2d32 upstream.

    For non-protection pXd_none() page faults in do_dat_exception(), we
    call do_exception() with access == (VM_READ | VM_WRITE | VM_EXEC).
    In do_exception(), vma->vm_flags is checked against that before
    calling handle_mm_fault().

    Since commit 92f842eac7ee3 ("[S390] store indication fault optimization"),
    we call handle_mm_fault() with FAULT_FLAG_WRITE, when recognizing that
    it was a write access. However, the vma flags check is still only
    checking against (VM_READ | VM_WRITE | VM_EXEC), and therefore also
    calling handle_mm_fault() with FAULT_FLAG_WRITE in cases where the vma
    does not allow VM_WRITE.

    Fix this by changing access check in do_exception() to VM_WRITE only,
    when recognizing write access.

    Link: https://lkml.kernel.org/r/20220811103435.188481-3-david@redhat.com
    Fixes: 92f842eac7ee3 ("[S390] store indication fault optimization")
    Cc: <stable@vger.kernel.org>
    Reported-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Heiko Carstens <hca@linux.ibm.com>
    Signed-off-by: Gerald Schaefer <gerald.schaefer@linux.ibm.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c035edae0dad1aff599ce2d3ecb8d91d90ec5da0
Author: Badari Pulavarty <badari.pulavarty@intel.com>
Date:   Sun Aug 21 18:08:53 2022 +0000

    mm/damon/dbgfs: avoid duplicate context directory creation

    commit d26f60703606ab425eee9882b32a1781a8bed74d upstream.

    When user tries to create a DAMON context via the DAMON debugfs interface
    with a name of an already existing context, the context directory creation
    fails but a new context is created and added in the internal data
    structure, due to absence of the directory creation success check.  As a
    result, memory could leak and DAMON cannot be turned on.  An example test
    case is as below:

        # cd /sys/kernel/debug/damon/
        # echo "off" >  monitor_on
        # echo paddr > target_ids
        # echo "abc" > mk_context
        # echo "abc" > mk_context
        # echo $$ > abc/target_ids
        # echo "on" > monitor_on  <<< fails

    Return value of 'debugfs_create_dir()' is expected to be ignored in
    general, but this is an exceptional case as DAMON feature is depending
    on the debugfs functionality and it has the potential duplicate name
    issue.  This commit therefore fixes the issue by checking the directory
    creation failure and immediately return the error in the case.

    Link: https://lkml.kernel.org/r/20220821180853.2400-1-sj@kernel.org
```

```
     Fixes: 75c1c2b53c78 ("mm/damon/dbgfs: support multiple contexts")
     Signed-off-by: Badari Pulavarty <badari.pulavarty@intel.com>
     Signed-off-by: SeongJae Park <sj@kernel.org>
     Cc: <stable@vger.kernel.org>    [ 5.15.x]
     Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit fe64e17d9b0120c6b1b02ec72ca5fc2d08cf1fcd
 Author: Quanyang Wang <quanyang.wang@windriver.com>
 Date:   Fri Aug 19 16:11:45 2022 +0800

     asm-generic: sections: refactor memory_intersects

     commit 0c7d7cc2b4fe2e74ef8728f030f0f1674f9f6aee upstream.

     There are two problems with the current code of memory_intersects:

     First, it doesn't check whether the region (begin, end) falls inside the
     region (virt, vend), that is (virt < begin && vend > end).

     The second problem is if vend is equal to begin, it will return true but
     this is wrong since vend (virt + size) is not the last address of the
     memory region but (virt + size -1) is.  The wrong determination will
     trigger the misreporting when the function check_for_illegal_area calls
     memory_intersects to check if the dma region intersects with stext region.

     The misreporting is as below (stext is at 0x80100000):
      WARNING: CPU: 0 PID: 77 at kernel/dma/debug.c:1073 check_for_illegal_area+0x130/0x168
      DMA-API: chipidea-usb2 e0002000.usb: device driver maps memory from kernel text or
 rodata [addr=800f0000] [len=65536]
      Modules linked in:
      CPU: 1 PID: 77 Comm: usb-storage Not tainted 5.19.0-yocto-standard #5
      Hardware name: Xilinx Zynq Platform
       unwind_backtrace from show_stack+0x18/0x1c
       show_stack from dump_stack_lvl+0x58/0x70
       dump_stack_lvl from __warn+0xb0/0x198
       __warn from warn_slowpath_fmt+0x80/0xb4
       warn_slowpath_fmt from check_for_illegal_area+0x130/0x168
       check_for_illegal_area from debug_dma_map_sg+0x94/0x368
       debug_dma_map_sg from __dma_map_sg_attrs+0x114/0x128
       __dma_map_sg_attrs from dma_map_sg_attrs+0x18/0x24
       dma_map_sg_attrs from usb_hcd_map_urb_for_dma+0x250/0x3b4
       usb_hcd_map_urb_for_dma from usb_hcd_submit_urb+0x194/0x214
       usb_hcd_submit_urb from usb_sg_wait+0xa4/0x118
       usb_sg_wait from usb_stor_bulk_transfer_sglist+0xa0/0xec
       usb_stor_bulk_transfer_sglist from usb_stor_bulk_srb+0x38/0x70
       usb_stor_bulk_srb from usb_stor_Bulk_transport+0x150/0x360
       usb_stor_Bulk_transport from usb_stor_invoke_transport+0x38/0x440
       usb_stor_invoke_transport from usb_stor_control_thread+0x1e0/0x238
       usb_stor_control_thread from kthread+0xf8/0x104
       kthread from ret_from_fork+0x14/0x2c

     Refactor memory_intersects to fix the two problems above.

     Before the 1d7db834a027e ("dma-debug: use memory_intersects()
     directly"), memory_intersects is called only by printk_late_init:

     printk_late_init -> init_section_intersects ->memory_intersects.

     There were few places where memory_intersects was called.

     When commit 1d7db834a027e ("dma-debug: use memory_intersects()
     directly") was merged and CONFIG_DMA_API_DEBUG is enabled, the DMA
     subsystem uses it to check for an illegal area and the calltrace above
     is triggered.
```

```
    [akpm@linux-foundation.org: fix nearby comment typo]
    Link: https://lkml.kernel.org/r/20220819081145.948016-1-quanyang.wang@windriver.com
    Fixes: 979559362516 ("asm/sections: add helpers to check for section data")
    Signed-off-by: Quanyang Wang <quanyang.wang@windriver.com>
    Cc: Ard Biesheuvel <ardb@kernel.org>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Cc: Thierry Reding <treding@nvidia.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 22ebb780d54ef1e9a1fdba41696ebf48ef99b96d
Author: Richard Guy Briggs <rgb@redhat.com>
Date:   Thu Aug 25 15:32:40 2022 -0400

    audit: move audit_return_fixup before the filters

    commit d4fefa4801a1c2f9c0c7a48fbb0fdf384e89a4ab upstream.

    The success and return_code are needed by the filters.  Move
    audit_return_fixup() before the filters.  This was causing syscall
    auditing events to be missed.

    Link: https://github.com/linux-audit/audit-kernel/issues/138
    Cc: stable@vger.kernel.org
    Fixes: 12c5e81d3fd0 ("audit: prepare audit_context for use in calling contexts beyond
syscalls")
    Signed-off-by: Richard Guy Briggs <rgb@redhat.com>
    [PM: manual merge required]
    Signed-off-by: Paul Moore <paul@paul-moore.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9a6c710f3bc10bc9cc23e1c080b53245b7f9d5b7
Author: Khazhismel Kumykov <khazhy@chromium.org>
Date:   Mon Aug 1 08:50:34 2022 -0700

    writeback: avoid use-after-free after removing device

    commit f87904c075515f3e1d8f4a7115869d3b914674fd upstream.

    When a disk is removed, bdi_unregister gets called to stop further
    writeback and wait for associated delayed work to complete.  However,
    wb_inode_writeback_end() may schedule bandwidth estimation dwork after
    this has completed, which can result in the timer attempting to access the
    just freed bdi_writeback.

    Fix this by checking if the bdi_writeback is alive, similar to when
    scheduling writeback work.

    Since this requires wb->work_lock, and wb_inode_writeback_end() may get
    called from interrupt, switch wb->work_lock to an irqsafe lock.

    Link: https://lkml.kernel.org/r/20220801155034.3772543-1-khazhy@google.com
    Fixes: 45a2966fd641 ("writeback: fix bandwidth estimate for spiky workload")
    Signed-off-by: Khazhismel Kumykov <khazhy@google.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Cc: Michael Stapelberg <stapelberg+linux@google.com>
    Cc: Wu Fengguang <fengguang.wu@intel.com>
    Cc: Alexander Viro <viro@zeniv.linux.org.uk>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9be7fa7ead18a48940df7b59d993bbc8b9055c15
```

```
Author: Siddh Raman Pant <code@siddh.me>
Date:   Tue Aug 23 21:38:10 2022 +0530

    loop: Check for overflow while configuring loop

    commit c490a0b5a4f36da3918181a8acdc6991d967c5f3 upstream.

    The userspace can configure a loop using an ioctl call, wherein
    a configuration of type loop_config is passed (see lo_ioctl()'s
    case on line 1550 of drivers/block/loop.c). This proceeds to call
    loop_configure() which in turn calls loop_set_status_from_info()
    (see line 1050 of loop.c), passing &config->info which is of type
    loop_info64*. This function then sets the appropriate values, like
    the offset.

    loop_device has lo_offset of type loff_t (see line 52 of loop.c),
    which is typdef-chained to long long, whereas loop_info64 has
    lo_offset of type __u64 (see line 56 of include/uapi/linux/loop.h).

    The function directly copies offset from info to the device as
    follows (See line 980 of loop.c):
            lo->lo_offset = info->lo_offset;

    This results in an overflow, which triggers a warning in iomap_iter()
    due to a call to iomap_iter_done() which has:
            WARN_ON_ONCE(iter->iomap.offset > iter->pos);

    Thus, check for negative value during loop_set_status_from_info().

    Bug report: https://syzkaller.appspot.com/bug?
id=c620fe14aac810396d3c3edc9ad73848bf69a29e

    Reported-and-tested-by: syzbot+a8e049cd3abd342936b6@syzkaller.appspotmail.com
    Cc: stable@vger.kernel.org
    Reviewed-by: Matthew Wilcox (Oracle) <willy@infradead.org>
    Signed-off-by: Siddh Raman Pant <code@siddh.me>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Link: https://lore.kernel.org/r/20220823160810.181275-1-code@siddh.me
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a210408b902465c20970c2abc1ef4391d1769cf6
Author: Jan Beulich <jbeulich@suse.com>
Date:   Thu Apr 28 16:50:29 2022 +0200

    x86/PAT: Have pat_enabled() properly reflect state when running on Xen

    commit 72cbc8f04fe2fa93443c0fcccb7ad91dfea3d9ce upstream.

    After commit ID in the Fixes: tag, pat_enabled() returns false (because
    of PAT initialization being suppressed in the absence of MTRRs being
    announced to be available).

    This has become a problem: the i915 driver now fails to initialize when
    running PV on Xen (i915_gem_object_pin_map() is where I located the
    induced failure), and its error handling is flaky enough to (at least
    sometimes) result in a hung system.

    Yet even beyond that problem the keying of the use of WC mappings to
    pat_enabled() (see arch_can_pci_mmap_wc()) means that in particular
    graphics frame buffer accesses would have been quite a bit less optimal
    than possible.

    Arrange for the function to return true in such environments, without
    undermining the rest of PAT MSR management logic considering PAT to be
```

```
    disabled: specifically, no writes to the PAT MSR should occur.

    For the new boolean to live in .init.data, init_cache_modes() also needs
    moving to .init.text (where it could/should have lived already before).

      [ bp: This is the "small fix" variant for stable. It'll get replaced
        with a proper PAT and MTRR detection split upstream but that is too
        involved for a stable backport.
        - additional touchups to commit msg. Use cpu_feature_enabled(). ]

    Fixes: bdd8b6c98239 ("drm/i915: replace X86_FEATURE_PAT with pat_enabled()")
    Signed-off-by: Jan Beulich <jbeulich@suse.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Acked-by: Ingo Molnar <mingo@kernel.org>
    Cc: <stable@vger.kernel.org>
    Cc: Juergen Gross <jgross@suse.com>
    Cc: Lucas De Marchi <lucas.demarchi@intel.com>
    Link: https://lore.kernel.org/r/9385fa60-fa5d-f559-a137-6608408f88b0@suse.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d9975eea5e6add825b18dadc8c13b0424f48ba4b
Author: Peter Zijlstra <peterz@infradead.org>
Date:   Tue Aug 16 14:28:36 2022 +0200

    x86/nospec: Unwreck the RSB stuffing

    commit 4e3aa9238277597c6c7624f302d81a7b568b6f2d upstream.

    Commit 2b1299322016 ("x86/speculation: Add RSB VM Exit protections")
    made a right mess of the RSB stuffing, rewrite the whole thing to not
    suck.

    Thanks to Andrew for the enlightening comment about Post-Barrier RSB
    things so we can make this code less magical.

    Cc: stable@vger.kernel.org
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Link: https://lkml.kernel.org/r/YvuNdDWoUZSBjYcm@worktop.programming.kicks-ass.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9d0a21053cf3a3c229e56e96464048aa3b9f657e
Author: Pawan Gupta <pawan.kumar.gupta@linux.intel.com>
Date:   Wed Aug 3 14:41:32 2022 -0700

    x86/bugs: Add "unknown" reporting for MMIO Stale Data

    commit 7df548840c496b0141fb2404b889c346380c2b22 upstream.

    Older Intel CPUs that are not in the affected processor list for MMIO
    Stale Data vulnerabilities currently report "Not affected" in sysfs,
    which may not be correct. Vulnerability status for these older CPUs is
    unknown.

    Add known-not-affected CPUs to the whitelist. Report "unknown"
    mitigation status for CPUs that are not in blacklist, whitelist and also
    don't enumerate MSR ARCH_CAPABILITIES bits that reflect hardware
    immunity to MMIO Stale Data vulnerabilities.

    Mitigation is not deployed when the status is unknown.

      [ bp: Massage, fixup. ]

    Fixes: 8d50cdf8b834 ("x86/speculation/mmio: Add sysfs reporting for Processor MMIO Stale
Data")
    Suggested-by: Andrew Cooper <andrew.cooper3@citrix.com>
```

```
    Suggested-by: Tony Luck <tony.luck@intel.com>
    Signed-off-by: Pawan Gupta <pawan.kumar.gupta@linux.intel.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: stable@vger.kernel.org
    Link:
https://lore.kernel.org/r/a932c154772f2121794a5f2eded1a11013114711.1657846269.git.pawan.kuma
r.gupta@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0666703c4be88fb576dab5bb109aa4f06c9ca073
Author: Tom Lendacky <thomas.lendacky@amd.com>
Date:   Tue Aug 23 16:55:51 2022 -0500

    x86/sev: Don't use cc_platform_has() for early SEV-SNP calls

    commit cdaa0a407f1acd3a44861e3aea6e3c7349e668f1 upstream.

    When running identity-mapped and depending on the kernel configuration,
    it is possible that the compiler uses jump tables when generating code
    for cc_platform_has().

    This causes a boot failure because the jump table uses un-mapped kernel
    virtual addresses, not identity-mapped addresses. This has been seen
    with CONFIG_RETPOLINE=n.

    Similar to sme_encrypt_kernel(), use an open-coded direct check for the
    status of SNP rather than trying to eliminate the jump table. This
    preserves any code optimization in cc_platform_has() that can be useful
    post boot. It also limits the changes to SEV-specific files so that
    future compiler features won't necessarily require possible build changes
    just because they are not compatible with running identity-mapped.

      [ bp: Massage commit message. ]

    Fixes: 5e5ccff60a29 ("x86/sev: Add helper for validating pages in early enc attribute
changes")
    Reported-by: Sean Christopherson <seanjc@google.com>
    Suggested-by: Sean Christopherson <seanjc@google.com>
    Signed-off-by: Tom Lendacky <thomas.lendacky@amd.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: <stable@vger.kernel.org> # 5.19.x
    Link: https://lore.kernel.org/all/YqfabnTRxFSM+LoX@google.com/
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a10290756e4fc89c1f2a9f39f5d27ed58dc895b5
Author: Chen Zhongjin <chenzhongjin@huawei.com>
Date:   Fri Aug 19 16:43:34 2022 +0800

    x86/unwind/orc: Unwind ftrace trampolines with correct ORC entry

    commit fc2e426b1161761561624ebd43ce8c8d2fa058da upstream.

    When meeting ftrace trampolines in ORC unwinding, unwinder uses address
    of ftrace_{regs_}call address to find the ORC entry, which gets next frame at
    sp+176.

    If there is an IRQ hitting at sub $0xa8,%rsp, the next frame should be
    sp+8 instead of 176. It makes unwinder skip correct frame and throw
    warnings such as "wrong direction" or "can't access registers", etc,
    depending on the content of the incorrect frame address.

    By adding the base address ftrace_{regs_}caller with the offset
    *ip - ops->trampoline*, we can get the correct address to find the ORC entry.

    Also change "caller" to "tramp_addr" to make variable name conform to
```

```
    its content.

    [ mingo: Clarified the changelog a bit. ]

    Fixes: 6be7fa3c74d1 ("ftrace, orc, x86: Handle ftrace dynamically allocated
trampolines")
    Signed-off-by: Chen Zhongjin <chenzhongjin@huawei.com>
    Signed-off-by: Ingo Molnar <mingo@kernel.org>
    Reviewed-by: Steven Rostedt (Google) <rostedt@goodmis.org>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20220819084334.244016-1-chenzhongjin@huawei.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d1a6d0a9631fe60bf113fa44a2074e577cb9a35e
Author: Juergen Gross <jgross@suse.com>
Date:   Tue Aug 16 09:11:37 2022 +0200

    x86/entry: Fix entry_INT80_compat for Xen PV guests

    commit 5b9f0c4df1c1152403c738373fb063e9ffdac0a1 upstream.

    Commit

      c89191ce67ef ("x86/entry: Convert SWAPGS to swapgs and remove the definition of
SWAPGS")

    missed one use case of SWAPGS in entry_INT80_compat(). Removing of
    the SWAPGS macro led to asm just using "swapgs", as it is accepting
    instructions in capital letters, too.

    This in turn leads to splats in Xen PV guests like:

      [   36.145223] general protection fault, maybe for address 0x2d: 0000 [#1] PREEMPT SMP
NOPTI
      [   36.145794] CPU: 2 PID: 1847 Comm: ld-linux.so.2 Not tainted 5.19.1-1-default #1 \
          openSUSE Tumbleweed f3b44bfb672cdb9f235aff53b57724eba8b9411b
      [   36.146608] Hardware name: HP ProLiant ML350p Gen8, BIOS P72 11/14/2013
      [   36.148126] RIP: e030:entry_INT80_compat+0x3/0xa3

    Fix that by open coding this single instance of the SWAPGS macro.

    Fixes: c89191ce67ef ("x86/entry: Convert SWAPGS to swapgs and remove the definition of
SWAPGS")
    Signed-off-by: Juergen Gross <jgross@suse.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Reviewed-by: Jan Beulich <jbeulich@suse.com>
    Cc: <stable@vger.kernel.org> # 5.19
    Link: https://lore.kernel.org/r/20220816071137.4893-1-jgross@suse.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 66f2f9f2772639e07b465d05f7e2a89eb6d66813
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Tue Aug 16 05:56:11 2022 -0700

    perf/x86/lbr: Enable the branch type for the Arch LBR by default

    commit 32ba156df1b1c8804a4e5be5339616945eafea22 upstream.

    On the platform with Arch LBR, the HW raw branch type encoding may leak
    to the perf tool when the SAVE_TYPE option is not set.

    In the intel_pmu_store_lbr(), the HW raw branch type is stored in
    lbr_entries[].type. If the SAVE_TYPE option is set, the
    lbr_entries[].type will be converted into the generic PERF_BR_* type
    in the intel_pmu_lbr_filter() and exposed to the user tools.
```

```
    But if the SAVE_TYPE option is NOT set by the user, the current perf
    kernel doesn't clear the field. The HW raw branch type leaks.

    There are two solutions to fix the issue for the Arch LBR.
    One is to clear the field if the SAVE_TYPE option is NOT set.
    The other solution is to unconditionally convert the branch type and
    expose the generic type to the user tools.

    The latter is implemented here, because
    - The branch type is valuable information. I don't see a case where
      you would not benefit from the branch type. (Stephane Eranian)
    - Not having the branch type DOES NOT save any space in the
      branch record (Stephane Eranian)
    - The Arch LBR HW can retrieve the common branch types from the
      LBR_INFO. It doesn't require the high overhead SW disassemble.

    Fixes: 47125db27e47 ("perf/x86/intel/lbr: Support Architectural LBR")
    Reported-by: Stephane Eranian <eranian@google.com>
    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20220816125612.2042397-1-kan.liang@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e31430b23603661b7c4751c212eef23b5b0be03e
Author: Kan Liang <kan.liang@linux.intel.com>
Date:   Thu Aug 18 11:44:29 2022 -0700

    perf/x86/intel: Fix pebs event constraints for ADL

    commit cde643ff75bc20c538dfae787ca3b587bab16b50 upstream.

    According to the latest event list, the LOAD_LATENCY PEBS event only
    works on the GP counter 0 and 1 for ADL and RPL.

    Update the pebs event constraints table.

    Fixes: f83d2f91d259 ("perf/x86/intel: Add Alder Lake Hybrid support")
    Reported-by: Ammy Yi <ammy.yi@intel.com>
    Signed-off-by: Kan Liang <kan.liang@linux.intel.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20220818184429.2355857-1-kan.liang@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ffbf5efde85e3fff2daeed3c9855b2861f932783
Author: Michael Roth <michael.roth@amd.com>
Date:   Tue Aug 23 11:07:34 2022 -0500

    x86/boot: Don't propagate uninitialized boot_params->cc_blob_address

    commit 4b1c742407571eff58b6de9881889f7ca7c4b4dc upstream.

    In some cases, bootloaders will leave boot_params->cc_blob_address
    uninitialized rather than zeroing it out. This field is only meant to be
    set by the boot/compressed kernel in order to pass information to the
    uncompressed kernel when SEV-SNP support is enabled.

    Therefore, there are no cases where the bootloader-provided values
    should be treated as anything other than garbage. Otherwise, the
    uncompressed kernel may attempt to access this bogus address, leading to
    a crash during early boot.

    Normally, sanitize_boot_params() would be used to clear out such fields
    but that happens too late: sev_enable() may have already initialized
```

```
    it to a valid value that should not be zeroed out. Instead, have
    sev_enable() zero it out unconditionally beforehand.

    Also ensure this happens for !CONFIG_AMD_MEM_ENCRYPT as well by also
    including this handling in the sev_enable() stub function.

      [ bp: Massage commit message and comments. ]

    Fixes: b190a043c49a ("x86/sev: Add SEV-SNP feature detection/setup")
    Reported-by: Jeremi Piotrowski <jpiotrowski@linux.microsoft.com>
    Reported-by: watnuss@gmx.de
    Signed-off-by: Michael Roth <michael.roth@amd.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: stable@vger.kernel.org
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216387
    Link: https://lore.kernel.org/r/20220823160734.89036-1-michael.roth@amd.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1fc82cdd90897417d2d2f472d14899c374c3b300
Author: Filipe Manana <fdmanana@suse.com>
Date:   Mon Aug 8 12:18:37 2022 +0100

    btrfs: update generation of hole file extent item when merging holes

    commit e6e3dec6c3c288d556b991a85d5d8e3ee71e9046 upstream.

    When punching a hole into a file range that is adjacent with a hole and we
    are not using the no-holes feature, we expand the range of the adjacent
    file extent item that represents a hole, to save metadata space.

    However we don't update the generation of hole file extent item, which
    means a full fsync will not log that file extent item if the fsync happens
    in a later transaction (since commit 7f30c07288bb9e ("btrfs: stop copying
    old file extents when doing a full fsync")).

    For example, if we do this:

        $ mkfs.btrfs -f -O ^no-holes /dev/sdb
        $ mount /dev/sdb /mnt
        $ xfs_io -f -c "pwrite -S 0xab 2M 2M" /mnt/foobar
        $ sync

    We end up with 2 file extent items in our file:

    1) One that represents the hole for the file range [0, 2M), with a
       generation of 7;

    2) Another one that represents an extent covering the range [2M, 4M).

    After that if we do the following:

        $ xfs_io -c "fpunch 2M 2M" /mnt/foobar

    We end up with a single file extent item in the file, which represents a
    hole for the range [0, 4M) and with a generation of 7 - because we end
    dropping the data extent for range [2M, 4M) and then update the file
    extent item that represented the hole at [0, 2M), by increasing
    length from 2M to 4M.

    Then doing a full fsync and power failing:

        $ xfs_io -c "fsync" /mnt/foobar
        <power failure>

    will result in the full fsync not logging the file extent item that
```

```
    represents the hole for the range [0, 4M), because its generation is 7,
    which is lower than the generation of the current transaction (8).
    As a consequence, after mounting again the filesystem (after log replay),
    the region [2M, 4M) does not have a hole, it still points to the
    previous data extent.

    So fix this by always updating the generation of existing file extent
    items representing holes when we merge/expand them. This solves the
    problem and it's the same approach as when we merge prealloc extents that
    got written (at btrfs_mark_extent_written()). Setting the generation to
    the current transaction's generation is also what we do when merging
    the new hole extent map with the previous one or the next one.

    A test case for fstests, covering both cases of hole file extent item
    merging (to the left and to the right), will be sent soon.

    Fixes: 7f30c07288bb9e ("btrfs: stop copying old file extents when doing a full fsync")
    CC: stable@vger.kernel.org # 5.18+
    Reviewed-by: Josef Bacik <josef@toxicpanda.com>
    Signed-off-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4b124ad87244cd7f0883c5eaa38d2326b2154cad
Author: Zixuan Fu <r33s3n6@gmail.com>
Date:   Mon Aug 15 23:16:06 2022 +0800

    btrfs: fix possible memory leak in btrfs_get_dev_args_from_path()

    commit 9ea0106a7a3d8116860712e3f17cd52ce99f6707 upstream.

    In btrfs_get_dev_args_from_path(), btrfs_get_bdev_and_sb() can fail if
    the path is invalid. In this case, btrfs_get_dev_args_from_path()
    returns directly without freeing args->uuid and args->fsid allocated
    before, which causes memory leak.

    To fix these possible leaks, when btrfs_get_bdev_and_sb() fails,
    btrfs_put_dev_args_from_path() is called to clean up the memory.

    Reported-by: TOTE Robot <oslab@tsinghua.edu.cn>
    Fixes: faa775c41d655 ("btrfs: add a btrfs_get_dev_args_from_path helper")
    CC: stable@vger.kernel.org # 5.16
    Reviewed-by: Boris Burkov <boris@bur.io>
    Signed-off-by: Zixuan Fu <r33s3n6@gmail.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 0f72e355c4a0737691610c9d3e6d1a23324a51a4
Author: Goldwyn Rodrigues <rgoldwyn@suse.de>
Date:   Tue Aug 16 16:42:56 2022 -0500

    btrfs: check if root is readonly while setting security xattr

    commit b51111271b0352aa596c5ae8faf06939e91b3b68 upstream.

    For a filesystem which has btrfs read-only property set to true, all
    write operations including xattr should be denied. However, security
    xattr can still be changed even if btrfs ro property is true.

    This happens because xattr_permission() does not have any restrictions
    on security.*, system.*  and in some cases trusted.* from VFS and
    the decision is left to the underlying filesystem. See comments in
    xattr_permission() for more details.
```

```
    This patch checks if the root is read-only before performing the set
    xattr operation.

    Testcase:

      DEV=/dev/vdb
      MNT=/mnt

      mkfs.btrfs -f $DEV
      mount $DEV $MNT
      echo "file one" > $MNT/f1

      setfattr -n "security.one" -v 2 $MNT/f1
      btrfs property set /mnt ro true

      setfattr -n "security.one" -v 1 $MNT/f1

      umount $MNT

    CC: stable@vger.kernel.org # 4.9+
    Reviewed-by: Qu Wenruo <wqu@suse.com>
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: Goldwyn Rodrigues <rgoldwyn@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 a2e54eb64229f07f917b05d0c323604fda9b89f7
Author: Omar Sandoval <osandov@fb.com>
Date:   Tue Aug 23 11:28:13 2022 -0700

    btrfs: fix space cache corruption and potential double allocations

    commit ced8ecf026fd8084cf175530ff85c76d6085d715 upstream.

    When testing space_cache v2 on a large set of machines, we encountered a
    few symptoms:

    1. "unable to add free space :-17" (EEXIST) errors.
    2. Missing free space info items, sometimes caught with a "missing free
       space info for X" error.
    3. Double-accounted space: ranges that were allocated in the extent tree
       and also marked as free in the free space tree, ranges that were
       marked as allocated twice in the extent tree, or ranges that were
       marked as free twice in the free space tree. If the latter made it
       onto disk, the next reboot would hit the BUG_ON() in
       add_new_free_space().
    4. On some hosts with no on-disk corruption or error messages, the
       in-memory space cache (dumped with drgn) disagreed with the free
       space tree.

    All of these symptoms have the same underlying cause: a race between
    caching the free space for a block group and returning free space to the
    in-memory space cache for pinned extents causes us to double-add a free
    range to the space cache. This race exists when free space is cached
    from the free space tree (space_cache=v2) or the extent tree
    (nospace_cache, or space_cache=v1 if the cache needs to be regenerated).
    struct btrfs_block_group::last_byte_to_unpin and struct
    btrfs_block_group::progress are supposed to protect against this race,
    but commit d0c2f4fa555e ("btrfs: make concurrent fsyncs wait less when
    waiting for a transaction commit") subtly broke this by allowing
    multiple transactions to be unpinning extents at the same time.

    Specifically, the race is as follows:
```

1. An extent is deleted from an uncached block group in transaction A.
2. btrfs_commit_transaction() is called for transaction A.
3. btrfs_run_delayed_refs() -> __btrfs_free_extent() runs the delayed ref for the deleted extent.
4. __btrfs_free_extent() -> do_free_extent_accounting() -> add_to_free_space_tree() adds the deleted extent back to the free space tree.
5. do_free_extent_accounting() -> btrfs_update_block_group() -> btrfs_cache_block_group() queues up the block group to get cached. block_group->progress is set to block_group->start.
6. btrfs_commit_transaction() for transaction A calls switch_commit_roots(). It sets block_group->last_byte_to_unpin to block_group->progress, which is block_group->start because the block group hasn't been cached yet.
7. The caching thread gets to our block group. Since the commit roots were already switched, load_free_space_tree() sees the deleted extent as free and adds it to the space cache. It finishes caching and sets block_group->progress to U64_MAX.
8. btrfs_commit_transaction() advances transaction A to TRANS_STATE_SUPER_COMMITTED.
9. fsync calls btrfs_commit_transaction() for transaction B. Since transaction A is already in TRANS_STATE_SUPER_COMMITTED and the commit is for fsync, it advances.
10. btrfs_commit_transaction() for transaction B calls switch_commit_roots(). This time, the block group has already been cached, so it sets block_group->last_byte_to_unpin to U64_MAX.
11. btrfs_commit_transaction() for transaction A calls btrfs_finish_extent_commit(), which calls unpin_extent_range() for the deleted extent. It sees last_byte_to_unpin set to U64_MAX (by transaction B!), so it adds the deleted extent to the space cache again!

This explains all of our symptoms above:

* If the sequence of events is exactly as described above, when the free space is re-added in step 11, it will fail with EEXIST.
* If another thread reallocates the deleted extent in between steps 7 and 11, then step 11 will silently re-add that space to the space cache as free even though it is actually allocated. Then, if that space is allocated *again*, the free space tree will be corrupted (namely, the wrong item will be deleted).
* If we don't catch this free space tree corruption, it will continue to get worse as extents are deleted and reallocated.

The v1 space_cache is synchronously loaded when an extent is deleted (btrfs_update_block_group() with alloc=0 calls btrfs_cache_block_group() with load_cache_only=1), so it is not normally affected by this bug. However, as noted above, if we fail to load the space cache, we will fall back to caching from the extent tree and may hit this bug.

The easiest fix for this race is to also make caching from the free space tree or extent tree synchronous. Josef tested this and found no performance regressions.

A few extra changes fall out of this change. Namely, this fix does the following, with step 2 being the crucial fix:

1. Factor btrfs_caching_ctl_wait_done() out of btrfs_wait_block_group_cache_done() to allow waiting on a caching_ctl that we already hold a reference to.
2. Change the call in btrfs_cache_block_group() of btrfs_wait_space_cache_v1_finished() to btrfs_caching_ctl_wait_done(), which makes us wait regardless of the space_cache option.
3. Delete the now unused btrfs_wait_space_cache_v1_finished() and

```
       space_cache_v1_done().
    4. Change btrfs_cache_block_group()'s `int load_cache_only` parameter to
       `bool wait` to more accurately describe its new meaning.
    5. Change a few callers which had a separate call to
       btrfs_wait_block_group_cache_done() to use wait = true instead.
    6. Make btrfs_wait_block_group_cache_done() static now that it's not
       used outside of block-group.c anymore.

    Fixes: d0c2f4fa555e ("btrfs: make concurrent fsyncs wait less when waiting for a
transaction commit")
    CC: stable@vger.kernel.org # 5.12+
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: Omar Sandoval <osandov@fb.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b4656b25c83f04cae2e947d417a50350131edf07
Author: Anand Jain <anand.jain@oracle.com>
Date:   Fri Aug 12 18:32:19 2022 +0800

    btrfs: add info when mount fails due to stale replace target

    commit f2c3bec215694fb8bc0ef5010f2a758d1906fc2d upstream.

    If the replace target device reappears after the suspended replace is
    cancelled, it blocks the mount operation as it can't find the matching
    replace-item in the metadata. As shown below,

       BTRFS error (device sda5): replace devid present without an active replace item

    To overcome this situation, the user can run the command

       btrfs device scan --forget <replace target device>

    and try the mount command again. And also, to avoid repeating the issue,
    superblock on the devid=0 must be wiped.

       wipefs -a device-path-to-devid=0.

    This patch adds some info when this situation occurs.

    Reported-by: Samuel Greiner <samuel@balkonien.org>
    Link: https://lore.kernel.org/linux-btrfs/b4f62b10-b295-26ea-71f9-
9a5c9299d42c@balkonien.org/T/
    CC: stable@vger.kernel.org # 5.0+
    Signed-off-by: Anand Jain <anand.jain@oracle.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 955d400e263d41255cf43c0fdae024116116f286
Author: Anand Jain <anand.jain@oracle.com>
Date:   Fri Aug 12 18:32:18 2022 +0800

    btrfs: replace: drop assert for suspended replace

    commit 59a3991984dbc1fc47e5651a265c5200bd85464e upstream.

    If the filesystem mounts with the replace-operation in a suspended state
    and try to cancel the suspended replace-operation, we hit the assert. The
    assert came from the commit fe97e2e173af ("btrfs: dev-replace: replace's
    scrub must not be running in suspended state") that was actually not
    required. So just remove it.

     $ mount /dev/sda5 /btrfs
```

```
         BTRFS info (device sda5): cannot continue dev_replace, tgtdev is missing
         BTRFS info (device sda5): you may cancel the operation after 'mount -o degraded'

      $ mount -o degraded /dev/sda5 /btrfs <-- success.

      $ btrfs replace cancel /btrfs

         kernel: assertion failed: ret != -ENOTCONN, in fs/btrfs/dev-replace.c:1131
         kernel: ------------[ cut here ]------------
         kernel: kernel BUG at fs/btrfs/ctree.h:3750!

     After the patch:

      $ btrfs replace cancel /btrfs

         BTRFS info (device sda5): suspended dev_replace from /dev/sda5 (devid 1) to <missing
 disk> canceled

     Fixes: fe97e2e173af ("btrfs: dev-replace: replace's scrub must not be running in
 suspended state")
     CC: stable@vger.kernel.org # 5.0+
     Signed-off-by: Anand Jain <anand.jain@oracle.com>
     Signed-off-by: David Sterba <dsterba@suse.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit e08fcb1284a998058643b454fddb59b4d1a85909
 Author: Filipe Manana <fdmanana@suse.com>
 Date:   Mon Aug 22 15:47:09 2022 +0100

     btrfs: fix silent failure when deleting root reference

     commit 47bf225a8d2cccb15f7e8d4a1ed9b757dd86afd7 upstream.

     At btrfs_del_root_ref(), if btrfs_search_slot() returns an error, we end
     up returning from the function with a value of 0 (success). This happens
     because the function returns the value stored in the variable 'err',
     which is 0, while the error value we got from btrfs_search_slot() is
     stored in the 'ret' variable.

     So fix it by setting 'err' with the error value.

     Fixes: 8289ed9f93bef2 ("btrfs: replace the BUG_ON in btrfs_del_root_ref with proper
 error handling")
     CC: stable@vger.kernel.org # 5.16+
     Reviewed-by: Qu Wenruo <wqu@suse.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 3ef2786e32d93e562cd40601248a14ae090de873
 Author: Aleksander Jan Bajkowski <olek2@wp.pl>
 Date:   Wed Aug 24 23:54:08 2022 +0200

     net: lantiq_xrx200: restore buffer if memory allocation failed

     [ Upstream commit c9c3b1775f80fa21f5bff874027d2ccb10f5d90c ]

     In a situation where memory allocation fails, an invalid buffer address
     is stored. When this descriptor is used again, the system panics in the
     build_skb() function when accessing memory.

     Fixes: 7ea6cd16f159 ("lantiq: net: fix duplicated skb in rx descriptor ring")
     Signed-off-by: Aleksander Jan Bajkowski <olek2@wp.pl>
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
```

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

commit 0d9981b0636dd54f5d17f6bac008d98163e9c509
Author: Aleksander Jan Bajkowski <olek2@wp.pl>
Date:   Wed Aug 24 23:54:07 2022 +0200

    net: lantiq_xrx200: fix lock under memory pressure

    [ Upstream commit c4b6e9341f930e4dd089231c0414758f5f1f9dbd ]

    When the xrx200_hw_receive() function returns -ENOMEM, the NAPI poll
    function immediately returns an error.
    This is incorrect for two reasons:
    * the function terminates without enabling interrupts or scheduling NAPI,
    * the error code (-ENOMEM) is returned instead of the number of received
    packets.

    After the first memory allocation failure occurs, packet reception is
    locked due to disabled interrupts from DMA..

    Fixes: fe1a56420cf2 ("net: lantiq: Add Lantiq / Intel VRX200 Ethernet driver")
    Signed-off-by: Aleksander Jan Bajkowski <olek2@wp.pl>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 73f475865269aa386563a9751f4e83615df12afe
Author: Aleksander Jan Bajkowski <olek2@wp.pl>
Date:   Wed Aug 24 23:54:06 2022 +0200

    net: lantiq_xrx200: confirm skb is allocated before using

    [ Upstream commit c8b043702dc0894c07721c5b019096cebc8c798f ]

    xrx200_hw_receive() assumes build_skb() always works and goes straight
    to skb_reserve(). However, build_skb() can fail under memory pressure.

    Add a check in case build_skb() failed to allocate and return NULL.

    Fixes: e015593573b3 ("net: lantiq_xrx200: convert to build_skb")
    Reported-by: Eric Dumazet <edumazet@google.com>
    Signed-off-by: Aleksander Jan Bajkowski <olek2@wp.pl>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 27a5ab8fec274b07b7a4d473d835a8ac49790d79
Author: Heiner Kallweit <hkallweit1@gmail.com>
Date:   Wed Aug 24 22:34:49 2022 +0200

    net: stmmac: work around sporadic tx issue on link-up

    [ Upstream commit a3a57bf07de23fe1ff779e0fdf710aa581c3ff73 ]

    This is a follow-up to the discussion in [0]. It seems to me that
    at least the IP version used on Amlogic SoC's sometimes has a problem
    if register MAC_CTRL_REG is written whilst the chip is still processing
    a previous write. But that's just a guess.
    Adding a delay between two writes to this register helps, but we can
    also simply omit the offending second write. This patch uses the second
    approach and is based on a suggestion from Qi Duan.
    Benefit of this approach is that we can save few register writes, also
    on not affected chip versions.

    [0] https://www.spinics.net/lists/netdev/msg831526.html

    Fixes: bfab27a146ed ("stmmac: add the experimental PCI support")
```

```
    Suggested-by: Qi Duan <qi.duan@amlogic.com>
    Suggested-by: Jerome Brunet <jbrunet@baylibre.com>
    Signed-off-by: Heiner Kallweit <hkallweit1@gmail.com>
    Link: https://lore.kernel.org/r/e99857ce-bd90-5093-ca8c-8cd480b5a0a2@gmail.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c830d7120137c460307c965cd80c09544ab92549
Author: R Mohamed Shah <mohamed@pensando.io>
Date:   Wed Aug 24 09:50:51 2022 -0700

    ionic: VF initial random MAC address if no assigned mac

    [ Upstream commit 19058be7c48ceb3e60fa3948e24da1059bd68ee4 ]

    Assign a random mac address to the VF interface station
    address if it boots with a zero mac address in order to match
    similar behavior seen in other VF drivers.  Handle the errors
    where the older firmware does not allow the VF to set its own
    station address.

    Newer firmware will allow the VF to set the station mac address
    if it hasn't already been set administratively through the PF.
    Setting it will also be allowed if the VF has trust.

    Fixes: fbb39807e9ae ("ionic: support sr-iov operations")
    Signed-off-by: R Mohamed Shah <mohamed@pensando.io>
    Signed-off-by: Shannon Nelson <snelson@pensando.io>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 79e77fb1565d00c76692166cfe614b38ed6b6e0f
Author: Shannon Nelson <snelson@pensando.io>
Date:   Wed Aug 24 09:50:50 2022 -0700

    ionic: fix up issues with handling EAGAIN on FW cmds

    [ Upstream commit 0fc4dd452d6c14828eed6369155c75c0ac15bab3 ]

    In looping on FW update tests we occasionally see the
    FW_ACTIVATE_STATUS command fail while it is in its EAGAIN loop
    waiting for the FW activate step to finsh inside the FW.  The
    firmware is complaining that the done bit is set when a new
    dev_cmd is going to be processed.

    Doing a clean on the cmd registers and doorbell before exiting
    the wait-for-done and cleaning the done bit before the sleep
    prevents this from occurring.

    Fixes: fbfb8031533c ("ionic: Add hardware init and device commands")
    Signed-off-by: Shannon Nelson <snelson@pensando.io>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 94d71d99e5dd5885d1a0bef8de5eabe21e7750f0
Author: Shannon Nelson <snelson@pensando.io>
Date:   Wed Aug 24 09:50:49 2022 -0700

    ionic: clear broken state on generation change

    [ Upstream commit 9cb9dadb8f45c67e4310e002c2f221b70312b293 ]

    There is a case found in heavy testing where a link flap happens just
    before a firmware Recovery event and the driver gets stuck in the
    BROKEN state.  This comes from the driver getting interrupted by a FW
```

```
    generation change when coming back up from the link flap, and the call
    to ionic_start_queues() in ionic_link_status_check() fails.  This can be
    addressed by having the fw_up code clear the BROKEN bit if seen, rather
    than waiting for a user to manually force the interface down and then
    back up.

    Fixes: 9e8eaf8427b6 ("ionic: stop watchdog when in broken state")
    Signed-off-by: Shannon Nelson <snelson@pensando.io>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 091dc91e119fdd61432347231724f4e861c6b465
Author: David Howells <dhowells@redhat.com>
Date:   Wed Aug 24 17:35:45 2022 +0100

    rxrpc: Fix locking in rxrpc's sendmsg

    [ Upstream commit b0f571ecd7943423c25947439045f0d352ca3dbf ]

    Fix three bugs in the rxrpc's sendmsg implementation:

     (1) rxrpc_new_client_call() should release the socket lock when returning
         an error from rxrpc_get_call_slot().

     (2) rxrpc_wait_for_tx_window_intr() will return without the call mutex
         held in the event that we're interrupted by a signal whilst waiting
         for tx space on the socket or relocking the call mutex afterwards.

         Fix this by: (a) moving the unlock/lock of the call mutex up to
         rxrpc_send_data() such that the lock is not held around all of
         rxrpc_wait_for_tx_window*() and (b) indicating to higher callers
         whether we're return with the lock dropped.  Note that this means
         recvmsg() will not block on this call whilst we're waiting.

     (3) After dropping and regaining the call mutex, rxrpc_send_data() needs
         to go and recheck the state of the tx_pending buffer and the
         tx_total_len check in case we raced with another sendmsg() on the same
         call.

    Thinking on this some more, it might make sense to have different locks for
    sendmsg() and recvmsg().  There's probably no need to make recvmsg() wait
    for sendmsg().  It does mean that recvmsg() can return MSG_EOR indicating
    that a call is dead before a sendmsg() to that call returns - but that can
    currently happen anyway.

    Without fix (2), something like the following can be induced:

            WARNING: bad unlock balance detected!
            5.16.0-rc6-syzkaller #0 Not tainted
            -------------------------------------
            syz-executor011/3597 is trying to release lock (&call->user_mutex) at:
            [<ffffffff885163a3>] rxrpc_do_sendmsg+0xc13/0x1350 net/rxrpc/sendmsg.c:748
            but there are no more locks to release!

            other info that might help us debug this:
            no locks held by syz-executor011/3597.
            ...
            Call Trace:
             <TASK>
             __dump_stack lib/dump_stack.c:88 [inline]
             dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106
             print_unlock_imbalance_bug include/trace/events/lock.h:58 [inline]
             __lock_release kernel/locking/lockdep.c:5306 [inline]
             lock_release.cold+0x49/0x4e kernel/locking/lockdep.c:5657
             __mutex_unlock_slowpath+0x99/0x5e0 kernel/locking/mutex.c:900
```

```
               rxrpc_do_sendmsg+0xc13/0x1350 net/rxrpc/sendmsg.c:748
               rxrpc_sendmsg+0x420/0x630 net/rxrpc/af_rxrpc.c:561
               sock_sendmsg_nosec net/socket.c:704 [inline]
               sock_sendmsg+0xcf/0x120 net/socket.c:724
               ____sys_sendmsg+0x6e8/0x810 net/socket.c:2409
               ___sys_sendmsg+0xf3/0x170 net/socket.c:2463
               __sys_sendmsg+0xe5/0x1b0 net/socket.c:2492
               do_syscall_x64 arch/x86/entry/common.c:50 [inline]
               do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80
               entry_SYSCALL_64_after_hwframe+0x44/0xae

    [Thanks to Hawkins Jiawei and Khalid Masum for their attempts to fix this]

    Fixes: bc5e3a546d55 ("rxrpc: Use MSG_WAITALL to tell sendmsg() to temporarily ignore
signals")
    Reported-by: syzbot+7f0483225d0c94cb3441@syzkaller.appspotmail.com
    Signed-off-by: David Howells <dhowells@redhat.com>
    Reviewed-by: Marc Dionne <marc.dionne@auristor.com>
    Tested-by: syzbot+7f0483225d0c94cb3441@syzkaller.appspotmail.com
    cc: Hawkins Jiawei <yin31149@gmail.com>
    cc: Khalid Masum <khalid.masum.92@gmail.com>
    cc: Dan Carpenter <dan.carpenter@oracle.com>
    cc: linux-afs@lists.infradead.org
    Link:
https://lore.kernel.org/r/166135894583.600315.7170979436768124075.stgit@warthog.procyon.org.
uk
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b886aebd0c3df87e75a6d1587990532a830000da
Author: Lorenzo Bianconi <lorenzo@kernel.org>
Date:   Tue Aug 23 14:24:07 2022 +0200

    net: ethernet: mtk_eth_soc: fix hw hash reporting for MTK_NETSYS_V2

    [ Upstream commit 0cf731f9ebb5bf6f252055bebf4463a5c0bd490b ]

    Properly report hw rx hash for mt7986 chipset accroding to the new dma
    descriptor layout.

    Fixes: 197c9e9b17b11 ("net: ethernet: mtk_eth_soc: introduce support for mt7986
chipset")
    Signed-off-by: Lorenzo Bianconi <lorenzo@kernel.org>
    Link:
https://lore.kernel.org/r/091394ea4e705fbb35f828011d98d0ba33808f69.1661257293.git.lorenzo@ke
rnel.org
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2f23757084678a074ef7188d34a86597fa5e5018
Author: Lorenzo Bianconi <lorenzo@kernel.org>
Date:   Mon Jun 6 21:49:00 2022 +0200

    net: ethernet: mtk_eth_soc: enable rx cksum offload for MTK_NETSYS_V2

    [ Upstream commit da6e113ff010815fdd21ee1e9af2e8d179a2680f ]

    Enable rx checksum offload for mt7986 chipset.

    Signed-off-by: Lorenzo Bianconi <lorenzo@kernel.org>
    Link:
https://lore.kernel.org/r/c8699805c18f7fd38315fcb8da2787676d83a32c.1654544585.git.lorenzo@ke
rnel.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>
```

```
commit 82fd14027677a8d19c24c48740dcf1b7b2a8ba0f
Author: Sylwester Dziedziuch <sylwesterx.dziedziuch@intel.com>
Date:   Fri Aug 19 12:45:52 2022 +0200

    i40e: Fix incorrect address type for IPv6 flow rules

    [ Upstream commit bcf3a156429306070afbfda5544f2b492d25e75b ]

    It was not possible to create 1-tuple flow director
    rule for IPv6 flow type. It was caused by incorrectly
    checking for source IP address when validating user provided
    destination IP address.

    Fix this by changing ip6src to correct ip6dst address
    in destination IP address validation for IPv6 flow type.

    Fixes: efca91e89b67 ("i40e: Add flow director support for IPv6")
    Signed-off-by: Sylwester Dziedziuch <sylwesterx.dziedziuch@intel.com>
    Tested-by: Gurucharan <gurucharanx.g@intel.com> (A Contingent worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2b99b2a249b3506ad7c6ffc55074510a911e506
Author: Jacob Keller <jacob.e.keller@intel.com>
Date:   Mon Aug 1 17:24:19 2022 -0700

    ixgbe: stop resetting SYSTIME in ixgbe_ptp_start_cyclecounter

    [ Upstream commit 25d7a5f5a6bb15a2dae0a3f39ea5dda215024726 ]

    The ixgbe_ptp_start_cyclecounter is intended to be called whenever the
    cyclecounter parameters need to be changed.

    Since commit a9763f3cb54c ("ixgbe: Update PTP to support X550EM_x
    devices"), this function has cleared the SYSTIME registers and reset the
    TSAUXC DISABLE_SYSTIME bit.

    While these need to be cleared during ixgbe_ptp_reset, it is wrong to clear
    them during ixgbe_ptp_start_cyclecounter. This function may be called
    during both reset and link status change. When link changes, the SYSTIME
    counter is still operating normally, but the cyclecounter should be updated
    to account for the possibly changed parameters.

    Clearing SYSTIME when link changes causes the timecounter to jump because
    the cycle counter now reads zero.

    Extract the SYSTIME initialization out to a new function and call this
    during ixgbe_ptp_reset. This prevents the timecounter adjustment and avoids
    an unnecessary reset of the current time.

    This also restores the original SYSTIME clearing that occurred during
    ixgbe_ptp_reset before the commit above.

    Reported-by: Steve Payne <spayne@aurora.tech>
    Reported-by: Ilya Evenbach <ievenbach@aurora.tech>
    Fixes: a9763f3cb54c ("ixgbe: Update PTP to support X550EM_x devices")
    Signed-off-by: Jacob Keller <jacob.e.keller@intel.com>
    Tested-by: Gurucharan <gurucharanx.g@intel.com> (A Contingent worker at Intel)
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 18a8b82643e791f7fed77d9525c7459e3ce4bd82
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:47:00 2022 -0700
```

```
    net: Fix a data-race around sysctl_somaxconn.

    [ Upstream commit 3c9ba81d72047f2e81bb535d42856517b613aba7 ]

    While reading sysctl_somaxconn, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 8a536935207a66b3cb1506b7b3ba5fdbad91525a
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:59 2022 -0700

    net: Fix a data-race around netdev_unregister_timeout_secs.

    [ Upstream commit 05e49cfc89e4f325eebbc62d24dd122e55f94c23 ]

    While reading netdev_unregister_timeout_secs, it can be changed
    concurrently.  Thus, we need to add READ_ONCE() to its reader.

    Fixes: 5aa3afe107d9 ("net: make unregister netdev warning timeout configurable")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Acked-by: Dmitry Vyukov <dvyukov@google.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 21c6c135354aca32676b0e94420b3b74c4194966
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:58 2022 -0700

    net: Fix a data-race around gro_normal_batch.

    [ Upstream commit 8db24af3f02ebdbf302196006ebb270c4c3a2706 ]

    While reading gro_normal_batch, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 323ebb61e32b ("net: use listified RX for handling GRO_NORMAL skbs")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Acked-by: Edward Cree <ecree.xilinx@gmail.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bdb33552e663f2aafa2678684c9b665cfac3e246
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:57 2022 -0700

    net: Fix data-races around sysctl_devconf_inherit_init_net.

    [ Upstream commit a5612ca10d1aa05624ebe72633e0c8c792970833 ]

    While reading sysctl_devconf_inherit_init_net, it can be changed
    concurrently.  Thus, we need to add READ_ONCE() to its readers.

    Fixes: 856c395cfa63 ("net: introduce a knob to control whether to inherit devconf
config")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e94dd3e960367fb32c09e36a95e3fd8a8a0b5af6
```

```
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:56 2022 -0700

    net: Fix data-races around sysctl_fb_tunnels_only_for_init_net.

    [ Upstream commit af67508ea6cbf0e4ea27f8120056fa2efce127dd ]

    While reading sysctl_fb_tunnels_only_for_init_net, it can be changed
    concurrently.  Thus, we need to add READ_ONCE() to its readers.

    Fixes: 79134e6ce2c9 ("net: do not create fallback tunnels for non-default namespaces")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d923063ba2d19eee7d044c3f30831b45e27c188e
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:55 2022 -0700

    net: Fix a data-race around netdev_budget_usecs.

    [ Upstream commit fa45d484c52c73f79db2c23b0cdfc6c6455093ad ]

    While reading netdev_budget_usecs, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 7acf8a1e8a28 ("Replace 2 jiffies with sysctl netdev_budget_usecs to enable
softirq tuning")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c34e06f05ab7163a461e550a70e81a7256079789
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:54 2022 -0700

    net: Fix data-races around sysctl_max_skb_frags.

    [ Upstream commit 657b991afb89d25fe6c4783b1b75a8ad4563670d ]

    While reading sysctl_max_skb_frags, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its readers.

    Fixes: 5f74f82ea34c ("net:Add sysctl_max_skb_frags")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 293ec6acc32a9ad2b4a8dd6e91dd43944876ee3e
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:53 2022 -0700

    net: Fix a data-race around netdev_budget.

    [ Upstream commit 2e0c42374ee32e72948559d2ae2f7ba3dc6b977c ]

    While reading netdev_budget, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 51b0bdedb8e7 ("[NET]: Separate two usages of netdev_max_backlog.")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6a520caf1f55f128592a3cde7294d133610a15b7
```

```
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:52 2022 -0700

    net: Fix a data-race around sysctl_net_busy_read.

    [ Upstream commit e59ef36f0795696ab229569c153936bfd068d21c ]

    While reading sysctl_net_busy_read, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 2d48d67fa8cd ("net: poll/select low latency socket support")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 05d92723f99cf5c15ae74cc196193c2dbaa9a12c
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:51 2022 -0700

    net: Fix a data-race around sysctl_net_busy_poll.

    [ Upstream commit c42b7cddea47503411bfb5f2f93a4154aaffa2d9 ]

    While reading sysctl_net_busy_poll, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its reader.

    Fixes: 060212928670 ("net: add low latency socket poll")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 6fc89f990716ec516dfd817c785daab5be6d98aa
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:50 2022 -0700

    net: Fix a data-race around sysctl_tstamp_allow_data.

    [ Upstream commit d2154b0afa73c0159b2856f875c6b4fe7cf6a95e ]

    While reading sysctl_tstamp_allow_data, it can be changed
    concurrently.  Thus, we need to add READ_ONCE() to its reader.

    Fixes: b245be1f4db1 ("net-timestamp: no-payload only sysctl")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 764352456e55004a8f7b23a71226e3e6a42843ad
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:49 2022 -0700

    net: Fix data-races around sysctl_optmem_max.

    [ Upstream commit 7de6d09f51917c829af2b835aba8bb5040f8e86a ]

    While reading sysctl_optmem_max, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its readers.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit f1bbd4c0966cdcdc60066c6f07ff06c94ec2634c
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
```

```
Date:   Tue Aug 23 10:46:48 2022 -0700

    ratelimit: Fix data-races in ___ratelimit().

    [ Upstream commit 6bae8ceb90ba76cdba39496db936164fa672b9be ]

    While reading rs->interval and rs->burst, they can be changed
    concurrently via sysctl (e.g. net_ratelimit_state).  Thus, we
    need to add READ_ONCE() to their readers.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c49b023e4e2eaa6401fba9fbb87867942ee53276
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:47 2022 -0700

    net: Fix data-races around netdev_tstamp_prequeue.

    [ Upstream commit 61adf447e38664447526698872e21c04623afb8e ]

    While reading netdev_tstamp_prequeue, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its readers.

    Fixes: 3b098e2d7c69 ("net: Consistent skb timestamping")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 32f8c816b92e5f7d556ca2f7f17082ce8c8286f0
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:46 2022 -0700

    net: Fix data-races around netdev_max_backlog.

    [ Upstream commit 5dcd08cd19912892586c6082d56718333e2d19db ]

    While reading netdev_max_backlog, it can be changed concurrently.
    Thus, we need to add READ_ONCE() to its readers.

    While at it, we remove the unnecessary spaces in the doc.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 01198bebd53e21a139f7d8eacd190f262fbd66eb
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:45 2022 -0700

    net: Fix data-races around weight_p and dev_weight_[rt]x_bias.

    [ Upstream commit bf955b5ab8f6f7b0632cdef8e36b14e4f6e77829 ]

    While reading weight_p, it can be changed concurrently.  Thus, we need
    to add READ_ONCE() to its reader.

    Also, dev_[rt]x_weight can be read/written at the same time.  So, we
    need to use READ_ONCE() and WRITE_ONCE() for its access.  Moreover, to
    use the same weight_p while changing dev_[rt]x_weight, we add a mutex
    in proc_do_dev_weight().
```

```
    Fixes: 3d48b53fb2ae ("net: dev_weight: TX/RX orthogonality")
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit a9de312f45141ffff8f618a528b183e468154aef
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Tue Aug 23 10:46:44 2022 -0700

    net: Fix data-races around sysctl_[rw]mem_(max|default).

    [ Upstream commit 1227c1771dd2ad44318aa3ab9e3a293b3f34ff2a ]

    While reading sysctl_[rw]mem_(max|default), they can be changed
    concurrently.  Thus, we need to add READ_ONCE() to its readers.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 89e135a36a9eb81412b5459df94a80995ce62eef
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Thu Nov 18 22:24:15 2021 +0100

    netfilter: flowtable: fix stuck flows on cleanup due to pending work

    [ Upstream commit 9afb4b27349a499483ae0134282cefd0c90f480f ]

    To clear the flow table on flow table free, the following sequence
    normally happens in order:

      1) gc_step work is stopped to disable any further stats/del requests.
      2) All flow table entries are set to teardown state.
      3) Run gc_step which will queue HW del work for each flow table entry.
      4) Waiting for the above del work to finish (flush).
      5) Run gc_step again, deleting all entries from the flow table.
      6) Flow table is freed.

    But if a flow table entry already has pending HW stats or HW add work
    step 3 will not queue HW del work (it will be skipped), step 4 will wait
    for the pending add/stats to finish, and step 5 will queue HW del work
    which might execute after freeing of the flow table.

    To fix the above, this patch flushes the pending work, then it sets the
    teardown flag to all flows in the flowtable and it forces a garbage
    collector run to queue work to remove the flows from hardware, then it
    flushes this new pending work and (finally) it forces another garbage
    collector run to remove the entry from the software flowtable.

    Stack trace:
    [47773.882335] BUG: KASAN: use-after-free in down_read+0x99/0x460
    [47773.883634] Write of size 8 at addr ffff888103b45aa8 by task kworker/u20:6/543704
    [47773.885634] CPU: 3 PID: 543704 Comm: kworker/u20:6 Not tainted 5.12.0-rc7+ #2
    [47773.886745] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009)
    [47773.888438] Workqueue: nf_ft_offload_del flow_offload_work_handler [nf_flow_table]
    [47773.889727] Call Trace:
    [47773.890214]  dump_stack+0xbb/0x107
    [47773.890818]  print_address_description.constprop.0+0x18/0x140
    [47773.892990]  kasan_report.cold+0x7c/0xd8
    [47773.894459]  kasan_check_range+0x145/0x1a0
    [47773.895174]  down_read+0x99/0x460
    [47773.899706]  nf_flow_offload_tuple+0x24f/0x3c0 [nf_flow_table]
    [47773.907137]  flow_offload_work_handler+0x72d/0xbe0 [nf_flow_table]
```

```
      [47773.913372]  process_one_work+0x8ac/0x14e0
      [47773.921325]
      [47773.921325] Allocated by task 592159:
      [47773.922031]  kasan_save_stack+0x1b/0x40
      [47773.922730]  __kasan_kmalloc+0x7a/0x90
      [47773.923411]  tcf_ct_flow_table_get+0x3cb/0x1230 [act_ct]
      [47773.924363]  tcf_ct_init+0x71c/0x1156 [act_ct]
      [47773.925207]  tcf_action_init_1+0x45b/0x700
      [47773.925987]  tcf_action_init+0x453/0x6b0
      [47773.926692]  tcf_exts_validate+0x3d0/0x600
      [47773.927419]  fl_change+0x757/0x4a51 [cls_flower]
      [47773.928227]  tc_new_tfilter+0x89a/0x2070
      [47773.936652]
      [47773.936652] Freed by task 543704:
      [47773.937303]  kasan_save_stack+0x1b/0x40
      [47773.938039]  kasan_set_track+0x1c/0x30
      [47773.938731]  kasan_set_free_info+0x20/0x30
      [47773.939467]  __kasan_slab_free+0xe7/0x120
      [47773.940194]  slab_free_freelist_hook+0x86/0x190
      [47773.941038]  kfree+0xce/0x3a0
      [47773.941644]  tcf_ct_flow_table_cleanup_work

      Original patch description and stack trace by Paul Blakey.

      Fixes: c29f74e0df7a ("netfilter: nf_flow_table: hardware offload support")
      Reported-by: Paul Blakey <paulb@nvidia.com>
      Tested-by: Paul Blakey <paulb@nvidia.com>
      Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit cbfc3a1b098dcb0b868d9fa8c58dd36d26361ab6
  Author: Pablo Neira Ayuso <pablo@netfilter.org>
  Date:   Mon Aug 22 23:13:00 2022 +0200

      netfilter: flowtable: add function to invoke garbage collection immediately

      [ Upstream commit 759eebbcfafcefa23b59e912396306543764bd3c ]

      Expose nf_flow_table_gc_run() to force a garbage collector run from the
      offload infrastructure.

      Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit fdca693fcf26c11596e7aa1e540af2b4a5288c76
  Author: Pablo Neira Ayuso <pablo@netfilter.org>
  Date:   Mon Aug 22 11:06:39 2022 +0200

      netfilter: nf_tables: disallow binding to already bound chain

      [ Upstream commit e02f0d3970404bfea385b6edb86f2d936db0ea2b ]

      Update nft_data_init() to report EINVAL if chain is already bound.

      Fixes: d0e2c7de92c7 ("netfilter: nf_tables: add NFT_CHAIN_BINDING")
      Reported-by: Gwangun Jung <exsociety@gmail.com>
      Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
      Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 486bfb68d7b1af1256a72033a7afe57dcc4995aa
  Author: Pablo Neira Ayuso <pablo@netfilter.org>
  Date:   Sun Aug 21 16:32:44 2022 +0200

      netfilter: nft_tunnel: restrict it to netdev family
```

```
    [ Upstream commit 01e4092d53bc4fe122a6e4b6d664adbd57528ca3 ]

    Only allow to use this expression from NFPROTO_NETDEV family.

    Fixes: af308b94a2a4 ("netfilter: nf_tables: add tunnel support")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 02e8575a86f48800f58292ebc5fd0d91745a8da9
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 16:25:07 2022 +0200

    netfilter: nft_osf: restrict osf to ipv4, ipv6 and inet families

    [ Upstream commit 5f3b7aae14a706d0d7da9f9e39def52ff5fc3d39 ]

    As it was originally intended, restrict extension to supported families.

    Fixes: b96af92d6eaf ("netfilter: nf_tables: implement Passive OS fingerprint module in
nft_osf")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 98a621ef45e3605c7487f7fa6fec7df94697d6a2
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 12:41:33 2022 +0200

    netfilter: nf_tables: do not leave chain stats enabled on error

    [ Upstream commit 43eb8949cfdffa764b92bc6c54b87cbe5b0003fe ]

    Error might occur later in the nf_tables_addchain() codepath, enable
    static key only after transaction has been created.

    Fixes: 9f08ea848117 ("netfilter: nf_tables: keep chain counters away from hot path")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 09cd8ecf3107efc4dcfb124a3079635c33965002
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 11:55:19 2022 +0200

    netfilter: nft_payload: do not truncate csum_offset and csum_type

    [ Upstream commit 7044ab281febae9e2fa9b0b247693d6026166293 ]

    Instead report ERANGE if csum_offset is too long, and EOPNOTSUPP if type
    is not support.

    Fixes: 7ec3f7b47b8d ("netfilter: nft_payload: add packet mangling support")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0ee5c638e108de9fcf699f975a26747b09a0a5ac
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 11:47:04 2022 +0200

    netfilter: nft_payload: report ERANGE for too long offset and length

    [ Upstream commit 94254f990c07e9ddf1634e0b727fab821c3b5bf9 ]

    Instead of offset and length are truncation to u8, report ERANGE.

    Fixes: 96518518cc41 ("netfilter: add nftables")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
```

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

commit c8ebc3b8635f6ceeb850099eb65226689ecd3168
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 10:52:48 2022 +0200

    netfilter: nf_tables: make table handle allocation per-netns friendly

    [ Upstream commit ab482c6b66a4a8c0a8c0b0f577a785cf9ff1c2e2 ]

    mutex is per-netns, move table_netns to the pernet area.

    *read-write* to 0xffffffff883a01e8 of 8 bytes by task 6542 on cpu 0:
     nf_tables_newtable+0x6dc/0xc00 net/netfilter/nf_tables_api.c:1221
     nfnetlink_rcv_batch net/netfilter/nfnetlink.c:513 [inline]
     nfnetlink_rcv_skb_batch net/netfilter/nfnetlink.c:634 [inline]
     nfnetlink_rcv+0xa6a/0x13a0 net/netfilter/nfnetlink.c:652
     netlink_unicast_kernel net/netlink/af_netlink.c:1319 [inline]
     netlink_unicast+0x652/0x730 net/netlink/af_netlink.c:1345
     netlink_sendmsg+0x643/0x740 net/netlink/af_netlink.c:1921

    Fixes: f102d66b335a ("netfilter: nf_tables: use dedicated mutex to guard transactions")
    Reported-by: Abhishek Shah <abhishek.shah@columbia.edu>
    Reviewed-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b7dfe042ecece0f1460aa17ce4c7910ba1d91911
Author: Pablo Neira Ayuso <pablo@netfilter.org>
Date:   Sun Aug 21 10:28:25 2022 +0200

    netfilter: nf_tables: disallow updates of implicit chain

    [ Upstream commit 5dc52d83baac30decf5f3b371d5eb41dfa1d1412 ]

    Updates on existing implicit chain make no sense, disallow this.

    Fixes: d0e2c7de92c7 ("netfilter: nf_tables: add NFT_CHAIN_BINDING")
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c5101ebeb2edb19b8659030e56ec0ef3e80b7f3b
Author: Vikas Gupta <vikas.gupta@broadcom.com>
Date:   Mon Aug 22 11:06:54 2022 -0400

    bnxt_en: fix LRO/GRO_HW features in ndo_fix_features callback

    [ Upstream commit 366c304741729e64d778c80555d9eb422cf5cc89 ]

    LRO/GRO_HW should be disabled if there is an attached XDP program.
    BNXT_FLAG_TPA is the current setting of the LRO/GRO_HW.  Using
    BNXT_FLAG_TPA to disable LRO/GRO_HW will cause these features to be
    permanently disabled once they are disabled.

    Fixes: 1dc4c557bfed ("bnxt: adding bnxt_xdp_build_skb to build skb from multibuffer
xdp_buff")
    Signed-off-by: Vikas Gupta <vikas.gupta@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 2ec3dc278d97a10879cf763644f90fdd1ddc159b
Author: Vikas Gupta <vikas.gupta@broadcom.com>
Date:   Mon Aug 22 11:06:53 2022 -0400
```

```
    bnxt_en: fix NQ resource accounting during vf creation on 57500 chips

    [ Upstream commit 09a89cc59ad67794a11e1d3dd13c5b3172adcc51 ]

    There are 2 issues:

    1. We should decrement hw_resc->max_nqs instead of hw_resc->max_irqs
       with the number of NQs assigned to the VFs.  The IRQs are fixed
       on each function and cannot be re-assigned.  Only the NQs are being
       assigned to the VFs.

    2. vf_msix is the total number of NQs to be assigned to the VFs.  So
       we should decrement vf_msix from hw_resc->max_nqs.

    Fixes: b16b68918674 ("bnxt_en: Add SR-IOV support for 57500 chips.")
    Signed-off-by: Vikas Gupta <vikas.gupta@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 46195aec8631af94b8afc34fb1eadb09f02f2ec0
Author: Vikas Gupta <vikas.gupta@broadcom.com>
Date:   Mon Aug 22 11:06:52 2022 -0400

    bnxt_en: set missing reload flag in devlink features

    [ Upstream commit 574b2bb9692fd3d45ed631ac447176d4679f3010 ]

    Add missing devlink_set_features() API for callbacks reload_down
    and reload_up to function.

    Fixes: 228ea8c187d8 ("bnxt_en: implement devlink dev reload driver_reinit")
    Reviewed-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Signed-off-by: Vikas Gupta <vikas.gupta@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 51ca62d3274c710441cdfea37cfe5ac4296867c7
Author: Pavan Chebbi <pavan.chebbi@broadcom.com>
Date:   Mon Aug 22 11:06:51 2022 -0400

    bnxt_en: Use PAGE_SIZE to init buffer when multi buffer XDP is not in use

    [ Upstream commit 7dd3de7cb1d657a918c6b2bc673c71e318aa0c05 ]

    Using BNXT_PAGE_MODE_BUF_SIZE + offset as buffer length value is not
    sufficient when running single buffer XDP programs doing redirect
    operations. The stack will complain on missing skb tail room. Fix it
    by using PAGE_SIZE when calling xdp_init_buff() for single buffer
    programs.

    Fixes: b231c3f3414c ("bnxt: refactor bnxt_rx_xdp to separate
xdp_init_buff/xdp_prepare_buff")
    Reviewed-by: Somnath Kotur <somnath.kotur@broadcom.com>
    Signed-off-by: Pavan Chebbi <pavan.chebbi@broadcom.com>
    Signed-off-by: Michael Chan <michael.chan@broadcom.com>
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 343fed6b0daeb528ae5c9d4d84d9ff763ac95619
Author: Florian Westphal <fw@strlen.de>
Date:   Sat Aug 20 17:54:06 2022 +0200

    netfilter: nft_tproxy: restrict to prerouting hook
```

```
    [ Upstream commit 18bbc3213383a82b05383827f4b1b882e3f0a5a5 ]

    TPROXY is only allowed from prerouting, but nft_tproxy doesn't check this.
    This fixes a crash (null dereference) when using tproxy from e.g. output.

    Fixes: 4ed8eb6570a4 ("netfilter: nf_tables: Add native tproxy support")
    Reported-by: Shell Chen <xierch@gmail.com>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e53cfa017bf4575d0b948a8f45313ef66d897136
Author: Florian Westphal <fw@strlen.de>
Date:   Sat Aug 20 17:38:37 2022 +0200

    netfilter: ebtables: reject blobs that don't provide all entry points

    [ Upstream commit 7997eff82828304b780dc0a39707e1946d6f1ebf ]

    Harshit Mogalapalli says:
     In ebt_do_table() function dereferencing 'private->hook_entry[hook]'
     can lead to NULL pointer dereference. [..] Kernel panic:

    general protection fault, probably for non-canonical address 0xdffffc0000000005: 0000
[#1] PREEMPT SMP KASAN
    KASAN: null-ptr-deref in range [0x0000000000000028-0x000000000000002f]
    [..]
    RIP: 0010:ebt_do_table+0x1dc/0x1ce0
    Code: 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 5c 16 00 00 48 b8 00 00 00 00 00 fc ff df 49
8b 6c df 08 48 8d 7d 2c 48 89 fa 48 c1 ea 03 <0f> b6 14 02 48 89 f8 83 e0 07 83 c0 03 38 d0
7c 08 84 d2 0f 85 88
    [..]
    Call Trace:
     nf_hook_slow+0xb1/0x170
     __br_forward+0x289/0x730
     maybe_deliver+0x24b/0x380
     br_flood+0xc6/0x390
     br_dev_xmit+0xa2e/0x12c0

    For some reason ebtables rejects blobs that provide entry points that are
    not supported by the table, but what it should instead reject is the
    opposite: blobs that DO NOT provide an entry point supported by the table.

    t->valid_hooks is the bitmask of hooks (input, forward ...) that will see
    packets.  Providing an entry point that is not support is harmless
    (never called/used), but the inverse isn't: it results in a crash
    because the ebtables traverser doesn't expect a NULL blob for a location
    its receiving packets for.

    Instead of fixing all the individual checks, do what iptables is doing and
    reject all blobs that differ from the expected hooks.

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Reported-by: Harshit Mogalapalli <harshit.m.mogalapalli@oracle.com>
    Reported-by: syzkaller <syzkaller@googlegroups.com>
    Signed-off-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 223fbc2f5039f3004d0cda0e12d43649983cb752
Author: Maciej Żenczykowski <maze@google.com>
Date:   Sun Aug 21 06:08:08 2022 -0700

    net: ipvtap - add __init/__exit annotations to module init/exit funcs

    [ Upstream commit 4b2e3a17e9f279325712b79fb01d1493f9e3e005 ]
```

```
    Looks to have been left out in an oversight.

    Cc: Mahesh Bandewar <maheshb@google.com>
    Cc: Sainath Grandhi <sainath.grandhi@intel.com>
    Fixes: 235a9d89da97 ('ipvtap: IP-VLAN based tap driver')
    Signed-off-by: Maciej Żenczykowski <maze@google.com>
    Link: https://lore.kernel.org/r/20220821130808.12143-1-zenczykowski@gmail.com
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3ff47e5994207fa0a5ee99b8063ad5a03397919d
Author: Jonathan Toppins <jtoppins@redhat.com>
Date:   Fri Aug 19 11:15:13 2022 -0400

    bonding: 802.3ad: fix no transmission of LACPDUs

    [ Upstream commit d745b5062ad2b5da90a5e728d7ca884fc07315fd ]

    This is caused by the global variable ad_ticks_per_sec being zero as
    demonstrated by the reproducer script discussed below. This causes
    all timer values in __ad_timer_to_ticks to be zero, resulting
    in the periodic timer to never fire.

    To reproduce:
    Run the script in
    `tools/testing/selftests/drivers/net/bonding/bond-break-lacpdu-tx.sh` which
    puts bonding into a state where it never transmits LACPDUs.

    line 44: ip link add fbond type bond mode 4 miimon 200 \
                xmit_hash_policy 1 ad_actor_sys_prio 65535 lacp_rate fast
    setting bond param: ad_actor_sys_prio
    given:
        params.ad_actor_system = 0
    call stack:
        bond_option_ad_actor_sys_prio()
        -> bond_3ad_update_ad_actor_settings()
           -> set ad.system.sys_priority = bond->params.ad_actor_sys_prio
           -> ad.system.sys_mac_addr = bond->dev->dev_addr; because
                params.ad_actor_system == 0
    results:
         ad.system.sys_mac_addr = bond->dev->dev_addr

    line 48: ip link set fbond address 52:54:00:3B:7C:A6
    setting bond MAC addr
    call stack:
        bond->dev->dev_addr = new_mac

    line 52: ip link set fbond type bond ad_actor_sys_prio 65535
    setting bond param: ad_actor_sys_prio
    given:
        params.ad_actor_system = 0
    call stack:
        bond_option_ad_actor_sys_prio()
        -> bond_3ad_update_ad_actor_settings()
           -> set ad.system.sys_priority = bond->params.ad_actor_sys_prio
           -> ad.system.sys_mac_addr = bond->dev->dev_addr; because
                params.ad_actor_system == 0
    results:
         ad.system.sys_mac_addr = bond->dev->dev_addr

    line 60: ip link set veth1-bond down master fbond
    given:
        params.ad_actor_system = 0
        params.mode = BOND_MODE_8023AD
```

```
         ad.system.sys_mac_addr == bond->dev->dev_addr
     call stack:
         bond_enslave
         -> bond_3ad_initialize(); because first slave
            -> if ad.system.sys_mac_addr != bond->dev->dev_addr
               return
     results:
          Nothing is run in bond_3ad_initialize() because dev_addr equals
          sys_mac_addr leaving the global ad_ticks_per_sec zero as it is
          never initialized anywhere else.

     The if check around the contents of bond_3ad_initialize() is no longer
     needed due to commit 5ee14e6d336f ("bonding: 3ad: apply ad_actor settings
     changes immediately") which sets ad.system.sys_mac_addr if any one of
     the bonding parameters whos set function calls
     bond_3ad_update_ad_actor_settings(). This is because if
     ad.system.sys_mac_addr is zero it will be set to the current bond mac
     address, this causes the if check to never be true.

     Fixes: 5ee14e6d336f ("bonding: 3ad: apply ad_actor settings changes immediately")
     Signed-off-by: Jonathan Toppins <jtoppins@redhat.com>
     Acked-by: Jay Vosburgh <jay.vosburgh@canonical.com>
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit b8bd96a46c933f101e3b91fe62a37b8cce42f204
 Author: Sergei Antonov <saproj@gmail.com>
 Date:   Fri Aug 19 14:05:19 2022 +0300

     net: moxa: get rid of asymmetry in DMA mapping/unmapping

     [ Upstream commit 0ee7828dfc56e97d71e51e6374dc7b4eb2b6e081 ]

     Since priv->rx_mapping[i] is maped in moxart_mac_open(), we
     should unmap it from moxart_mac_stop(). Fixes 2 warnings.

     1. During error unwinding in moxart_mac_probe(): "goto init_fail;",
     then moxart_mac_free_memory() calls dma_unmap_single() with
     priv->rx_mapping[i] pointers zeroed.

     WARNING: CPU: 0 PID: 1 at kernel/dma/debug.c:963 check_unmap+0x704/0x980
     DMA-API: moxart-ethernet 92000000.mac: device driver tries to free DMA memory it has not
 allocated [device address=0x0000000000000000] [size=1600 bytes]
     CPU: 0 PID: 1 Comm: swapper Not tainted 5.19.0+ #60
     Hardware name: Generic DT based system
      unwind_backtrace from show_stack+0x10/0x14
      show_stack from dump_stack_lvl+0x34/0x44
      dump_stack_lvl from __warn+0xbc/0x1f0
      __warn from warn_slowpath_fmt+0x94/0xc8
      warn_slowpath_fmt from check_unmap+0x704/0x980
      check_unmap from debug_dma_unmap_page+0x8c/0x9c
      debug_dma_unmap_page from moxart_mac_free_memory+0x3c/0xa8
      moxart_mac_free_memory from moxart_mac_probe+0x190/0x218
      moxart_mac_probe from platform_probe+0x48/0x88
      platform_probe from really_probe+0xc0/0x2e4

     2. After commands:
      ip link set dev eth0 down
      ip link set dev eth0 up

     WARNING: CPU: 0 PID: 55 at kernel/dma/debug.c:570 add_dma_entry+0x204/0x2ec
     DMA-API: moxart-ethernet 92000000.mac: cacheline tracking EEXIST, overlapping mappings
 aren't supported
     CPU: 0 PID: 55 Comm: ip Not tainted 5.19.0+ #57
     Hardware name: Generic DT based system
```

```
       unwind_backtrace from show_stack+0x10/0x14
       show_stack from dump_stack_lvl+0x34/0x44
       dump_stack_lvl from __warn+0xbc/0x1f0
       __warn from warn_slowpath_fmt+0x94/0xc8
       warn_slowpath_fmt from add_dma_entry+0x204/0x2ec
       add_dma_entry from dma_map_page_attrs+0x110/0x328
       dma_map_page_attrs from moxart_mac_open+0x134/0x320
       moxart_mac_open from __dev_open+0x11c/0x1ec
       __dev_open from __dev_change_flags+0x194/0x22c
       __dev_change_flags from dev_change_flags+0x14/0x44
       dev_change_flags from devinet_ioctl+0x6d4/0x93c
       devinet_ioctl from inet_ioctl+0x1ac/0x25c

     v1 -> v2:
     Extraneous change removed.

     Fixes: 6c821bd9edc9 ("net: Add MOXA ART SoCs ethernet driver")
     Signed-off-by: Sergei Antonov <saproj@gmail.com>
     Reviewed-by: Andrew Lunn <andrew@lunn.ch>
     Link: https://lore.kernel.org/r/20220819110519.1230877-1-saproj@gmail.com
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 162571b77486beb0ecda05d7c2c23620269dffc7
  Author: Xiaolei Wang <xiaolei.wang@windriver.com>
  Date:   Fri Aug 19 16:24:51 2022 +0800

     net: phy: Don't WARN for PHY_READY state in mdio_bus_phy_resume()

     [ Upstream commit 6dbe852c379ff032a70a6b13a91914918c82cb07 ]

     For some MAC drivers, they set the mac_managed_pm to true in its
     ->ndo_open() callback. So before the mac_managed_pm is set to true,
     we still want to leverage the mdio_bus_phy_suspend()/resume() for
     the phy device suspend and resume. In this case, the phy device is
     in PHY_READY, and we shouldn't warn about this. It also seems that
     the check of mac_managed_pm in WARN_ON is redundant since we already
     check this in the entry of mdio_bus_phy_resume(), so drop it.

     Fixes: 744d23c71af3 ("net: phy: Warn about incorrect mdio_bus_phy_resume() state")
     Signed-off-by: Xiaolei Wang <xiaolei.wang@windriver.com>
     Acked-by: Florian Fainelli <f.fainelli@gmail.com>
     Link: https://lore.kernel.org/r/20220819082451.1992102-1-xiaolei.wang@windriver.com
     Signed-off-by: Jakub Kicinski <kuba@kernel.org>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

  commit 834a5483bfe08feac5cff1d561b01653b7b4eb10
  Author: Alex Elder <elder@linaro.org>
  Date:   Thu Aug 18 08:42:05 2022 -0500

     net: ipa: don't assume SMEM is page-aligned

     [ Upstream commit b8d4380365c515d8e0351f2f46d371738dd19be1 ]

     In ipa_smem_init(), a Qualcomm SMEM region is allocated (if needed)
     and then its virtual address is fetched using qcom_smem_get().  The
     physical address associated with that region is also fetched.

     The physical address is adjusted so that it is page-aligned, and an
     attempt is made to update the size of the region to compensate for
     any non-zero adjustment.

     But that adjustment isn't done properly.  The physical address is
     aligned twice, and as a result the size is never actually adjusted.
```

```
    Fix this by *not* aligning the "addr" local variable, and instead
    making the "phys" local variable be the adjusted "addr" value.

    Fixes: a0036bb413d5b ("net: ipa: define SMEM memory region for IPA")
    Signed-off-by: Alex Elder <elder@linaro.org>
    Link: https://lore.kernel.org/r/20220818134206.567618-1-elder@linaro.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 67426e99a1ac5c9ea61059614371bfecdcb44eea
Author: Vladimir Oltean <vladimir.oltean@nxp.com>
Date:   Thu Aug 18 17:32:50 2022 +0300

    net: dsa: microchip: keep compatibility with device tree blobs with no phy-mode

    [ Upstream commit 5fbb08eb7f945c7e8896ea39f03143ce66dfa4c7 ]

    DSA has multiple ways of specifying a MAC connection to an internal PHY.
    One requires a DT description like this:

            port@0 {
                    reg = <0>;
                    phy-handle = <&internal_phy>;
                    phy-mode = "internal";
            };

    (which is IMO the recommended approach, as it is the clearest
    description)

    but it is also possible to leave the specification as just:

            port@0 {
                    reg = <0>;
            }

    and if the driver implements ds->ops->phy_read and ds->ops->phy_write,
    the DSA framework "knows" it should create a ds->slave_mii_bus, and it
    should connect to a non-OF-based internal PHY on this MDIO bus, at an
    MDIO address equal to the port address.

    There is also an intermediary way of describing things:

            port@0 {
                    reg = <0>;
                    phy-handle = <&internal_phy>;
            };

    In case 2, DSA calls phylink_connect_phy() and in case 3, it calls
    phylink_of_phy_connect(). In both cases, phylink_create() has been
    called with a phy_interface_t of PHY_INTERFACE_MODE_NA, and in both
    cases, PHY_INTERFACE_MODE_NA is translated into phy->interface.

    It is important to note that phy_device_create() initializes
    dev->interface = PHY_INTERFACE_MODE_GMII, and so, when we use
    phylink_create(PHY_INTERFACE_MODE_NA), no one will override this, and we
    will end up with a PHY_INTERFACE_MODE_GMII interface inherited from the
    PHY.

    All this means that in order to maintain compatibility with device tree
    blobs where the phy-mode property is missing, we need to allow the
    "gmii" phy-mode and treat it as "internal".

    Fixes: 2c709e0bdad4 ("net: dsa: microchip: ksz8795: add phylink support")
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216320
    Reported-by: Craig McQueen <craig@mcqueen.id.au>
```

```
    Signed-off-by: Vladimir Oltean <vladimir.oltean@nxp.com>
    Reviewed-by: Alvin Šipraga <alsi@bang-olufsen.dk>
    Tested-by: Rasmus Villemoes <rasmus.villemoes@prevas.dk>
    Link: https://lore.kernel.org/r/20220818143250.2797111-1-vladimir.oltean@nxp.com
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit dfee8aec730dc9f2e74909fc9f150750160321ff
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:52 2022 +0530

    net: dsa: microchip: update the ksz_phylink_get_caps

    [ Upstream commit 7012033ce10e0968e6cb82709aa0ed7f2080b61e ]

    This patch assigns the phylink_get_caps in ksz8795 and ksz9477 to
    ksz_phylink_get_caps. And update their mac_capabilities in the
    respective ksz_dev_ops.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Reviewed-by: Vladimir Oltean <olteanv@gmail.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit bb015bf77a9e61541b3a2d3e4790965d2aa22d8d
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:50 2022 +0530

    net: dsa: microchip: move the port mirror to ksz_common

    [ Upstream commit 00a298bbc23876288b1cd04c38752d8e7ed53ae2 ]

    This patch updates the common port mirror add/del dsa_switch_ops in
    ksz_common.c. The individual switches implementation is executed based
    on the ksz_dev_ops function pointers.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Reviewed-by: Vladimir Oltean <olteanv@gmail.com>
    Reviewed-by: Florian Fainelli <f.fainelli@gmail.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d719d680a557810a34d8de0ac3431dc7bd79b3d1
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:49 2022 +0530

    net: dsa: microchip: move vlan functionality to ksz_common

    [ Upstream commit f0d997e31bb307c7aa046c4992c568547fd25195 ]

    This patch moves the vlan dsa_switch_ops such as vlan_add, vlan_del and
    vlan_filtering from the individual files ksz8795.c, ksz9477.c to
    ksz_common.c file.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Reviewed-by: Vladimir Oltean <olteanv@gmail.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 23fcd5216540d23c65588b5c6f60769f209ad8eb
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:47 2022 +0530

    net: dsa: microchip: move tag_protocol to ksz_common
```

```
    [ Upstream commit 534a0431e9e68959e2c0d71c141d5b911d66ad7c ]

    This patch move the dsa hook get_tag_protocol to ksz_common file. And
    the tag_protocol is returned based on the dev->chip_id.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eafb01efb8c745608d27bc851dbfd6fee3fc1b6b
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:46 2022 +0530

    net: dsa: microchip: move switch chip_id detection to ksz_common

    [ Upstream commit 91a98917a8839923d404a77c21646ca5fc9e330a ]

    KSZ87xx and KSZ88xx have chip_id representation at reg location 0. And
    KSZ9477 compatible switch and LAN937x switch have same chip_id detection
    at location 0x01 and 0x02. To have the common switch detect
    functionality for ksz switches, ksz_switch_detect function is
    introduced.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 422e808ba8a58b76e16dc7083e002d3cd7ffc9db
Author: Arun Ramadoss <arun.ramadoss@microchip.com>
Date:   Fri Jun 17 14:12:45 2022 +0530

    net: dsa: microchip: ksz9477: cleanup the ksz9477_switch_detect

    [ Upstream commit 27faa0aa85f6696d411bbbebaed9f0f723c2a175 ]

    The ksz9477_switch_detect performs the detecting the chip id from the
    location 0x00 and also check gigabit compatibility check & number of
    ports based on the register global_options0. To prepare the common ksz
    switch detect function, routine other than chip id read is moved to
    ksz9477_switch_init.

    Signed-off-by: Arun Ramadoss <arun.ramadoss@microchip.com>
    Reviewed-by: Vladimir Oltean <olteanv@gmail.com>
    Signed-off-by: Paolo Abeni <pabeni@redhat.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit eaa08e3c5abd1701c888438ae07d8edbce077642
Author: Maor Dickman <maord@nvidia.com>
Date:   Thu Aug 4 15:28:42 2022 +0300

    net/mlx5e: Fix wrong tc flag used when set hw-tc-offload off

    [ Upstream commit 550f96432e6f6770efdaee0e65239d61431062a1 ]

    The cited commit reintroduced the ability to set hw-tc-offload
    in switchdev mode by reusing NIC mode calls without modifying it
    to support both modes, this can cause an illegal memory access
    when trying to turn hw-tc-offload off.

    Fix this by using the right TC_FLAG when checking if tc rules
    are installed while disabling hw-tc-offload.

    Fixes: d3cbd4254df8 ("net/mlx5e: Add ndo_set_feature for uplink representor")
    Signed-off-by: Maor Dickman <maord@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
```

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

commit 160967199c5ee01b72d6769815bcdf624c3f45fb
Author: Aya Levin <ayal@nvidia.com>
Date:   Wed Jun 8 18:38:37 2022 +0300

    net/mlx5e: Fix wrong application of the LRO state

    [ Upstream commit 7b3707fc79044871ab8f3d5fa5e9603155bb5577 ]

    Driver caches packet merge type in mlx5e_params instance which must be
    in perfect sync with the netdev_feature's bit.
    Prior to this patch, in certain conditions (*) LRO state was set in
    mlx5e_params, while netdev_feature's bit was off. Causing the LRO to
    be applied on the RQs (HW level).

    (*) This can happen only on profile init (mlx5e_build_nic_params()),
    when RQ expect non-linear SKB and PCI is fast enough in comparison to
    link width.

    Solution: remove setting of packet merge type from
    mlx5e_build_nic_params() as netdev features are not updated.

    Fixes: 619a8f2a42f1 ("net/mlx5e: Use linear SKB in Striding RQ")
    Signed-off-by: Aya Levin <ayal@nvidia.com>
    Reviewed-by: Tariq Toukan <tariqt@nvidia.com>
    Reviewed-by: Maxim Mikityanskiy <maximmi@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit b0faef51599e2e848bce046b7c769a4f8dbeac33
Author: Moshe Shemesh <moshe@nvidia.com>
Date:   Wed Aug 3 10:49:23 2022 +0300

    net/mlx5: Avoid false positive lockdep warning by adding lock_class_key

    [ Upstream commit d59b73a66e5e0682442b6d7b4965364e57078b80 ]

    Add a lock_class_key per mlx5 device to avoid a false positive
    "possible circular locking dependency" warning by lockdep, on flows
    which lock more than one mlx5 device, such as adding SF.

    kernel log:
     ======================================================
     WARNING: possible circular locking dependency detected
     5.19.0-rc8+ #2 Not tainted
     ------------------------------------------------------
     kworker/u20:0/8 is trying to acquire lock:
     ffff88812dfe0d98 (&dev->intf_state_mutex){+.+.}-{3:3}, at: mlx5_init_one+0x2e/0x490
[mlx5_core]

     but task is already holding lock:
     ffff888101aa7898 (&(&notifier->n_head)->rwsem){++++}-{3:3}, at:
blocking_notifier_call_chain+0x5a/0x130

     which lock already depends on the new lock.

     the existing dependency chain (in reverse order) is:

     -> #1 (&(&notifier->n_head)->rwsem){++++}-{3:3}:
            down_write+0x90/0x150
            blocking_notifier_chain_register+0x53/0xa0
            mlx5_sf_table_init+0x369/0x4a0 [mlx5_core]
            mlx5_init_one+0x261/0x490 [mlx5_core]
            probe_one+0x430/0x680 [mlx5_core]
```

```
              local_pci_probe+0xd6/0x170
              work_for_cpu_fn+0x4e/0xa0
              process_one_work+0x7c2/0x1340
              worker_thread+0x6f6/0xec0
              kthread+0x28f/0x330
              ret_from_fork+0x1f/0x30

       -> #0 (&dev->intf_state_mutex){+.+.}-{3:3}:
              __lock_acquire+0x2fc7/0x6720
              lock_acquire+0x1c1/0x550
              __mutex_lock+0x12c/0x14b0
              mlx5_init_one+0x2e/0x490 [mlx5_core]
              mlx5_sf_dev_probe+0x29c/0x370 [mlx5_core]
              auxiliary_bus_probe+0x9d/0xe0
              really_probe+0x1e0/0xaa0
              __driver_probe_device+0x219/0x480
              driver_probe_device+0x49/0x130
              __device_attach_driver+0x1b8/0x280
              bus_for_each_drv+0x123/0x1a0
              __device_attach+0x1a3/0x460
              bus_probe_device+0x1a2/0x260
              device_add+0x9b1/0x1b40
              __auxiliary_device_add+0x88/0xc0
              mlx5_sf_dev_state_change_handler+0x67e/0x9d0 [mlx5_core]
              blocking_notifier_call_chain+0xd5/0x130
              mlx5_vhca_state_work_handler+0x2b0/0x3f0 [mlx5_core]
              process_one_work+0x7c2/0x1340
              worker_thread+0x59d/0xec0
              kthread+0x28f/0x330
              ret_from_fork+0x1f/0x30

       other info that might help us debug this:

       Possible unsafe locking scenario:

              CPU0                    CPU1
              ----                    ----
         lock(&(&notifier->n_head)->rwsem);
                                      lock(&dev->intf_state_mutex);
                                      lock(&(&notifier->n_head)->rwsem);
         lock(&dev->intf_state_mutex);

       *** DEADLOCK ***

      4 locks held by kworker/u20:0/8:
       #0: ffff888150612938 ((wq_completion)mlx5_events){+.+.}-{0:0}, at:
  process_one_work+0x6e2/0x1340
       #1: ffff888100cafdb8 ((work_completion)(&work->work)#3){+.+.}-{0:0}, at:
  process_one_work+0x70f/0x1340
       #2: ffff888101aa7898 (&(&notifier->n_head)->rwsem){++++}-{3:3}, at:
  blocking_notifier_call_chain+0x5a/0x130
       #3: ffff88813682d0e8 (&dev->mutex){....}-{3:3}, at:__device_attach+0x76/0x460

      stack backtrace:
      CPU: 6 PID: 8 Comm: kworker/u20:0 Not tainted 5.19.0-rc8+
      Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-
  prebuilt.qemu.org 04/01/2014
      Workqueue: mlx5_events mlx5_vhca_state_work_handler [mlx5_core]
      Call Trace:
       <TASK>
       dump_stack_lvl+0x57/0x7d
       check_noncircular+0x278/0x300
       ? print_circular_bug+0x460/0x460
       ? lock_chain_count+0x20/0x20
       ? register_lock_class+0x1880/0x1880
```

```
        __lock_acquire+0x2fc7/0x6720
        ? register_lock_class+0x1880/0x1880
        ? register_lock_class+0x1880/0x1880
        lock_acquire+0x1c1/0x550
        ? mlx5_init_one+0x2e/0x490 [mlx5_core]
        ? lockdep_hardirqs_on_prepare+0x400/0x400
        __mutex_lock+0x12c/0x14b0
        ? mlx5_init_one+0x2e/0x490 [mlx5_core]
        ? mlx5_init_one+0x2e/0x490 [mlx5_core]
        ? _raw_read_unlock+0x1f/0x30
        ? mutex_lock_io_nested+0x1320/0x1320
        ? __ioremap_caller.constprop.0+0x306/0x490
        ? mlx5_sf_dev_probe+0x269/0x370 [mlx5_core]
        ? iounmap+0x160/0x160
        mlx5_init_one+0x2e/0x490 [mlx5_core]
        mlx5_sf_dev_probe+0x29c/0x370 [mlx5_core]
        ? mlx5_sf_dev_remove+0x130/0x130 [mlx5_core]
        auxiliary_bus_probe+0x9d/0xe0
        really_probe+0x1e0/0xaa0
        __driver_probe_device+0x219/0x480
        ? auxiliary_match_id+0xe9/0x140
        driver_probe_device+0x49/0x130
        __device_attach_driver+0x1b8/0x280
        ? driver_allows_async_probing+0x140/0x140
        bus_for_each_drv+0x123/0x1a0
        ? bus_for_each_dev+0x1a0/0x1a0
        ? lockdep_hardirqs_on_prepare+0x286/0x400
        ? trace_hardirqs_on+0x2d/0x100
        __device_attach+0x1a3/0x460
        ? device_driver_attach+0x1e0/0x1e0
        ? kobject_uevent_env+0x22d/0xf10
        bus_probe_device+0x1a2/0x260
        device_add+0x9b1/0x1b40
        ? dev_set_name+0xab/0xe0
        ? __fw_devlink_link_to_suppliers+0x260/0x260
        ? memset+0x20/0x40
        ? lockdep_init_map_type+0x21a/0x7d0
        __auxiliary_device_add+0x88/0xc0
        ? auxiliary_device_init+0x86/0xa0
        mlx5_sf_dev_state_change_handler+0x67e/0x9d0 [mlx5_core]
        blocking_notifier_call_chain+0xd5/0x130
        mlx5_vhca_state_work_handler+0x2b0/0x3f0 [mlx5_core]
        ? mlx5_vhca_event_arm+0x100/0x100 [mlx5_core]
        ? lock_downgrade+0x6e0/0x6e0
        ? lockdep_hardirqs_on_prepare+0x286/0x400
        process_one_work+0x7c2/0x1340
        ? lockdep_hardirqs_on_prepare+0x400/0x400
        ? pwq_dec_nr_in_flight+0x230/0x230
        ? rwlock_bug.part.0+0x90/0x90
        worker_thread+0x59d/0xec0
        ? process_one_work+0x1340/0x1340
        kthread+0x28f/0x330
        ? kthread_complete_and_exit+0x20/0x20
        ret_from_fork+0x1f/0x30
        </TASK>

    Fixes: 6a3273217469 ("net/mlx5: SF, Port function state change support")
    Signed-off-by: Moshe Shemesh <moshe@nvidia.com>
    Reviewed-by: Shay Drory <shayd@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 0ea1abf797f01826fe90916053dbe8942274854c
Author: Roy Novich <royno@nvidia.com>
Date:   Wed Mar 30 17:59:27 2022 +0300
```

```
    net/mlx5: Fix cmd error logging for manage pages cmd

    [ Upstream commit 090f3e4f4089ab8041ed7d632c7851c2a42fcc10 ]

    When the driver unloads, give/reclaim_pages may fail as PF driver in
    teardown flow, current code will lead to the following kernel log print
    'failed reclaiming pages: err 0'.

    Fix it to get same behavior as before the cited commits,
    by calling mlx5_cmd_check before handling error state.
    mlx5_cmd_check will verify if the returned error is an actual error
    needed to be handled by the driver or not and will return an
    appropriate value.

    Fixes: 8d564292a166 ("net/mlx5: Remove redundant error on reclaim pages")
    Fixes: 4dac2f10ada0 ("net/mlx5: Remove redundant notify fail on give pages")
    Signed-off-by: Roy Novich <royno@nvidia.com>
    Reviewed-by: Moshe Shemesh <moshe@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit cddad6c98f5c7a48c9278673bd9391e5beadce81
Author: Vlad Buslov <vladbu@nvidia.com>
Date:   Thu Aug 11 13:46:36 2022 +0200

    net/mlx5: Disable irq when locking lag_lock

    [ Upstream commit 8e93f29422ffe968d7161f91acdf0d47f5323727 ]

    The lag_lock is taken from both process and softirq contexts which results
    lockdep warning[0] about potential deadlock. However, just disabling
    softirqs by using *_bh spinlock API is not enough since it will cause
    warning in some contexts where the lock is obtained with hard irqs
    disabled. To fix the issue save current irq state, disable them before
    obtaining the lock an re-enable irqs from saved state after releasing it.

    [0]:

    [Sun Aug  7 13:12:29 2022] ================================
    [Sun Aug  7 13:12:29 2022] WARNING: inconsistent lock state
    [Sun Aug  7 13:12:29 2022] 5.19.0_for_upstream_debug_2022_08_04_16_06 #1 Not tainted
    [Sun Aug  7 13:12:29 2022] --------------------------------
    [Sun Aug  7 13:12:29 2022] inconsistent {SOFTIRQ-ON-W} -> {IN-SOFTIRQ-W} usage.
    [Sun Aug  7 13:12:29 2022] swapper/0/0 [HC0[0]:SC1[1]:HE1:SE0] takes:
    [Sun Aug  7 13:12:29 2022] ffffffffa06dc0d8 (lag_lock){+.?.}-{2:2}, at:
mlx5_lag_is_shared_fdb+0x1f/0x120 [mlx5_core]
    [Sun Aug  7 13:12:29 2022] {SOFTIRQ-ON-W} state was registered at:
    [Sun Aug  7 13:12:29 2022]   lock_acquire+0x1c1/0x550
    [Sun Aug  7 13:12:29 2022]   _raw_spin_lock+0x2c/0x40
    [Sun Aug  7 13:12:29 2022]   mlx5_lag_add_netdev+0x13b/0x480 [mlx5_core]
    [Sun Aug  7 13:12:29 2022]   mlx5e_nic_enable+0x114/0x470 [mlx5_core]
    [Sun Aug  7 13:12:29 2022]   mlx5e_attach_netdev+0x30e/0x6a0 [mlx5_core]
    [Sun Aug  7 13:12:29 2022]   mlx5e_resume+0x105/0x160 [mlx5_core]
    [Sun Aug  7 13:12:29 2022]   mlx5e_probe+0xac3/0x14f0 [mlx5_core]
    [Sun Aug  7 13:12:29 2022]   auxiliary_bus_probe+0x9d/0xe0
    [Sun Aug  7 13:12:29 2022]   really_probe+0x1e0/0xaa0
    [Sun Aug  7 13:12:29 2022]   __driver_probe_device+0x219/0x480
    [Sun Aug  7 13:12:29 2022]   driver_probe_device+0x49/0x130
    [Sun Aug  7 13:12:29 2022]   __driver_attach+0x1e4/0x4d0
    [Sun Aug  7 13:12:29 2022]   bus_for_each_dev+0x11e/0x1a0
    [Sun Aug  7 13:12:29 2022]   bus_add_driver+0x3f4/0x5a0
    [Sun Aug  7 13:12:29 2022]   driver_register+0x20f/0x390
    [Sun Aug  7 13:12:29 2022]   __auxiliary_driver_register+0x14e/0x260
    [Sun Aug  7 13:12:29 2022]   mlx5e_init+0x38/0x90 [mlx5_core]
```

```
     [Sun Aug  7 13:12:29 2022]   vhost_iotlb_itree_augment_rotate+0xcb/0x180 [vhost_iotlb]
     [Sun Aug  7 13:12:29 2022]   do_one_initcall+0xc4/0x400
     [Sun Aug  7 13:12:29 2022]   do_init_module+0x18a/0x620
     [Sun Aug  7 13:12:29 2022]   load_module+0x563a/0x7040
     [Sun Aug  7 13:12:29 2022]   __do_sys_finit_module+0x122/0x1d0
     [Sun Aug  7 13:12:29 2022]   do_syscall_64+0x3d/0x90
     [Sun Aug  7 13:12:29 2022]   entry_SYSCALL_64_after_hwframe+0x46/0xb0
     [Sun Aug  7 13:12:29 2022] irq event stamp: 3596508
     [Sun Aug  7 13:12:29 2022] hardirqs last  enabled at (3596508): [<ffffffff813687c2>]
 __local_bh_enable_ip+0xa2/0x100
     [Sun Aug  7 13:12:29 2022] hardirqs last disabled at (3596507): [<ffffffff813687da>]
 __local_bh_enable_ip+0xba/0x100
     [Sun Aug  7 13:12:29 2022] softirqs last  enabled at (3596488): [<ffffffff81368a2a>]
 irq_exit_rcu+0x11a/0x170
     [Sun Aug  7 13:12:29 2022] softirqs last disabled at (3596495): [<ffffffff81368a2a>]
 irq_exit_rcu+0x11a/0x170
     [Sun Aug  7 13:12:29 2022]
                                other info that might help us debug this:
     [Sun Aug  7 13:12:29 2022]  Possible unsafe locking scenario:

     [Sun Aug  7 13:12:29 2022]        CPU0
     [Sun Aug  7 13:12:29 2022]        ----
     [Sun Aug  7 13:12:29 2022]   lock(lag_lock);
     [Sun Aug  7 13:12:29 2022]   <Interrupt>
     [Sun Aug  7 13:12:29 2022]     lock(lag_lock);
     [Sun Aug  7 13:12:29 2022]
                                 *** DEADLOCK ***

     [Sun Aug  7 13:12:29 2022] 4 locks held by swapper/0/0:
     [Sun Aug  7 13:12:29 2022]  #0: ffffffff84643260 (rcu_read_lock){....}-{1:2}, at:
 mlx5e_napi_poll+0x43/0x20a0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  #1: ffffffff84643260 (rcu_read_lock){....}-{1:2}, at:
 netif_receive_skb_list_internal+0x2d7/0xd60
     [Sun Aug  7 13:12:29 2022]  #2: ffff888144a18b58 (&br->hash_lock){+.-.}-{2:2}, at:
 br_fdb_update+0x301/0x570
     [Sun Aug  7 13:12:29 2022]  #3: ffffffff84643260 (rcu_read_lock){....}-{1:2}, at:
 atomic_notifier_call_chain+0x5/0x1d0
     [Sun Aug  7 13:12:29 2022]
                                stack backtrace:
     [Sun Aug  7 13:12:29 2022] CPU: 0 PID: 0 Comm: swapper/0 Not tainted
 5.19.0_for_upstream_debug_2022_08_04_16_06 #1
     [Sun Aug  7 13:12:29 2022] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-
 1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014
     [Sun Aug  7 13:12:29 2022] Call Trace:
     [Sun Aug  7 13:12:29 2022]  <IRQ>
     [Sun Aug  7 13:12:29 2022]  dump_stack_lvl+0x57/0x7d
     [Sun Aug  7 13:12:29 2022]  mark_lock.part.0.cold+0x5f/0x92
     [Sun Aug  7 13:12:29 2022]  ? lock_chain_count+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? unwind_next_frame+0x1c4/0x1b50
     [Sun Aug  7 13:12:29 2022]  ? secondary_startup_64_no_verify+0xcd/0xdb
     [Sun Aug  7 13:12:29 2022]  ? mlx5e_napi_poll+0x4e9/0x20a0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? mlx5e_napi_poll+0x4e9/0x20a0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? stack_access_ok+0x1d0/0x1d0
     [Sun Aug  7 13:12:29 2022]  ? start_kernel+0x3a7/0x3c5
     [Sun Aug  7 13:12:29 2022]  __lock_acquire+0x1260/0x6720
     [Sun Aug  7 13:12:29 2022]  ? lock_chain_count+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? lock_chain_count+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? register_lock_class+0x1880/0x1880
     [Sun Aug  7 13:12:29 2022]  ? mark_lock.part.0+0xed/0x3060
     [Sun Aug  7 13:12:29 2022]  ? stack_trace_save+0x91/0xc0
     [Sun Aug  7 13:12:29 2022]  lock_acquire+0x1c1/0x550
     [Sun Aug  7 13:12:29 2022]  ? mlx5_lag_is_shared_fdb+0x1f/0x120 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? lockdep_hardirqs_on_prepare+0x400/0x400
     [Sun Aug  7 13:12:29 2022]  ? __lock_acquire+0xd6f/0x6720
     [Sun Aug  7 13:12:29 2022]  _raw_spin_lock+0x2c/0x40
```

```
     [Sun Aug  7 13:12:29 2022]  ? mlx5_lag_is_shared_fdb+0x1f/0x120 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  mlx5_lag_is_shared_fdb+0x1f/0x120 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  mlx5_esw_bridge_rep_vport_num_vhca_id_get+0x1a0/0x600
  [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? mlx5_esw_bridge_update_work+0x90/0x90 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? lock_acquire+0x1c1/0x550
     [Sun Aug  7 13:12:29 2022]  mlx5_esw_bridge_switchdev_event+0x185/0x8f0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? mlx5_esw_bridge_port_obj_attr_set+0x3e0/0x3e0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? check_chain_key+0x24a/0x580
     [Sun Aug  7 13:12:29 2022]  atomic_notifier_call_chain+0xd7/0x1d0
     [Sun Aug  7 13:12:29 2022]  br_switchdev_fdb_notify+0xea/0x100
     [Sun Aug  7 13:12:29 2022]  ? br_switchdev_set_port_flag+0x310/0x310
     [Sun Aug  7 13:12:29 2022]  fdb_notify+0x11b/0x150
     [Sun Aug  7 13:12:29 2022]  br_fdb_update+0x34c/0x570
     [Sun Aug  7 13:12:29 2022]  ? lock_chain_count+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? br_fdb_add_local+0x50/0x50
     [Sun Aug  7 13:12:29 2022]  ? br_allowed_ingress+0x5f/0x1070
     [Sun Aug  7 13:12:29 2022]  ? check_chain_key+0x24a/0x580
     [Sun Aug  7 13:12:29 2022]  br_handle_frame_finish+0x786/0x18e0
     [Sun Aug  7 13:12:29 2022]  ? check_chain_key+0x24a/0x580
     [Sun Aug  7 13:12:29 2022]  ? br_handle_local_finish+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? __lock_acquire+0xd6f/0x6720
     [Sun Aug  7 13:12:29 2022]  ? sctp_inet_bind_verify+0x4d/0x190
     [Sun Aug  7 13:12:29 2022]  ? xlog_unpack_data+0x2e0/0x310
     [Sun Aug  7 13:12:29 2022]  ? br_handle_local_finish+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  br_nf_hook_thresh+0x227/0x380 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? setup_pre_routing+0x460/0x460 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? br_handle_local_finish+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? br_nf_pre_routing_ipv6+0x48b/0x69c [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  br_nf_pre_routing_finish_ipv6+0x5c2/0xbf0 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? br_handle_local_finish+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  br_nf_pre_routing_ipv6+0x4c6/0x69c [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? br_validate_ipv6+0x9e0/0x9e0 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? br_nf_forward_arp+0xb70/0xb70 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  ? br_nf_pre_routing+0xacf/0x1160 [br_netfilter]
     [Sun Aug  7 13:12:29 2022]  br_handle_frame+0x8a9/0x1270
     [Sun Aug  7 13:12:29 2022]  ? br_handle_frame_finish+0x18e0/0x18e0
     [Sun Aug  7 13:12:29 2022]  ? register_lock_class+0x1880/0x1880
     [Sun Aug  7 13:12:29 2022]  ? br_handle_local_finish+0x20/0x20
     [Sun Aug  7 13:12:29 2022]  ? bond_handle_frame+0xf9/0xac0 [bonding]
     [Sun Aug  7 13:12:29 2022]  ? br_handle_frame_finish+0x18e0/0x18e0
     [Sun Aug  7 13:12:29 2022]  __netif_receive_skb_core+0x7c0/0x2c70
     [Sun Aug  7 13:12:29 2022]  ? check_chain_key+0x24a/0x580
     [Sun Aug  7 13:12:29 2022]  ? generic_xdp_tx+0x5b0/0x5b0
     [Sun Aug  7 13:12:29 2022]  ? __lock_acquire+0xd6f/0x6720
     [Sun Aug  7 13:12:29 2022]  ? register_lock_class+0x1880/0x1880
     [Sun Aug  7 13:12:29 2022]  ? check_chain_key+0x24a/0x580
     [Sun Aug  7 13:12:29 2022]  __netif_receive_skb_list_core+0x2d7/0x8a0
     [Sun Aug  7 13:12:29 2022]  ? lock_acquire+0x1c1/0x550
     [Sun Aug  7 13:12:29 2022]  ? process_backlog+0x960/0x960
     [Sun Aug  7 13:12:29 2022]  ? lockdep_hardirqs_on_prepare+0x129/0x400
     [Sun Aug  7 13:12:29 2022]  ? kvm_clock_get_cycles+0x14/0x20
     [Sun Aug  7 13:12:29 2022]  netif_receive_skb_list_internal+0x5f4/0xd60
     [Sun Aug  7 13:12:29 2022]  ? do_xdp_generic+0x150/0x150
     [Sun Aug  7 13:12:29 2022]  ? mlx5e_poll_rx_cq+0xf6b/0x2960 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? mlx5e_poll_ico_cq+0x3d/0x1590 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  napi_complete_done+0x188/0x710
     [Sun Aug  7 13:12:29 2022]  mlx5e_napi_poll+0x4e9/0x20a0 [mlx5_core]
     [Sun Aug  7 13:12:29 2022]  ? __queue_work+0x53c/0xeb0
     [Sun Aug  7 13:12:29 2022]  __napi_poll+0x9f/0x540
     [Sun Aug  7 13:12:29 2022]  net_rx_action+0x420/0xb70
     [Sun Aug  7 13:12:29 2022]  ? napi_threaded_poll+0x470/0x470
     [Sun Aug  7 13:12:29 2022]  ? __common_interrupt+0x79/0x1a0
     [Sun Aug  7 13:12:29 2022]  __do_softirq+0x271/0x92c
     [Sun Aug  7 13:12:29 2022]  irq_exit_rcu+0x11a/0x170
```

```
     [Sun Aug  7 13:12:29 2022]  common_interrupt+0x7d/0xa0
     [Sun Aug  7 13:12:29 2022]  </IRQ>
     [Sun Aug  7 13:12:29 2022]  <TASK>
     [Sun Aug  7 13:12:29 2022]  asm_common_interrupt+0x22/0x40
     [Sun Aug  7 13:12:29 2022] RIP: 0010:default_idle+0x42/0x60
     [Sun Aug  7 13:12:29 2022] Code: c1 83 e0 07 48 c1 e9 03 83 c0 03 0f b6 14 11 38 d0 7c
 04 84 d2 75 14 8b 05 6b f1 22 02 85 c0 7e 07 0f 00 2d 80 3b 4a 00 fb f4 <c3> 48 c7 c7 e0 07
 7e 85 e8 21 bd 40 fe eb de 66 66 2e 0f 1f 84 00
     [Sun Aug  7 13:12:29 2022] RSP: 0018:ffffffff84407e18 EFLAGS: 00000242
     [Sun Aug  7 13:12:29 2022] RAX: 0000000000000001 RBX: ffffffff84ec4a68 RCX:
 1fffffff0afc0fc
     [Sun Aug  7 13:12:29 2022] RDX: 0000000000000004 RSI: 0000000000000000 RDI:
 ffffffff835b1fac
     [Sun Aug  7 13:12:29 2022] RBP: 0000000000000000 R08: 0000000000000001 R09:
 ffff8884d2c44ac3
     [Sun Aug  7 13:12:29 2022] R10: ffffed109a588958 R11: 00000000ffffffff R12:
 0000000000000000
     [Sun Aug  7 13:12:29 2022] R13: ffffffff84efac20 R14: 0000000000000000 R15:
 dffffc0000000000
     [Sun Aug  7 13:12:29 2022]  ? default_idle_call+0xcc/0x460
     [Sun Aug  7 13:12:29 2022]  default_idle_call+0xec/0x460
     [Sun Aug  7 13:12:29 2022]  do_idle+0x394/0x450
     [Sun Aug  7 13:12:29 2022]  ? arch_cpu_idle_exit+0x40/0x40
     [Sun Aug  7 13:12:29 2022]  cpu_startup_entry+0x19/0x20
     [Sun Aug  7 13:12:29 2022]  rest_init+0x156/0x250
     [Sun Aug  7 13:12:29 2022]  arch_call_rest_init+0xf/0x15
     [Sun Aug  7 13:12:29 2022]  start_kernel+0x3a7/0x3c5
     [Sun Aug  7 13:12:29 2022]  secondary_startup_64_no_verify+0xcd/0xdb
     [Sun Aug  7 13:12:29 2022]  </TASK>

     Fixes: ff9b7521468b ("net/mlx5: Bridge, support LAG")
     Signed-off-by: Vlad Buslov <vladbu@nvidia.com>
     Reviewed-by: Mark Bloch <mbloch@nvidia.com>
     Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 3325cb4f2d071b87d33cc8bc3ec271fbe4564aa8
 Author: Eli Cohen <elic@nvidia.com>
 Date:   Sun Aug 7 08:25:28 2022 +0300

     net/mlx5: Eswitch, Fix forwarding decision to uplink

     [ Upstream commit 942fca7e762be39204e5926e91a288a343a97c72 ]

     Make sure to modify the rule for uplink forwarding only for the case
     where destination vport number is MLX5_VPORT_UPLINK.

     Fixes: 94db33177819 ("net/mlx5: Support multiport eswitch mode")
     Signed-off-by: Eli Cohen <elic@nvidia.com>
     Reviewed-by: Maor Dickman <maord@nvidia.com>
     Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
     Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 4c040acf5744e87a7b3490f9ec8bedd0d15c9f29
 Author: Eli Cohen <elic@nvidia.com>
 Date:   Tue Aug 2 19:45:36 2022 +0300

     net/mlx5: LAG, fix logic over MLX5_LAG_FLAG_NDEVS_READY

     [ Upstream commit a6e675a66175869b7d87c0e1dd0ddf93e04f8098 ]

     Only set MLX5_LAG_FLAG_NDEVS_READY if both netdevices are registered.
     Doing so guarantees that both ldev->pf[MLX5_LAG_P0].dev and
     ldev->pf[MLX5_LAG_P1].dev have valid pointers when
     MLX5_LAG_FLAG_NDEVS_READY is set.
```

```
    The core issue is asymmetry in setting MLX5_LAG_FLAG_NDEVS_READY and
    clearing it. Setting it is done wrongly when both
    ldev->pf[MLX5_LAG_P0].dev and ldev->pf[MLX5_LAG_P1].dev are set;
    clearing it is done right when either of ldev->pf[i].netdev is cleared.

    Consider the following scenario:
    1. PF0 loads and sets ldev->pf[MLX5_LAG_P0].dev to a valid pointer
    2. PF1 loads and sets both ldev->pf[MLX5_LAG_P1].dev and
       ldev->pf[MLX5_LAG_P1].netdev with valid pointers. This results in
       MLX5_LAG_FLAG_NDEVS_READY is set.
    3. PF0 is unloaded before setting dev->pf[MLX5_LAG_P0].netdev.
       MLX5_LAG_FLAG_NDEVS_READY remains set.

    Further execution of mlx5_do_bond() will result in null pointer
    dereference when calling mlx5_lag_is_multipath()

    This patch fixes the following call trace actually encountered:

    [ 1293.475195] BUG: kernel NULL pointer dereference, address: 00000000000009a8
    [ 1293.478756] #PF: supervisor read access in kernel mode
    [ 1293.481320] #PF: error_code(0x0000) - not-present page
    [ 1293.483686] PGD 0 P4D 0
    [ 1293.484434] Oops: 0000 [#1] SMP PTI
    [ 1293.485377] CPU: 1 PID: 23690 Comm: kworker/u16:2 Not tainted 5.18.0-
rc5_for_upstream_min_debug_2022_05_05_10_13 #1
    [ 1293.488039] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-
gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014
    [ 1293.490836] Workqueue: mlx5_lag mlx5_do_bond_work [mlx5_core]
    [ 1293.492448] RIP: 0010:mlx5_lag_is_multipath+0x5/0x50 [mlx5_core]
    [ 1293.494044] Code: e8 70 40 ff e0 48 8b 14 24 48 83 05 5c 1a 1b 00 01 e9 19 ff ff ff
48 83 05 47 1a 1b 00 01 eb d7 0f 1f 44 00 00 0f 1f 44 00 00 <48> 8b 87 a8 09 00 00 48 85 c0
74 26 48 83 05 a7 1b 1b 00 01 41 b8
    [ 1293.498673] RSP: 0018:ffff88811b2fbe40 EFLAGS: 00010202
    [ 1293.500152] RAX: ffff88818a94e1c0 RBX: ffff888165eca6c0 RCX: 0000000000000000
    [ 1293.501841] RDX: 0000000000000001 RSI: ffff88818a94e1c0 RDI: 0000000000000000
    [ 1293.503585] RBP: 0000000000000000 R08: ffff888119886740 R09: ffff888165eca73c
    [ 1293.505286] R10: 0000000000000018 R11: 0000000000000018 R12: ffff88818a94e1c0
    [ 1293.506979] R13: ffff888112729800 R14: 0000000000000000 R15: ffff888112729858
    [ 1293.508753] FS:  0000000000000000(0000) GS:ffff88852cc40000(0000)
knlGS:0000000000000000
    [ 1293.510782] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [ 1293.512265] CR2: 00000000000009a8 CR3: 00000001032d4002 CR4: 0000000000370ea0
    [ 1293.514001] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [ 1293.515806] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

    Fixes: 8a66e4585979 ("net/mlx5: Change ownership model for lag")
    Signed-off-by: Eli Cohen <elic@nvidia.com>
    Reviewed-by: Maor Dickman <maord@nvidia.com>
    Reviewed-by: Mark Bloch <mbloch@nvidia.com>
    Signed-off-by: Saeed Mahameed <saeedm@nvidia.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 1155eb7baf1b9b0ddc50fd1a8fb8b928fc5961dd
Author: Vlad Buslov <vladbu@nvidia.com>
Date:   Fri Jul 15 21:41:48 2022 +0200

    net/mlx5e: Properly disable vlan strip on non-UL reps

    [ Upstream commit f37044fd759b6bc40b6398a978e0b1acdf717372 ]

    When querying mlx5 non-uplink representors capabilities with ethtool
    rx-vlan-offload is marked as "off [fixed]". However, it is actually always
    enabled because mlx5e_params->vlan_strip_disable is 0 by default when
    initializing struct mlx5e_params instance. Fix the issue by explicitly
```

```
    setting the vlan_strip_disable to 'true' for non-uplink representors.

    Fixes: cb67b832921c ("net/mlx5e: Introduce SRIOV VF representors")
    Signed-off-by: Vlad Buslov <vladbu@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 952efbc7a06f60eb6e7d2813dc4d29174fe255c4
Author: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
Date:   Thu Aug 11 20:21:49 2022 +0200

    ice: xsk: use Rx ring's XDP ring when picking NAPI context

    [ Upstream commit 9ead7e74bfd6dd54db12ef133b8604add72511de ]

    Ice driver allocates per cpu XDP queues so that redirect path can safely
    use smp_processor_id() as an index to the array. At the same time
    though, XDP rings are used to pick NAPI context to call napi_schedule()
    or set NAPIF_STATE_MISSED. When user reduces queue count, say to 8, and
    num_possible_cpus() of underlying platform is 44, then this means queue
    vectors with correlated NAPI contexts will carry several XDP queues.

    This in turn can result in a broken behavior where NAPI context of
    interest will never be scheduled and AF_XDP socket will not process any
    traffic.

    To fix this, let us change the way how XDP rings are assigned to Rx
    rings and use this information later on when setting
    ice_tx_ring::xsk_pool pointer. For each Rx ring, grab the associated
    queue vector and walk through Tx ring's linked list. Once we stumble
    upon XDP ring in it, assign this ring to ice_rx_ring::xdp_ring.

    Previous [0] approach of fixing this issue was for txonly scenario
    because of the described grouping of XDP rings across queue vectors. So,
    relying on Rx ring meant that NAPI context could be scheduled with a
    queue vector without XDP ring with associated XSK pool.

    [0]: https://lore.kernel.org/netdev/20220707161128.54215-1-maciej.fijalkowski@intel.com/

    Fixes: 2d4238f55697 ("ice: Add support for AF_XDP")
    Fixes: 22bf877e528f ("ice: introduce XDP_TX fallback path")
    Signed-off-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Tested-by: George Kuruvinakunnel <george.kuruvinakunnel@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 03a3f29fe5b1751ad9b5c892c894183e75a6e4c4
Author: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
Date:   Thu Aug 11 20:21:48 2022 +0200

    ice: xsk: prohibit usage of non-balanced queue id

    [ Upstream commit 5a42f112d367bb4700a8a41f5c12724fde6bfbb9 ]

    Fix the following scenario:
    1. ethtool -L $IFACE rx 8 tx 96
    2. xdpsock -q 10 -t -z

    Above refers to a case where user would like to attach XSK socket in
    txonly mode at a queue id that does not have a corresponding Rx queue.
    At this moment ice's XSK logic is tightly bound to act on a "queue pair",
    e.g. both Tx and Rx queues at a given queue id are disabled/enabled and
    both of them will get XSK pool assigned, which is broken for the presented
    queue configuration. This results in the splat included at the bottom,
```

```
    which is basically an OOB access to Rx ring array.

    To fix this, allow using the ids only in scope of "combined" queues
    reported by ethtool. However, logic should be rewritten to allow such
    configurations later on, which would end up as a complete rewrite of the
    control path, so let us go with this temporary fix.

    [420160.558008] BUG: kernel NULL pointer dereference, address: 0000000000000082
    [420160.566359] #PF: supervisor read access in kernel mode
    [420160.572657] #PF: error_code(0x0000) - not-present page
    [420160.579002] PGD 0 P4D 0
    [420160.582756] Oops: 0000 [#1] PREEMPT SMP NOPTI
    [420160.588396] CPU: 10 PID: 21232 Comm: xdpsock Tainted: G           OE     5.19.0-rc7+
#10
    [420160.597893] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS
SE5C620.86B.02.01.0008.031920191559 03/19/2019
    [420160.609894] RIP: 0010:ice_xsk_pool_setup+0x44/0x7d0 [ice]
    [420160.616968] Code: f3 48 83 ec 40 48 8b 4f 20 48 8b 3f 65 48 8b 04 25 28 00 00 00 48
89 44 24 38 31 c0 48 8d 04 ed 00 00 00 00 48 01 c1 48 8b 11 <0f> b7 92 82 00 00 00 48 85 d2
0f 84 2d 75 00 00 48 8d 72 ff 48 85
    [420160.639421] RSP: 0018:ffffc9002d2afd48 EFLAGS: 00010282
    [420160.646650] RAX: 0000000000000050 RBX: ffff88811d8bdd00 RCX: ffff888112c14ff8
    [420160.655893] RDX: 0000000000000000 RSI: ffff88811d8bdd00 RDI: ffff888109861000
    [420160.665166] RBP: 000000000000000a R08: 000000000000000a R09: 0000000000000000
    [420160.674493] R10: 000000000000889f R11: 0000000000000000 R12: 000000000000000a
    [420160.683833] R13: 000000000000000a R14: 0000000000000000 R15: ffff888117611828
    [420160.693211] FS:  00007fa869fc1f80(0000) GS:ffff8897e0880000(0000)
knlGS:0000000000000000
    [420160.703645] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [420160.711783] CR2: 0000000000000082 CR3: 00000001d076c001 CR4: 00000000007706e0
    [420160.721399] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [420160.731045] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [420160.740707] PKRU: 55555554
    [420160.745960] Call Trace:
    [420160.750962]  <TASK>
    [420160.755597]  ? kmalloc_large_node+0x79/0x90
    [420160.762703]  ? __kmalloc_node+0x3f5/0x4b0
    [420160.769341]  xp_assign_dev+0xfd/0x210
    [420160.775661]  ? shmem_file_read_iter+0x29a/0x420
    [420160.782896]  xsk_bind+0x152/0x490
    [420160.788943]  __sys_bind+0xd0/0x100
    [420160.795097]  ? exit_to_user_mode_prepare+0x20/0x120
    [420160.802801]  __x64_sys_bind+0x16/0x20
    [420160.809298]  do_syscall_64+0x38/0x90
    [420160.815741]  entry_SYSCALL_64_after_hwframe+0x63/0xcd
    [420160.823731] RIP: 0033:0x7fa86a0dd2fb
    [420160.830264] Code: c3 66 0f 1f 44 00 00 48 8b 15 69 8b 0c 00 f7 d8 64 89 02 b8 ff ff
ff ff eb bc 0f 1f 44 00 00 f3 0f 1e fa b8 31 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48
8b 0d 3d 8b 0c 00 f7 d8 64 89 01 48
    [420160.855410] RSP: 002b:00007ffc1146f618 EFLAGS: 00000246 ORIG_RAX: 0000000000000031
    [420160.866366] RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007fa86a0dd2fb
    [420160.876957] RDX: 0000000000000010 RSI: 00007ffc1146f680 RDI: 0000000000000003
    [420160.887604] RBP: 000055d7113a0520 R08: 00007fa868fb8000 R09: 0000000080000000
    [420160.898293] R10: 0000000000008001 R11: 0000000000000246 R12: 000055d7113a04e0
    [420160.909038] R13: 000055d7113a0320 R14: 000000000000000a R15: 0000000000000000
    [420160.919817]  </TASK>
    [420160.925659] Modules linked in: ice(OE) af_packet binfmt_misc nls_iso8859_1 ipmi_ssif
intel_rapl_msr intel_rapl_common x86_pkg_temp_thermal intel_powerclamp mei_me coretemp
ioatdma mei ipmi_si wmi ipmi_msghandler acpi_pad acpi_power_meter ip_tables x_tables autofs4
ixgbe i40e crct10dif_pclmul crc32_pclmul ghash_clmulni_intel aesni_intel crypto_simd cryptd
ahci mdio dca libahci lpc_ich [last unloaded: ice]
    [420160.977576] CR2: 0000000000000082
    [420160.985037] ---[ end trace 0000000000000000 ]---
    [420161.097724] RIP: 0010:ice_xsk_pool_setup+0x44/0x7d0 [ice]
    [420161.107341] Code: f3 48 83 ec 40 48 8b 4f 20 48 8b 3f 65 48 8b 04 25 28 00 00 00 48
```

```
 89 44 24 38 31 c0 48 8d 04 ed 00 00 00 00 48 01 c1 48 8b 11 <0f> b7 92 82 00 00 00 48 85 d2
 0f 84 2d 75 00 00 48 8d 72 ff 48 85
    [420161.134741] RSP: 0018:ffffc9002d2afd48 EFLAGS: 00010282
    [420161.144274] RAX: 0000000000000050 RBX: ffff88811d8bdd00 RCX: ffff888112c14ff8
    [420161.155690] RDX: 0000000000000000 RSI: ffff88811d8bdd00 RDI: ffff888109861000
    [420161.168088] RBP: 000000000000000a R08: 000000000000000a R09: 0000000000000000
    [420161.179295] R10: 000000000000889f R11: 0000000000000000 R12: 000000000000000a
    [420161.190420] R13: 000000000000000a R14: 0000000000000000 R15: ffff888117611828
    [420161.201505] FS:  00007fa869fc1f80(0000) GS:ffff8897e0880000(0000)
 knlGS:0000000000000000
    [420161.213628] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    [420161.223413] CR2: 0000000000000082 CR3: 00000001d076c001 CR4: 00000000007706e0
    [420161.234653] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    [420161.245893] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    [420161.257052] PKRU: 55555554

    Fixes: 2d4238f55697 ("ice: Add support for AF_XDP")
    Signed-off-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
    Tested-by: George Kuruvinakunnel <george.kuruvinakunnel@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 2c71f5d55a86fd5969428abf525c1ae6b1c7b0f5
 Author: Duoming Zhou <duoming@zju.edu.cn>
 Date:   Thu Aug 18 17:06:21 2022 +0800

    nfc: pn533: Fix use-after-free bugs caused by pn532_cmd_timeout

    [ Upstream commit f1e941dbf80a9b8bab0bffbc4cbe41cc7f4c6fb6 ]

    When the pn532 uart device is detaching, the pn532_uart_remove()
    is called. But there are no functions in pn532_uart_remove() that
    could delete the cmd_timeout timer, which will cause use-after-free
    bugs. The process is shown below:

        (thread 1)                  |        (thread 2)
                                    |  pn532_uart_send_frame
    pn532_uart_remove               |    mod_timer(&pn532->cmd_timeout,...)
      ...                           |    (wait a time)
      kfree(pn532) //FREE           |    pn532_cmd_timeout
                                    |      pn532_uart_send_frame
                                    |        pn532->... //USE

    This patch adds del_timer_sync() in pn532_uart_remove() in order to
    prevent the use-after-free bugs. What's more, the pn53x_unregister_nfc()
    is well synchronized, it sets nfc_dev->shutting_down to true and there
    are no syscalls could restart the cmd_timeout timer.

    Fixes: c656aa4c27b1 ("nfc: pn533: add UART phy driver")
    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 edbcbe37c31ab27583b354134a65ea5d5a975346
 Author: Hayes Wang <hayeswang@realtek.com>
 Date:   Thu Aug 18 16:06:20 2022 +0800

    r8152: fix the RX FIFO settings when suspending

    [ Upstream commit b75d612014447e04abdf0e37ffb8f2fd8b0b49d6 ]

    The RX FIFO would be changed when suspending, so the related settings
    have to be modified, too. Otherwise, the flow control would work
    abnormally.
```

```
    BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=216333
    Reported-by: Mark Blakeney <mark.blakeney@bullet-systems.net>
    Fixes: cdf0b86b250f ("r8152: fix a WOL issue")
    Signed-off-by: Hayes Wang <hayeswang@realtek.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 3eb8eb6e2e2a1e6420f96de7544f62fe67444bfa
Author: Hayes Wang <hayeswang@realtek.com>
Date:   Thu Aug 18 16:06:19 2022 +0800

    r8152: fix the units of some registers for RTL8156A

    [ Upstream commit 6dc4df12d741c0fe8f885778a43039e0619b9cd9 ]

    The units of PLA_RX_FIFO_FULL and PLA_RX_FIFO_EMPTY are 16 bytes.

    Fixes: 195aae321c82 ("r8152: support new chips")
    Signed-off-by: Hayes Wang <hayeswang@realtek.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9197ca40fd9de265caedba70d0cb5814c4e45952
Author: Bernard Pidoux <f6bvp@free.fr>
Date:   Thu Aug 18 02:02:13 2022 +0200

    rose: check NULL rose_loopback_neigh->loopback

    [ Upstream commit 3c53cd65dece47dd1f9d3a809f32e59d1d87b2b8 ]

    Commit 3b3fd068c56e3fbea30090859216a368398e39bf added NULL check for
    `rose_loopback_neigh->dev` in rose_loopback_timer() but omitted to
    check rose_loopback_neigh->loopback.

    It thus prevents *all* rose connect.

    The reason is that a special rose_neigh loopback has a NULL device.

    /proc/net/rose_neigh illustrates it via rose_neigh_show() function :
    [...]
    seq_printf(seq, "%05d %-9s %-4s   %3d %3d  %3s     %3s %3lu %3lu",
               rose_neigh->number,
               (rose_neigh->loopback) ? "RSLOOP-0" : ax2asc(buf, &rose_neigh->callsign),
               rose_neigh->dev ? rose_neigh->dev->name : "???",
               rose_neigh->count,

    /proc/net/rose_neigh displays special rose_loopback_neigh->loopback as
    callsign RSLOOP-0:

    addr  callsign  dev  count use mode restart  t0  tf digipeaters
    00001 RSLOOP-0  ???      1   2  DCE     yes   0   0

    By checking rose_loopback_neigh->loopback, rose_rx_call_request() is called
    even in case rose_loopback_neigh->dev is NULL. This repairs rose connections.

    Verification with rose client application FPAC:

    FPAC-Node v 4.1.3 (built Aug  5 2022) for LINUX (help = h)
    F6BVP-4 (Commands = ?) : u
    Users - AX.25 Level 2 sessions :
    Port   Callsign     Callsign  AX.25 state  ROSE state  NetRom status
    axudp  F6BVP-5   -> F6BVP-9   Connected    Connected   ---------

    Fixes: 3b3fd068c56e ("rose: Fix Null pointer dereference in rose_send_frame()")
    Signed-off-by: Bernard Pidoux <f6bvp@free.fr>
```

```
    Suggested-by: Francois Romieu <romieu@fr.zoreil.com>
    Cc: Thomas DL9SAU Osterried <thomas@osterried.de>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 546443d8886d003ad09e1e59a0b76d8311d3fec9
Author: Christian Brauner <brauner@kernel.org>
Date:   Wed Jul 20 14:32:52 2022 +0200

    ntfs: fix acl handling

    [ Upstream commit 0c3bc7899e6dfb52df1c46118a5a670ae619645f ]

    While looking at our current POSIX ACL handling in the context of some
    overlayfs work I went through a range of other filesystems checking how they
    handle them currently and encountered ntfs3.

    The posic_acl_{from,to}_xattr() helpers always need to operate on the
    filesystem idmapping. Since ntfs3 can only be mounted in the initial user
    namespace the relevant idmapping is init_user_ns.

    The posix_acl_{from,to}_xattr() helpers are concerned with translating between
    the kernel internal struct posix_acl{_entry} and the uapi struct
    posix_acl_xattr_{header,entry} and the kernel internal data structure is cached
    filesystem wide.

    Additional idmappings such as the caller's idmapping or the mount's idmapping
    are handled higher up in the VFS. Individual filesystems usually do not need to
    concern themselves with these.

    The posix_acl_valid() helper is concerned with checking whether the values in
    the kernel internal struct posix_acl can be represented in the filesystem's
    idmapping. IOW, if they can be written to disk. So this helper too needs to
    take the filesystem's idmapping.

    Fixes: be71b5cba2e6 ("fs/ntfs3: Add attrib operations")
    Cc: Konstantin Komarov <almaz.alexandrovich@paragon-software.com>
    Cc: ntfs3@lists.linux.dev
    Signed-off-by: Christian Brauner (Microsoft) <brauner@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 02ab2b234c58149bec7c494d6b7f11ec2956838b
Author: Peter Xu <peterx@redhat.com>
Date:   Fri Aug 5 12:00:03 2022 -0400

    mm/smaps: don't access young/dirty bit if pte unpresent

    [ Upstream commit efd4149342db2df41b1bbe68972ead853b30e444 ]

    These bits should only be valid when the ptes are present.  Introducing
    two booleans for it and set it to false when !pte_present() for both pte
    and pmd accountings.

    The bug is found during code reading and no real world issue reported, but
    logically such an error can cause incorrect readings for either smaps or
    smaps_rollup output on quite a few fields.

    For example, it could cause over-estimate on values like Shared_Dirty,
    Private_Dirty, Referenced.  Or it could also cause under-estimate on
    values like LazyFree, Shared_Clean, Private_Clean.

    Link: https://lkml.kernel.org/r/20220805160003.58929-1-peterx@redhat.com
    Fixes: b1d4d9e0cbd0 ("proc/smaps: carefully handle migration entries")
    Fixes: c94b6923fa0a ("/proc/PID/smaps: Add PMD migration entry parsing")
    Signed-off-by: Peter Xu <peterx@redhat.com>
```

```
    Reviewed-by: Vlastimil Babka <vbabka@suse.cz>
    Reviewed-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Yang Shi <shy828301@gmail.com>
    Cc: Konstantin Khlebnikov <khlebnikov@openvz.org>
    Cc: Huang Ying <ying.huang@intel.com>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 86ebf313929fbcc4bbf098158973d69691b6e4a6
Author: Trond Myklebust <trond.myklebust@hammerspace.com>
Date:   Wed Aug 3 14:55:03 2022 -0400

    SUNRPC: RPC level errors should set task->tk_rpc_status

    [ Upstream commit ed06fce0b034b2e25bd93430f5c4cbb28036cc1a ]

    Fix up a case in call_encode() where we're failing to set
    task->tk_rpc_status when an RPC level error occurred.

    Fixes: 9c5948c24869 ("SUNRPC: task should be exit if encode return EKEYEXPIRED more
times")
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit c2a47f6903e270c308c40ad4a23c17b30a54373c
Author: Olga Kornievskaia <kolga@netapp.com>
Date:   Thu Aug 18 15:07:05 2022 -0400

    NFSv4.2 fix problems with __nfs42_ssc_open

    [ Upstream commit fcfc8be1e9cf2f12b50dce8b579b3ae54443a014 ]

    A destination server while doing a COPY shouldn't accept using the
    passed in filehandle if its not a regular filehandle.

    If alloc_file_pseudo() has failed, we need to decrement a reference
    on the newly created inode, otherwise it leaks.

    Reported-by: Al Viro <viro@zeniv.linux.org.uk>
    Fixes: ec4b092508982 ("NFS: inter ssc open")
    Signed-off-by: Olga Kornievskaia <kolga@netapp.com>
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4eac2ff103b904afcfb97221fdaf8a43565036d5
Author: Sabrina Dubroca <sd@queasysnail.net>
Date:   Wed Aug 17 14:54:36 2022 +0200

    Revert "net: macsec: update SCI upon MAC address change."

    [ Upstream commit e82c649e851c9c25367fb7a2a6cf3479187de467 ]

    This reverts commit 6fc498bc82929ee23aa2f35a828c6178dfd3f823.

    Commit 6fc498bc8292 states:

        SCI should be updated, because it contains MAC in its first 6
        octets.

    That's not entirely correct. The SCI can be based on the MAC address,
    but doesn't have to be. We can also use any 64-bit number as the
    SCI. When the SCI based on the MAC address, it uses a 16-bit "port
    number" provided by userspace, which commit 6fc498bc8292 overwrites
    with 1.
```

```
    In addition, changing the SCI after macsec has been setup can just
    confuse the receiver. If we configure the RXSC on the peer based on
    the original SCI, we should keep the same SCI on TX.

    When the macsec device is being managed by a userspace key negotiation
    daemon such as wpa_supplicant, commit 6fc498bc8292 would also
    overwrite the SCI defined by userspace.

    Fixes: 6fc498bc8292 ("net: macsec: update SCI upon MAC address change.")
    Signed-off-by: Sabrina Dubroca <sd@queasysnail.net>
    Link:
https://lore.kernel.org/r/9b1a9d28327e7eb54550a92eebda45d25e54dd0d.1660667033.git.sd@queasys
nail.net
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9a852bcf90fbe631f87d1d17fa99c58ee53bb92e
Author: Seth Forshee <sforshee@digitalocean.com>
Date:   Tue Aug 16 11:47:52 2022 -0500

    fs: require CAP_SYS_ADMIN in target namespace for idmapped mounts

    [ Upstream commit bf1ac16edf6770a92bc75cf2373f1f9feea398a4 ]

    Idmapped mounts should not allow a user to map file ownsership into a
    range of ids which is not under the control of that user. However, we
    currently don't check whether the mounter is privileged wrt to the
    target user namespace.

    Currently no FS_USERNS_MOUNT filesystems support idmapped mounts, thus
    this is not a problem as only CAP_SYS_ADMIN in init_user_ns is allowed
    to set up idmapped mounts. But this could change in the future, so add a
    check to refuse to create idmapped mounts when the mounter does not have
    CAP_SYS_ADMIN in the target user namespace.

    Fixes: bd303368b776 ("fs: support mapped mounts of mapped filesystems")
    Signed-off-by: Seth Forshee <sforshee@digitalocean.com>
    Reviewed-by: Christian Brauner (Microsoft) <brauner@kernel.org>
    Link: https://lore.kernel.org/r/20220816164752.2595240-1-sforshee@digitalocean.com
    Signed-off-by: Christian Brauner (Microsoft) <brauner@kernel.org>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit e26d676c1f9f335510780b566a10475c47ce03d0
Author: Nikolay Aleksandrov <razor@blackwall.org>
Date:   Tue Aug 16 18:30:50 2022 +0300

    xfrm: policy: fix metadata dst->dev xmit null pointer dereference

    [ Upstream commit 17ecd4a4db4783392edd4944f5e8268205083f70 ]

    When we try to transmit an skb with metadata_dst attached (i.e. dst->dev
    == NULL) through xfrm interface we can hit a null pointer dereference[1]
    in xfrmi_xmit2() -> xfrm_lookup_with_ifid() due to the check for a
    loopback skb device when there's no policy which dereferences dst->dev
    unconditionally. Not having dst->dev can be interepreted as it not being
    a loopback device, so just add a check for a null dst_orig->dev.

    With this fix xfrm interface's Tx error counters go up as usual.

    [1] net-next calltrace captured via netconsole:
      BUG: kernel NULL pointer dereference, address: 00000000000000c0
      #PF: supervisor read access in kernel mode
      #PF: error_code(0x0000) - not-present page
      PGD 0 P4D 0
      Oops: 0000 [#1] PREEMPT SMP
```

```
       CPU: 1 PID: 7231 Comm: ping Kdump: loaded Not tainted 5.19.0+ #24
       Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.0-1.fc36 04/01/2014
       RIP: 0010:xfrm_lookup_with_ifid+0x5eb/0xa60
       Code: 8d 74 24 38 e8 26 a4 37 00 48 89 c1 e9 12 fc ff ff 49 63 ed 41 83 fd be 0f 85 be
 01 00 00 41 be ff ff ff ff 45 31 ed 48 8b 03 <f6> 80 c0 00 00 00 08 75 0f 41 80 bc 24 19 0d
 00 00 01 0f 84 1e 02
       RSP: 0018:ffffb0db82c679f0 EFLAGS: 00010246
       RAX: 0000000000000000 RBX: ffffd0db7fcad430 RCX: ffffb0db82c67a10
       RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffffb0db82c67a80
       RBP: ffffb0db82c67a80 R08: ffffb0db82c67a14 R09: 0000000000000000
       R10: 0000000000000000 R11: ffff8fa449667dc8 R12: ffffffff966db880
       R13: 0000000000000000 R14: 00000000ffffffff R15: 0000000000000000
       FS:  00007ff35c83f000(0000) GS:ffff8fa478480000(0000) knlGS:0000000000000000
       CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
       CR2: 00000000000000c0 CR3: 000000001ebb7000 CR4: 0000000000350ee0
       Call Trace:
        <TASK>
        xfrmi_xmit+0xde/0x460
        ? tcf_bpf_act+0x13d/0x2a0
        dev_hard_start_xmit+0x72/0x1e0
        __dev_queue_xmit+0x251/0xd30
        ip_finish_output2+0x140/0x550
        ip_push_pending_frames+0x56/0x80
        raw_sendmsg+0x663/0x10a0
        ? try_charge_memcg+0x3fd/0x7a0
        ? __mod_memcg_lruvec_state+0x93/0x110
        ? sock_sendmsg+0x30/0x40
        sock_sendmsg+0x30/0x40
        __sys_sendto+0xeb/0x130
        ? handle_mm_fault+0xae/0x280
        ? do_user_addr_fault+0x1e7/0x680
        ? kvm_read_and_reset_apf_flags+0x3b/0x50
        __x64_sys_sendto+0x20/0x30
        do_syscall_64+0x34/0x80
        entry_SYSCALL_64_after_hwframe+0x46/0xb0
       RIP: 0033:0x7ff35cac1366
       Code: eb 0b 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 41 89 ca 64 8b 04 25
 18 00 00 00 85 c0 75 11 b8 2c 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 72 c3 90 55 48 83 ec 30
 44 89 4c 24 2c 4c 89
       RSP: 002b:00007fff738e4028 EFLAGS: 00000246 ORIG_RAX: 000000000000002c
       RAX: ffffffffffffffda RBX: 00007fff738e57b0 RCX: 00007ff35cac1366
       RDX: 0000000000000040 RSI: 0000557164e4b450 RDI: 0000000000000003
       RBP: 0000557164e4b450 R08: 00007fff738e7a2c R09: 0000000000000010
       R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000040
       R13: 00007fff738e5770 R14: 00007fff738e4030 R15: 0000001d00000001
        </TASK>
       Modules linked in: netconsole veth br_netfilter bridge bonding virtio_net [last
 unloaded: netconsole]
       CR2: 00000000000000c0

    CC: Steffen Klassert <steffen.klassert@secunet.com>
    CC: Daniel Borkmann <daniel@iogearbox.net>
    Fixes: 2d151d39073a ("xfrm: Add possibility to set the default to block if we have no
 policy")
    Signed-off-by: Nikolay Aleksandrov <razor@blackwall.org>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

 commit 6901885656c029c976498290b52f67f2c251e6a0
 Author: Herbert Xu <herbert@gondor.apana.org.au>
 Date:   Thu Aug 4 18:03:46 2022 +0800

    af_key: Do not call xfrm_probe_algs in parallel

    [ Upstream commit ba953a9d89a00c078b85f4b190bc1dde66fe16b5 ]
```

```
    When namespace support was added to xfrm/afkey, it caused the
    previously single-threaded call to xfrm_probe_algs to become
    multi-threaded.  This is buggy and needs to be fixed with a mutex.

    Reported-by: Abhishek Shah <abhishek.shah@columbia.edu>
    Fixes: 283bc9f35bbb ("xfrm: Namespacify xfrm state/policy locks")
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 87b7ef9e760b50b3b2fbd76038ad1ca790094fe1
Author: Antony Antony <antony.antony@secunet.com>
Date:   Wed Jul 27 17:41:22 2022 +0200

    xfrm: clone missing x->lastused in xfrm_do_migrate

    [ Upstream commit 6aa811acdb76facca0b705f4e4c1d948ccb6af8b ]

    x->lastused was not cloned in xfrm_do_migrate. Add it to clone during
    migrate.

    Fixes: 80c9abaabf42 ("[XFRM]: Extension for dynamic update of endpoint address(es)")
    Signed-off-by: Antony Antony <antony.antony@secunet.com>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 9156a7b65c9d74e7effee70236ce93bfe21e58f9
Author: Antony Antony <antony.antony@secunet.com>
Date:   Wed Jul 27 17:38:35 2022 +0200

    Revert "xfrm: update SA curlft.use_time"

    [ Upstream commit 717ada9f10f2de8c4f4d72ad045f3b67a7ced715 ]

    This reverts commit af734a26a1a95a9fda51f2abb0c22a7efcafd5ca.

    The abvoce commit is a regression according RFC 2367. A better fix would be
    use x->lastused. Which will be propsed later.

    according to RFC 2367 use_time == sadb_lifetime_usetime.

    "sadb_lifetime_usetime
                        For CURRENT, the time, in seconds, when association
                        was first used. For HARD and SOFT, the number of
                        seconds after the first use of the association until
                        it expires."

    Fixes: af734a26a1a9 ("xfrm: update SA curlft.use_time")
    Signed-off-by: Antony Antony <antony.antony@secunet.com>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit d66c052879791313f90c0584420f196a038fb8b8
Author: Xin Xiong <xiongx18@fudan.edu.cn>
Date:   Sun Jul 24 17:55:58 2022 +0800

    xfrm: fix refcount leak in __xfrm_policy_check()

    [ Upstream commit 9c9cb23e00ddf45679b21b4dacc11d1ae7961ebe ]

    The issue happens on an error path in __xfrm_policy_check(). When the
    fetching process of the object `pols[1]` fails, the function simply
    returns 0, forgetting to decrement the reference count of `pols[0]`,
    which is incremented earlier by either xfrm_sk_policy_lookup() or
```

```
    xfrm_policy_lookup(). This may result in memory leaks.

    Fix it by decreasing the reference count of `pols[0]` in that path.

    Fixes: 134b0fc544ba ("IPsec: propagate security module errors up from
flow_cache_lookup")
    Signed-off-by: Xin Xiong <xiongx18@fudan.edu.cn>
    Signed-off-by: Xin Tan <tanxin.ctf@gmail.com>
    Signed-off-by: Steffen Klassert <steffen.klassert@secunet.com>
    Signed-off-by: Sasha Levin <sashal@kernel.org>

commit 4500aa11358d2a797ea3250ad9cb70ac26e98587
Author: Deren Wu <deren.wu@mediatek.com>
Date:   Wed Jun 8 20:53:26 2022 +0800

    mt76: mt7921: fix command timeout in AP stop period

    commit 9d958b60ebc2434f2b7eae83d77849e22d1059eb upstream.

    Due to AP stop improperly, mt7921 driver would face random command timeout
    by chip fw problem. Migrate AP start/stop process to .start_ap/.stop_ap and
    congiure BSS network settings in both hooks.

    The new flow is shown below.
    * AP start
        .start_ap()
          configure BSS network resource
          set BSS to connected state
        .bss_info_changed()
          enable fw beacon offload

    * AP stop
        .bss_info_changed()
          disable fw beacon offload (skip this command)
        .stop_ap()
          set BSS to disconnected state (beacon offload disabled automatically)
          destroy BSS network resource

    Fixes: 116c69603b01 ("mt76: mt7921: Add AP mode support")
    Signed-off-by: Sean Wang <sean.wang@mediatek.com>
    Signed-off-by: Deren Wu <deren.wu@mediatek.com>
    Signed-off-by: Felix Fietkau <nbd@nbd.name>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 49f05dfd2412401574ce6885078fec79f1e878b2
Author: David Hildenbrand <david@redhat.com>
Date:   Thu Aug 11 12:34:35 2022 +0200

    mm/hugetlb: support write-faults in shared mappings

    commit 1d8d14641fd94a01b20a4abbf2749fd8eddcf57b upstream.

    If we ever get a write-fault on a write-protected page in a shared
    mapping, we'd be in trouble (again).  Instead, we can simply map the page
    writable.

    And in fact, there is even a way right now to trigger that code via
    uffd-wp ever since we stared to support it for shmem in 5.19:

    -------------------------------------------------------------------------
     #include <stdio.h>
     #include <stdlib.h>
     #include <string.h>
     #include <fcntl.h>
     #include <unistd.h>
```

```
	#include <errno.h>
	#include <sys/mman.h>
	#include <sys/syscall.h>
	#include <sys/ioctl.h>
	#include <linux/userfaultfd.h>

	#define HUGETLB_SIZE (2 * 1024 * 1024u)

	static char *map;
	int uffd;

	static int temp_setup_uffd(void)
	{
		struct uffdio_api uffdio_api;
		struct uffdio_register uffdio_register;
		struct uffdio_writeprotect uffd_writeprotect;
		struct uffdio_range uffd_range;

		uffd = syscall(__NR_userfaultfd,
			       O_CLOEXEC | O_NONBLOCK | UFFD_USER_MODE_ONLY);
		if (uffd < 0) {
			fprintf(stderr, "syscall() failed: %d\n", errno);
			return -errno;
		}

		uffdio_api.api = UFFD_API;
		uffdio_api.features = UFFD_FEATURE_PAGEFAULT_FLAG_WP;
		if (ioctl(uffd, UFFDIO_API, &uffdio_api) < 0) {
			fprintf(stderr, "UFFDIO_API failed: %d\n", errno);
			return -errno;
		}

		if (!(uffdio_api.features & UFFD_FEATURE_PAGEFAULT_FLAG_WP)) {
			fprintf(stderr, "UFFD_FEATURE_WRITEPROTECT missing\n");
			return -ENOSYS;
		}

		/* Register UFFD-WP */
		uffdio_register.range.start = (unsigned long) map;
		uffdio_register.range.len = HUGETLB_SIZE;
		uffdio_register.mode = UFFDIO_REGISTER_MODE_WP;
		if (ioctl(uffd, UFFDIO_REGISTER, &uffdio_register) < 0) {
			fprintf(stderr, "UFFDIO_REGISTER failed: %d\n", errno);
			return -errno;
		}

		/* Writeprotect a single page. */
		uffd_writeprotect.range.start = (unsigned long) map;
		uffd_writeprotect.range.len = HUGETLB_SIZE;
		uffd_writeprotect.mode = UFFDIO_WRITEPROTECT_MODE_WP;
		if (ioctl(uffd, UFFDIO_WRITEPROTECT, &uffd_writeprotect)) {
			fprintf(stderr, "UFFDIO_WRITEPROTECT failed: %d\n", errno);
			return -errno;
		}

		/* Unregister UFFD-WP without prior writeunprotection. */
		uffd_range.start = (unsigned long) map;
		uffd_range.len = HUGETLB_SIZE;
		if (ioctl(uffd, UFFDIO_UNREGISTER, &uffd_range)) {
			fprintf(stderr, "UFFDIO_UNREGISTER failed: %d\n", errno);
			return -errno;
		}

		return 0;
	}
```

```
 int main(int argc, char **argv)
 {
        int fd;

        fd = open("/dev/hugepages/tmp", O_RDWR | O_CREAT);
        if (!fd) {
                fprintf(stderr, "open() failed\n");
                return -errno;
        }
        if (ftruncate(fd, HUGETLB_SIZE)) {
                fprintf(stderr, "ftruncate() failed\n");
                return -errno;
        }

        map = mmap(NULL, HUGETLB_SIZE, PROT_READ|PROT_WRITE, MAP_SHARED, fd, 0);
        if (map == MAP_FAILED) {
                fprintf(stderr, "mmap() failed\n");
                return -errno;
        }

        *map = 0;

        if (temp_setup_uffd())
                return 1;

        *map = 0;

        return 0;
 }
-------------------------------------------------------------------------

Above test fails with SIGBUS when there is only a single free hugetlb page.
 # echo 1 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
 # ./test
 Bus error (core dumped)

And worse, with sufficient free hugetlb pages it will map an anonymous page
into a shared mapping, for example, messing up accounting during unmap
and breaking MAP_SHARED semantics:
 # echo 2 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
 # ./test
 # cat /proc/meminfo | grep HugePages_
 HugePages_Total:       2
 HugePages_Free:        1
 HugePages_Rsvd:    18446744073709551615
 HugePages_Surp:        0

Reason is that uffd-wp doesn't clear the uffd-wp PTE bit when
unregistering and consequently keeps the PTE writeprotected.  Reason for
this is to avoid the additional overhead when unregistering.  Note that
this is the case also for !hugetlb and that we will end up with writable
PTEs that still have the uffd-wp PTE bit set once we return from
hugetlb_wp().  I'm not touching the uffd-wp PTE bit for now, because it
seems to be a generic thing -- wp_page_reuse() also doesn't clear it.

VM_MAYSHARE handling in hugetlb_fault() for FAULT_FLAG_WRITE indicates
that MAP_SHARED handling was at least envisioned, but could never have
worked as expected.

While at it, make sure that we never end up in hugetlb_wp() on write
faults without VM_WRITE, because we don't support maybe_mkwrite()
semantics as commonly used in the !hugetlb case -- for example, in
wp_page_reuse().
```

```
    Note that there is no need to do any kind of reservation in
    hugetlb_fault() in this case ...  because we already have a hugetlb page
    mapped R/O that we will simply map writable and we are not dealing with
    COW/unsharing.

    Link: https://lkml.kernel.org/r/20220811103435.188481-3-david@redhat.com
    Fixes: b1f9e876862d ("mm/uffd: enable write protection for shmem & hugetlbfs")
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Bjorn Helgaas <bhelgaas@google.com>
    Cc: Cyrill Gorcunov <gorcunov@openvz.org>
    Cc: Hugh Dickins <hughd@google.com>
    Cc: Jamie Liu <jamieliu@google.com>
    Cc: Kirill A. Shutemov <kirill.shutemov@linux.intel.com>
    Cc: Muchun Song <songmuchun@bytedance.com>
    Cc: Naoya Horiguchi <n-horiguchi@ah.jp.nec.com>
    Cc: Pavel Emelyanov <xemul@parallels.com>
    Cc: Peter Feiner <pfeiner@google.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: <stable@vger.kernel.org>    [5.19]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3e2747c3ddfa717697c3cc2aa6ab989e48d6587d
Author: Peter Xu <peterx@redhat.com>
Date:   Thu Aug 11 16:13:40 2022 -0400

    mm/uffd: reset write protection when unregister with wp-mode

    commit f369b07c861435bd812a9d14493f71b34132ed6f upstream.

    The motivation of this patch comes from a recent report and patchfix from
    David Hildenbrand on hugetlb shared handling of wr-protected page [1].

    With the reproducer provided in commit message of [1], one can leverage
    the uffd-wp lazy-reset of ptes to trigger a hugetlb issue which can affect
    not only the attacker process, but also the whole system.

    The lazy-reset mechanism of uffd-wp was used to make unregister faster,
    meanwhile it has an assumption that any leftover pgtable entries should
    only affect the process on its own, so not only the user should be aware
    of anything it does, but also it should not affect outside of the process.

    But it seems that this is not true, and it can also be utilized to make
    some exploit easier.

    So far there's no clue showing that the lazy-reset is important to any
    userfaultfd users because normally the unregister will only happen once
    for a specific range of memory of the lifecycle of the process.

    Considering all above, what this patch proposes is to do explicit pte
    resets when unregister an uffd region with wr-protect mode enabled.

    It should be the same as calling ioctl(UFFDIO_WRITEPROTECT, wp=false)
    right before ioctl(UFFDIO_UNREGISTER) for the user.  So potentially it'll
    make the unregister slower.  From that pov it's a very slight abi change,
    but hopefully nothing should break with this change either.

    Regarding to the change itself - core of uffd write [un]protect operation
    is moved into a separate function (uffd_wp_range()) and it is reused in
    the unregister code path.

    Note that the new function will not check for anything, e.g.  ranges or
    memory types, because they should have been checked during the previous
    UFFDIO_REGISTER or it should have failed already.  It also doesn't check
```

```
    mmap_changing because we're with mmap write lock held anyway.

    I added a Fixes upon introducing of uffd-wp shmem+hugetlbfs because that's
    the only issue reported so far and that's the commit David's reproducer
    will start working (v5.19+).  But the whole idea actually applies to not
    only file memories but also anonymous.  It's just that we don't need to
    fix anonymous prior to v5.19- because there's no known way to exploit.

    IOW, this patch can also fix the issue reported in [1] as the patch 2 does.

    [1] https://lore.kernel.org/all/20220811103435.188481-3-david@redhat.com/

    Link: https://lkml.kernel.org/r/20220811201340.39342-1-peterx@redhat.com
    Fixes: b1f9e876862d ("mm/uffd: enable write protection for shmem & hugetlbfs")
    Signed-off-by: Peter Xu <peterx@redhat.com>
    Cc: David Hildenbrand <david@redhat.com>
    Cc: Mike Rapoport <rppt@linux.vnet.ibm.com>
    Cc: Mike Kravetz <mike.kravetz@oracle.com>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: Nadav Amit <nadav.amit@gmail.com>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bc3188d8a3b8c08c306a4c851ddb2c92ba4599ca
Author: Kuniyuki Iwashima <kuniyu@amazon.com>
Date:   Fri Aug 12 19:05:09 2022 -0700

    kprobes: don't call disarm_kprobe() for disabled kprobes

    commit 9c80e79906b4ca440d09e7f116609262bb747909 upstream.

    The assumption in __disable_kprobe() is wrong, and it could try to disarm
    an already disarmed kprobe and fire the WARN_ONCE() below. [0]  We can
    easily reproduce this issue.

    1. Write 0 to /sys/kernel/debug/kprobes/enabled.

      # echo 0 > /sys/kernel/debug/kprobes/enabled

    2. Run execsnoop.  At this time, one kprobe is disabled.

      # /usr/share/bcc/tools/execsnoop &
      [1] 2460
      PCOMM            PID    PPID   RET ARGS

      # cat /sys/kernel/debug/kprobes/list
      ffffffff91345650  r  __x64_sys_execve+0x0    [FTRACE]
      ffffffff91345650  k  __x64_sys_execve+0x0    [DISABLED][FTRACE]

    3. Write 1 to /sys/kernel/debug/kprobes/enabled, which changes
       kprobes_all_disarmed to false but does not arm the disabled kprobe.

      # echo 1 > /sys/kernel/debug/kprobes/enabled

      # cat /sys/kernel/debug/kprobes/list
      ffffffff91345650  r  __x64_sys_execve+0x0    [FTRACE]
      ffffffff91345650  k  __x64_sys_execve+0x0    [DISABLED][FTRACE]

    4. Kill execsnoop, when __disable_kprobe() calls disarm_kprobe() for the
       disabled kprobe and hits the WARN_ONCE() in __disarm_kprobe_ftrace().

      # fg
      /usr/share/bcc/tools/execsnoop
```

```
       ^C

    Actually, WARN_ONCE() is fired twice, and __unregister_kprobe_top() misses
    some cleanups and leaves the aggregated kprobe in the hash table.  Then,
    __unregister_trace_kprobe() initialises tk->rp.kp.list and creates an
    infinite loop like this.

      aggregated kprobe.list -> kprobe.list -.
                                        ^    |
                                        '.__.'

    In this situation, these commands fall into the infinite loop and result
    in RCU stall or soft lockup.

      cat /sys/kernel/debug/kprobes/list : show_kprobe_addr() enters into the
                                           infinite loop with RCU.

      /usr/share/bcc/tools/execsnoop : warn_kprobe_rereg() holds kprobe_mutex,
                                       and __get_valid_kprobe() is stuck in
                                       the loop.

    To avoid the issue, make sure we don't call disarm_kprobe() for disabled
    kprobes.

    [0]
    Failed to disarm kprobe-ftrace at __x64_sys_execve+0x0/0x40 (error -2)
    WARNING: CPU: 6 PID: 2460 at kernel/kprobes.c:1130 __disarm_kprobe_ftrace.isra.19
 (kernel/kprobes.c:1129)
    Modules linked in: ena
    CPU: 6 PID: 2460 Comm: execsnoop Not tainted 5.19.0+ #28
    Hardware name: Amazon EC2 c5.2xlarge/, BIOS 1.0 10/16/2017
    RIP: 0010:__disarm_kprobe_ftrace.isra.19 (kernel/kprobes.c:1129)
    Code: 24 8b 02 eb c1 80 3d c4 83 f2 01 00 75 d4 48 8b 75 00 89 c2 48 c7 c7 90 fa 0f 92
 89 04 24 c6 05 ab 83 01 e8 e4 94 f0 ff <0f> 0b 8b 04 24 eb b1 89 c6 48 c7 c7 60 fa 0f 92 89
 04 24 e8 cc 94
    RSP: 0018:ffff9e6ec154bd98 EFLAGS: 00010282
    RAX: 0000000000000000 RBX: ffffffff930f7b00 RCX: 0000000000000001
    RDX: 0000000080000001 RSI: ffffffff921461c5 RDI: 00000000ffffffff
    RBP: ffff89c504286da8 R08: 0000000000000000 R09: c0000000fffeffff
    R10: 0000000000000000 R11: ffff9e6ec154bc28 R12: ffff89c502394e40
    R13: ffff89c502394c00 R14: ffff9e6ec154bc00 R15: 0000000000000000
    FS:  00007fe800398740(0000) GS:ffff89c812d80000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 000000c00057f010 CR3: 0000000103b54006 CR4: 00000000007706e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    PKRU: 55555554
    Call Trace:
    <TASK>
     __disable_kprobe (kernel/kprobes.c:1716)
     disable_kprobe (kernel/kprobes.c:2392)
     __disable_trace_kprobe (kernel/trace/trace_kprobe.c:340)
     disable_trace_kprobe (kernel/trace/trace_kprobe.c:429)
     perf_trace_event_unreg.isra.2 (./include/linux/tracepoint.h:93
 kernel/trace/trace_event_perf.c:168)
     perf_kprobe_destroy (kernel/trace/trace_event_perf.c:295)
     _free_event (kernel/events/core.c:4971)
     perf_event_release_kernel (kernel/events/core.c:5176)
     perf_release (kernel/events/core.c:5186)
     __fput (fs/file_table.c:321)
     task_work_run (./include/linux/sched.h:2056 (discriminator 1) kernel/task_work.c:179
 (discriminator 1))
     exit_to_user_mode_prepare (./include/linux/resume_user_mode.h:49
 kernel/entry/common.c:169 kernel/entry/common.c:201)
     syscall_exit_to_user_mode (./arch/x86/include/asm/jump_label.h:55
```

```
 ./arch/x86/include/asm/nospec-branch.h:384 ./arch/x86/include/asm/entry-common.h:94
 kernel/entry/common.c:133 kernel/entry/common.c:296)
      do_syscall_64 (arch/x86/entry/common.c:87)
      entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120)
     RIP: 0033:0x7fe7ff210654
     Code: 15 79 89 20 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb be 0f 1f 00 8b 05 9a cd 20
 00 48 63 ff 85 c0 75 11 b8 03 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 3a f3 c3 48 83 ec 18 48
 89 7c 24 08 e8 34 fc
     RSP: 002b:00007ffdbd1d3538 EFLAGS: 00000246 ORIG_RAX: 0000000000000003
     RAX: 0000000000000000 RBX: 0000000000000008 RCX: 00007fe7ff210654
     RDX: 0000000000000000 RSI: 0000000000002401 RDI: 0000000000000008
     RBP: 0000000000000000 R08: 94ae31d6fda838a4 R0900007fe8001c9d30
     R10: 00007ffdbd1d34b0 R11: 0000000000000246 R12: 00007ffdbd1d3600
     R13: 0000000000000000 R14: fffffffffffffffc R15: 00007ffdbd1d3560
     </TASK>

     Link: https://lkml.kernel.org/r/20220813020509.90805-1-kuniyu@amazon.com
     Fixes: 69d54b916d83 ("kprobes: makes kprobes/enabled works correctly for optimized
 kprobes.")
     Signed-off-by: Kuniyuki Iwashima <kuniyu@amazon.com>
     Reported-by: Ayushman Dutta <ayudutta@amazon.com>
     Cc: "Naveen N. Rao" <naveen.n.rao@linux.ibm.com>
     Cc: Anil S Keshavamurthy <anil.s.keshavamurthy@intel.com>
     Cc: "David S. Miller" <davem@davemloft.net>
     Cc: Masami Hiramatsu <mhiramat@kernel.org>
     Cc: Wang Nan <wangnan0@huawei.com>
     Cc: Kuniyuki Iwashima <kuniyu@amazon.com>
     Cc: Kuniyuki Iwashima <kuni1840@gmail.com>
     Cc: Ayushman Dutta <ayudutta@amazon.com>
     Cc: <stable@vger.kernel.org>
     Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 79ce0b1445f92f4cf1a706101c41543f56059ce8
 Author: Randy Dunlap <rdunlap@infradead.org>
 Date:   Sun Aug 7 15:09:34 2022 -0700

     kernel/sys_ni: add compat entry for fadvise64_64

     commit a8faed3a02eeb75857a3b5d660fa80fe79db77a3 upstream.

     When CONFIG_ADVISE_SYSCALLS is not set/enabled and CONFIG_COMPAT is
     set/enabled, the riscv compat_syscall_table references
     'compat_sys_fadvise64_64', which is not defined:

     riscv64-linux-ld: arch/riscv/kernel/compat_syscall_table.o:(.rodata+0x6f8):
     undefined reference to `compat_sys_fadvise64_64'

     Add 'fadvise64_64' to kernel/sys_ni.c as a conditional COMPAT function so
     that when CONFIG_ADVISE_SYSCALLS is not set, there is a fallback function
     available.

     Link: https://lkml.kernel.org/r/20220807220934.5689-1-rdunlap@infradead.org
     Fixes: d3ac21cacc24 ("mm: Support compiling out madvise and fadvise")
     Signed-off-by: Randy Dunlap <rdunlap@infradead.org>
     Suggested-by: Arnd Bergmann <arnd@arndb.de>
     Reviewed-by: Arnd Bergmann <arnd@arndb.de>
     Cc: Josh Triplett <josh@joshtriplett.org>
     Cc: Paul Walmsley <paul.walmsley@sifive.com>
     Cc: Palmer Dabbelt <palmer@dabbelt.com>
     Cc: Albert Ou <aou@eecs.berkeley.edu>
     Cc: <stable@vger.kernel.org>
     Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 760dc9f658d7fb133007eba63fa012618752fd48
Author: Helge Deller <deller@gmx.de>
Date:   Sat Aug 20 17:59:17 2022 +0200

    parisc: Fix exception handler for fldw and fstw instructions

    commit 7ae1f5508d9a33fd58ed3059bd2d569961e3b8bd upstream.

    The exception handler is broken for unaligned memory acceses with fldw
    and fstw instructions, because it trashes or uses randomly some other
    floating point register than the one specified in the instruction word
    on loads and stores.

    The instruction "fldw 0(addr),%fr22L" (and the other fldw/fstw
    instructions) encode the target register (%fr22) in the rightmost 5 bits
    of the instruction word. The 7th rightmost bit of the instruction word
    defines if the left or right half of %fr22 should be used.

    While processing unaligned address accesses, the FR3() define is used to
    extract the offset into the local floating-point register set.  But the
    calculation in FR3() was buggy, so that for example instead of %fr22,
    register %fr12 [((22 * 2) & 0x1f) = 12] was used.

    This bug has been since forever in the parisc kernel and I wonder why it
    wasn't detected earlier. Interestingly I noticed this bug just because
    the libime debian package failed to build on *native* hardware, while it
    successfully built in qemu.

    This patch corrects the bitshift and masking calculation in FR3().

    Signed-off-by: Helge Deller <deller@gmx.de>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 012dad4b908a1b7ee3b4c1a52f5deb12dc446c46
Author: Helge Deller <deller@gmx.de>
Date:   Fri Aug 19 19:30:50 2022 +0200

    parisc: Make CONFIG_64BIT available for ARCH=parisc64 only

    commit 3dcfb729b5f4a0c9b50742865cd5e6c4dbcc80dc upstream.

    With this patch the ARCH= parameter decides if the
    CONFIG_64BIT option will be set or not. This means, the
    ARCH= parameter will give:

            ARCH=parisc     -> 32-bit kernel
            ARCH=parisc64   -> 64-bit kernel

    This simplifies the usage of the other config options like
    randconfig, allmodconfig and allyesconfig a lot and produces
    the output which is expected for parisc64 (64-bit) vs. parisc (32-bit).

    Suggested-by: Masahiro Yamada <masahiroy@kernel.org>
    Signed-off-by: Helge Deller <deller@gmx.de>
    Tested-by: Randy Dunlap <rdunlap@infradead.org>
    Reviewed-by: Randy Dunlap <rdunlap@infradead.org>
    Cc: <stable@vger.kernel.org> # 5.15+
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b0d2e414bcc751e0823fcb5091c9967fe2474c65
Author: Jing-Ting Wu <Jing-Ting.Wu@mediatek.com>
Date:   Tue Aug 23 13:41:46 2022 +0800

    cgroup: Fix race condition at rebind_subsystems()
```

```
    commit 763f4fb76e24959c370cdaa889b2492ba6175580 upstream.

    Root cause:
    The rebind_subsystems() is no lock held when move css object from A
    list to B list,then let B's head be treated as css node at
    list_for_each_entry_rcu().

    Solution:
    Add grace period before invalidating the removed rstat_css_node.

    Reported-by: Jing-Ting Wu <jing-ting.wu@mediatek.com>
    Suggested-by: Michal Koutný <mkoutny@suse.com>
    Signed-off-by: Jing-Ting Wu <jing-ting.wu@mediatek.com>
    Tested-by: Jing-Ting Wu <jing-ting.wu@mediatek.com>
    Link: https://lore.kernel.org/linux-arm-
kernel/d8f0bc5e2fb6ed259f9334c83279b4c011283c41.camel@mediatek.com/T/
    Acked-by: Mukesh Ojha <quic_mojha@quicinc.com>
    Fixes: a7df69b81aac ("cgroup: rstat: support cgroup1")
    Cc: stable@vger.kernel.org # v5.13+
    Signed-off-by: Tejun Heo <tj@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 24a943ca4c454dda12596c474e0185cb807cbd63
Author: Gaosheng Cui <cuigaosheng1@huawei.com>
Date:   Mon Aug 22 10:29:05 2022 +0800

    audit: fix potential double free on error path from fsnotify_add_inode_mark

    commit ad982c3be4e60c7d39c03f782733503cbd88fd2a upstream.

    Audit_alloc_mark() assign pathname to audit_mark->path, on error path
    from fsnotify_add_inode_mark(), fsnotify_put_mark will free memory
    of audit_mark->path, but the caller of audit_alloc_mark will free
    the pathname again, so there will be double free problem.

    Fix this by resetting audit_mark->path to NULL pointer on error path
    from fsnotify_add_inode_mark().

    Cc: stable@vger.kernel.org
    Fixes: 7b1293234084d ("fsnotify: Add group pointer in fsnotify_init_mark()")
    Signed-off-by: Gaosheng Cui <cuigaosheng1@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Paul Moore <paul@paul-moore.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 72440414b10b7d323b15e5c2ccfb5fa368833235
Author: Trond Myklebust <trond.myklebust@hammerspace.com>
Date:   Sat Aug 13 08:22:25 2022 -0400

    NFS: Fix another fsync() issue after a server reboot

    commit 67f4b5dc49913abcdb5cc736e73674e2f352f81d upstream.

    Currently, when the writeback code detects a server reboot, it redirties
    any pages that were not committed to disk, and it sets the flag
    NFS_CONTEXT_RESEND_WRITES in the nfs_open_context of the file descriptor
    that dirtied the file. While this allows the file descriptor in question
    to redrive its own writes, it violates the fsync() requirement that we
    should be synchronising all writes to disk.
    While the problem is infrequent, we do see corner cases where an
    untimely server reboot causes the fsync() call to abandon its attempt to
    sync data to disk and causing data corruption issues due to missed error
    conditions or similar.
```

```
    In order to tighted up the client's ability to deal with this situation
    without introducing livelocks, add a counter that records the number of
    times pages are redirtied due to a server reboot-like condition, and use
    that in fsync() to redrive the sync to disk.

    Fixes: 2197e9b06c22 ("NFS: Fix up fsync() when the server rebooted")
    Cc: stable@vger.kernel.org
    Signed-off-by: Trond Myklebust <trond.myklebust@hammerspace.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9def52eb10baab3b700858003d462fcf17d62873
Author: David Hildenbrand <david@redhat.com>
Date:   Wed Aug 24 21:23:33 2022 +0200

    mm/gup: fix FOLL_FORCE COW security issue and remove FOLL_COW

    commit 5535be3099717646781ce1540cf725965d680e7b upstream.

    Ever since the Dirty COW (CVE-2016-5195) security issue happened, we know
    that FOLL_FORCE can be possibly dangerous, especially if there are races
    that can be exploited by user space.

    Right now, it would be sufficient to have some code that sets a PTE of a
    R/O-mapped shared page dirty, in order for it to erroneously become
    writable by FOLL_FORCE.  The implications of setting a write-protected PTE
    dirty might not be immediately obvious to everyone.

    And in fact ever since commit 9ae0f87d009c ("mm/shmem: unconditionally set
    pte dirty in mfill_atomic_install_pte"), we can use UFFDIO_CONTINUE to map
    a shmem page R/O while marking the pte dirty.  This can be used by
    unprivileged user space to modify tmpfs/shmem file content even if the
    user does not have write permissions to the file, and to bypass memfd
    write sealing -- Dirty COW restricted to tmpfs/shmem (CVE-2022-2590).

    To fix such security issues for good, the insight is that we really only
    need that fancy retry logic (FOLL_COW) for COW mappings that are not
    writable (!VM_WRITE).  And in a COW mapping, we really only broke COW if
    we have an exclusive anonymous page mapped.  If we have something else
    mapped, or the mapped anonymous page might be shared (!PageAnonExclusive),
    we have to trigger a write fault to break COW.  If we don't find an
    exclusive anonymous page when we retry, we have to trigger COW breaking
    once again because something intervened.

    Let's move away from this mandatory-retry + dirty handling and rely on our
    PageAnonExclusive() flag for making a similar decision, to use the same
    COW logic as in other kernel parts here as well.  In case we stumble over
    a PTE in a COW mapping that does not map an exclusive anonymous page, COW
    was not properly broken and we have to trigger a fake write-fault to break
    COW.

    Just like we do in can_change_pte_writable() added via commit 64fe24a3e05e
    ("mm/mprotect: try avoiding write faults for exclusive anonymous pages
    when changing protection") and commit 76aefad628aa ("mm/mprotect: fix
    soft-dirty check in can_change_pte_writable()"), take care of softdirty
    and uffd-wp manually.

    For example, a write() via /proc/self/mem to a uffd-wp-protected range has
    to fail instead of silently granting write access and bypassing the
    userspace fault handler.  Note that FOLL_FORCE is not only used for debug
    access, but also triggered by applications without debug intentions, for
    example, when pinning pages via RDMA.

    This fixes CVE-2022-2590. Note that only x86_64 and aarch64 are
    affected, because only those support CONFIG_HAVE_ARCH_USERFAULTFD_MINOR.
```

```
    Fortunately, FOLL_COW is no longer required to handle FOLL_FORCE. So
    let's just get rid of it.

    Thanks to Nadav Amit for pointing out that the pte_dirty() check in
    FOLL_FORCE code is problematic and might be exploitable.

    Note 1: We don't check for the PTE being dirty because it doesn't matter
            for making a "was COWed" decision anymore, and whoever modifies the
            page has to set the page dirty either way.

    Note 2: Kernels before extended uffd-wp support and before
            PageAnonExclusive (< 5.19) can simply revert the problematic
            commit instead and be safe regarding UFFDIO_CONTINUE. A backport to
            v5.19 requires minor adjustments due to lack of
            vma_soft_dirty_enabled().

    Link: https://lkml.kernel.org/r/20220809205640.70916-1-david@redhat.com
    Fixes: 9ae0f87d009c ("mm/shmem: unconditionally set pte dirty in
mfill_atomic_install_pte")
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: Axel Rasmussen <axelrasmussen@google.com>
    Cc: Nadav Amit <nadav.amit@gmail.com>
    Cc: Peter Xu <peterx@redhat.com>
    Cc: Hugh Dickins <hughd@google.com>
    Cc: Andrea Arcangeli <aarcange@redhat.com>
    Cc: Matthew Wilcox <willy@infradead.org>
    Cc: Vlastimil Babka <vbabka@suse.cz>
    Cc: John Hubbard <jhubbard@nvidia.com>
    Cc: Jason Gunthorpe <jgg@nvidia.com>
    Cc: David Laight <David.Laight@ACULAB.COM>
    Cc: <stable@vger.kernel.org>    [5.16]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: David Hildenbrand <david@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```