```
commit 449718782a469dba0922fa97c39ff6897efd7026
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Feb 28 17:23:45 2020 +0100

    Linux 5.5.7

commit 858b4f944a67b921b0c0e81310a376a0663291a2
Author: Sathyanarayana Nujella <sathyanarayana.nujella@intel.com>
Date:   Fri Dec 20 11:10:36 2019 -0600

    ASoC: SOF: Intel: hda: Add iDisp4 DAI

    commit e68d6696575e1af3f92125e842f2853708f34589 upstream.

    TGL supports more than three iDisp DAI's.
    Add support for iDisp4 CPU DAI.

    Without this patch, we saw the below error on our TGL DUT:
       sof_rt5682 tgl_max98357a_rt5682: ASoC: CPU DAI iDisp4 Pin not registered

    Signed-off-by: Sathyanarayana Nujella <sathyanarayana.nujella@intel.com>
    Signed-off-by: Jairaj Arava <jairaj.arava@intel.com>
    Signed-off-by: Pierre-Louis Bossart <pierre-louis.bossart@linux.intel.com>
    Link: https://lore.kernel.org/r/20191220171037.10689-2-pierre-
louis.bossart@linux.intel.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Cc: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9b8eac9eefa6a3ba8de079db8f143ec195f34917
Author: John Fastabend <john.fastabend@gmail.com>
Date:   Sun Feb 9 21:44:37 2020 -0800

    bpf: Selftests build error in sockmap_basic.c

    commit f2e97dc126b712c0d21219ed0c42710006c1cf52 upstream.

    Fix following build error. We could push a tcp.h header into one of the
    include paths, but I think its easy enough to simply pull in the three
    defines we need here. If we end up using more of tcp.h at some point
    we can pull it in later.

    /home/john/git/bpf/tools/testing/selftests/bpf/prog_tests/sockmap_basic.c: In function
‘connected_socket_v4’:
    /home/john/git/bpf/tools/testing/selftests/bpf/prog_tests/sockmap_basic.c:20:11: error:
‘TCP_REPAIR_ON’ undeclared (first use in this function)
      repair = TCP_REPAIR_ON;
               ^
    /home/john/git/bpf/tools/testing/selftests/bpf/prog_tests/sockmap_basic.c:20:11: note:
each undeclared identifier is reported only once for each function it appears in
    /home/john/git/bpf/tools/testing/selftests/bpf/prog_tests/sockmap_basic.c:29:11: error:
‘TCP_REPAIR_OFF_NO_WP’ undeclared (first use in this function)
      repair = TCP_REPAIR_OFF_NO_WP;

    Then with fix,

    $ ./test_progs -n 44
    #44/1 sockmap create_update_free:OK
    #44/2 sockhash create_update_free:OK
    #44 sockmap_basic:OK

    Fixes: 5d3919a953c3c ("selftests/bpf: Test freeing sockmap/sockhash with a socket in
it")
    Signed-off-by: John Fastabend <john.fastabend@gmail.com>
```

```
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Reviewed-by: Jakub Sitnicki <jakub@cloudflare.com>
    Link: https://lore.kernel.org/bpf/158131347731.21414.12120493483848386652.stgit@john-
Precision-5820-Tower
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 92d75fda52144e11e042d4e8fff2d0128f7559fe
Author: Nathan Chancellor <natechancellor@gmail.com>
Date:   Thu Feb 13 23:42:07 2020 -0700

    s390/mm: Explicitly compare PAGE_DEFAULT_KEY against zero in storage_key_init_range

    commit 380324734956c64cd060e1db4304f3117ac15809 upstream.

    Clang warns:

     In file included from ../arch/s390/purgatory/purgatory.c:10:
     In file included from ../include/linux/kexec.h:18:
     In file included from ../include/linux/crash_core.h:6:
     In file included from ../include/linux/elfcore.h:5:
     In file included from ../include/linux/user.h:1:
     In file included from ../arch/s390/include/asm/user.h:11:
     ../arch/s390/include/asm/page.h:45:6: warning: converting the result of
     '<<' to a boolean always evaluates to false
     [-Wtautological-constant-compare]
             if (PAGE_DEFAULT_KEY)
                ^
     ../arch/s390/include/asm/page.h:23:44: note: expanded from macro
     'PAGE_DEFAULT_KEY'
     #define PAGE_DEFAULT_KEY        (PAGE_DEFAULT_ACC << 4)
                                                        ^
     1 warning generated.

    Explicitly compare this against zero to silence the warning as it is
    intended to be used in a boolean context.

    Fixes: de3fa841e429 ("s390/mm: fix compile for PAGE_DEFAULT_KEY != 0")
    Link: https://github.com/ClangBuiltLinux/linux/issues/860
    Link: https://lkml.kernel.org/r/20200214064207.10381-1-natechancellor@gmail.com
    Acked-by: Christian Borntraeger <borntraeger@de.ibm.com>
    Signed-off-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ff7b36cf9bef2d358c50c4a3c5aa7398b42ece93
Author: Nathan Chancellor <natechancellor@gmail.com>
Date:   Sat Feb 8 07:10:52 2020 -0700

    s390/kaslr: Fix casts in get_random

    commit 788d671517b5c81efbed9310ccbadb8cca86a08e upstream.

    Clang warns:

    ../arch/s390/boot/kaslr.c:78:25: warning: passing 'char *' to parameter
    of type 'const u8 *' (aka 'const unsigned char *') converts between
    pointers to integer
    types with different sign [-Wpointer-sign]
                                    (char *) entropy, (char *) entropy,
                                                      ^~~~~~~~~~~~~~~~
    ../arch/s390/include/asm/cpacf.h:280:28: note: passing argument to
    parameter 'src' here
                               u8 *dest, const u8 *src, long src_len)
                                                  ^
    2 warnings generated.
```

```
    Fix the cast to match what else is done in this function.

    Fixes: b2d24b97b2a9 ("s390/kernel: add support for kernel address space layout
randomization (KASLR)")
    Link: https://github.com/ClangBuiltLinux/linux/issues/862
    Link: https://lkml.kernel.org/r/20200208141052.48476-1-natechancellor@gmail.com
    Signed-off-by: Nathan Chancellor <natechancellor@gmail.com>
    Signed-off-by: Vasily Gorbik <gor@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a0230b7957f2397857ea3a97edeb16ef2142a74e
Author: Aya Levin <ayal@mellanox.com>
Date:   Wed Feb 12 15:17:25 2020 +0200

    net/mlx5e: Fix crash in recovery flow without devlink reporter

    commit 1ad6c43c6a7b8627240c6cc19c69e31fedc596a7 upstream.

    When health reporters are not supported, recovery function is invoked
    directly, not via devlink health reporters.

    In this direct flow, the recover function input parameter was passed
    incorrectly and is causing a kernel oops. This patch is fixing the input
    parameter.

    Following call trace is observed on rx error health reporting.

    Internal error: Oops: 96000007 [#1] PREEMPT SMP
    Process kworker/u16:4 (pid: 4584, stack limit = 0x00000000c9e45703)
    Call trace:
    mlx5e_rx_reporter_err_rq_cqe_recover+0x30/0x164 [mlx5_core]
    mlx5e_health_report+0x60/0x6c [mlx5_core]
    mlx5e_reporter_rq_cqe_err+0x6c/0x90 [mlx5_core]
    mlx5e_rq_err_cqe_work+0x20/0x2c [mlx5_core]
    process_one_work+0x168/0x3d0
    worker_thread+0x58/0x3d0
    kthread+0x108/0x134

    Fixes: c50de4af1d63 ("net/mlx5e: Generalize tx reporter's functionality")
    Signed-off-by: Aya Levin <ayal@mellanox.com>
    Signed-off-by: Parav Pandit <parav@mellanox.com>
    Reviewed-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db5d741f50e23ad01adf1c8ab430f2d83f215851
Author: Dmytro Linkin <dmitrolin@mellanox.com>
Date:   Wed Feb 12 11:32:39 2020 +0200

    net/mlx5e: Don't clear the whole vf config when switching modes

    commit 383de108157c881074f32914b61125e299820bd2 upstream.

    There is no need to reset all vf config (except link state) between
    legacy and switchdev modes changes.
    Also, set link state to AUTO, when legacy enabled.

    Fixes: 3b83b6c2e024 ("net/mlx5e: Clear VF config when switching modes")
    Signed-off-by: Dmytro Linkin <dmitrolin@mellanox.com>
    Reviewed-by: Roi Dayan <roid@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cb50a7b490822d16912d15d20fc04bbe92b1a72f
```

```
Author: Huy Nguyen <huyn@mellanox.com>
Date:   Mon Feb 3 16:32:18 2020 -0600

    net/mlx5: Fix sleep while atomic in mlx5_eswitch_get_vepa

    commit 3d9c5e023a0dbf3e117bb416cfefd9405bf5af0c upstream.

    rtnl_bridge_getlink is protected by rcu lock, so mlx5_eswitch_get_vepa
    cannot take mutex lock. Two possible issues can happen:
    1. User at the same time change vepa mode via RTM_SETLINK command.
    2. User at the same time change the switchdev mode via devlink netlink
    interface.

    Case 1 cannot happen because rtnl executes one message in order.
    Case 2 can happen but we do not expect user to change the switchdev mode
    when changing vepa. Even if a user does it, so he will read a value
    which is no longer valid.

    Fixes: 8da202b24913 ("net/mlx5: E-Switch, Add support for VEPA in legacy mode.")
    Signed-off-by: Huy Nguyen <huyn@mellanox.com>
    Reviewed-by: Mark Bloch <markb@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e8a83728b349a78ed983f192ec369cccd3f0cfd0
Author: Aya Levin <ayal@mellanox.com>
Date:   Mon Dec 9 14:08:18 2019 +0200

    net/mlx5e: Reset RQ doorbell counter before moving RQ state from RST to RDY

    commit 5ee090ed0da649b1febae2b7c285ac77d1e55a0c upstream.

    Initialize RQ doorbell counters to zero prior to moving an RQ from RST
    to RDY state. Per HW spec, when RQ is back to RDY state, the descriptor
    ID on the completion is reset. The doorbell record must comply.

    Fixes: 8276ea1353a4 ("net/mlx5e: Report and recover from CQE with error on RQ")
    Signed-off-by: Aya Levin <ayal@mellanox.com>
    Reported-by: Tariq Toukan <tariqt@mellanox.com>
    Reviewed-by: Tariq Toukan <tariqt@mellanox.com>
    Signed-off-by: Saeed Mahameed <saeedm@mellanox.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fbee04b21c3136118d487d7cd94e28a5ed572d39
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Wed Feb 19 18:30:26 2020 +0100

    xen: Enable interrupts when calling _cond_resched()

    commit 8645e56a4ad6dcbf504872db7f14a2f67db88ef2 upstream.

    xen_maybe_preempt_hcall() is called from the exception entry point
    xen_do_hypervisor_callback with interrupts disabled.

    _cond_resched() evades the might_sleep() check in cond_resched() which
    would have caught that and schedule_debug() unfortunately lacks a check
    for irqs_disabled().

    Enable interrupts around the call and use cond_resched() to catch future
    issues.

    Fixes: fdfd811ddde3 ("x86/xen: allow privcmd hypercalls to be preempted")
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Link: https://lore.kernel.org/r/878skypjrh.fsf@nanos.tec.linutronix.de
    Reviewed-by: Juergen Gross <jgross@suse.com>
```

```
    Signed-off-by: Boris Ostrovsky <boris.ostrovsky@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5edacd0e1d0e0216a716a1e0c09eb4a490c42b9a
Author: Prabhakar Kushwaha <pkushwaha@marvell.com>
Date:   Sat Jan 25 03:37:29 2020 +0000

    ata: ahci: Add shutdown to freeze hardware resources of ahci

    commit 10a663a1b15134a5a714aa515e11425a44d4fdf7 upstream.

    device_shutdown() called from reboot or power_shutdown expect
    all devices to be shutdown. Same is true for even ahci pci driver.
    As no ahci shutdown function is implemented, the ata subsystem
    always remains alive with DMA & interrupt support. File system
    related calls should not be honored after device_shutdown().

    So defining ahci pci driver shutdown to freeze hardware (mask
    interrupt, stop DMA engine and free DMA resources).

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

commit 2b984480a8bfa01400c3d4a2ce114df66a37c644
Author: David Howells <dhowells@redhat.com>
Date:   Thu Feb 6 13:57:40 2020 +0000

    rxrpc: Fix call RCU cleanup using non-bh-safe locks

    commit 963485d436ccc2810177a7b08af22336ec2af67b upstream.

    rxrpc_rcu_destroy_call(), which is called as an RCU callback to clean up a
    put call, calls rxrpc_put_connection() which, deep in its bowels, takes a
    number of spinlocks in a non-BH-safe way, including rxrpc_conn_id_lock and
    local->client_conns_lock.  RCU callbacks, however, are normally called from
    softirq context, which can cause lockdep to notice the locking
    inconsistency.

    To get lockdep to detect this, it's necessary to have the connection
    cleaned up on the put at the end of the last of its calls, though normally
    the clean up is deferred.  This can be induced, however, by starting a call
    on an AF_RXRPC socket and then closing the socket without reading the
    reply.

    Fix this by having rxrpc_rcu_destroy_call() punt the destruction to a
    workqueue if in softirq-mode and defer the destruction to process context.

    Note that another way to fix this could be to add a bunch of bh-disable
    annotations to the spinlocks concerned - and there might be more than just
    those two - but that means spending more time with BHs disabled.

    Note also that some of these places were covered by bh-disable spinlocks
    belonging to the rxrpc_transport object, but these got removed without the
    _bh annotation being retained on the next lock in.

    Fixes: 999b69f89241 ("rxrpc: Kill the client connection bundle concept")
    Reported-by: syzbot+d82f3ac8d87e7ccbb2c9@syzkaller.appspotmail.com
    Reported-by: syzbot+3f1fd6b8cbf8702d134e@syzkaller.appspotmail.com
    Signed-off-by: David Howells <dhowells@redhat.com>
    cc: Hillf Danton <hdanton@sina.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 571b72af81f5065d43be3d57e22e8846b86fd839
```

```
Author: Cong Wang <xiyou.wangcong@gmail.com>
Date:   Sun Feb 2 20:30:53 2020 -0800

    netfilter: xt_hashlimit: limit the max size of hashtable

    commit 8d0015a7ab76b8b1e89a3e5f5710a6e5103f2dd5 upstream.

    The user-specified hashtable size is unbound, this could
    easily lead to an OOM or a hung task as we hold the global
    mutex while allocating and initializing the new hashtable.

    Add a max value to cap both cfg->size and cfg->max, as
    suggested by Florian.

    Reported-and-tested-by: syzbot+adf6c6c2be1c3a718121@syzkaller.appspotmail.com
    Signed-off-by: Cong Wang <xiyou.wangcong@gmail.com>
    Reviewed-by: Florian Westphal <fw@strlen.de>
    Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e63c496794a805ad2414be51f42f54af4b27d1be
Author: Takashi Iwai <tiwai@suse.de>
Date:   Fri Feb 14 12:13:15 2020 +0100

    ALSA: seq: Fix concurrent access to queue current tick/time

    commit dc7497795e014d84699c3b8809ed6df35352dd74 upstream.

    snd_seq_check_queue() passes the current tick and time of the given
    queue as a pointer to snd_seq_prioq_cell_out(), but those might be
    updated concurrently by the seq timer update.

    Fix it by retrieving the current tick and time via the proper helper
    functions at first, and pass those values to snd_seq_prioq_cell_out()
    later in the loops.

    snd_seq_timer_get_cur_time() takes a new argument and adjusts with the
    current system time only when it's requested so; this update isn't
    needed for snd_seq_check_queue(), as it's called either from the
    interrupt handler or right after queuing.

    Also, snd_seq_timer_get_cur_tick() is changed to read the value in the
    spinlock for the concurrency, too.

    Reported-by: syzbot+fd5e0eaa1a32999173b2@syzkaller.appspotmail.com
    Link: https://lore.kernel.org/r/20200214111316.26939-3-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2d0cff7c50670a08fdae8afed3b8eba0c48b610c
Author: Takashi Iwai <tiwai@suse.de>
Date:   Fri Feb 14 12:13:14 2020 +0100

    ALSA: seq: Avoid concurrent access to queue flags

    commit bb51e669fa49feb5904f452b2991b240ef31bc97 upstream.

    The queue flags are represented in bit fields and the concurrent
    access may result in unexpected results.  Although the current code
    should be mostly OK as it's only reading a field while writing other
    fields as KCSAN reported, it's safer to cover both with a proper
    spinlock protection.

    This patch fixes the possible concurrent read by protecting with
    q->owner_lock.  Also the queue owner field is protected as well since
```

```
    it's the field to be protected by the lock itself.

    Reported-by: syzbot+65c6c92d04304d0a8efc@syzkaller.appspotmail.com
    Reported-by: syzbot+e60ddfa48717579799dd@syzkaller.appspotmail.com
    Link: https://lore.kernel.org/r/20200214111316.26939-2-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c1cf36fe8e6ad2a015179f9d182582bf29064fe1
Author: Takashi Iwai <tiwai@suse.de>
Date:   Fri Feb 14 12:13:16 2020 +0100

    ALSA: rawmidi: Avoid bit fields for state flags

    commit dfa9a5efe8b932a84b3b319250aa3ac60c20f876 upstream.

    The rawmidi state flags (opened, append, active_sensing) are stored in
    bit fields that can be potentially racy when concurrently accessed
    without any locks.  Although the current code should be fine, there is
    also no any real benefit by keeping the bitfields for this kind of
    short number of members.

    This patch changes those bit fields flags to the simple bool fields.
    There should be no size increase of the snd_rawmidi_substream by this
    change.

    Reported-by: syzbot+576cc007eb9f2c968200@syzkaller.appspotmail.com
    Link: https://lore.kernel.org/r/20200214111316.26939-4-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0b1fd8dc6210aa00c2c550986868d1f0fc9b208c
Author: Christoph Hellwig <hch@lst.de>
Date:   Mon Feb 3 18:11:10 2020 +0100

    dma-direct: relax addressability checks in dma_direct_supported

    commit 91ef26f914171cf753330f13724fd9142b5b1640 upstream.

    dma_direct_supported tries to find the minimum addressable bitmask
    based on the end pfn and optional magic that architectures can use
    to communicate the size of the magic ZONE_DMA that can be used
    for bounce buffering.  But between the DMA offsets that can change
    per device (or sometimes even region), the fact the ZONE_DMA isn't
    even guaranteed to be the lowest addresses and failure of having
    proper interfaces to the MM code this fails at least for one
    arm subarchitecture.

    As all the legacy DMA implementations have supported 32-bit DMA
    masks, and 32-bit masks are guranteed to always work by the API
    contract (using bounce buffers if needed), we can short cut the
    complicated check and always return true without breaking existing
    assumptions.  Hopefully we can properly clean up the interaction
    with the arch defined zones and the bootmem allocator eventually.

    Fixes: ad3c7b18c5b3 ("arm: use swiotlb for bounce buffering on LPAE configs")
    Reported-by: Peter Ujfalusi <peter.ujfalusi@ti.com>
    Signed-off-by: Christoph Hellwig <hch@lst.de>
    Tested-by: Peter Ujfalusi <peter.ujfalusi@ti.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 219e50d184139cc0064abfcbfc02c2414baba18f
Author: Xiaoguang Wang <xiaoguang.wang@linux.alibaba.com>
Date:   Sat Feb 22 14:46:05 2020 +0800
```

```
    io_uring: fix __io_iopoll_check deadlock in io_sq_thread

    commit c7849be9cc2dd2754c48ddbaca27c2de6d80a95d upstream.

    Since commit a3a0e43fd770 ("io_uring: don't enter poll loop if we have
    CQEs pending"), if we already events pending, we won't enter poll loop.
    In case SETUP_IOPOLL and SETUP_SQPOLL are both enabled, if app has
    been terminated and don't reap pending events which are already in cq
    ring, and there are some reqs in poll_list, io_sq_thread will enter
    __io_iopoll_check(), and find pending events, then return, this loop
    will never have a chance to exit.

    I have seen this issue in fio stress tests, to fix this issue, let
    io_sq_thread call io_iopoll_getevents() with argument 'min' being zero,
    and remove __io_iopoll_check().

    Fixes: a3a0e43fd770 ("io_uring: don't enter poll loop if we have CQEs pending")
    Signed-off-by: Xiaoguang Wang <xiaoguang.wang@linux.alibaba.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2ac6220a3343e6efe7b1f6ddfa4c39e348591f18
Author: Stefano Garzarella <sgarzare@redhat.com>
Date:   Fri Feb 21 16:42:16 2020 +0100

    io_uring: prevent sq_thread from spinning when it should stop

    commit 7143b5ac5750f404ff3a594b34fdf3fc2f99f828 upstream.

    This patch drops 'cur_mm' before calling cond_resched(), to prevent
    the sq_thread from spinning even when the user process is finished.

    Before this patch, if the user process ended without closing the
    io_uring fd, the sq_thread continues to spin until the
    'sq_thread_idle' timeout ends.

    In the worst case where the 'sq_thread_idle' parameter is bigger than
    INT_MAX, the sq_thread will spin forever.

    Fixes: 6c271ce2f1d5 ("io_uring: add submission polling")
    Signed-off-by: Stefano Garzarella <sgarzare@redhat.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8083cc04592545b9e9019d7e7893f892c529f12c
Author: Douglas Anderson <dianders@chromium.org>
Date:   Thu Feb 20 20:04:12 2020 -0800

    scripts/get_maintainer.pl: deprioritize old Fixes: addresses

    commit 0ef82fcefb99300ede6f4d38a8100845b2dc8e30 upstream.

    Recently, I found that get_maintainer was causing me to send emails to
    the old addresses for maintainers.  Since I usually just trust the
    output of get_maintainer to know the right email address, I didn't even
    look carefully and fired off two patch series that went to the wrong
    place.  Oops.

    The problem was introduced recently when trying to add signatures from
    Fixes.  The problem was that these email addresses were added too early
    in the process of compiling our list of places to send.  Things added to
    the list earlier are considered more canonical and when we later added
    maintainer entries we ended up deduplicating to the old address.

    Here are two examples using mainline commits (to make it easier to
```

```
    replicate) for the two maintainers that I messed up recently:

      $ git format-patch d8549bcd0529~..d8549bcd0529
      $ ./scripts/get_maintainer.pl 0001-clk-Add-clk_hw*.patch | grep Boyd
      Stephen Boyd <sboyd@codeaurora.org>...

      $ git format-patch 6d1238aa3395~..6d1238aa3395
      $ ./scripts/get_maintainer.pl 0001-arm64-dts-qcom-qcs404*.patch | grep Andy
      Andy Gross <andy.gross@linaro.org>

    Let's move the adding of addresses from Fixes: to the end since the
    email addresses from these are much more likely to be older.

    After this patch the above examples get the right addresses for the two
    examples.

    Link:
http://lkml.kernel.org/r/20200127095001.1.I41fba9f33590bfd92cd01960161d8384268c6569@changeid
    Fixes: 2f5bd343694e ("scripts/get_maintainer.pl: add signatures from Fixes: <badcommit>
lines in commit message")
    Signed-off-by: Douglas Anderson <dianders@chromium.org>
    Acked-by: Joe Perches <joe@perches.com>
    Cc: Stephen Boyd <sboyd@kernel.org>
    Cc: Bjorn Andersson <bjorn.andersson@linaro.org>
    Cc: Andy Gross <agross@kernel.org>
    Cc: Kees Cook <keescook@chromium.org>
    Cc: Dan Carpenter <dan.carpenter@oracle.com>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e81d244d482eb98cc8bf0a021e6c8fa689a447de
Author: Vincenzo Frascino <vincenzo.frascino@arm.com>
Date:   Tue Feb 18 16:49:06 2020 +0000

    arm64: lse: Fix LSE atomics with LLVM

    commit dd1f6308b28edf0452dd5dc7877992903ec61e69 upstream.

    Commit e0d5896bd356 ("arm64: lse: fix LSE atomics with LLVM's integrated
    assembler") broke the build when clang is used in connjunction with the
    binutils assembler ("-no-integrated-as"). This happens because
    __LSE_PREAMBLE is defined as ".arch armv8-a+lse", which overrides the
    version of the CPU architecture passed via the "-march" paramter to gas:

    $ aarch64-none-linux-gnu-as -EL -I ./arch/arm64/include
                                    -I ./arch/arm64/include/generated
                                    -I ./include -I ./include
                                    -I ./arch/arm64/include/uapi
                                    -I ./arch/arm64/include/generated/uapi
                                    -I ./include/uapi -I ./include/generated/uapi
                                    -I ./init -I ./init
                                    -march=armv8.3-a -o init/do_mounts.o
                                    /tmp/do_mounts-d7992a.s
    /tmp/do_mounts-d7992a.s: Assembler messages:
    /tmp/do_mounts-d7992a.s:1959: Error: selected processor does not support `autiasp'
    /tmp/do_mounts-d7992a.s:2021: Error: selected processor does not support `paciasp'
    /tmp/do_mounts-d7992a.s:2157: Error: selected processor does not support `autiasp'
    /tmp/do_mounts-d7992a.s:2175: Error: selected processor does not support `paciasp'
    /tmp/do_mounts-d7992a.s:2494: Error: selected processor does not support `autiasp'

    Fix the issue by replacing ".arch armv8-a+lse" with ".arch_extension lse".
    Sami confirms that the clang integrated assembler does now support the
    '.arch_extension' directive, so this change will be fine even for LTO
```

```
    builds in future.

    Fixes: e0d5896bd356cd ("arm64: lse: fix LSE atomics with LLVM's integrated assembler")
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: Will Deacon <will@kernel.org>
    Reported-by: Amit Kachhap <Amit.Kachhap@arm.com>
    Tested-by: Sami Tolvanen <samitolvanen@google.com>
    Signed-off-by: Vincenzo Frascino <vincenzo.frascino@arm.com>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4c3583cc299750915b0a4497d1b8b308859d76f3
Author: Johannes Krude <johannes@krude.de>
Date:   Wed Feb 12 20:32:27 2020 +0100

    bpf, offload: Replace bitwise AND by logical AND in bpf_prog_offload_info_fill

    commit e20d3a055a457a10a4c748ce5b7c2ed3173a1324 upstream.

    This if guards whether user-space wants a copy of the offload-jited
    bytecode and whether this bytecode exists. By erroneously doing a bitwise
    AND instead of a logical AND on user- and kernel-space buffer-size can lead
    to no data being copied to user-space especially when user-space size is a
    power of two and bigger then the kernel-space buffer.

    Fixes: fcfb126defda ("bpf: add new jited info fields in bpf_dev_offload and
bpf_prog_info")
    Signed-off-by: Johannes Krude <johannes@krude.de>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Jakub Kicinski <kuba@kernel.org>
    Link: https://lore.kernel.org/bpf/20200212193227.GA3769@phlox.h.transitiv.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b46c8fdba51271e6ce6b021265ac8b69f8aba907
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Wed Feb 12 12:19:41 2020 +0100

    genirq/proc: Reject invalid affinity masks (again)

    commit cba6437a1854fde5934098ec3bd0ee83af3129f5 upstream.

    Qian Cai reported that the WARN_ON() in the x86/msi affinity setting code,
    which catches cases where the affinity setting is not done on the CPU which
    is the current target of the interrupt, triggers during CPU hotplug stress
    testing.

    It turns out that the warning which was added with the commit addressing
    the MSI affinity race unearthed yet another long standing bug.

    If user space writes a bogus affinity mask, i.e. it contains no online CPUs,
    then it calls irq_select_affinity_usr(). This was introduced for ALPHA in

      eee45269b0f5 ("[PATCH] Alpha: convert to generic irq framework (generic part)")

    and subsequently made available for all architectures in

      18404756765c ("genirq: Expose default irq affinity mask (take 3)")

    which introduced the circumvention of the affinity setting restrictions for
    interrupt which cannot be moved in process context.

    The whole exercise is bogus in various aspects:

      1) If the interrupt is already started up then there is absolutely
         no point to honour a bogus interrupt affinity setting from user
```

```
          space. The interrupt is already assigned to an online CPU and it
          does not make any sense to reassign it to some other randomly
          chosen online CPU.

       2) If the interupt is not yet started up then there is no point
          either. A subsequent startup of the interrupt will invoke
          irq_setup_affinity() anyway which will chose a valid target CPU.

    So the only correct solution is to just return -EINVAL in case user space
    wrote an affinity mask which does not contain any online CPUs, except for
    ALPHA which has it's own magic sauce for this.

    Fixes: 18404756765c ("genirq: Expose default irq affinity mask (take 3)")
    Reported-by: Qian Cai <cai@lca.pw>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Tested-by: Qian Cai <cai@lca.pw>
    Link: https://lkml.kernel.org/r/878sl8xdbm.fsf@nanos.tec.linutronix.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f778537486d5a4ec41ed9b7d35ea385bf18b4070
Author: Tianjia Zhang <tianjia.zhang@linux.alibaba.com>
Date:   Mon Feb 10 20:44:39 2020 +0800

    crypto: rename sm3-256 to sm3 in hash_algo_name

    commit 6a30e1b1dcad0ba94fae757f797812d7d8dcb72c upstream.

    The name sm3-256 is defined in hash_algo_name in hash_info, but the
    algorithm name implemented in sm3_generic.c is sm3, which will cause
    the sm3-256 algorithm to be not found in some application scenarios of
    the hash algorithm, and an ENOENT error will occur. For example,
    IMA, keys, and other subsystems that reference hash_algo_name all use
    the hash algorithm of sm3.

    Fixes: 5ca4c20cfd37 ("keys, trusted: select hash algorithm for TPM2 chips")
    Signed-off-by: Tianjia Zhang <tianjia.zhang@linux.alibaba.com>
    Reviewed-by: Pascal van Leeuwen <pvanleeuwen@rambus.com>
    Signed-off-by: Mimi Zohar <zohar@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7549b2fc3de608577c2ace781bcc11ac45b200c7
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 10 10:36:56 2020 +0100

    iommu/vt-d: Fix compile warning from intel-svm.h

    commit e7598fac323aad0e502415edeffd567315994dd6 upstream.

    The intel_svm_is_pasid_valid() needs to be marked inline, otherwise it
    causes the compile warning below:

      CC [M]  drivers/dma/idxd/cdev.o
    In file included from drivers/dma/idxd/cdev.c:9:0:
    ./include/linux/intel-svm.h:125:12: warning: ‘intel_svm_is_pasid_valid’ defined but not
used [-Wunused-function]
     static int intel_svm_is_pasid_valid(struct device *dev, int pasid)
                ^~~~~~~~~~~~~~~~~~~~~~~~

    Reported-by: Borislav Petkov <bp@alien8.de>
    Fixes: 15060aba71711 ('iommu/vt-d: Helper function to query if a pasid has any active
users')
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0cdc01b0ad58e95b5b89f84195e0595a2fbbaf5d
```

```
Author: Aditya Pakki <pakki001@umn.edu>
Date:   Fri Feb 14 12:21:01 2020 -0600

    ecryptfs: replace BUG_ON with error handling code

    commit 2c2a7552dd6465e8fde6bc9cccf8d66ed1c1eb72 upstream.

    In crypt_scatterlist, if the crypt_stat argument is not set up
    correctly, the kernel crashes. Instead, by returning an error code
    upstream, the error is handled safely.

    The issue is detected via a static analysis tool written by us.

    Fixes: 237fead619984 (ecryptfs: fs/Makefile and fs/Kconfig)
    Signed-off-by: Aditya Pakki <pakki001@umn.edu>
    Signed-off-by: Tyler Hicks <code@tyhicks.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5eb48fd9cbc68e18f9357c4b1c4ab2f22448cb9e
Author: Oleksandr Suvorov <oleksandr.suvorov@toradex.com>
Date:   Wed Feb 5 18:04:36 2020 +0200

    ASoC: fsl_sai: Fix exiting path on probing failure

    commit d1520889782dff58610c0b6b54d4cf3211ceb690 upstream.

    If the imx-sdma driver is built as a module, the fsl-sai device doesn't
    disable on probing failure, which causes the warning in the next probing:

    ================================================================
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    fsl-sai 308a0000.sai: Unbalanced pm_runtime_enable!
    ================================================================

    Disabling the device properly fixes the issue.

    Fixes: 812ad463e089 ("ASoC: fsl_sai: Add support for runtime pm")
    Signed-off-by: Oleksandr Suvorov <oleksandr.suvorov@toradex.com>
    Link: https://lore.kernel.org/r/20200205160436.3813642-1-oleksandr.suvorov@toradex.com
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f1bd94354c3bf3d97f334cb6b71a820e217ccc49
Author: Arnd Bergmann <arnd@arndb.de>
Date:   Thu Jan 30 15:05:45 2020 +0200

    ASoC: atmel: fix atmel_ssc_set_audio link failure

    commit 9437bfda00f3b26eb5f475737ddaaf4dc07fee4f upstream.

    The ssc audio driver can call into both pdc and dma backends.  With the
    latest rework, the logic to do this in a safe way avoiding link errors
    was removed, bringing back link errors that were fixed long ago in commit
    061981ff8cc8 ("ASoC: atmel: properly select dma driver state") such as

    sound/soc/atmel/atmel_ssc_dai.o: In function `atmel_ssc_set_audio':
    atmel_ssc_dai.c:(.text+0xac): undefined reference to `atmel_pcm_pdc_platform_register'

    Fix it this time using Makefile hacks and a comment to prevent this
    from accidentally getting removed again rather than Kconfig hacks.
```

```
    Fixes: 18291410557f ("ASoC: atmel: enable SOC_SSC_PDC and SOC_SSC_DMA in Kconfig")
    Signed-off-by: Arnd Bergmann <arnd@arndb.de>
    Signed-off-by: Codrin Ciubotariu <codrin.ciubotariu@microchip.com>
    Link: https://lore.kernel.org/r/20200130130545.31148-1-codrin.ciubotariu@microchip.com
    Reviewed-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a9f659d837fedca7518b6968e26133fe5af40935
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Wed Feb 5 15:32:17 2020 +0300

    staging: greybus: use after free in gb_audio_manager_remove_all()

    commit b7db58105b80fa9232719c8329b995b3addfab55 upstream.

    When we call kobject_put() and it's the last reference to the kobject
    then it calls gb_audio_module_release() and frees module.  We dereference
    "module" on the next line which is a use after free.

    Fixes: c77f85bbc91a ("greybus: audio: Fix incorrect counting of 'ida'")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Acked-by: Viresh Kumar <viresh.kumar@linaro.org>
    Reviewed-by: Vaibhav Agarwal <vaibhav.sr@gmail.com>
    Link: https://lore.kernel.org/r/20200205123217.jreendkyxulqsool@kili.mountain
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c0a6ee82b267372f0ac630ae4b9f4c56479c382b
Author: Colin Ian King <colin.king@canonical.com>
Date:   Sun Jan 26 22:05:49 2020 +0000

    staging: rtl8723bs: fix copy of overlapping memory

    commit 8ae9a588ca35eb9c32dc03299c5e1f4a1e9a9617 upstream.

    Currently the rtw_sprintf prints the contents of thread_name
    onto thread_name and this can lead to a potential copy of a
    string over itself. Avoid this by printing the literal string RTWHALXT
    instread of the contents of thread_name.

    Addresses-Coverity: ("copy of overlapping memory")
    Fixes: 554c0a3abf21 ("staging: Add rtl8723bs sdio wifi driver")
    Signed-off-by: Colin Ian King <colin.king@canonical.com>
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Link: https://lore.kernel.org/r/20200126220549.9849-1-colin.king@canonical.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4404095458de4c49095f112db4e057b4d5735964
Author: Minas Harutyunyan <Minas.Harutyunyan@synopsys.com>
Date:   Tue Jan 21 14:24:04 2020 +0400

    usb: dwc2: Fix in ISOC request length checking

    commit 860ef6cd3f90b84a1832f8a6485c90c34d3b588b upstream.

    Moved ISOC request length checking from dwc2_hsotg_start_req() function to
    dwc2_hsotg_ep_queue().

    Fixes: 4fca54aa58293 ("usb: gadget: s3c-hsotg: add multi count support")
    Signed-off-by: Minas Harutyunyan <hminas@synopsys.com>
    Signed-off-by: Felipe Balbi <balbi@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cf242e3b992c93253ed73d4e319c67a3268a6e17
```

```
Author: Jack Pham <jackp@codeaurora.org>
Date:   Thu Jan 30 19:10:35 2020 -0800

    usb: gadget: composite: Fix bMaxPower for SuperSpeedPlus

    commit c724417baf162bd3e035659e22cdf990cfb0d917 upstream.

    SuperSpeedPlus peripherals must report their bMaxPower of the
    configuration descriptor in units of 8mA as per the USB 3.2
    specification. The current switch statement in encode_bMaxPower()
    only checks for USB_SPEED_SUPER but not USB_SPEED_SUPER_PLUS so
    the latter falls back to USB 2.0 encoding which uses 2mA units.
    Replace the switch with a simple if/else.

    Fixes: eae5820b852f ("usb: gadget: composite: Write SuperSpeedPlus config descriptors")
    Signed-off-by: Jack Pham <jackp@codeaurora.org>
    Signed-off-by: Felipe Balbi <balbi@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4fc010340bc162cf147712fd4d7ee83d4d764692
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Wed Feb 12 21:09:00 2020 -0800

    scsi: Revert "target: iscsi: Wait for all commands to finish before freeing a session"

    commit 807b9515b7d044cf77df31f1af9d842a76ecd5cb upstream.

    Since commit e9d3009cb936 introduced a regression and since the fix for
    that regression was not perfect, revert this commit.

    Link: https://marc.info/?l=target-devel&m=158157054906195
    Cc: Rahul Kundu <rahul.kundu@chelsio.com>
    Cc: Mike Marciniszyn <mike.marciniszyn@intel.com>
    Cc: Sagi Grimberg <sagi@grimberg.me>
    Reported-by: Dakshaja Uppalapati <dakshaja@chelsio.com>
    Fixes: e9d3009cb936 ("scsi: target: iscsi: Wait for all commands to finish before
freeing a session")
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e473567240c086efa4427c644f3a8fbe78cef144
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Wed Feb 12 21:08:59 2020 -0800

    scsi: Revert "RDMA/isert: Fix a recently introduced regression related to logout"

    commit 76261ada16dcc3be610396a46d35acc3efbda682 upstream.

    Since commit 04060db41178 introduces soft lockups when toggling network
    interfaces, revert it.

    Link: https://marc.info/?l=target-devel&m=158157054906196
    Cc: Rahul Kundu <rahul.kundu@chelsio.com>
    Cc: Mike Marciniszyn <mike.marciniszyn@intel.com>
    Cc: Sagi Grimberg <sagi@grimberg.me>
    Reported-by: Dakshaja Uppalapati <dakshaja@chelsio.com>
    Fixes: 04060db41178 ("scsi: RDMA/isert: Fix a recently introduced regression related to
logout")
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b5feb0031e6fc03858ec665c92bb0e2044ea4da5
Author: Rob Clark <robdclark@chromium.org>
```

```
Date:   Thu Feb 13 12:01:35 2020 -0800

    drm/msm/dpu: fix BGR565 vs RGB565 confusion

    commit 8fc7036ee652207ca992fbb9abb64090c355a9e0 upstream.

    The component order between the two was swapped, resulting in incorrect
    color when games with 565 visual hit the overlay path instead of GPU
    composition.

    Fixes: 25fdd5933e4c ("drm/msm: Add SDM845 DPU support")
    Signed-off-by: Rob Clark <robdclark@chromium.org>
    Reviewed-by: Sean Paul <seanpaul@chromium.org>
    Signed-off-by: Rob Clark <robdclark@chromium.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 74b0c1ce3c1b2f242470aefb9380b51e343a2860
Author: Matt Roper <matthew.d.roper@intel.com>
Date:   Thu Feb 6 16:14:16 2020 -0800

    drm/i915/ehl: Update port clock voltage level requirements

    commit 58e9121c32a245fab47f29ab4ad29dd62470a7e8 upstream.

    Voltage level depends not only on the cdclk, but also on the DDI clock.
    Last time the bspec voltage level table for EHL was updated, we only
    updated the cdclk requirements, but forgot to account for the new port
    clock criteria.

    Bspec: 21809
    Fixes: d147483884ed ("drm/i915/ehl: Update voltage level checks")
    Cc: José Roberto de Souza <jose.souza@intel.com>
    Signed-off-by: Matt Roper <matthew.d.roper@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200207001417.1229251-1-
matthew.d.roper@intel.com
    Reviewed-by: José Roberto de Souza <jose.souza@intel.com>
    (cherry picked from commit 9d5fd37ed7e26efdbe90f492d7eb8b53dcdb61d6)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8f23cafc7dab16ff42b90873c7855f0d6d18a168
Author: Chris Wilson <chris@chris-wilson.co.uk>
Date:   Thu Feb 6 20:49:13 2020 +0000

    drm/i915/gt: Protect defer_request() from new waiters

    commit 19b5f3b419a61808ff2713f1f30b8a88fe14ac9b upstream.

    Mika spotted

    <4>[17436.705441] general protection fault: 0000 [#1] PREEMPT SMP PTI
    <4>[17436.705447] CPU: 2 PID: 0 Comm: swapper/2 Not tainted 5.5.0+ #1
    <4>[17436.705449] Hardware name: System manufacturer System Product Name/Z170M-PLUS,
BIOS 3805 05/16/2018
    <4>[17436.705512] RIP: 0010:__execlists_submission_tasklet+0xc4d/0x16e0 [i915]
    <4>[17436.705516] Code: c5 4c 8d 60 e0 75 17 e9 8c 07 00 00 49 8b 44 24 20 49 39 c5 4c
8d 60 e0 0f 84 7a 07 00 00 49 8b 5c 24 08 49 8b 87 80 00 00 00 <48> 39 83 d8 fe ff ff 75 d9
48 8b 83 88 fe ff ff a8 01 0f 84 b6 05
    <4>[17436.705518] RSP: 0018:ffffc9000012ce80 EFLAGS: 00010083
    <4>[17436.705521] RAX: ffff88822ae42000 RBX: 5a5a5a5a5a5a5a5a RCX: dead000000000122
    <4>[17436.705523] RDX: ffff88822ae42588 RSI: ffff8881e32a7908 RDI: ffff8881c429fd48
    <4>[17436.705525] RBP: ffffc9000012cf00 R08: ffff88822ae42588 R09: 00000000fffffffe
    <4>[17436.705527] R10: ffff8881c429fb80 R11: 00000000a677cf08 R12: ffff8881c42a0aa8
    <4>[17436.705529] R13: ffff8881c429fd38 R14: ffff88822ae42588 R15: ffff8881c429fb80
    <4>[17436.705532] FS:  0000000000000000(0000) GS:ffff88822ed00000(0000)
```

```
 knlGS:0000000000000000
     <4>[17436.705534] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
     <4>[17436.705536] CR2: 00007f858c76d000 CR3: 0000000005610003 CR4: 00000000003606e0
     <4>[17436.705538] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
     <4>[17436.705540] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
     <4>[17436.705542] Call Trace:
     <4>[17436.705545]  <IRQ>
     <4>[17436.705603]  execlists_submission_tasklet+0xc0/0x130 [i915]

     which is us consuming a partially initialised new waiter in
     defer_requests(). We can prevent this by initialising the i915_dependency
     prior to making it visible, and since we are using a concurrent
     list_add/iterator mark them up to the compiler.

     Fixes: 8ee36e048c98 ("drm/i915/execlists: Minimalistic timeslicing")
     Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
     Cc: Mika Kuoppala <mika.kuoppala@linux.intel.com>
     Reviewed-by: Mika Kuoppala <mika.kuoppala@linux.intel.com>
     Link: https://patchwork.freedesktop.org/patch/msgid/20200206204915.2636606-2-
 chris@chris-wilson.co.uk
     (cherry picked from commit f14f27b1663269a81ed62d3961fe70250a1a0623)
     Signed-off-by: Jani Nikula <jani.nikula@intel.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit 9029ddeeb104ac9147b0ca851f171631cd882003
 Author: Chris Wilson <chris@chris-wilson.co.uk>
 Date:   Thu Jan 30 16:45:53 2020 +0000

     drm/i915/gem: Require per-engine reset support for non-persistent contexts

     commit dea8d5ce46d7e7f7270b9804df7d1174f88bfd99 upstream.

     To enable non-persistent contexts, we require a means of cancelling any
     inflight work from that context. This is first done "gracefully" by
     using preemption to kick the active context off the engine, and then
     forcefully by resetting the engine if it is active. If we are unable to
     reset the engine to remove hostile userspace, we should not allow
     userspace to opt into using non-persistent contexts.

     If the per-engine reset fails, we still do a full GPU reset, but that is
     rare and usually indicative of much deeper issues. The damage is already
     done. However, the goal of the interface to allow long running compute
     jobs without causing collateral damage elsewhere, and if we are unable
     to support that we should make that known by not providing the
     interface (and falsely pretending we can).

     Fixes: a0e047156cde ("drm/i915/gem: Make context persistence optional")
     Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
     Cc: Joonas Lahtinen <joonas.lahtinen@linux.intel.com>
     Cc: Jon Bloomfield <jon.bloomfield@intel.com>
     Reviewed-by: Joonas Lahtinen <joonas.lahtinen@linux.intel.com>
     Link: https://patchwork.freedesktop.org/patch/msgid/20200130164553.1937718-1-
 chris@chris-wilson.co.uk
     (cherry picked from commit d1b9b5f127bc3797fc274cfa4f363e039f045c3a)
     Signed-off-by: Jani Nikula <jani.nikula@intel.com>
     Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 commit d562a06a010a2a3be3e0982cdc7b77c4869ddfb7
 Author: Tomi Valkeinen <tomi.valkeinen@ti.com>
 Date:   Mon Dec 9 10:27:07 2019 +0200

     drm/bridge: tc358767: fix poll timeouts

     commit 8a6483ac634acda3f599f50082c652d2d37199c7 upstream.
```

```
    Link training fails with:

      Link training timeout waiting for LT_LOOPDONE!
      main link enable error: -110

    This is caused by too tight timeouts, which were changed recently in
    aa92213f388b ("drm/bridge: tc358767: Simplify polling in tc_link_training()").

    With a quick glance, the commit does not change the timeouts. However,
    the method of delaying/sleeping is different, and as the timeout in the
    previous implementation was not explicit, the new version in practice
    has much tighter timeout.

    The same change was made to other parts in the driver, but the link
    training timeout is the only one I have seen causing issues.
    Nevertheless, 1 us sleep is not very sane, and the timeouts look pretty
    tight, so lets fix all the timeouts.

    One exception was the aux busy poll, where the poll sleep was much
    longer than necessary (or optimal).

    I measured the times on my setup, and now the sleep times are set to
    such values that they result in multiple loops, but not too many (say,
    5-10 loops). The timeouts were all increased to 100ms, which should be
    more than enough for all of these, but in case of bad errors, shouldn't
    stop the driver as multi-second timeouts could do.

    Signed-off-by: Tomi Valkeinen <tomi.valkeinen@ti.com>
    Fixes: aa92213f388b ("drm/bridge: tc358767: Simplify polling in tc_link_training()")
    Tested-by: Andrey Smirnov <andrew.smirnov@gmail.com>
    Reviewed-by: Neil Armstrong <narmstrong@baylibre.com>
    Signed-off-by: Neil Armstrong <narmstrong@baylibre.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20191209082707.24531-1-
tomi.valkeinen@ti.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit db7fe57548e2de3b1b846952cdb10630f080d800
Author: Tony Nguyen <anthony.l.nguyen@intel.com>
Date:   Thu Feb 6 01:20:08 2020 -0800

    ice: Remove possible null dereference

    commit 0a6ea04e3bbd20833d2b49296e5adc1c5bb86386 upstream.

    Commit 1f45ebe0d8fb ("ice: add extra check for null Rx descriptor") moved
    the call to ice_construct_skb() under a null check as Coverity reported a
    possible use of null skb. However, the original call was not deleted, do so
    now.

    Fixes: 1f45ebe0d8fb ("ice: add extra check for null Rx descriptor")
    Reported-by: Bruce Allan <bruce.w.allan@intel.com>
    Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
    Tested-by: Andrew Bowers <andrewx.bowers@intel.com>
    Signed-off-by: Jeff Kirsher <jeffrey.t.kirsher@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 21ceee581d1ea24adda1320f5fb5bb2fe5470fc2
Author: Igor Druzhinin <igor.druzhinin@citrix.com>
Date:   Mon Feb 3 15:07:01 2020 +0000

    drm/i915/gvt: more locking for ppgtt mm LRU list

    commit 0e9d7bb293f3f9c3ee376b126141407efb265f31 upstream.

    When the lock was introduced in commit 72aabfb862e40 ("drm/i915/gvt: Add mutual
```

```
    lock for ppgtt mm LRU list") one place got lost.

    Fixes: 72aabfb862e4 ("drm/i915/gvt: Add mutual lock for ppgtt mm LRU list")
    Signed-off-by: Igor Druzhinin <igor.druzhinin@citrix.com>
    Reviewed-by: Zhenyu Wang <zhenyuw@linux.intel.com>
    Signed-off-by: Zhenyu Wang <zhenyuw@linux.intel.com>
    Link: http://patchwork.freedesktop.org/patch/msgid/1580742421-25194-1-git-send-email-
igor.druzhinin@citrix.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a5ec9799f87a4b52439073bd2a00466cdc696c5f
Author: Chris Wilson <chris@chris-wilson.co.uk>
Date:   Tue Jan 14 16:00:30 2020 +0000

    drm/i915/selftests: Add a mock i915_vma to the mock_ring

    commit 1fdea0cb0dba0d42ffcfb619b349c1a2afa2492e upstream.

    Add a i915_vma to the mock_engine/mock_ring so that the core code can
    always assume the presence of ring->vma.

    Fixes: 8ccfc20a7d56 ("drm/i915/gt: Mark ring->vma as active while pinned")
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Reviewed-by: Mika Kuoppala <mika.kuoppala@linux.intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200114160030.2468927-1-
chris@chris-wilson.co.uk
    (cherry picked from commit b63b4feaef7363d2cf46dd76bb6e87e060b2b0de)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 56f9e8137206ee3d9754cc2cbf589d23654e4b09
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Thu Feb 27 10:45:54 2020 +0100

    Revert "dmaengine: imx-sdma: Fix memory leak"

    This reverts commit 7ac78dd1e0992fd6d2ae375cc0dd6456c632f605 which is
    commit 02939cd167095f16328a1bd5cab5a90b550606df upstream.

    Andreas writes:
            This patch breaks our imx6 board with the attached trace.
            Reverting the patch makes it boot again.

    Reported-by: Andreas Tobler <andreas.tobler@onway.ch>
    Cc: Sascha Hauer <s.hauer@pengutronix.de>
    Cc: Robin Gong <yibin.gong@nxp.com>
    Cc: Vinod Koul <vkoul@kernel.org>
    Cc: Sasha Levin <sashal@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c3e6951b732154174be6e755c77d9eecc3c6c69a
Author: Filipe Manana <fdmanana@suse.com>
Date:   Thu Feb 20 13:29:49 2020 +0000

    Btrfs: fix deadlock during fast fsync when logging prealloc extents beyond eof

    commit a5ae50dea9111db63d30d700766dd5509602f7ad upstream.

    While logging the prealloc extents of an inode during a fast fsync we call
    btrfs_truncate_inode_items(), through btrfs_log_prealloc_extents(), while
    holding a read lock on a leaf of the inode's root (not the log root, the
    fs/subvol root), and then that function locks the file range in the inode's
    iotree. This can lead to a deadlock when:

    * the fsync is ranged
```

```
* the file has prealloc extents beyond eof

* writeback for a range different from the fsync range starts
  during the fsync

* the size of the file is not sector size aligned

Because when finishing an ordered extent we lock first a file range and
then try to COW the fs/subvol tree to insert an extent item.

The following diagram shows how the deadlock can happen.

           CPU 1                                          CPU 2

  btrfs_sync_file()
    --> for range [0, 1MiB)

    --> inode has a size of
        1MiB and has 1 prealloc
        extent beyond the
        i_size, starting at offset
        4MiB

    flushes all delalloc for the
    range [0MiB, 1MiB) and waits
    for the respective ordered
    extents to complete

                                                  --> before task at CPU 1 locks the
                                                      inode, a write into file range
                                                      [1MiB, 2MiB + 1KiB) is made

                                                  --> i_size is updated to 2MiB + 1KiB

                                                  --> writeback is started for that
                                                      range, [1MiB, 2MiB + 4KiB)
                                                      --> end offset rounded up to
                                                          be sector size aligned

    btrfs_log_dentry_safe()
      btrfs_log_inode_parent()
        btrfs_log_inode()

          btrfs_log_changed_extents()
            btrfs_log_prealloc_extents()
              --> does a search on the
                  inode's root
              --> holds a read lock on
                  leaf X

                                                  btrfs_finish_ordered_io()
                                                    --> locks range [1MiB, 2MiB + 4KiB)
                                                        --> end offset rounded up
                                                            to be sector size aligned

                                                    --> tries to cow leaf X, through
                                                        insert_reserved_file_extent()
                                                        --> already locked by the
                                                            task at CPU 1

              btrfs_truncate_inode_items()

                --> gets an i_size of
                    2MiB + 1KiB, which is
```

```
                        not sector size
                        aligned

                    --> tries to lock file
                        range [2MiB, (u64)-1)
                        --> the start range
                            is rounded down
                            from 2MiB + 1K
                            to 2MiB to be sector
                            size aligned

                        --> but the subrange
                            [2MiB, 2MiB + 4KiB) is
                            already locked by
                            task at CPU 2 which
                            is waiting to get a
                            write lock on leaf X
                            for which we are
                            holding a read lock

                                    *** deadlock ***
```

This results in a stack trace like the following, triggered by test case generic/561 from fstests:

```
  [ 2779.973608] INFO: task kworker/u8:6:247 blocked for more than 120 seconds.
  [ 2779.979536]       Not tainted 5.6.0-rc2-btrfs-next-53 #1
  [ 2779.984503] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this
message.
  [ 2779.990136] kworker/u8:6    D    0   247      2 0x80004000
  [ 2779.990457] Workqueue: btrfs-endio-write btrfs_work_helper [btrfs]
  [ 2779.990466] Call Trace:
  [ 2779.990491]  ? __schedule+0x384/0xa30
  [ 2779.990521]  schedule+0x33/0xe0
  [ 2779.990616]  btrfs_tree_read_lock+0x19e/0x2e0 [btrfs]
  [ 2779.990632]  ? remove_wait_queue+0x60/0x60
  [ 2779.990730]  btrfs_read_lock_root_node+0x2f/0x40 [btrfs]
  [ 2779.990782]  btrfs_search_slot+0x510/0x1000 [btrfs]
  [ 2779.990869]  btrfs_lookup_file_extent+0x4a/0x70 [btrfs]
  [ 2779.990944]  __btrfs_drop_extents+0x161/0x1060 [btrfs]
  [ 2779.990987]  ? mark_held_locks+0x6d/0xc0
  [ 2779.990994]  ? __slab_alloc.isra.49+0x99/0x100
  [ 2779.991060]  ? insert_reserved_file_extent.constprop.19+0x64/0x300 [btrfs]
  [ 2779.991145]  insert_reserved_file_extent.constprop.19+0x97/0x300 [btrfs]
  [ 2779.991222]  ? start_transaction+0xdd/0x5c0 [btrfs]
  [ 2779.991291]  btrfs_finish_ordered_io+0x4f4/0x840 [btrfs]
  [ 2779.991405]  btrfs_work_helper+0xaa/0x720 [btrfs]
  [ 2779.991432]  process_one_work+0x26d/0x6a0
  [ 2779.991460]  worker_thread+0x4f/0x3e0
  [ 2779.991481]  ? process_one_work+0x6a0/0x6a0
  [ 2779.991489]  kthread+0x103/0x140
  [ 2779.991499]  ? kthread_create_worker_on_cpu+0x70/0x70
  [ 2779.991515]  ret_from_fork+0x3a/0x50
  (...)
  [ 2780.026211] INFO: task fsstress:17375 blocked for more than 120 seconds.
  [ 2780.027480]       Not tainted 5.6.0-rc2-btrfs-next-53 #1
  [ 2780.028482] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this
message.
  [ 2780.030035] fsstress        D    0 17375  17373 0x00004000
  [ 2780.030038] Call Trace:
  [ 2780.030044]  ? __schedule+0x384/0xa30
  [ 2780.030052]  schedule+0x33/0xe0
  [ 2780.030075]  lock_extent_bits+0x20c/0x320 [btrfs]
  [ 2780.030094]  ? btrfs_truncate_inode_items+0xf4/0x1150 [btrfs]
  [ 2780.030098]  ? rcu_read_lock_sched_held+0x59/0xa0
```

```
        [ 2780.030102]  ? remove_wait_queue+0x60/0x60
        [ 2780.030122]  btrfs_truncate_inode_items+0x133/0x1150 [btrfs]
        [ 2780.030151]  ? btrfs_set_path_blocking+0xb2/0x160 [btrfs]
        [ 2780.030165]  ? btrfs_search_slot+0x379/0x1000 [btrfs]
        [ 2780.030195]  btrfs_log_changed_extents.isra.8+0x841/0x93e [btrfs]
        [ 2780.030202]  ? do_raw_spin_unlock+0x49/0xc0
        [ 2780.030215]  ? btrfs_get_num_csums+0x10/0x10 [btrfs]
        [ 2780.030239]  btrfs_log_inode+0xf83/0x1124 [btrfs]
        [ 2780.030251]  ? __mutex_unlock_slowpath+0x45/0x2a0
        [ 2780.030275]  btrfs_log_inode_parent+0x2a0/0xe40 [btrfs]
        [ 2780.030282]  ? dget_parent+0xa1/0x370
        [ 2780.030309]  btrfs_log_dentry_safe+0x4a/0x70 [btrfs]
        [ 2780.030329]  btrfs_sync_file+0x3f3/0x490 [btrfs]
        [ 2780.030339]  do_fsync+0x38/0x60
        [ 2780.030343]  __x64_sys_fdatasync+0x13/0x20
        [ 2780.030345]  do_syscall_64+0x5c/0x280
        [ 2780.030348]  entry_SYSCALL_64_after_hwframe+0x49/0xbe
        [ 2780.030356] RIP: 0033:0x7f2d80f6d5f0
        [ 2780.030361] Code: Bad RIP value.
        [ 2780.030362] RSP: 002b:00007ffdba3c8548 EFLAGS: 00000246 ORIG_RAX: 000000000000004b
        [ 2780.030364] RAX: ffffffffffffffda RBX: 0000000000000003 RCX: 00007f2d80f6d5f0
        [ 2780.030365] RDX: 00007ffdba3c84b0 RSI: 00007ffdba3c84b0 RDI: 0000000000000003
        [ 2780.030367] RBP: 000000000000004a R08: 0000000000000001 R09: 00007ffdba3c855c
        [ 2780.030368] R10: 0000000000000078 R11: 0000000000000246 R12: 00000000000001f4
        [ 2780.030369] R13: 0000000051eb851f R14: 00007ffdba3c85f0 R15: 0000557a49220d90

    So fix this by making btrfs_truncate_inode_items() not lock the range in
    the inode's iotree when the target root is a log root, since it's not
    needed to lock the range for log roots as the protection from the inode's
    lock and log_mutex are all that's needed.

    Fixes: 28553fa992cb28 ("Btrfs: fix race between shrinking truncate and fiemap")
    CC: stable@vger.kernel.org # 4.4+
    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 999479d6d65f2ec493d89dd294a9a7061493167f
Author: Filipe Manana <fdmanana@suse.com>
Date:   Thu Feb 13 12:29:50 2020 +0000

    Btrfs: fix btrfs_wait_ordered_range() so that it waits for all ordered extents

    commit e75fd33b3f744f644061a4f9662bd63f5434f806 upstream.

    In btrfs_wait_ordered_range() once we find an ordered extent that has
    finished with an error we exit the loop and don't wait for any other
    ordered extents that might be still in progress.

    All the users of btrfs_wait_ordered_range() expect that there are no more
    ordered extents in progress after that function returns. So past fixes
    such like the ones from the two following commits:

      ff612ba7849964 ("btrfs: fix panic during relocation after ENOSPC before
                       writeback happens")

      28aeeac1dd3080 ("Btrfs: fix panic when starting bg cache writeout after
                       IO error")

    don't work when there are multiple ordered extents in the range.

    Fix that by making btrfs_wait_ordered_range() wait for all ordered extents
    even after it finds one that had an error.
```

```
    Link: https://github.com/kdave/btrfs-progs/issues/228#issuecomment-569777554
    CC: stable@vger.kernel.org # 4.4+
    Reviewed-by: Qu Wenruo <wqu@suse.com>
    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 148e2a1bf1685a9d6694c522f3022164939474e5
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Thu Feb 13 10:47:29 2020 -0500

    btrfs: do not check delayed items are empty for single transaction cleanup

    commit 1e90315149f3fe148e114a5de86f0196d1c21fa5 upstream.

    btrfs_assert_delayed_root_empty() will check if the delayed root is
    completely empty, but this is a filesystem-wide check.  On cleanup we
    may have allowed other transactions to begin, for whatever reason, and
    thus the delayed root is not empty.

    So remove this check from cleanup_one_transation().  This however can
    stay in btrfs_cleanup_transaction(), because it checks only after all of
    the transactions have been properly cleaned up, and thus is valid.

    CC: stable@vger.kernel.org # 4.4+
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Reviewed-by: Nikolay Borisov <nborisov@suse.com>
    Reviewed-by: Qu Wenruo <wqu@suse.com>
    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a2c971c017132e09ea7cb333f31577d07edb0a55
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Thu Feb 13 10:47:28 2020 -0500

    btrfs: reset fs_root to NULL on error in open_ctree

    commit 315bf8ef914f31d51d084af950703aa1e09a728c upstream.

    While running my error injection script I hit a panic when we tried to
    clean up the fs_root when freeing the fs_root.  This is because
    fs_info->fs_root == PTR_ERR(-EIO), which isn't great.  Fix this by
    setting fs_info->fs_root = NULL; if we fail to read the root.

    CC: stable@vger.kernel.org # 4.4+
    Reviewed-by: Nikolay Borisov <nborisov@suse.com>
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Reviewed-by: Qu Wenruo <wqu@suse.com>
    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2fdba5d3b2cbb911d86b68eea7ba45014d88f0e0
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Thu Feb 13 10:47:31 2020 -0500

    btrfs: fix bytes_may_use underflow in prealloc error condtition

    commit b778cf962d71a0e737923d55d0432f3bd287258e upstream.

    I hit the following warning while running my error injection stress
```

```
      testing:

        WARNING: CPU: 3 PID: 1453 at fs/btrfs/space-info.h:108
  btrfs_free_reserved_data_space_noquota+0xfd/0x160 [btrfs]
        RIP: 0010:btrfs_free_reserved_data_space_noquota+0xfd/0x160 [btrfs]
        Call Trace:
        btrfs_free_reserved_data_space+0x4f/0x70 [btrfs]
        __btrfs_prealloc_file_range+0x378/0x470 [btrfs]
        elfcorehdr_read+0x40/0x40
        ? elfcorehdr_read+0x40/0x40
        ? btrfs_commit_transaction+0xca/0xa50 [btrfs]
        ? dput+0xb4/0x2a0
        ? btrfs_log_dentry_safe+0x55/0x70 [btrfs]
        ? btrfs_sync_file+0x30e/0x420 [btrfs]
        ? do_fsync+0x38/0x70
        ? __x64_sys_fdatasync+0x13/0x20
        ? do_syscall_64+0x5b/0x1b0
        ? entry_SYSCALL_64_after_hwframe+0x44/0xa9

      This happens if we fail to insert our reserved file extent.  At this
      point we've already converted our reservation from ->bytes_may_use to
      ->bytes_reserved.  However once we break we will attempt to free
      everything from [cur_offset, end] from ->bytes_may_use, but our extent
      reservation will overlap part of this.

      Fix this problem by adding ins.offset (our extent allocation size) to
      cur_offset so we remove the actual remaining part from ->bytes_may_use.

      I validated this fix using my inject-error.py script

      python inject-error.py -o should_fail_bio -t cache_save_setup -t \
              __btrfs_prealloc_file_range \
              -t insert_reserved_file_extent.constprop.0 \
              -r "-5" ./run-fsstress.sh

      where run-fsstress.sh simply mounts and runs fsstress on a disk.

      CC: stable@vger.kernel.org # 4.4+
      Reviewed-by: Qu Wenruo <wqu@suse.com>
      Signed-off-by: Josef Bacik <josef@toxicpanda.com>
      Reviewed-by: David Sterba <dsterba@suse.com>
      Signed-off-by: David Sterba <dsterba@suse.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 4e2e49d4211db43e0ec932579dab6a969e7e8df1
  Author: Jeff Mahoney <jeffm@suse.com>
  Date:   Tue Feb 11 15:25:37 2020 +0800

      btrfs: destroy qgroup extent records on transaction abort

      commit 81f7eb00ff5bb8326e82503a32809421d14abb8a upstream.

      We clean up the delayed references when we abort a transaction but we
      leave the pending qgroup extent records behind, leaking memory.

      This patch destroys the extent records when we destroy the delayed refs
      and makes sure ensure they're gone before releasing the transaction.

      Fixes: 3368d001ba5d ("btrfs: qgroup: Record possible quota-related extent for qgroup.")
      CC: stable@vger.kernel.org # 4.4+
      Reviewed-by: Josef Bacik <josef@toxicpanda.com>
      Signed-off-by: Jeff Mahoney <jeffm@suse.com>
      [ Rebased to latest upstream, remove to_qgroup() helper, use
        rbtree_postorder_for_each_entry_safe() wrapper ]
      Signed-off-by: Qu Wenruo <wqu@suse.com>
```

```
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f5122114c190615f10b1889918c34f94667c5b88
Author: Jason A. Donenfeld <Jason@zx2c4.com>
Date:   Thu Feb 6 12:42:01 2020 +0100

    crypto: chacha20poly1305 - prevent integer overflow on large input

    commit c9cc0517bba9f0213f1e55172feceb99e5512daf upstream.

    This code assigns src_len (size_t) to sl (int), which causes problems
    when src_len is very large. Probably nobody in the kernel should be
    passing this much data to chacha20poly1305 all in one go anyway, so I
    don't think we need to change the algorithm or introduce larger types
    or anything. But we should at least error out early in this case and
    print a warning so that we get reports if this does happen and can look
    into why anybody is possibly passing it that much data or if they're
    accidently passing -1 or similar.

    Fixes: d95312a3ccc0 ("crypto: lib/chacha20poly1305 - reimplement crypt_from_sg()
routine")
    Cc: Ard Biesheuvel <ardb@kernel.org>
    Cc: stable@vger.kernel.org # 5.5+
    Signed-off-by: Jason A. Donenfeld <Jason@zx2c4.com>
    Acked-by: Ard Biesheuvel <ardb@kernel.org>
    Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7cad6ccee697b8f425ecd2c9222784c7c16561ca
Author: Miaohe Lin <linmiaohe@huawei.com>
Date:   Fri Feb 21 22:04:46 2020 +0800

    KVM: apic: avoid calculating pending eoi from an uninitialized val

    commit 23520b2def95205f132e167cf5b25c609975e959 upstream.

    When pv_eoi_get_user() fails, 'val' may remain uninitialized and the return
    value of pv_eoi_get_pending() becomes random. Fix the issue by initializing
    the variable.

    Reviewed-by: Vitaly Kuznetsov <vkuznets@redhat.com>
    Signed-off-by: Miaohe Lin <linmiaohe@huawei.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 15def96af9e3cbca5b9d7ed9859bcb45cb5bd3f9
Author: Vitaly Kuznetsov <vkuznets@redhat.com>
Date:   Thu Feb 20 18:22:05 2020 +0100

    KVM: nVMX: handle nested posted interrupts when apicv is disabled for L1

    commit 91a5f413af596ad01097e59bf487eb07cb3f1331 upstream.

    Even when APICv is disabled for L1 it can (and, actually, is) still
    available for L2, this means we need to always call
    vmx_deliver_nested_posted_interrupt() when attempting an interrupt
    delivery.

    Suggested-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Vitaly Kuznetsov <vkuznets@redhat.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
```

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

commit 69a01e17f9f39bbc5a63b6666379265e6fe3978e
Author: Vitaly Kuznetsov <vkuznets@redhat.com>
Date:   Thu Feb 20 18:22:04 2020 +0100

    KVM: nVMX: clear PIN_BASED_POSTED_INTR from nested pinbased_ctls only when apicv is
globally disabled

    commit a4443267800af240072280c44521caab61924e55 upstream.

    When apicv is disabled on a vCPU (e.g. by enabling KVM_CAP_HYPERV_SYNIC*),
    nothing happens to VMX MSRs on the already existing vCPUs, however, all new
    ones are created with PIN_BASED_POSTED_INTR filtered out. This is very
    confusing and results in the following picture inside the guest:

    $ rdmsr -ax 0x48d
    ff00000016
    7f00000016
    7f00000016
    7f00000016

    This is observed with QEMU and 4-vCPU guest: QEMU creates vCPU0, does
    KVM_CAP_HYPERV_SYNIC2 and then creates the remaining three.

    L1 hypervisor may only check CPU0's controls to find out what features
    are available and it will be very confused later. Switch to setting
    PIN_BASED_POSTED_INTR control based on global 'enable_apicv' setting.

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

commit 189ae6a1a14c39976ee8a838eea8cbb88afa1927
Author: Oliver Upton <oupton@google.com>
Date:   Tue Feb 4 15:26:31 2020 -0800

    KVM: nVMX: Check IO instruction VM-exit conditions

    commit 35a571346a94fb93b5b3b6a599675ef3384bc75c upstream.

    Consult the 'unconditional IO exiting' and 'use IO bitmaps' VM-execution
    controls when checking instruction interception. If the 'use IO bitmaps'
    VM-execution control is 1, check the instruction access against the IO
    bitmaps to determine if the instruction causes a VM-exit.

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

commit f074932d326c94c94b4d832fb176f5a031705f35
Author: Oliver Upton <oupton@google.com>
Date:   Tue Feb 4 15:26:30 2020 -0800

    KVM: nVMX: Refactor IO bitmap checks into helper function

    commit e71237d3ff1abf9f3388337cfebf53b96df2020d upstream.

    Checks against the IO bitmap are useful for both instruction emulation
    and VM-exit reflection. Refactor the IO bitmap checks into a helper
    function.

    Signed-off-by: Oliver Upton <oupton@google.com>
    Reviewed-by: Vitaly Kuznetsov <vkuznets@redhat.com>
```

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

commit 48c6262bd0a9f4768dd740d329da20a9f742b47f
Author: Chris Wilson <chris@chris-wilson.co.uk>
Date:   Fri Feb 7 21:14:52 2020 +0000

    drm/i915/execlists: Always force a context reload when rewinding RING_TAIL

    commit b1339ecac661e1cf3e1dc78ac56bff3aeeaeb92c upstream.

    If we rewind the RING_TAIL on a context, due to a preemption event, we
    must force the context restore for the RING_TAIL update to be properly
    handled. Rather than note which preemption events may cause us to rewind
    the tail, compare the new request's tail with the previously submitted
    RING_TAIL, as it turns out that timeslicing was causing unexpected
    rewinds.

       <idle>-0       0d.s2 1280851190us : __execlists_submission_tasklet: 0000:00:02.0
rcs0: expired last=130:4698, prio=3, hint=3
       <idle>-0       0d.s2 1280851192us : __i915_request_unsubmit: 0000:00:02.0 rcs0: fence
66:119966, current 119964
       <idle>-0       0d.s2 1280851195us : __i915_request_unsubmit: 0000:00:02.0 rcs0: fence
130:4698, current 4695
       <idle>-0       0d.s2 1280851198us : __i915_request_unsubmit: 0000:00:02.0 rcs0: fence
130:4696, current 4695
    ^----  Note we unwind 2 requests from the same context

       <idle>-0       0d.s2 1280851208us : __i915_request_submit: 0000:00:02.0 rcs0: fence
130:4696, current 4695
       <idle>-0       0d.s2 1280851213us : __i915_request_submit: 0000:00:02.0 rcs0: fence
134:1508, current 1506
    ^---- But to apply the new timeslice, we have to replay the first request
          before the new client can start -- the unexpected RING_TAIL rewind

       <idle>-0       0d.s2 1280851219us : trace_ports: 0000:00:02.0 rcs0: submit {
130:4696*, 134:1508 }
     synmark2-5425    2..s. 1280851239us : process_csb: 0000:00:02.0 rcs0: cs-irq head=5,
tail=0
     synmark2-5425    2..s. 1280851240us : process_csb: 0000:00:02.0 rcs0: csb[0]:
status=0x00008002:0x00000000
    ^---- Preemption event for the ELSP update; note the lite-restore

     synmark2-5425    2..s. 1280851243us : trace_ports: 0000:00:02.0 rcs0: preempted {
130:4698, 66:119966 }
     synmark2-5425    2..s. 1280851246us : trace_ports: 0000:00:02.0 rcs0: promote {
130:4696*, 134:1508 }
     synmark2-5425    2.... 1280851462us : __i915_request_commit: 0000:00:02.0 rcs0: fence
130:4700, current 4695
     synmark2-5425    2.... 1280852111us : __i915_request_commit: 0000:00:02.0 rcs0: fence
130:4702, current 4695
     synmark2-5425    2.Ns1 1280852296us : process_csb: 0000:00:02.0 rcs0: cs-irq head=0,
tail=2
     synmark2-5425    2.Ns1 1280852297us : process_csb: 0000:00:02.0 rcs0: csb[1]:
status=0x00000814:0x00000000
     synmark2-5425    2.Ns1 1280852299us : trace_ports: 0000:00:02.0 rcs0: completed {
130:4696!, 134:1508 }
     synmark2-5425    2.Ns1 1280852301us : process_csb: 0000:00:02.0 rcs0: csb[2]:
status=0x00000818:0x00000040
     synmark2-5425    2.Ns1 1280852302us : trace_ports: 0000:00:02.0 rcs0: completed {
134:1508, 0:0 }
     synmark2-5425    2.Ns1 1280852313us : process_csb: process_csb:2336
GEM_BUG_ON(!i915_request_completed(*execlists->active) && !reset_in_progress(execlists))

    Fixes: 8ee36e048c98 ("drm/i915/execlists: Minimalistic timeslicing")
```

```
    Referenecs: 82c69bf58650 ("drm/i915/gt: Detect if we miss WaIdleLiteRestore")
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: Mika Kuoppala <mika.kuoppala@linux.intel.com>
    Reviewed-by: Mika Kuoppala <mika.kuoppala@linux.intel.com>
    Cc: <stable@vger.kernel.org> # v5.4+
    Link: https://patchwork.freedesktop.org/patch/msgid/20200207211452.2860634-1-
chris@chris-wilson.co.uk
    (cherry picked from commit 5ba32c7be81e53ea8a27190b0f6be98e6c6779af)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 69f8a7991fd93c49096ddd11574db3e7df238b7b
Author: Eric Biggers <ebiggers@google.com>
Date:   Wed Feb 19 10:30:47 2020 -0800

    ext4: fix race between writepages and enabling EXT4_EXTENTS_FL

    commit cb85f4d23f794e24127f3e562cb3b54b0803f456 upstream.

    If EXT4_EXTENTS_FL is set on an inode while ext4_writepages() is running
    on it, the following warning in ext4_add_complete_io() can be hit:

    WARNING: CPU: 1 PID: 0 at fs/ext4/page-io.c:234 ext4_put_io_end_defer+0xf0/0x120

    Here's a minimal reproducer (not 100% reliable) (root isn't required):

            while true; do
                    sync
            done &
            while true; do
                    rm -f file
                    touch file
                    chattr -e file
                    echo X >> file
                    chattr +e file
            done

    The problem is that in ext4_writepages(), ext4_should_dioread_nolock()
    (which only returns true on extent-based files) is checked once to set
    the number of reserved journal credits, and also again later to select
    the flags for ext4_map_blocks() and copy the reserved journal handle to
    ext4_io_end::handle.  But if EXT4_EXTENTS_FL is being concurrently set,
    the first check can see dioread_nolock disabled while the later one can
    see it enabled, causing the reserved handle to unexpectedly be NULL.

    Since changing EXT4_EXTENTS_FL is uncommon, and there may be other races
    related to doing so as well, fix this by synchronizing changing
    EXT4_EXTENTS_FL with ext4_writepages() via the existing
    s_writepages_rwsem (previously called s_journal_flag_rwsem).

    This was originally reported by syzbot without a reproducer at
    https://syzkaller.appspot.com/bug?extid=2202a584a00fffd19fbf,
    but now that dioread_nolock is the default I also started seeing this
    when running syzkaller locally.

    Link: https://lore.kernel.org/r/20200219183047.47417-3-ebiggers@kernel.org
    Reported-by: syzbot+2202a584a00fffd19fbf@syzkaller.appspotmail.com
    Fixes: 6b523df4fb5a ("ext4: use transaction reservation for extent conversion in
ext4_end_io")
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 70d42a675eac5648dbc9bb9dbd6b5d19c1e7a882
Author: Eric Biggers <ebiggers@google.com>
Date:   Wed Feb 19 10:30:46 2020 -0800

    ext4: rename s_journal_flag_rwsem to s_writepages_rwsem

    commit bbd55937de8f2754adc5792b0f8e5ff7d9c0420e upstream.

    In preparation for making s_journal_flag_rwsem synchronize
    ext4_writepages() with changes to both the EXTENTS and JOURNAL_DATA
    flags (rather than just JOURNAL_DATA as it does currently), rename it to
    s_writepages_rwsem.

    Link: https://lore.kernel.org/r/20200219183047.47417-2-ebiggers@kernel.org
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d896c49be24566903a9e502b79f750e35638989d
Author: Jan Kara <jack@suse.cz>
Date:   Fri Feb 21 11:08:35 2020 +0100

    ext4: fix mount failure with quota configured as module

    commit 9db176bceb5c5df4990486709da386edadc6bd1d upstream.

    When CONFIG_QFMT_V2 is configured as a module, the test in
    ext4_feature_set_ok() fails and so mount of filesystems with quota or
    project features fails. Fix the test to use IS_ENABLED macro which
    works properly even for modules.

    Link: https://lore.kernel.org/r/20200221100835.9332-1-jack@suse.cz
    Fixes: d65d87a07476 ("ext4: improve explanation of a mount failure caused by a
misconfigured kernel")
    Signed-off-by: Jan Kara <jack@suse.cz>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 056d1b18a58f7a65298e6501b4e97810de865648
Author: Suraj Jitindar Singh <surajjs@amazon.com>
Date:   Tue Feb 18 19:08:51 2020 -0800

    ext4: fix potential race between s_flex_groups online resizing and access

    commit 7c990728b99ed6fbe9c75fc202fce1172d9916da upstream.

    During an online resize an array of s_flex_groups structures gets replaced
    so it can get enlarged. If there is a concurrent access to the array and
    this memory has been reused then this can lead to an invalid memory access.

    The s_flex_group array has been converted into an array of pointers rather
    than an array of structures. This is to ensure that the information
    contained in the structures cannot get out of sync during a resize due to
    an accessor updating the value in the old structure after it has been
    copied but before the array pointer is updated. Since the structures them-
    selves are no longer copied but only the pointers to them this case is
    mitigated.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206443
    Link: https://lore.kernel.org/r/20200221053458.730016-4-tytso@mit.edu
    Signed-off-by: Suraj Jitindar Singh <surajjs@amazon.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
```

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

commit 79e23a5c4eb38ef85b9a7e8299abec7cb5f07cb6
Author: Suraj Jitindar Singh <surajjs@amazon.com>
Date:   Tue Feb 18 19:08:50 2020 -0800

    ext4: fix potential race between s_group_info online resizing and access

    commit df3da4ea5a0fc5d115c90d5aa6caa4dd433750a7 upstream.

    During an online resize an array of pointers to s_group_info gets replaced
    so it can get enlarged. If there is a concurrent access to the array in
    ext4_get_group_info() and this memory has been reused then this can lead to
    an invalid memory access.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206443
    Link: https://lore.kernel.org/r/20200221053458.730016-3-tytso@mit.edu
    Signed-off-by: Suraj Jitindar Singh <surajjs@amazon.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Reviewed-by: Balbir Singh <sblbir@amazon.com>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9866b5910b83bd5dbd50a81ae5ee0a433c4e8071
Author: Theodore Ts'o <tytso@mit.edu>
Date:   Sat Feb 15 16:40:37 2020 -0500

    ext4: fix potential race between online resizing and write operations

    commit 1d0c3924a92e69bfa91163bda83c12a994b4d106 upstream.

    During an online resize an array of pointers to buffer heads gets
    replaced so it can get enlarged.  If there is a racing block
    allocation or deallocation which uses the old array, and the old array
    has gotten reused this can lead to a GPF or some other random kernel
    memory getting modified.

    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206443
    Link: https://lore.kernel.org/r/20200221053458.730016-2-tytso@mit.edu
    Reported-by: Suraj Jitindar Singh <surajjs@amazon.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ea147965416c3726d63b346cb60264ef9f6e9d4f
Author: Shijie Luo <luoshijie1@huawei.com>
Date:   Sat Feb 15 03:02:06 2020 -0500

    ext4: add cond_resched() to __ext4_find_entry()

    commit 9424ef56e13a1f14c57ea161eed3ecfdc7b2770e upstream.

    We tested a soft lockup problem in linux 4.19 which could also
    be found in linux 5.x.

    When dir inode takes up a large number of blocks, and if the
    directory is growing when we are searching, it's possible the
    restart branch could be called many times, and the do while loop
    could hold cpu a long time.

    Here is the call trace in linux 4.19.

    [  473.756186] Call trace:
    [  473.756196]  dump_backtrace+0x0/0x198
```

```
      [  473.756199]  show_stack+0x24/0x30
      [  473.756205]  dump_stack+0xa4/0xcc
      [  473.756210]  watchdog_timer_fn+0x300/0x3e8
      [  473.756215]  __hrtimer_run_queues+0x114/0x358
      [  473.756217]  hrtimer_interrupt+0x104/0x2d8
      [  473.756222]  arch_timer_handler_virt+0x38/0x58
      [  473.756226]  handle_percpu_devid_irq+0x90/0x248
      [  473.756231]  generic_handle_irq+0x34/0x50
      [  473.756234]  __handle_domain_irq+0x68/0xc0
      [  473.756236]  gic_handle_irq+0x6c/0x150
      [  473.756238]  el1_irq+0xb8/0x140
      [  473.756286]  ext4_es_lookup_extent+0xdc/0x258 [ext4]
      [  473.756310]  ext4_map_blocks+0x64/0x5c0 [ext4]
      [  473.756333]  ext4_getblk+0x6c/0x1d0 [ext4]
      [  473.756356]  ext4_bread_batch+0x7c/0x1f8 [ext4]
      [  473.756379]  ext4_find_entry+0x124/0x3f8 [ext4]
      [  473.756402]  ext4_lookup+0x8c/0x258 [ext4]
      [  473.756407]  __lookup_hash+0x8c/0xe8
      [  473.756411]  filename_create+0xa0/0x170
      [  473.756413]  do_mkdirat+0x6c/0x140
      [  473.756415]  __arm64_sys_mkdirat+0x28/0x38
      [  473.756419]  el0_svc_common+0x78/0x130
      [  473.756421]  el0_svc_handler+0x38/0x78
      [  473.756423]  el0_svc+0x8/0xc
      [  485.755156] watchdog: BUG: soft lockup - CPU#2 stuck for 22s! [tmp:5149]

    Add cond_resched() to avoid soft lockup and to provide a better
    system responding.

    Link: https://lore.kernel.org/r/20200215080206.13293-1-luoshijie1@huawei.com
    Signed-off-by: Shijie Luo <luoshijie1@huawei.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e6ddbd4311949ee7e7b8d12681b194abdb6b0b81
Author: Qian Cai <cai@lca.pw>
Date:   Fri Feb 7 09:29:11 2020 -0500

    ext4: fix a data race in EXT4_I(inode)->i_disksize

    commit 35df4299a6487f323b0aca120ea3f485dfee2ae3 upstream.

    EXT4_I(inode)->i_disksize could be accessed concurrently as noticed by
    KCSAN,

     BUG: KCSAN: data-race in ext4_write_end [ext4] / ext4_writepages [ext4]

     write to 0xffff91c6713b00f8 of 8 bytes by task 49268 on cpu 127:
      ext4_write_end+0x4e3/0x750 [ext4]
      ext4_update_i_disksize at fs/ext4/ext4.h:3032
      (inlined by) ext4_update_inode_size at fs/ext4/ext4.h:3046
      (inlined by) ext4_write_end at fs/ext4/inode.c:1287
      generic_perform_write+0x208/0x2a0
      ext4_buffered_write_iter+0x11f/0x210 [ext4]
      ext4_file_write_iter+0xce/0x9e0 [ext4]
      new_sync_write+0x29c/0x3b0
      __vfs_write+0x92/0xa0
      vfs_write+0x103/0x260
      ksys_write+0x9d/0x130
      __x64_sys_write+0x4c/0x60
      do_syscall_64+0x91/0xb47
      entry_SYSCALL_64_after_hwframe+0x49/0xbe
```

```
      read to 0xffff91c6713b00f8 of 8 bytes by task 24872 on cpu 37:
       ext4_writepages+0x10ac/0x1d00 [ext4]
       mpage_map_and_submit_extent at fs/ext4/inode.c:2468
       (inlined by) ext4_writepages at fs/ext4/inode.c:2772
       do_writepages+0x5e/0x130
       __writeback_single_inode+0xeb/0xb20
       writeback_sb_inodes+0x429/0x900
       __writeback_inodes_wb+0xc4/0x150
       wb_writeback+0x4bd/0x870
       wb_workfn+0x6b4/0x960
       process_one_work+0x54c/0xbe0
       worker_thread+0x80/0x650
       kthread+0x1e0/0x200
       ret_from_fork+0x27/0x50

      Reported by Kernel Concurrency Sanitizer on:
      CPU: 37 PID: 24872 Comm: kworker/u261:2 Tainted: G        W  O L 5.5.0-next-20200204+
#5
      Hardware name: HPE ProLiant DL385 Gen10/ProLiant DL385 Gen10, BIOS A40 07/10/2019
      Workqueue: writeback wb_workfn (flush-7:0)

    Since only the read is operating as lockless (outside of the
    "i_data_sem"), load tearing could introduce a logic bug. Fix it by
    adding READ_ONCE() for the read and WRITE_ONCE() for the write.

    Signed-off-by: Qian Cai <cai@lca.pw>
    Link: https://lore.kernel.org/r/1581085751-31793-1-git-send-email-cai@lca.pw
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 843a9ae8073377633905d4dd055ad18c2430375d
Author: Miaohe Lin <linmiaohe@huawei.com>
Date:   Fri Feb 14 10:32:38 2020 +0800

    KVM: x86: don't notify userspace IOAPIC on edge-triggered interrupt EOI

    commit 7455a8327674e1a7c9a1f5dd1b0743ab6713f6d1 upstream.

    Commit 13db77347db1 ("KVM: x86: don't notify userspace IOAPIC on edge
    EOI") said, edge-triggered interrupts don't set a bit in TMR, which means
    that IOAPIC isn't notified on EOI. And var level indicates level-triggered
    interrupt.
    But commit 3159d36ad799 ("KVM: x86: use generic function for MSI parsing")
    replace var level with irq.level by mistake. Fix it by changing irq.level
    to irq.trig_mode.

    Cc: stable@vger.kernel.org
    Fixes: 3159d36ad799 ("KVM: x86: use generic function for MSI parsing")
    Signed-off-by: Miaohe Lin <linmiaohe@huawei.com>
    Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bfdac9d7632d29fab54cb4488d208a89c9ac48c7
Author: Paolo Bonzini <pbonzini@redhat.com>
Date:   Tue Feb 4 15:26:29 2020 -0800

    KVM: nVMX: Don't emulate instructions in guest mode

    commit 07721feee46b4b248402133228235318199b05ec upstream.

    vmx_check_intercept is not yet fully implemented. To avoid emulating
    instructions disallowed by the L1 hypervisor, refuse to emulate
    instructions by default.
```

```
      Cc: stable@vger.kernel.org
      [Made commit, added commit msg - Oliver]
      Signed-off-by: Oliver Upton <oupton@google.com>
      Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 9ce51fc82bbc8632cb55643ce309bd6732b3d5db
  Author: Suren Baghdasaryan <surenb@google.com>
  Date:   Mon Feb 3 13:22:16 2020 -0800

      sched/psi: Fix OOB write when writing 0 bytes to PSI files

      commit 6fcca0fa48118e6d63733eb4644c6cd880c15b8f upstream.

      Issuing write() with count parameter set to 0 on any file under
      /proc/pressure/ will cause an OOB write because of the access to
      buf[buf_size-1] when NUL-termination is performed. Fix this by checking
      for buf_size to be non-zero.

      Signed-off-by: Suren Baghdasaryan <surenb@google.com>
      Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
      Signed-off-by: Ingo Molnar <mingo@kernel.org>
      Acked-by: Johannes Weiner <hannes@cmpxchg.org>
      Link: https://lkml.kernel.org/r/20200203212216.7076-1-surenb@google.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 659b375f9b41c22e676fb4343e74267916c4d6b6
  Author: Jani Nikula <jani.nikula@intel.com>
  Date:   Wed Feb 12 18:04:34 2020 +0200

      drm/i915: Update drm/i915 bug filing URL

      commit 7ddc7005a0aa2f43a826b71f5d6bd7d4b90f8f2a upstream.

      We've moved from bugzilla to gitlab.

      Cc: stable@vger.kernel.org
      Reviewed-by: Chris Wilson <chris@chris-wilson.co.uk>
      Signed-off-by: Jani Nikula <jani.nikula@intel.com>
      Link: https://patchwork.freedesktop.org/patch/msgid/20200212160434.6437-2-
  jani.nikula@intel.com
      (cherry picked from commit ddae4d7af0bbe3b2051f1603459a8b24e9a19324)
      Signed-off-by: Jani Nikula <jani.nikula@intel.com>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit e348f6c0fbf321579b24c372122e6c91bcc56dc5
  Author: Chris Wilson <chris@chris-wilson.co.uk>
  Date:   Sun Feb 2 15:39:34 2020 +0000

      drm/i915: Wean off drm_pci_alloc/drm_pci_free

      commit aa3146193ae25d0fe4b96d815169a135db2e8f01 upstream.

      drm_pci_alloc and drm_pci_free are just very thin wrappers around
      dma_alloc_coherent, with a note that we should be removing them.
      Furthermore since

      commit de09d31dd38a50fdce106c15abd68432eebbd014
      Author: Kirill A. Shutemov <kirill.shutemov@linux.intel.com>
      Date:   Fri Jan 15 16:51:42 2016 -0800

          page-flags: define PG_reserved behavior on compound pages

          As far as I can see there's no users of PG_reserved on compound pages.
          Let's use PF_NO_COMPOUND here.
```

```
    drm_pci_alloc has been declared broken since it mixes GFP_COMP and
    SetPageReserved. Avoid this conflict by weaning ourselves off using the
    abstraction and using the dma functions directly.

    Reported-by: Taketo Kabe
    Closes: https://gitlab.freedesktop.org/drm/intel/issues/1027
    Fixes: de09d31dd38a ("page-flags: define PG_reserved behavior on compound pages")
    Signed-off-by: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: <stable@vger.kernel.org> # v4.5+
    Reviewed-by: Daniel Vetter <daniel.vetter@ffwll.ch>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200202153934.3899472-1-
chris@chris-wilson.co.uk
    (cherry picked from commit c6790dc22312f592c1434577258b31c48c72d52a)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d55e33d12a6983fe99f3f8569436e149346f8e9e
Author: Lyude Paul <lyude@redhat.com>
Date:   Wed Feb 12 18:11:49 2020 -0500

    drm/nouveau/kms/gv100-: Re-set LUT after clearing for modesets

    commit f287d3d19769b1d22cba4e51fa0487f2697713c9 upstream.

    While certain modeset operations on gv100+ need us to temporarily
    disable the LUT, we make the mistake of sometimes neglecting to
    reprogram the LUT after such modesets. In particular, moving a head from
    one encoder to another seems to trigger this quite often. GV100+ is very
    picky about having a LUT in most scenarios, so this causes the display
    engine to hang with the following error code:

    disp: chid 1 stat 00005080 reason 5 [INVALID_STATE] mthd 0200 data
    00000001 code 0000002d)

    So, fix this by always re-programming the LUT if we're clearing it in a
    state where the wndw is still visible, and has a XLUT handle programmed.

    Signed-off-by: Lyude Paul <lyude@redhat.com>
    Fixes: facaed62b4cb ("drm/nouveau/kms/gv100: initial support")
    Cc: <stable@vger.kernel.org> # v4.18+
    Signed-off-by: Ben Skeggs <bskeggs@redhat.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b534e6803d2df194a172299bff000a758e6ec8ab
Author: Alex Deucher <alexander.deucher@amd.com>
Date:   Wed Feb 12 08:52:32 2020 -0500

    drm/amdgpu/gfx10: disable gfxoff when reading rlc clock

    commit b08c3ed609aabc4e76e74edc4404f0c26279d7ed upstream.

    Otherwise we readback all ones.  Fixes rlc counter
    readback while gfxoff is active.

    Reviewed-by: Xiaojie Yuan <xiaojie.yuan@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 00908efff00720a331af29500937683d4c4d58de
Author: Alex Deucher <alexander.deucher@amd.com>
Date:   Wed Feb 12 08:51:29 2020 -0500

    drm/amdgpu/gfx9: disable gfxoff when reading rlc clock
```

```
    commit 120cf959308e1bda984e40a9edd25ee2d6262efd upstream.

    Otherwise we readback all ones.  Fixes rlc counter
    readback while gfxoff is active.

    Reviewed-by: Xiaojie Yuan <xiaojie.yuan@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2b38454b3e871737b77db428b210a45ec18db41d
Author: Alex Deucher <alexander.deucher@amd.com>
Date:   Wed Feb 12 01:46:16 2020 -0500

    drm/amdgpu/soc15: fix xclk for raven

    commit c657b936ea98630ef5ba4f130ab1ad5c534d0165 upstream.

    It's 25 Mhz (refclk / 4).  This fixes the interpretation
    of the rlc clock counter.

    Acked-by: Evan Quan <evan.quan@amd.com>
    Signed-off-by: Alex Deucher <alexander.deucher@amd.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c1947a09073350073f73e7024bda4cfdc240dc8f
Author: Catalin Marinas <catalin.marinas@arm.com>
Date:   Wed Feb 19 12:31:56 2020 +0000

    mm: Avoid creating virtual address aliases in brk()/mmap()/mremap()

    commit dcde237319e626d1ec3c9d8b7613032f0fd4663a upstream.

    Currently the arm64 kernel ignores the top address byte passed to brk(),
    mmap() and mremap(). When the user is not aware of the 56-bit address
    limit or relies on the kernel to return an error, untagging such
    pointers has the potential to create address aliases in user-space.
    Passing a tagged address to munmap(), madvise() is permitted since the
    tagged pointer is expected to be inside an existing mapping.

    The current behaviour breaks the existing glibc malloc() implementation
    which relies on brk() with an address beyond 56-bit to be rejected by
    the kernel.

    Remove untagging in the above functions by partially reverting commit
    ce18d171cb73 ("mm: untag user pointers in mmap/munmap/mremap/brk"). In
    addition, update the arm64 tagged-address-abi.rst document accordingly.

    Link: https://bugzilla.redhat.com/1797052
    Fixes: ce18d171cb73 ("mm: untag user pointers in mmap/munmap/mremap/brk")
    Cc: <stable@vger.kernel.org> # 5.4.x-
    Cc: Florian Weimer <fweimer@redhat.com>
    Reviewed-by: Andrew Morton <akpm@linux-foundation.org>
    Reported-by: Victor Stinner <vstinner@redhat.com>
    Acked-by: Will Deacon <will@kernel.org>
    Acked-by: Andrey Konovalov <andreyknvl@google.com>
    Signed-off-by: Catalin Marinas <catalin.marinas@arm.com>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f46690ff97981d7dac1a082381d3aa78b134cd80
Author: Alexander Potapenko <glider@google.com>
Date:   Thu Feb 20 20:04:30 2020 -0800
```

```
    lib/stackdepot.c: fix global out-of-bounds in stack_slabs

    commit 305e519ce48e935702c32241f07d393c3c8fed3e upstream.

    Walter Wu has reported a potential case in which init_stack_slab() is
    called after stack_slabs[STACK_ALLOC_MAX_SLABS - 1] has already been
    initialized.  In that case init_stack_slab() will overwrite
    stack_slabs[STACK_ALLOC_MAX_SLABS], which may result in a memory
    corruption.

    Link: http://lkml.kernel.org/r/20200218102950.260263-1-glider@google.com
    Fixes: cd11016e5f521 ("mm, kasan: stackdepot implementation. Enable stackdepot for
SLAB")
    Signed-off-by: Alexander Potapenko <glider@google.com>
    Reported-by: Walter Wu <walter-zh.wu@mediatek.com>
    Cc: Dmitry Vyukov <dvyukov@google.com>
    Cc: Matthias Brugger <matthias.bgg@gmail.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Josh Poimboeuf <jpoimboe@redhat.com>
    Cc: Kate Stewart <kstewart@linuxfoundation.org>
    Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ad3c79df5e76b30dd8742643219eda2e77d9c937
Author: Wei Yang <richardw.yang@linux.intel.com>
Date:   Thu Feb 20 20:04:27 2020 -0800

    mm/sparsemem: pfn_to_page is not valid yet on SPARSEMEM

    commit 18e19f195cd888f65643a77a0c6aee8f5be6439a upstream.

    When we use SPARSEMEM instead of SPARSEMEM_VMEMMAP, pfn_to_page()
    doesn't work before sparse_init_one_section() is called.

    This leads to a crash when hotplug memory:

        BUG: unable to handle page fault for address: 0000000006400000
        #PF: supervisor write access in kernel mode
        #PF: error_code(0x0002) - not-present page
        PGD 0 P4D 0
        Oops: 0002 [#1] SMP PTI
        CPU: 3 PID: 221 Comm: kworker/u16:1 Tainted: G        W         5.5.0-next-20200205+
#343
        Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 0.0.0 02/06/2015
        Workqueue: kacpi_hotplug acpi_hotplug_work_fn
        RIP: 0010:__memset+0x24/0x30
        Code: cc cc cc cc cc cc 0f 1f 44 00 00 49 89 f9 48 89 d1 83 e2 07 48 c1 e9 03 40 0f
b6 f6 48 b8 01 01 01 01 01 01 01 01 48 0f af c6 <f3> 48 ab 89 d1 f3 aa 4c 89 c8 c3 90 49 89
f9 40 88 f0 48 89 d1 f3
        RSP: 0018:ffffb43ac0373c80 EFLAGS: 00010a87
        RAX: ffffffffffffffff RBX: ffff8a1518800000 RCX: 0000000000050000
        RDX: 0000000000000000 RSI: 00000000000000ff RDI: 0000000006400000
        RBP: 0000000000140000 R08: 0000000000100000 R09: 0000000006400000
        R10: 0000000000000000 R11: 0000000000000002 R12: 0000000000000000
        R13: 0000000000000028 R14: 0000000000000000 R15: ffff8a153ffd9280
        FS:  0000000000000000(0000) GS:ffff8a153ab00000(0000) knlGS:0000000000000000
        CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
        CR2: 0000000006400000 CR3: 0000000136fca000 CR4: 00000000000006e0
        DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
        DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
        Call Trace:
```

```
          sparse_add_section+0x1c9/0x26a
          __add_pages+0xbf/0x150
          add_pages+0x12/0x60
          add_memory_resource+0xc8/0x210
          __add_memory+0x62/0xb0
          acpi_memory_device_add+0x13f/0x300
          acpi_bus_attach+0xf6/0x200
          acpi_bus_scan+0x43/0x90
          acpi_device_hotplug+0x275/0x3d0
          acpi_hotplug_work_fn+0x1a/0x30
          process_one_work+0x1a7/0x370
          worker_thread+0x30/0x380
          kthread+0x112/0x130
          ret_from_fork+0x35/0x40

    We should use memmap as it did.

    On x86 the impact is limited to x86_32 builds, or x86_64 configurations
    that override the default setting for SPARSEMEM_VMEMMAP.

    Other memory hotplug archs (arm64, ia64, and ppc) also default to
    SPARSEMEM_VMEMMAP=y.

    [dan.j.williams@intel.com: changelog update]
    {rppt@linux.ibm.com: changelog update]
    Link: http://lkml.kernel.org/r/20200219030454.4844-1-bhe@redhat.com
    Fixes: ba72b4c8cf60 ("mm/sparsemem: support sub-section hotplug")
    Signed-off-by: Wei Yang <richardw.yang@linux.intel.com>
    Signed-off-by: Baoquan He <bhe@redhat.com>
    Acked-by: David Hildenbrand <david@redhat.com>
    Reviewed-by: Baoquan He <bhe@redhat.com>
    Reviewed-by: Dan Williams <dan.j.williams@intel.com>
    Acked-by: Michal Hocko <mhocko@suse.com>
    Cc: Mike Rapoport <rppt@linux.ibm.com>
    Cc: Oscar Salvador <osalvador@suse.de>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 79f131a8fc24300028db514378fc5d45fd100af3
Author: Gavin Shan <gshan@redhat.com>
Date:   Thu Feb 20 20:04:24 2020 -0800

    mm/vmscan.c: don't round up scan size for online memory cgroup

    commit 76073c646f5f4999d763f471df9e38a5a912d70d upstream.

    Commit 68600f623d69 ("mm: don't miss the last page because of round-off
    error") makes the scan size round up to @denominator regardless of the
    memory cgroup's state, online or offline.  This affects the overall
    reclaiming behavior: the corresponding LRU list is eligible for
    reclaiming only when its size logically right shifted by @sc->priority
    is bigger than zero in the former formula.

    For example, the inactive anonymous LRU list should have at least 0x4000
    pages to be eligible for reclaiming when we have 60/12 for
    swappiness/priority and without taking scan/rotation ratio into account.

    After the roundup is applied, the inactive anonymous LRU list becomes
    eligible for reclaiming when its size is bigger than or equal to 0x1000
    in the same condition.

        (0x4000 >> 12) * 60 / (60 + 140 + 1) = 1
        ((0x1000 >> 12) * 60) + 200) / (60 + 140 + 1) = 1
```

```
    aarch64 has 512MB huge page size when the base page size is 64KB.  The
    memory cgroup that has a huge page is always eligible for reclaiming in
    that case.

    The reclaiming is likely to stop after the huge page is reclaimed,
    meaing the further iteration on @sc->priority and the silbing and child
    memory cgroups will be skipped.  The overall behaviour has been changed.
    This fixes the issue by applying the roundup to offlined memory cgroups
    only, to give more preference to reclaim memory from offlined memory
    cgroup.  It sounds reasonable as those memory is unlikedly to be used by
    anyone.

    The issue was found by starting up 8 VMs on a Ampere Mustang machine,
    which has 8 CPUs and 16 GB memory.  Each VM is given with 2 vCPUs and
    2GB memory.  It took 264 seconds for all VMs to be completely up and
    784MB swap is consumed after that.  With this patch applied, it took 236
    seconds and 60MB swap to do same thing.  So there is 10% performance
    improvement for my case.  Note that KSM is disable while THP is enabled
    in the testing.

                  total     used    free   shared  buff/cache   available
       Mem:  16196    10065    2049       16        4081        3749
       Swap:  8175      784    7391
                  total     used    free   shared  buff/cache   available
       Mem:  16196    11324    3656       24        1215        2936
       Swap:  8175       60    8115

    Link: http://lkml.kernel.org/r/20200211024514.8730-1-gshan@redhat.com
    Fixes: 68600f623d69 ("mm: don't miss the last page because of round-off error")
    Signed-off-by: Gavin Shan <gshan@redhat.com>
    Acked-by: Roman Gushchin <guro@fb.com>
    Cc: <stable@vger.kernel.org>    [4.20+]
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5b6a0b4385e55c858172ac23592a1b75cc87fd89
Author: Zenghui Yu <yuzenghui@huawei.com>
Date:   Fri Feb 21 10:07:25 2020 +0800

    genirq/irqdomain: Make sure all irq domain flags are distinct

    commit 2546287c5fb363a0165933ae2181c92f03e701d0 upstream.

    This was noticed when printing debugfs for MSIs on my ARM64 server.  The
    new dstate IRQD_MSI_NOMASK_QUIRK came out surprisingly while it should only
    be the x86 stuff for the time being...

    The new MSI quirk flag uses the same bit as IRQ_DOMAIN_NAME_ALLOCATED which
    is oddly defined as bit 6 for no good reason.

    Switch it to the non used bit 1.

    Fixes: 6f1a4891a592 ("x86/apic/msi: Plug non-maskable MSI affinity race")
    Signed-off-by: Zenghui Yu <yuzenghui@huawei.com>
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Cc: stable@vger.kernel.org
    Link: https://lkml.kernel.org/r/20200221020725.2038-1-yuzenghui@huawei.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ea7f30c453729141aa6169104cdeeb2f5b29d07c
Author: Logan Gunthorpe <logang@deltatee.com>
Date:   Thu Feb 20 13:29:53 2020 -0700
```

```
    nvme-multipath: Fix memory leak with ana_log_buf

    commit 3b7830904e17202524bad1974505a9bfc718d31f upstream.

    kmemleak reports a memory leak with the ana_log_buf allocated by
    nvme_mpath_init():

    unreferenced object 0xffff888120e94000 (size 8208):
      comm "nvme", pid 6884, jiffies 4295020435 (age 78786.312s)
        hex dump (first 32 bytes):
          00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
          01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
        backtrace:
          [<00000000e2360188>] kmalloc_order+0x97/0xc0
          [<0000000079b18dd4>] kmalloc_order_trace+0x24/0x100
          [<00000000f50c0406>] __kmalloc+0x24c/0x2d0
          [<00000000f31a10b9>] nvme_mpath_init+0x23c/0x2b0
          [<000000005802589e>] nvme_init_identify+0x75f/0x1600
          [<0000000058ef911b>] nvme_loop_configure_admin_queue+0x26d/0x280
          [<0000000673774b9>] nvme_loop_create_ctrl+0x2a7/0x710
          [<00000000f1c7a233>] nvmf_dev_write+0xc66/0x10b9
          [<000000004199f8d0>] __vfs_write+0x50/0xa0
          [<0000000065466fef>] vfs_write+0xf3/0x280
          [<00000000b0db9a8b>] ksys_write+0xc6/0x160
          [<0000000082156b91>] __x64_sys_write+0x43/0x50
          [<00000000c34fbb6d>] do_syscall_64+0x77/0x2f0
          [<00000000bbc574c9>] entry_SYSCALL_64_after_hwframe+0x49/0xbe

    nvme_mpath_init() is called by nvme_init_identify() which is called in
    multiple places (nvme_reset_work(), nvme_passthru_end(), etc). This
    means nvme_mpath_init() may be called multiple times before
    nvme_mpath_uninit() (which is only called on nvme_free_ctrl()).

    When nvme_mpath_init() is called multiple times, it overwrites the
    ana_log_buf pointer with a new allocation, thus leaking the previous
    allocation.

    To fix this, free ana_log_buf before allocating a new one.

    Fixes: 0d0b660f214dc490 ("nvme: add ANA support")
    Cc: <stable@vger.kernel.org>
    Reviewed-by: Sagi Grimberg <sagi@grimberg.me>
    Reviewed-by: Christoph Hellwig <hch@lst.de>
    Signed-off-by: Logan Gunthorpe <logang@deltatee.com>
    Signed-off-by: Keith Busch <kbusch@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a12c65562db42e70c386a70a1b70cec6c9ca34af
Author: Vasily Averin <vvs@virtuozzo.com>
Date:   Thu Feb 20 20:04:18 2020 -0800

    mm/memcontrol.c: lost css_put in memcg_expand_shrinker_maps()

    commit 75866af62b439859d5146b7093ceb6b482852683 upstream.

    for_each_mem_cgroup() increases css reference counter for memory cgroup
    and requires to use mem_cgroup_iter_break() if the walk is cancelled.

    Link: http://lkml.kernel.org/r/c98414fb-7e1f-da0f-867a-9340ec4bd30b@virtuozzo.com
    Fixes: 0a4465d34028 ("mm, memcg: assign memcg-aware shrinkers bitmap to memcg")
    Signed-off-by: Vasily Averin <vvs@virtuozzo.com>
    Acked-by: Kirill Tkhai <ktkhai@virtuozzo.com>
    Acked-by: Michal Hocko <mhocko@suse.com>
    Reviewed-by: Roman Gushchin <guro@fb.com>
    Cc: Johannes Weiner <hannes@cmpxchg.org>
```

```
    Cc: Vladimir Davydov <vdavydov.dev@gmail.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 790200cc15df4733dd50c0dd3774367c74364e7f
Author: Ioanna Alifieraki <ioanna-maria.alifieraki@canonical.com>
Date:   Thu Feb 20 20:04:00 2020 -0800

    Revert "ipc,sem: remove uneeded sem_undo_list lock usage in exit_sem()"

    commit edf28f4061afe4c2d9eb1c3323d90e882c1d6800 upstream.

    This reverts commit a97955844807e327df11aa33869009d14d6b7de0.

    Commit a97955844807 ("ipc,sem: remove uneeded sem_undo_list lock usage
    in exit_sem()") removes a lock that is needed.  This leads to a process
    looping infinitely in exit_sem() and can also lead to a crash.  There is
    a reproducer available in [1] and with the commit reverted the issue
    does not reproduce anymore.

    Using the reproducer found in [1] is fairly easy to reach a point where
    one of the child processes is looping infinitely in exit_sem between
    for(;;) and if (semid == -1) block, while it's trying to free its last
    sem_undo structure which has already been freed by freeary().

    Each sem_undo struct is on two lists: one per semaphore set (list_id)
    and one per process (list_proc).  The list_id list tracks undos by
    semaphore set, and the list_proc by process.

    Undo structures are removed either by freeary() or by exit_sem().  The
    freeary function is invoked when the user invokes a syscall to remove a
    semaphore set.  During this operation freeary() traverses the list_id
    associated with the semaphore set and removes the undo structures from
    both the list_id and list_proc lists.

    For this case, exit_sem() is called at process exit.  Each process
    contains a struct sem_undo_list (referred to as "ulp") which contains
    the head for the list_proc list.  When the process exits, exit_sem()
    traverses this list to remove each sem_undo struct.  As in freeary(),
    whenever a sem_undo struct is removed from list_proc, it is also removed
    from the list_id list.

    Removing elements from list_id is safe for both exit_sem() and freeary()
    due to sem_lock().  Removing elements from list_proc is not safe;
    freeary() locks &un->ulp->lock when it performs
    list_del_rcu(&un->list_proc) but exit_sem() does not (locking was
    removed by commit a97955844807 ("ipc,sem: remove uneeded sem_undo_list
    lock usage in exit_sem()").

    This can result in the following situation while executing the
    reproducer [1] : Consider a child process in exit_sem() and the parent
    in freeary() (because of semctl(sid[i], NSEM, IPC_RMID)).

     - The list_proc for the child contains the last two undo structs A and
       B (the rest have been removed either by exit_sem() or freeary()).

     - The semid for A is 1 and semid for B is 2.

     - exit_sem() removes A and at the same time freeary() removes B.

     - Since A and B have different semid sem_lock() will acquire different
       locks for each process and both can proceed.
```

```
    The bug is that they remove A and B from the same list_proc at the same
    time because only freeary() acquires the ulp lock. When exit_sem()
    removes A it makes ulp->list_proc.next to point at B and at the same
    time freeary() removes B setting B->semid=-1.

    At the next iteration of for(;;) loop exit_sem() will try to remove B.

    The only way to break from for(;;) is for (&un->list_proc ==
    &ulp->list_proc) to be true which is not. Then exit_sem() will check if
    B->semid=-1 which is and will continue looping in for(;;) until the
    memory for B is reallocated and the value at B->semid is changed.

    At that point, exit_sem() will crash attempting to unlink B from the
    lists (this can be easily triggered by running the reproducer [1] a
    second time).

    To prove this scenario instrumentation was added to keep information
    about each sem_undo (un) struct that is removed per process and per
    semaphore set (sma).

              CPU0                                  CPU1
      [caller holds sem_lock(sma for A)]      ...
      freeary()                               exit_sem()
      ...                                     ...
      ...                                     sem_lock(sma for B)
      spin_lock(A->ulp->lock)                 ...
      list_del_rcu(un_A->list_proc)           list_del_rcu(un_B->list_proc)

    Undo structures A and B have different semid and sem_lock() operations
    proceed.  However they belong to the same list_proc list and they are
    removed at the same time.  This results into ulp->list_proc.next
    pointing to the address of B which is already removed.

    After reverting commit a97955844807 ("ipc,sem: remove uneeded
    sem_undo_list lock usage in exit_sem()") the issue was no longer
    reproducible.

    [1] https://bugzilla.redhat.com/show_bug.cgi?id=1694779

    Link: http://lkml.kernel.org/r/20191211191318.11860-1-ioanna-
maria.alifieraki@canonical.com
    Fixes: a97955844807 ("ipc,sem: remove uneeded sem_undo_list lock usage in exit_sem()")
    Signed-off-by: Ioanna Alifieraki <ioanna-maria.alifieraki@canonical.com>
    Acked-by: Manfred Spraul <manfred@colorfullife.com>
    Acked-by: Herton R. Krzesinski <herton@redhat.com>
    Cc: Arnd Bergmann <arnd@arndb.de>
    Cc: Catalin Marinas <catalin.marinas@arm.com>
    Cc: <malat@debian.org>
    Cc: Joel Fernandes (Google) <joel@joelfernandes.org>
    Cc: Davidlohr Bueso <dave@stgolabs.net>
    Cc: Jay Vosburgh <jay.vosburgh@canonical.com>
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b1ad4bac542aea0ad076936be87f20e7d058c427
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Date:   Fri Feb 21 01:46:18 2020 +0100

    ACPI: PM: s2idle: Check fixed wakeup events in acpi_s2idle_wake()

    commit 63fb9623427fbb44e3782233b6e4714057b76ff2 upstream.

    Commit fdde0ff8590b ("ACPI: PM: s2idle: Prevent spurious SCIs from
```

```
    waking up the system") overlooked the fact that fixed events can wake
    up the system too and broke RTC wakeup from suspend-to-idle as a
    result.

    Fix this issue by checking the fixed events in acpi_s2idle_wake() in
    addition to checking wakeup GPEs and break out of the suspend-to-idle
    loop if the status bits of any enabled fixed events are set then.

    Fixes: fdde0ff8590b ("ACPI: PM: s2idle: Prevent spurious SCIs from waking up the
system")
    Reported-and-tested-by: Chris Wilson <chris@chris-wilson.co.uk>
    Cc: 5.4+ <stable@vger.kernel.org> # 5.4+
    Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b07d9425b98a97ce1df26b63ee670a8d3dc1a06f
Author: Brendan Higgins <brendanhiggins@google.com>
Date:   Fri Jan 31 14:18:32 2020 +1030

    fsi: aspeed: add unspecified HAS_IOMEM dependency

    commit ea3d147a474cb522bfdfe68f1f2557750dcf41dd upstream.

    Currently CONFIG_FSI_MASTER_ASPEED=y implicitly depends on
    CONFIG_HAS_IOMEM=y; consequently, on architectures without IOMEM we get
    the following build error:

    ld: drivers/fsi/fsi-master-aspeed.o: in function `fsi_master_aspeed_probe':
    drivers/fsi/fsi-master-aspeed.c:436: undefined reference to `devm_ioremap_resource'

    Fix the build error by adding the unspecified dependency.

    Fixes: 606397d67f41 ("fsi: Add ast2600 master driver")
    Cc: stable@vger.kernel.org
    Reported-by: Brendan Higgins <brendanhiggins@google.com>
    Signed-off-by: Brendan Higgins <brendanhiggins@google.com>
    Reviewed-by: Joel Stanley <joel@jms.id.au>
    Signed-off-by: Joel Stanley <joel@jms.id.au>
    Link: https://lore.kernel.org/r/20200131034832.294268-1-joel@jms.id.au
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7946f92552d09bd45a4888865a02ea50e5cb6bd8
Author: Jani Nikula <jani.nikula@intel.com>
Date:   Wed Feb 12 18:04:33 2020 +0200

    MAINTAINERS: Update drm/i915 bug filing URL

    commit 96228b7df33f8eb9006f8ae96949400aed9bd303 upstream.

    We've moved from bugzilla to gitlab.

    Cc: stable@vger.kernel.org
    Reviewed-by: Chris Wilson <chris@chris-wilson.co.uk>
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200212160434.6437-1-
jani.nikula@intel.com
    (cherry picked from commit 3a6a4f0810c8ade6f1ff63c34aa9834176b9d88b)
    Signed-off-by: Jani Nikula <jani.nikula@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit df7c6540f82b9f8930e91c5261650a4dca1de472
Author: Johan Hovold <johan@kernel.org>
Date:   Mon Feb 10 15:57:30 2020 +0100
```

```
    serdev: ttyport: restore client ops on deregistration

    commit 0c5aae59270fb1f827acce182786094c9ccf598e upstream.

    The serdev tty-port controller driver should reset the tty-port client
    operations also on deregistration to avoid a NULL-pointer dereference in
    case the port is later re-registered as a normal tty device.

    Note that this can only happen with tty drivers such as 8250 which have
    statically allocated port structures that can end up being reused and
    where a later registration would not register a serdev controller (e.g.
    due to registration errors or if the devicetree has been changed in
    between).

    Specifically, this can be an issue for any statically defined ports that
    would be registered by 8250 core when an 8250 driver is being unbound.

    Fixes: bed35c6dfa6a ("serdev: add a tty port controller driver")
    Cc: stable <stable@vger.kernel.org>     # 4.11
    Reported-by: Loic Poulain <loic.poulain@linaro.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Link: https://lore.kernel.org/r/20200210145730.22762-1-johan@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 09a40f7d65b93306d11b407086904362baf3b172
Author: satya priya <skakit@codeaurora.org>
Date:   Tue Feb 11 15:43:02 2020 +0530

    tty: serial: qcom_geni_serial: Fix RX cancel command failure

    commit 679aac5ead2f18d223554a52b543e1195e181811 upstream.

    RX cancel command fails when BT is switched on and off multiple times.

    To handle this, poll for the cancel bit in SE_GENI_S_IRQ_STATUS register
    instead of SE_GENI_S_CMD_CTRL_REG.

    As per the HPG update, handle the RX last bit after cancel command
    and flush out the RX FIFO buffer.

    Signed-off-by: satya priya <skakit@codeaurora.org>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/1581415982-8793-1-git-send-email-skakit@codeaurora.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 861200f2836d21cca5b86b7f481076c950f5cddd
Author: Fugang Duan <fugang.duan@nxp.com>
Date:   Tue Feb 11 14:16:01 2020 +0800

    tty: serial: imx: setup the correct sg entry for tx dma

    commit f76707831829530ffdd3888bebc108aecefccaa0 upstream.

    There has oops as below happen on i.MX8MP EVK platform that has
    6G bytes DDR memory.

    when (xmit->tail < xmit->head) && (xmit->head == 0),
    it setups one sg entry with sg->length is zero:
            sg_set_buf(sgl + 1, xmit->buf, xmit->head);

    if xmit->buf is allocated from >4G address space, and SDMA only
    support <4G address space, then dma_map_sg() will call swiotlb_map()
    to do bounce buffer copying and mapping.

    But swiotlb_map() don't allow sg entry's length is zero, otherwise
```

```
report BUG_ON().

So the patch is to correct the tx DMA scatter list.

Oops:
[  287.675715] kernel BUG at kernel/dma/swiotlb.c:497!
[  287.680592] Internal error: Oops - BUG: 0 [#1] PREEMPT SMP
[  287.686075] Modules linked in:
[  287.689133] CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.4.3-00016-g3fdc4e0-dirty #10
[  287.696872] Hardware name: FSL i.MX8MP EVK (DT)
[  287.701402] pstate: 80000085 (Nzcv daIf -PAN -UAO)
[  287.706199] pc : swiotlb_tbl_map_single+0x1fc/0x310
[  287.711076] lr : swiotlb_map+0x60/0x148
[  287.714909] sp : ffff800010003c00
[  287.718221] x29: ffff800010003c00 x28: 0000000000000000
[  287.723533] x27: 0000000000000040 x26: ffff800011ae0000
[  287.728844] x25: ffff800011ae09f8 x24: 0000000000000000
[  287.734155] x23: 00000001b7af9000 x22: 0000000000000000
[  287.739465] x21: ffff000176409c10 x20: 00000000001f7ffe
[  287.744776] x19: ffff000176409c10 x18: 000000000000002e
[  287.750087] x17: 0000000000000000 x16: 0000000000000000
[  287.755397] x15: 0000000000000000 x14: 0000000000000000
[  287.760707] x13: ffff00017f334000 x12: 0000000000000001
[  287.766018] x11: 00000000001fffff x10: 0000000000000000
[  287.771328] x9 : 0000000000000003 x8 : 0000000000000000
[  287.776638] x7 : 0000000000000000 x6 : 0000000000000000
[  287.781949] x5 : 0000000000200000 x4 : 0000000000000000
[  287.787259] x3 : 0000000000000001 x2 : 00000001b7af9000
[  287.792570] x1 : 00000000fbfff000 x0 : 0000000000000000
[  287.797881] Call trace:
[  287.800328]  swiotlb_tbl_map_single+0x1fc/0x310
[  287.804859]  swiotlb_map+0x60/0x148
[  287.808347]  dma_direct_map_page+0xf0/0x130
[  287.812530]  dma_direct_map_sg+0x78/0xe0
[  287.816453]  imx_uart_dma_tx+0x134/0x2f8
[  287.820374]  imx_uart_dma_tx_callback+0xd8/0x168
[  287.824992]  vchan_complete+0x194/0x200
[  287.828828]  tasklet_action_common.isra.0+0x154/0x1a0
[  287.833879]  tasklet_action+0x24/0x30
[  287.837540]  __do_softirq+0x120/0x23c
[  287.841202]  irq_exit+0xb8/0xd8
[  287.844343]  __handle_domain_irq+0x64/0xb8
[  287.848438]  gic_handle_irq+0x5c/0x148
[  287.852185]  el1_irq+0xb8/0x180
[  287.855327]  cpuidle_enter_state+0x84/0x360
[  287.859508]  cpuidle_enter+0x34/0x48
[  287.863083]  call_cpuidle+0x18/0x38
[  287.866571]  do_idle+0x1e0/0x280
[  287.869798]  cpu_startup_entry+0x20/0x40
[  287.873721]  rest_init+0xd4/0xe0
[  287.876949]  arch_call_rest_init+0xc/0x14
[  287.880958]  start_kernel+0x420/0x44c
[  287.884622] Code: 9124c021 9417aff8 a94363f7 17ffffd5 (d4210000)
[  287.890718] ---[ end trace 5bc44c4ab6b009ce ]---
[  287.895334] Kernel panic - not syncing: Fatal exception in interrupt
[  287.901686] SMP: stopping secondary CPUs
[  288.905607] SMP: failed to stop secondary CPUs 0-1
[  288.910395] Kernel Offset: disabled
[  288.913882] CPU features: 0x0002,2000200c
[  288.917888] Memory Limit: none
[  288.920944] ---[ end Kernel panic - not syncing: Fatal exception in interrupt ]---

Reported-by: Eagle Zhou <eagle.zhou@nxp.com>
Tested-by: Eagle Zhou <eagle.zhou@nxp.com>
Signed-off-by: Fugang Duan <fugang.duan@nxp.com>
```

```
    Cc: stable <stable@vger.kernel.org>
    Fixes: 7942f8577f2a ("serial: imx: TX DMA: clean up sg initialization")
    Reviewed-by: Uwe Kleine-König <u.kleine-koenig@pengutronix.de>
    Link: https://lore.kernel.org/r/1581401761-6378-1-git-send-email-fugang.duan@nxp.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a4b70bbd06bbaf6f361bf2d8e08d16a358114444
Author: Nicolas Ferre <nicolas.ferre@microchip.com>
Date:   Mon Feb 10 16:20:53 2020 +0100

    tty/serial: atmel: manage shutdown in case of RS485 or ISO7816 mode

    commit 04b5bfe3dc94e64d0590c54045815cb5183fb095 upstream.

    In atmel_shutdown() we call atmel_stop_rx() and atmel_stop_tx() functions.
    Prevent the rx restart that is implemented in RS485 or ISO7816 modes when
    calling atmel_stop_tx() by using the atomic information tasklet_shutdown
    that is already in place for this purpose.

    Fixes: 98f2082c3ac4 ("tty/serial: atmel: enforce tasklet init and termination
sequences")
    Signed-off-by: Nicolas Ferre <nicolas.ferre@microchip.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200210152053.8289-1-nicolas.ferre@microchip.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3379ddd8cbbd152afa6777b34b016f003ac59a7f
Author: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
Date:   Tue Feb 11 15:55:59 2020 +0200

    serial: 8250: Check UPF_IRQ_SHARED in advance

    commit 7febbcbc48fc92e3f33863b32ed715ba4aff18c4 upstream.

    The commit 54e53b2e8081
      ("tty: serial: 8250: pass IRQ shared flag to UART ports")
    nicely explained the problem:

    ---8<---8<---

    On some systems IRQ lines between multiple UARTs might be shared. If so, the
    irqflags have to be configured accordingly. The reason is: The 8250 port startup
    code performs IRQ tests *before* the IRQ handler for that particular port is
    registered. This is performed in serial8250_do_startup(). This function checks
    whether IRQF_SHARED is configured and only then disables the IRQ line while
    testing.

    This test is performed upon each open() of the UART device. Imagine two UARTs
    share the same IRQ line: On is already opened and the IRQ is active. When the
    second UART is opened, the IRQ line has to be disabled while performing IRQ
    tests. Otherwise an IRQ might handler might be invoked, but the IRQ itself
    cannot be handled, because the corresponding handler isn't registered,
    yet. That's because the 8250 code uses a chain-handler and invokes the
    corresponding port's IRQ handling routines himself.

    Unfortunately this IRQF_SHARED flag isn't configured for UARTs probed via device
    tree even if the IRQs are shared. This way, the actual and shared IRQ line isn't
    disabled while performing tests and the kernel correctly detects a spurious
    IRQ. So, adding this flag to the DT probe solves the issue.

    Note: The UPF_SHARE_IRQ flag is configured unconditionally. Therefore, the
    IRQF_SHARED flag can be set unconditionally as well.

    Example stack trace by performing `echo 1 > /dev/ttyS2` on a non-patched system:
```

```
      |irq 85: nobody cared (try booting with the "irqpoll" option)
      | [...]
      |handlers:
      |[<ffff0000080fc628>] irq_default_primary_handler threaded [<ffff00000855fbb8>]
  serial8250_interrupt
      |Disabling IRQ #85

      ---8<---8<---

      But unfortunately didn't fix the root cause. Let's try again here by moving
      IRQ flag assignment from serial_link_irq_chain() to serial8250_do_startup().

      This should fix the similar issue reported for 8250_pnp case.

      Since this change we don't need to have custom solutions in 8250_aspeed_vuart
      and 8250_of drivers, thus, drop them.

      Fixes: 1c2f04937b3e ("serial: 8250: add IRQ trigger support")
      Reported-by: Li RongQing <lirongqing@baidu.com>
      Cc: Kurt Kanzenbach <kurt@linutronix.de>
      Cc: Vikram Pandita <vikram.pandita@ti.com>
      Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
      Cc: stable <stable@vger.kernel.org>
      Acked-by: Kurt Kanzenbach <kurt@linutronix.de>
      Link: https://lore.kernel.org/r/20200211135559.85960-1-andriy.shevchenko@linux.intel.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit ab8d411190d7119b38c4b3db39d0908e96cb7558
  Author: Paul Cercueil <paul@crapouillou.net>
  Date:   Tue Feb 11 11:53:37 2020 -0300

      MIPS: ingenic: DTS: Fix watchdog nodes

      commit 11479e8e3cd896673a15af21cd0f145a4752f01a upstream.

      The devicetree ABI was broken on purpose by commit 6d532143c915
      ("watchdog: jz4740: Use regmap provided by TCU driver"), and
      commit 1d9c30745455 ("watchdog: jz4740: Use WDT clock provided
      by TCU driver"). The commit message of the latter explains why the ABI
      was broken.

      However, the current devicetree files were not updated to the new ABI
      described in Documentation/devicetree/bindings/timer/ingenic,tcu.txt,
      so the watchdog driver would not probe.

      Fix this problem by updating the watchdog nodes to comply with the new
      ABI.

      Fixes: 6d532143c915 ("watchdog: jz4740: Use regmap provided by TCU driver")
      Signed-off-by: Paul Cercueil <paul@crapouillou.net>
      Reviewed-by: Philippe Mathieu-Daudé <f4bug@amsat.org>
      Signed-off-by: Paul Burton <paulburton@kernel.org>
      Cc: Ralf Baechle <ralf@linux-mips.org>
      Cc: Rob Herring <robh+dt@kernel.org>
      Cc: Mark Rutland <mark.rutland@arm.com>
      Cc: Zhou Yanjie <zhouyanjie@wanyeetech.com>
      Cc: od@zcrc.me
      Cc: linux-mips@vger.kernel.org
      Cc: devicetree@vger.kernel.org
      Cc: linux-kernel@vger.kernel.org
      Cc: <stable@vger.kernel.org> # v5.5+
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 30220a9b690a9d955631bed6c657a7e8a0958064
  Author: Kim Phillips <kim.phillips@amd.com>
```

```
Date:   Wed Feb 19 18:52:43 2020 +0100

    x86/cpu/amd: Enable the fixed Instructions Retired counter IRPERF

    commit 21b5ee59ef18e27d85810584caf1f7ddc705ea83 upstream.

    Commit

      aaf248848db50 ("perf/x86/msr: Add AMD IRPERF (Instructions Retired)
                     performance counter")

    added support for access to the free-running counter via 'perf -e
    msr/irperf/', but when exercised, it always returns a 0 count:

    BEFORE:

      $ perf stat -e instructions,msr/irperf/ true

       Performance counter stats for 'true':

                 624,833      instructions
                       0      msr/irperf/

    Simply set its enable bit - HWCR bit 30 - to make it start counting.

    Enablement is restricted to all machines advertising IRPERF capability,
    except those susceptible to an erratum that makes the IRPERF return
    bad values.

    That erratum occurs in Family 17h models 00-1fh [1], but not in F17h
    models 20h and above [2].

    AFTER (on a family 17h model 31h machine):

      $ perf stat -e instructions,msr/irperf/ true

       Performance counter stats for 'true':

                 621,690      instructions
                 622,490      msr/irperf/

    [1] Revision Guide for AMD Family 17h Models 00h-0Fh Processors
    [2] Revision Guide for AMD Family 17h Models 30h-3Fh Processors

    The revision guides are available from the bugzilla Link below.

     [ bp: Massage commit message. ]

    Fixes: aaf248848db50 ("perf/x86/msr: Add AMD IRPERF (Instructions Retired) performance
counter")
    Signed-off-by: Kim Phillips <kim.phillips@amd.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: Peter Zijlstra <peterz@infradead.org>
    Cc: stable@vger.kernel.org
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=206537
    Link: http://lkml.kernel.org/r/20200214201805.13830-1-kim.phillips@amd.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9c5c16c32ed2ef260349e9a481e7a955d7e661ba
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Thu Feb 13 19:01:34 2020 +0100

    x86/mce/amd: Fix kobject lifetime

    commit 51dede9c05df2b78acd6dcf6a17d21f0877d2d7b upstream.
```

```
    Accessing the MCA thresholding controls in sysfs concurrently with CPU
    hotplug can lead to a couple of KASAN-reported issues:

      BUG: KASAN: use-after-free in sysfs_file_ops+0x155/0x180
      Read of size 8 at addr ffff888367578940 by task grep/4019

    and

      BUG: KASAN: use-after-free in show_error_count+0x15c/0x180
      Read of size 2 at addr ffff888368a05514 by task grep/4454

    for example. Both result from the fact that the threshold block
    creation/teardown code frees the descriptor memory itself instead of
    defining proper ->release function and leaving it to the driver core to
    take care of that, after all sysfs accesses have completed.

    Do that and get rid of the custom freeing code, fixing the above UAFs in
    the process.

      [ bp: write commit message. ]

    Fixes: 95268664390b ("[PATCH] x86_64: mce_amd support for family 0x10 processors")
    Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: <stable@vger.kernel.org>
    Link: https://lkml.kernel.org/r/20200214082801.13836-1-bp@alien8.de
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c1ed043193396e5cbb4494c2f44b338b271bbd43
Author: Borislav Petkov <bp@suse.de>
Date:   Tue Feb 4 13:28:41 2020 +0100

    x86/mce/amd: Publish the bank pointer only after setup has succeeded

    commit 6e5cf31fbe651bed7ba1df768f2e123531132417 upstream.

    threshold_create_bank() creates a bank descriptor per MCA error
    thresholding counter which can be controlled over sysfs. It publishes
    the pointer to that bank in a per-CPU variable and then goes on to
    create additional thresholding blocks if the bank has such.

    However, that creation of additional blocks in
    allocate_threshold_blocks() can fail, leading to a use-after-free
    through the per-CPU pointer.

    Therefore, publish that pointer only after all blocks have been setup
    successfully.

    Fixes: 019f34fccfd5 ("x86, MCE, AMD: Move shared bank to node descriptor")
    Reported-by: Saar Amar <Saar.Amar@microsoft.com>
    Reported-by: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Borislav Petkov <bp@suse.de>
    Cc: <stable@vger.kernel.org>
    Link: http://lkml.kernel.org/r/20200128140846.phctkvx5btiexvbx@kili.mountain
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 907972d73048d113197c4e25beb083af8be2297c
Author: Ard Biesheuvel <ardb@kernel.org>
Date:   Sat Feb 1 09:32:21 2020 +0100

    x86/ima: use correct identifier for SetupMode variable

    commit ff5ac61ee83c13f516544d29847d28be093a40ee upstream.
```

```
    The IMA arch code attempts to inspect the "SetupMode" EFI variable
    by populating a variable called efi_SetupMode_name with the string
    "SecureBoot" and passing that to the EFI GetVariable service, which
    obviously does not yield the expected result.

    Given that the string is only referenced a single time, let's get
    rid of the intermediate variable, and pass the correct string as
    an immediate argument. While at it, do the same for "SecureBoot".

    Fixes: 399574c64eaf ("x86/ima: retry detecting secure boot mode")
    Fixes: 980ef4d22a95 ("x86/ima: check EFI SetupMode too")
    Cc: Matthew Garrett <mjg59@google.com>
    Signed-off-by: Ard Biesheuvel <ardb@kernel.org>
    Cc: stable@vger.kernel.org # v5.3
    Signed-off-by: Mimi Zohar <zohar@linux.ibm.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2a168f7252c7e2982d224217a83f5d3ffc7f894b
Author: wangyan <wangyan122@huawei.com>
Date:   Thu Feb 20 21:46:14 2020 +0800

    jbd2: fix ocfs2 corrupt when clearing block group bits

    commit 8eedabfd66b68a4623beec0789eac54b8c9d0fb6 upstream.

    I found a NULL pointer dereference in ocfs2_block_group_clear_bits().
    The running environment:
            kernel version: 4.19
            A cluster with two nodes, 5 luns mounted on two nodes, and do some
            file operations like dd/fallocate/truncate/rm on every lun with storage
            network disconnection.

    The fallocate operation on dm-23-45 caused an null pointer dereference.

    The information of NULL pointer dereference as follows:
            [577992.878282] JBD2: Error -5 detected when updating journal superblock for dm-
23-45.
            [577992.878290] Aborting journal on device dm-23-45.
            ...
            [577992.890778] JBD2: Error -5 detected when updating journal superblock for dm-
24-46.
            [577992.890908] __journal_remove_journal_head: freeing b_committed_data
            [577992.890916] (fallocate,88392,52):ocfs2_extend_trans:474 ERROR: status = -30
            [577992.890918] __journal_remove_journal_head: freeing b_committed_data
            [577992.890920] (fallocate,88392,52):ocfs2_rotate_tree_right:2500 ERROR: status
= -30
            [577992.890922] __journal_remove_journal_head: freeing b_committed_data
            [577992.890924] (fallocate,88392,52):ocfs2_do_insert_extent:4382 ERROR: status =
-30
            [577992.890928] (fallocate,88392,52):ocfs2_insert_extent:4842 ERROR: status =
-30
            [577992.890928] __journal_remove_journal_head: freeing b_committed_data
            [577992.890930] (fallocate,88392,52):ocfs2_add_clusters_in_btree:4947 ERROR:
status = -30
            [577992.890933] __journal_remove_journal_head: freeing b_committed_data
            [577992.890939] __journal_remove_journal_head: freeing b_committed_data
            [577992.890949] Unable to handle kernel NULL pointer dereference at virtual
address 0000000000000020
            [577992.890950] Mem abort info:
            [577992.890951]   ESR = 0x96000004
            [577992.890952]   Exception class = DABT (current EL), IL = 32 bits
            [577992.890952]   SET = 0, FnV = 0
            [577992.890953]   EA = 0, S1PTW = 0
            [577992.890954] Data abort info:
            [577992.890955]   ISV = 0, ISS = 0x00000004
```

```
            [577992.890956]   CM = 0, WnR = 0
            [577992.890958] user pgtable: 4k pages, 48-bit VAs, pgdp = 00000000f8da07a9
            [577992.890960] [0000000000000020] pgd=0000000000000000
            [577992.890964] Internal error: Oops: 96000004 [#1] SMP
            [577992.890965] Process fallocate (pid: 88392, stack limit = 0x00000000013db2fd)
            [577992.890968] CPU: 52 PID: 88392 Comm: fallocate Kdump: loaded Tainted: G
 W  OE     4.19.36 #1
            [577992.890969] Hardware name: Huawei TaiShan 2280 V2/BC82AMDD, BIOS 0.98
 08/25/2019
            [577992.890971] pstate: 60400009 (nZCv daif +PAN -UAO)
            [577992.891054] pc : _ocfs2_free_suballoc_bits+0x63c/0x968 [ocfs2]
            [577992.891082] lr : _ocfs2_free_suballoc_bits+0x618/0x968 [ocfs2]
            [577992.891084] sp : ffff0000c8e2b810
            [577992.891085] x29: ffff0000c8e2b820 x28: 0000000000000000
            [577992.891087] x27: 00000000000006f3 x26: ffffa07957b02e70
            [577992.891089] x25: ffff807c59d50000 x24: 00000000000006f2
            [577992.891091] x23: 0000000000000001 x22: ffff807bd39abc30
            [577992.891093] x21: ffff0000811d9000 x20: ffffa07535d6a000
            [577992.891097] x19: ffff000001681638 x18: ffffffffffffffff
            [577992.891098] x17: 0000000000000000 x16: ffff000080a03df0
            [577992.891100] x15: ffff0000811d9708 x14: 203d207375746174
            [577992.891101] x13: 73203a524f525245 x12: 20373439343a6565
            [577992.891103] x11: 0000000000000038 x10: 0101010101010101
            [577992.891106] x9 : ffffa07c68a85d70 x8 : 7f7f7f7f7f7f7f7f
            [577992.891109] x7 : 0000000000000000 x6 : 0000000000000080
            [577992.891110] x5 : 0000000000000000 x4 : 0000000000000002
            [577992.891112] x3 : ffff000001713390 x2 : 2ff90f88b1c22f00
            [577992.891114] x1 : ffff807bd39abc30 x0 : 0000000000000000
            [577992.891116] Call trace:
            [577992.891139]  _ocfs2_free_suballoc_bits+0x63c/0x968 [ocfs2]
            [577992.891162]  _ocfs2_free_clusters+0x100/0x290 [ocfs2]
            [577992.891185]  ocfs2_free_clusters+0x50/0x68 [ocfs2]
            [577992.891206]  ocfs2_add_clusters_in_btree+0x198/0x5e0 [ocfs2]
            [577992.891227]  ocfs2_add_inode_data+0x94/0xc8 [ocfs2]
            [577992.891248]  ocfs2_extend_allocation+0x1bc/0x7a8 [ocfs2]
            [577992.891269]  ocfs2_allocate_extents+0x14c/0x338 [ocfs2]
            [577992.891290]  __ocfs2_change_file_space+0x3f8/0x610 [ocfs2]
            [577992.891309]  ocfs2_fallocate+0xe4/0x128 [ocfs2]
            [577992.891316]  vfs_fallocate+0x11c/0x250
            [577992.891317]  ksys_fallocate+0x54/0x88
            [577992.891319]  __arm64_sys_fallocate+0x28/0x38
            [577992.891323]  el0_svc_common+0x78/0x130
            [577992.891325]  el0_svc_handler+0x38/0x78
            [577992.891327]  el0_svc+0x8/0xc

    My analysis process as follows:
    ocfs2_fallocate
      __ocfs2_change_file_space
        ocfs2_allocate_extents
          ocfs2_extend_allocation
            ocfs2_add_inode_data
              ocfs2_add_clusters_in_btree
                ocfs2_insert_extent
                  ocfs2_do_insert_extent
                    ocfs2_rotate_tree_right
                      ocfs2_extend_rotate_transaction
                        ocfs2_extend_trans
                          jbd2_journal_restart
                            jbd2__journal_restart
                              /* handle->h_transaction is NULL,
                               * is_handle_aborted(handle) is true
                               */
                              handle->h_transaction = NULL;
                              start_this_handle
                                return -EROFS;
```

```
                ocfs2_free_clusters
                  _ocfs2_free_clusters
                    _ocfs2_free_suballoc_bits
                      ocfs2_block_group_clear_bits
                        ocfs2_journal_access_gd
                          __ocfs2_journal_access
                            jbd2_journal_get_undo_access
                              /* I think jbd2_write_access_granted() will
                               * return true, because do_get_write_access()
                               * will return -EROFS.
                               */
                              if (jbd2_write_access_granted(...)) return 0;
                              do_get_write_access
                                /* handle->h_transaction is NULL, it will
                                 * return -EROFS here, so do_get_write_access()
                                 * was not called.
                                 */
                                if (is_handle_aborted(handle)) return -EROFS;
                        /* bh2jh(group_bh) is NULL, caused NULL
                           pointer dereference */
                        undo_bg = (struct ocfs2_group_desc *)
                                    bh2jh(group_bh)->b_committed_data;
```

    If handle->h_transaction == NULL, then jbd2_write_access_granted()
    does not really guarantee that journal_head will stay around,
    not even speaking of its b_committed_data. The bh2jh(group_bh)
    can be removed after ocfs2_journal_access_gd() and before call
    "bh2jh(group_bh)->b_committed_data". So, we should move
    is_handle_aborted() check from do_get_write_access() into
    jbd2_journal_get_undo_access() and jbd2_journal_get_write_access()
    before the call to jbd2_write_access_granted().

    Link: https://lore.kernel.org/r/f72a623f-b3f1-381a-d91d-d22a1c83a336@huawei.com
    Signed-off-by: Yan Wang <wangyan122@huawei.com>
    Signed-off-by: Theodore Ts'o <tytso@mit.edu>
    Reviewed-by: Jun Piao <piaojun@huawei.com>
    Reviewed-by: Jan Kara <jack@suse.cz>
    Cc: stable@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7f745f5c3ac9eea25584b911b6fe3236f4c6d1cf
Author: Will Deacon <will@kernel.org>
Date: Wed Feb 19 10:19:13 2020 +0000

    arm64: memory: Add missing brackets to untagged_addr() macro

    commit d0022c0ef29b78bcbe8a5c5894bd2307143afce1 upstream.

    Add brackets around the evaluation of the 'addr' parameter to the
    untagged_addr() macro so that the cast to 'u64' applies to the result
    of the expression.

    Cc: <stable@vger.kernel.org>
    Fixes: 597399d0cb91 ("arm64: tags: Preserve tags for addresses translated via TTBR1")
    Reported-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Will Deacon <will@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 314f70006e6c7bc615680ac8d3b7fde79af2fd04
Author: Christophe Leroy <christophe.leroy@c-s.fr>
Date: Sun Feb 9 16:02:41 2020 +0000

    powerpc/hugetlb: Fix 8M hugepages on 8xx

    commit 50a175dd18de7a647e72aca7daf4744e3a5a81e3 upstream.

```
    With HW assistance all page tables must be 4k aligned, the 8xx drops
    the last 12 bits during the walk.

    Redefine HUGEPD_SHIFT_MASK to mask last 12 bits out. HUGEPD_SHIFT_MASK
    is used to for alignment of page table cache.

    Fixes: 22569b881d37 ("powerpc/8xx: Enable 8M hugepage support with HW assistance")
    Cc: stable@vger.kernel.org # v5.0+
    Signed-off-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
https://lore.kernel.org/r/778b1a248c4c7ca79640eeff7740044da6a220a0.1581264115.git.christophe
.leroy@c-s.fr
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d72a5348ae2e89ebb3ca26fa27a113c0f2f0bd6a
Author: Christophe Leroy <christophe.leroy@c-s.fr>
Date:   Thu Feb 6 13:50:28 2020 +0000

    powerpc/hugetlb: Fix 512k hugepages on 8xx with 16k page size

    commit f2b67ef90b0d5eca0f2255e02cf2f620bc0ddcdb upstream.

    Commit 55c8fc3f4930 ("powerpc/8xx: reintroduce 16K pages with HW
    assistance") redefined pte_t as a struct of 4 pte_basic_t, because
    in 16K pages mode there are four identical entries in the
    page table. But the size of hugepage tables is calculated based
    of the size of (void *). Therefore, we end up with page tables
    of size 1k instead of 4k for 512k pages.

    As 512k hugepage tables are the same size as standard page tables,
    ie 4k, use the standard page tables instead of PGT_CACHE tables.

    Fixes: 3fb69c6a1a13 ("powerpc/8xx: Enable 512k hugepage support with HW assistance")
    Cc: stable@vger.kernel.org # v5.0+
    Signed-off-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
https://lore.kernel.org/r/90ec56a2315be602494619ed0223bba3b0b8d619.1580997007.git.christophe
.leroy@c-s.fr
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a3a4482f814431274c8792cf0a889799a426669
Author: Christophe Leroy <christophe.leroy@c-s.fr>
Date:   Tue Feb 18 14:09:29 2020 +0000

    powerpc/entry: Fix an #if which should be an #ifdef in entry_32.S

    commit 9eb425b2e04e0e3006adffea5bf5f227a896f128 upstream.

    Fixes: 12c3f1fd87bf ("powerpc/32s: get rid of CPU_FTR_601 feature")
    Cc: stable@vger.kernel.org # v5.4+
    Signed-off-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
https://lore.kernel.org/r/a99fc0ad65b87a1ba51cfa3e0e9034ee294c3e07.1582034961.git.christophe
.leroy@c-s.fr
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit fe307d260c36b1fac4b417c4eab6bd0f8752b97e
Author: Gustavo Luiz Duarte <gustavold@linux.ibm.com>
Date:   Tue Feb 11 00:38:29 2020 -0300

    powerpc/tm: Fix clearing MSR[TS] in current when reclaiming on signal delivery
```

```
    commit 2464cc4c345699adea52c7aef75707207cb8a2f6 upstream.

    After a treclaim, we expect to be in non-transactional state. If we
    don't clear the current thread's MSR[TS] before we get preempted, then
    tm_recheckpoint_new_task() will recheckpoint and we get rescheduled in
    suspended transaction state.

    When handling a signal caught in transactional state,
    handle_rt_signal64() calls get_tm_stackpointer() that treclaims the
    transaction using tm_reclaim_current() but without clearing the
    thread's MSR[TS]. This can cause the TM Bad Thing exception below if
    later we pagefault and get preempted trying to access the user's
    sigframe, using __put_user(). Afterwards, when we are rescheduled back
    into do_page_fault() (but now in suspended state since the thread's
    MSR[TS] was not cleared), upon executing 'rfid' after completion of
    the page fault handling, the exception is raised because a transition
    from suspended to non-transactional state is invalid.

      Unexpected TM Bad Thing exception at c00000000000de44 (msr 0x8000000302a03031)
tm_scratch=800000010280b033
      Oops: Unrecoverable exception, sig: 6 [#1]
      LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries
      CPU: 25 PID: 15547 Comm: a.out Not tainted 5.4.0-rc2 #32
      NIP:  c00000000000de44 LR: c000000000034728 CTR: 0000000000000000
      REGS: c00000003fe7bd70 TRAP: 0700   Not tainted  (5.4.0-rc2)
      MSR:  8000000302a03031 <SF,VEC,VSX,FP,ME,IR,DR,LE,TM[SE]>  CR: 44000884  XER: 00000000
      CFAR: c00000000000dda4 IRQMASK: 0
      PACATMSCRATCH: 800000010280b033
      GPR00: c000000000034728 c000000f65a17c80 c000000001662800 00007fffacf3fd78
      GPR04: 0000000000001000 0000000000001000 0000000000000000 c000000f611f8af0
      GPR08: 0000000000000000 0000000078006001 0000000000000000 000c000000000000
      GPR12: c000000f611f84b0 c00000003ffcb200 0000000000000000 0000000000000000
      GPR16: 0000000000000000 0000000000000000 0000000000000000 0000000000000000
      GPR20: 0000000000000000 0000000000000000 0000000000000000 c000000f611f8140
      GPR24: 0000000000000000 00007fffacf3fd68 c000000f65a17d90 c000000f611f7800
      GPR28: c000000f65a17e90 c000000f65a17e90 c000000001685e18 00007fffacf3f000
      NIP [c00000000000de44] fast_exception_return+0xf4/0x1b0
      LR [c000000000034728] handle_rt_signal64+0x78/0xc50
      Call Trace:
      [c000000f65a17c80] [c000000000034710] handle_rt_signal64+0x60/0xc50 (unreliable)
      [c000000f65a17d30] [c000000000023640] do_notify_resume+0x330/0x460
      [c000000f65a17e20] [c00000000000dcc4] ret_from_except_lite+0x70/0x74
      Instruction dump:
      7c4ff120 e8410170 7c5a03a6 38400000 f8410060 e8010070 e8410080 e8610088
      60000000 60000000 e8810090 e8210078 <4c000024> 48000000 e8610178 88ed0989
      ---[ end trace 93094aa44b442f87 ]---

    The simplified sequence of events that triggers the above exception is:

      ...                           # userspace in NON-TRANSACTIONAL state
      tbegin                        # userspace in TRANSACTIONAL state
      signal delivery               # kernelspace in SUSPENDED state
      handle_rt_signal64()
        get_tm_stackpointer()
          treclaim                  # kernelspace in NON-TRANSACTIONAL state
        __put_user()
          page fault happens. We will never get back here because of the TM Bad Thing
exception.

      page fault handling kicks in and we voluntarily preempt ourselves
      do_page_fault()
        __schedule()
          __switch_to(other_task)
```

```
       our task is rescheduled and we recheckpoint because the thread's MSR[TS] was not
  cleared
       __switch_to(our_task)
         switch_to_tm()
           tm_recheckpoint_new_task()
             trechkpt                        # kernelspace in SUSPENDED state

       The page fault handling resumes, but now we are in suspended transaction state
       do_page_fault()    completes
       rfid     <----- trying to get back where the page fault happened (we were non-
  transactional back then)
       TM Bad Thing                  # illegal transition from suspended to non-transactional

    This patch fixes that issue by clearing the current thread's MSR[TS]
    just after treclaim in get_tm_stackpointer() so that we stay in
    non-transactional state in case we are preempted. In order to make
    treclaim and clearing the thread's MSR[TS] atomic from a preemption
    perspective when CONFIG_PREEMPT is set, preempt_disable/enable() is
    used. It's also necessary to save the previous value of the thread's
    MSR before get_tm_stackpointer() is called so that it can be exposed
    to the signal handler later in setup_tm_sigcontexts() to inform the
    userspace MSR at the moment of the signal delivery.

    Found with tm-signal-context-force-tm kernel selftest.

    Fixes: 2b0a576d15e0 ("powerpc: Add new transactional memory state to the signal
  context")
    Cc: stable@vger.kernel.org # v3.9
    Signed-off-by: Gustavo Luiz Duarte <gustavold@linux.ibm.com>
    Acked-by: Michael Neuling <mikey@neuling.org>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link: https://lore.kernel.org/r/20200211033831.11165-1-gustavold@linux.ibm.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 326b50796a4405a4ef60c92ef559177d7e6429ca
  Author: Sam Bobroff <sbobroff@linux.ibm.com>
  Date:   Fri Feb 7 15:57:31 2020 +1100

    powerpc/eeh: Fix deadlock handling dead PHB

    commit d4f194ed9eb9841a8f978710e4d24296f791a85b upstream.

    Recovering a dead PHB can currently cause a deadlock as the PCI
    rescan/remove lock is taken twice.

    This is caused as part of an existing bug in
    eeh_handle_special_event(). The pe is processed while traversing the
    PHBs even though the pe is unrelated to the loop. This causes the pe
    to be, incorrectly, processed more than once.

    Untangling this section can move the pe processing out of the loop and
    also outside the locked section, correcting both problems.

    Fixes: 2e25505147b8 ("powerpc/eeh: Fix crash when edev->pdev changes")
    Cc: stable@vger.kernel.org # 5.4+
    Signed-off-by: Sam Bobroff <sbobroff@linux.ibm.com>
    Reviewed-by: Frederic Barrat <fbarrat@linux.ibm.com>
    Tested-by: Frederic Barrat <fbarrat@linux.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
  https://lore.kernel.org/r/0547e82dbf90ee0729a2979a8cac5c91665c621f.1581051445.git.sbobroff@l
  inux.ibm.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit a863f612309ce9744c6ab6ec37c388cb54081669
```

```
Author: Christophe Leroy <christophe.leroy@c-s.fr>
Date:   Sun Feb 9 18:14:42 2020 +0000

    powerpc/8xx: Fix clearing of bits 20-23 in ITLB miss

    commit a4031afb9d10d97f4d0285844abbc0ab04245304 upstream.

    In ITLB miss handled the line supposed to clear bits 20-23 on the L2
    ITLB entry is buggy and does indeed nothing, leading to undefined
    value which could allow execution when it shouldn't.

    Properly do the clearing with the relevant instruction.

    Fixes: 74fabcadfd43 ("powerpc/8xx: don't use r12/SPRN_SPRG_SCRATCH2 in TLB Miss
handlers")
    Cc: stable@vger.kernel.org # v5.0+
    Signed-off-by: Christophe Leroy <christophe.leroy@c-s.fr>
    Reviewed-by: Leonardo Bras <leonardo@linux.ibm.com>
    Signed-off-by: Michael Ellerman <mpe@ellerman.id.au>
    Link:
https://lore.kernel.org/r/4f70c2778163affce8508a210f65d140e84524b4.1581272050.git.christophe
.leroy@c-s.fr
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1179bfbde856e11beb28d738458510e0764ca4b0
Author: Boris Brezillon <boris.brezillon@collabora.com>
Date:   Thu Feb 6 15:13:27 2020 +0100

    drm/panfrost: perfcnt: Reserve/use the AS attached to the perfcnt MMU context

    commit dde2bb2da01e96c17f0a44b4a3cf72a30e66e3ef upstream.

    We need to use the AS attached to the opened FD when dumping counters.

    Reported-by: Antonio Caggiano <antonio.caggiano@collabora.com>
    Fixes: 7282f7645d06 ("drm/panfrost: Implement per FD address spaces")
    Cc: <stable@vger.kernel.org>
    Signed-off-by: Boris Brezillon <boris.brezillon@collabora.com>
    Reviewed-by: Steven Price <steven.price@arm.com>
    Tested-by: Antonio Caggiano <antonio.caggiano@collabora.com>
    Signed-off-by: Rob Herring <robh@kernel.org>
    Link: https://patchwork.freedesktop.org/patch/msgid/20200206141327.446127-1-
boris.brezillon@collabora.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 84d064535079cca6b4c1040e6cbcc1332a3cbde2
Author: Guenter Roeck <linux@roeck-us.net>
Date:   Wed Feb 19 14:36:14 2020 -0800

    hwmon: (acpi_power_meter) Fix lockdep splat

    commit badcd4546d52ae4318f2bcfda0e47a1394b60e38 upstream.

    Damien Le Moal reports a lockdep splat with the acpi_power_meter,
    observed with Linux v5.5 and later.

    ======================================================
    WARNING: possible circular locking dependency detected
    5.6.0-rc2+ #629 Not tainted
    ------------------------------------------------------
    python/1397 is trying to acquire lock:
    ffff888619080070 (&resource->lock){+.+.}, at: show_power+0x3c/0xa0 [acpi_power_meter]

                   but task is already holding lock:
    ffff88881643f188 (kn->count#119){++++}, at: kernfs_seq_start+0x6a/0x160
```

```
                which lock already depends on the new lock.

                the existing dependency chain (in reverse order) is:

                -> #1 (kn->count#119){++++}:
        __kernfs_remove+0x626/0x7e0
        kernfs_remove_by_name_ns+0x41/0x80
        remove_attrs+0xcb/0x3c0 [acpi_power_meter]
        acpi_power_meter_notify+0x1f7/0x310 [acpi_power_meter]
        acpi_ev_notify_dispatch+0x198/0x1f3
        acpi_os_execute_deferred+0x4d/0x70
        process_one_work+0x7c8/0x1340
        worker_thread+0x94/0xc70
        kthread+0x2ed/0x3f0
        ret_from_fork+0x24/0x30

                -> #0 (&resource->lock){+.+.}:
        __lock_acquire+0x20be/0x49b0
        lock_acquire+0x127/0x340
        __mutex_lock+0x15b/0x1350
        show_power+0x3c/0xa0 [acpi_power_meter]
        dev_attr_show+0x3f/0x80
        sysfs_kf_seq_show+0x216/0x410
        seq_read+0x407/0xf90
        vfs_read+0x152/0x2c0
        ksys_read+0xf3/0x1d0
        do_syscall_64+0x95/0x1010
        entry_SYSCALL_64_after_hwframe+0x49/0xbe

                other info that might help us debug this:

 Possible unsafe locking scenario:

       CPU0                    CPU1
       ----                    ----
  lock(kn->count#119);
                               lock(&resource->lock);
                               lock(kn->count#119);
  lock(&resource->lock);

                *** DEADLOCK ***
4 locks held by python/1397:
 #0: ffff8890242d64e0 (&f->f_pos_lock){+.+.}, at: __fdget_pos+0x9b/0xb0
 #1: ffff889040be74e0 (&p->lock){+.+.}, at: seq_read+0x6b/0xf90
 #2: ffff8890448eb880 (&of->mutex){+.+.}, at: kernfs_seq_start+0x47/0x160
 #3: ffff88881643f188 (kn->count#119){++++}, at: kernfs_seq_start+0x6a/0x160

                stack backtrace:
CPU: 10 PID: 1397 Comm: python Not tainted 5.6.0-rc2+ #629
Hardware name: Supermicro Super Server/X11DPL-i, BIOS 3.1 05/21/2019
Call Trace:
 dump_stack+0x97/0xe0
 check_noncircular+0x32e/0x3e0
 ? print_circular_bug.isra.0+0x1e0/0x1e0
 ? unwind_next_frame+0xb9a/0x1890
 ? entry_SYSCALL_64_after_hwframe+0x49/0xbe
 ? graph_lock+0x79/0x170
 ? __lockdep_reset_lock+0x3c0/0x3c0
 ? mark_lock+0xbc/0x1150
 __lock_acquire+0x20be/0x49b0
 ? mark_held_locks+0xe0/0xe0
 ? stack_trace_save+0x91/0xc0
 lock_acquire+0x127/0x340
 ? show_power+0x3c/0xa0 [acpi_power_meter]
```

```
 ? device_remove_bin_file+0x10/0x10
 ? device_remove_bin_file+0x10/0x10
 __mutex_lock+0x15b/0x1350
 ? show_power+0x3c/0xa0 [acpi_power_meter]
 ? show_power+0x3c/0xa0 [acpi_power_meter]
 ? mutex_lock_io_nested+0x11f0/0x11f0
 ? lock_downgrade+0x6a0/0x6a0
 ? kernfs_seq_start+0x47/0x160
 ? lock_acquire+0x127/0x340
 ? kernfs_seq_start+0x6a/0x160
 ? device_remove_bin_file+0x10/0x10
 ? show_power+0x3c/0xa0 [acpi_power_meter]
 show_power+0x3c/0xa0 [acpi_power_meter]
 dev_attr_show+0x3f/0x80
 ? memset+0x20/0x40
 sysfs_kf_seq_show+0x216/0x410
 seq_read+0x407/0xf90
 ? security_file_permission+0x16f/0x2c0
 vfs_read+0x152/0x2c0
```

Problem is that reading an attribute takes the kernfs lock in the kernfs code, then resource->lock in the driver. During an ACPI notification, the opposite happens: The resource lock is taken first, followed by the kernfs lock when sysfs attributes are removed and re-created. Presumably this is now seen due to some locking related changes in kernfs after v5.4, but it was likely always a problem.

Fix the problem by not blindly acquiring the lock in the notification function. It is only needed to protect the various update functions. However, those update functions are called anyway when sysfs attributes are read. This means that we can just stop calling those functions from the notifier, and the resource lock in the notifier function is no longer needed.

That leaves two situations:

First, METER_NOTIFY_CONFIG removes and re-allocates capability strings. While it did so under the resource lock, _displaying_ those strings was not protected, creating a race condition. To solve this problem, selectively protect both removal/creation and reporting of capability attributes with the resource lock.

Second, removing and re-creating the attribute files is no longer protected by the resource lock. That doesn't matter since access to each individual attribute is protected by the kernfs lock. Userspace may get messed up if attributes disappear and reappear under its nose, but that is not different than today, and there is nothing we can do about it without major driver restructuring.

Last but not least, when removing the driver, remove attribute functions first, then release capability strings. This avoids yet another race condition.

Reported-by: Damien Le Moal <Damien.LeMoal@wdc.com>
Cc: Damien Le Moal <Damien.LeMoal@wdc.com>
Cc: stable@vger.kernel.org # v5.5+
Tested-by: Damien Le Moal <damien.lemoal@wdc.com>
Signed-off-by: Guenter Roeck <linux@roeck-us.net>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 564bcdd09c0a00cf8a08ed24605af0c4a8cae507
Author: Larry Finger <Larry.Finger@lwfinger.net>
Date: Mon Feb 10 12:02:33 2020 -0600

staging: rtl8723bs: Fix potential overuse of kernel memory

```
    commit 23954cb078febfc63a755301fe77e06bccdb4d2a upstream.

    In routine wpa_supplicant_ioctl(), the user-controlled p->length is
    checked to be at least the size of struct ieee_param size, but the code
    does not detect the case where p->length is greater than the size
    of the struct, thus a malicious user could be wasting kernel memory.
    Fixes commit 554c0a3abf216 ("staging: Add rtl8723bs sdio wifi driver").

    Reported by: Pietro Oliva <pietroliva@gmail.com>
    Cc: Pietro Oliva <pietroliva@gmail.com>
    Cc: Stable <stable@vger.kernel.org>
    Fixes: 554c0a3abf216 ("staging: Add rtl8723bs sdio wifi driver").
    Signed-off-by: Larry Finger <Larry.Finger@lwfinger.net>
    Link: https://lore.kernel.org/r/20200210180235.21691-5-Larry.Finger@lwfinger.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 238f71c70ff3ef8ad7f18c571a6bb7ad511ad284
Author: Larry Finger <Larry.Finger@lwfinger.net>
Date:   Mon Feb 10 12:02:31 2020 -0600

    staging: rtl8723bs: Fix potential security hole

    commit ac33597c0c0d1d819dccfe001bcd0acef7107e7c upstream.

    In routine rtw_hostapd_ioctl(), the user-controlled p->length is assumed
    to be at least the size of struct ieee_param size, but this assumption is
    never checked. This could result in out-of-bounds read/write on kernel
    heap in case a p->length less than the size of struct ieee_param is
    specified by the user. If p->length is allowed to be greater than the size
    of the struct, then a malicious user could be wasting kernel memory.
    Fixes commit 554c0a3abf216 ("0taging: Add rtl8723bs sdio wifi driver").

    Reported by: Pietro Oliva <pietroliva@gmail.com>
    Cc: Pietro Oliva <pietroliva@gmail.com>
    Cc: Stable <stable@vger.kernel.org>
    Fixes 554c0a3abf216 ("0taging: Add rtl8723bs sdio wifi driver").
    Signed-off-by: Larry Finger <Larry.Finger@lwfinger.net>
    Link: https://lore.kernel.org/r/20200210180235.21691-3-Larry.Finger@lwfinger.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b8cfb9547b10753494db327800a46a43d9eed04a
Author: Larry Finger <Larry.Finger@lwfinger.net>
Date:   Mon Feb 10 12:02:32 2020 -0600

    staging: rtl8188eu: Fix potential overuse of kernel memory

    commit 4ddf8ab8d15ddbc52eefb44eb64e38466ce1f70f upstream.

    In routine wpa_supplicant_ioctl(), the user-controlled p->length is
    checked to be at least the size of struct ieee_param size, but the code
    does not detect the case where p->length is greater than the size
    of the struct, thus a malicious user could be wasting kernel memory.
    Fixes commit a2c60d42d97c ("Add files for new driver - part 16").

    Reported by: Pietro Oliva <pietroliva@gmail.com>
    Cc: Pietro Oliva <pietroliva@gmail.com>
    Cc: Stable <stable@vger.kernel.org>
    Fixes commit a2c60d42d97c ("Add files for new driver - part 16").
    Signed-off-by: Larry Finger <Larry.Finger@lwfinger.net>
    Link: https://lore.kernel.org/r/20200210180235.21691-4-Larry.Finger@lwfinger.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e4f246e6381a519fc7d3fe5f959c690962861e74
Author: Larry Finger <Larry.Finger@lwfinger.net>
```

```
Date:   Mon Feb 10 12:02:30 2020 -0600

    staging: rtl8188eu: Fix potential security hole

    commit 499c405b2b80bb3a04425ba3541d20305e014d3e upstream.

    In routine rtw_hostapd_ioctl(), the user-controlled p->length is assumed
    to be at least the size of struct ieee_param size, but this assumption is
    never checked. This could result in out-of-bounds read/write on kernel
    heap in case a p->length less than the size of struct ieee_param is
    specified by the user. If p->length is allowed to be greater than the size
    of the struct, then a malicious user could be wasting kernel memory.
    Fixes commit a2c60d42d97c ("Add files for new driver - part 16").

    Reported by: Pietro Oliva <pietroliva@gmail.com>
    Cc: Pietro Oliva <pietroliva@gmail.com>
    Cc: Stable <stable@vger.kernel.org>
    Fixes: a2c60d42d97c ("staging: r8188eu: Add files for new driver - part 16")
    Signed-off-by: Larry Finger <Larry.Finger@lwfinger.net>
    Link: https://lore.kernel.org/r/20200210180235.21691-2-Larry.Finger@lwfinger.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b7dbfcebd3cfc669fb5fcda26701b81911b2bfce
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Sun Feb 9 21:12:02 2020 -0800

    scsi: Revert "target/core: Inline transport_lun_remove_cmd()"

    commit c14335ebb92a98646ddbf447e6cacc66de5269ad upstream.

    Commit 83f85b8ec305 postponed the percpu_ref_put(&se_cmd->se_lun->lun_ref)
    call from command completion to the time when the final command reference
    is dropped. That approach is not compatible with the iSCSI target driver
    because the iSCSI target driver keeps the command with the highest stat_sn
    after it has completed until the next command is received (see also
    iscsit_ack_from_expstatsn()). Fix this regression by reverting commit
    83f85b8ec305.

    Fixes: 83f85b8ec305 ("scsi: target/core: Inline transport_lun_remove_cmd()")
    Cc: Pavel Zakharov <pavel.zakharov@delphix.com>
    Cc: Mike Christie <mchristi@redhat.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200210051202.12934-1-bvanassche@acm.org
    Reported-by: Pavel Zakharov <pavel.zakharov@delphix.com>
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Signed-off-by: Martin K. Petersen <martin.petersen@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a471638e6bb6d77128c64dc9f3d4cf84c058c2b8
Author: Colin Ian King <colin.king@canonical.com>
Date:   Mon Feb 10 09:51:39 2020 +0000

    usb: dwc3: debug: fix string position formatting mixup with ret and len

    commit b32196e35bd7bbc8038db1aba1fbf022dc469b6a upstream.

    Currently the string formatting is mixing up the offset of ret and
    len. Re-work the code to use just len, remove ret and use scnprintf
    instead of snprintf and len position accumulation where required.
    Remove the -ve return check since scnprintf never returns a failure
    -ve size. Also break overly long lines to clean up checkpatch
    warnings.

    Addresses-Coverity: ("Unused value")
    Fixes: 1381a5113caf ("usb: dwc3: debug: purge usage of strcat")
```

```
    Signed-off-by: Colin Ian King <colin.king@canonical.com>
    Reviewed-by: Dan Carpenter <dan.carpenter@oracle.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200210095139.328711-1-colin.king@canonical.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 358404d816863f6add2e4793b51e598b6c113007
Author: Anurag Kumar Vulisha <anurag.kumar.vulisha@xilinx.com>
Date:   Mon Jan 27 19:30:46 2020 +0000

    usb: dwc3: gadget: Check for IOC/LST bit in TRB->ctrl fields

    commit 5ee858975b13a9b40db00f456989a689fdbb296c upstream.

    The current code in dwc3_gadget_ep_reclaim_completed_trb() will
    check for IOC/LST bit in the event->status and returns if
    IOC/LST bit is set. This logic doesn't work if multiple TRBs
    are queued per request and the IOC/LST bit is set on the last
    TRB of that request.

    Consider an example where a queued request has multiple queued
    TRBs and IOC/LST bit is set only for the last TRB. In this case,
    the core generates XferComplete/XferInProgress events only for
    the last TRB (since IOC/LST are set only for the last TRB). As
    per the logic in dwc3_gadget_ep_reclaim_completed_trb()
    event->status is checked for IOC/LST bit and returns on the
    first TRB. This leaves the remaining TRBs left unhandled.

    Similarly, if the gadget function enqueues an unaligned request
    with sglist already in it, it should fail the same way, since we
    will append another TRB to something that already uses more than
    one TRB.

    To aviod this, this patch changes the code to check for IOC/LST
    bits in TRB->ctrl instead.

    At a practical level, this patch resolves USB transfer stalls seen
    with adb on dwc3 based HiKey960 after functionfs gadget added
    scatter-gather support around v4.20.

    Cc: Felipe Balbi <balbi@kernel.org>
    Cc: Yang Fei <fei.yang@intel.com>
    Cc: Thinh Nguyen <thinhn@synopsys.com>
    Cc: Tejas Joglekar <tejas.joglekar@synopsys.com>
    Cc: Andrzej Pietrasiewicz <andrzej.p@collabora.com>
    Cc: Jack Pham <jackp@codeaurora.org>
    Cc: Todd Kjos <tkjos@google.com>
    Cc: Greg KH <gregkh@linuxfoundation.org>
    Cc: Linux USB List <linux-usb@vger.kernel.org>
    Cc: stable <stable@vger.kernel.org>
    Tested-by: Tejas Joglekar <tejas.joglekar@synopsys.com>
    Reviewed-by: Thinh Nguyen <thinhn@synopsys.com>
    Signed-off-by: Anurag Kumar Vulisha <anurag.kumar.vulisha@xilinx.com>
    [jstultz: forward ported to mainline, reworded commit log, reworked
     to only check trb->ctrl as suggested by Felipe]
    Signed-off-by: John Stultz <john.stultz@linaro.org>
    Signed-off-by: Felipe Balbi <balbi@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 135948a50b32ae09fbf0304f925a0c1d55a0f4da
Author: Minas Harutyunyan <Minas.Harutyunyan@synopsys.com>
Date:   Tue Jan 21 14:17:07 2020 +0400

    usb: dwc2: Fix SET/CLEAR_FEATURE and GET_STATUS flows
```

```
      commit 9a0d6f7c0a83844baae1d6d85482863d2bf3b7a7 upstream.

      SET/CLEAR_FEATURE for Remote Wakeup allowance not handled correctly.
      GET_STATUS handling provided not correct data on DATA Stage.
      Issue seen when gadget's dr_mode set to "otg" mode and connected
      to MacOS.
      Both are fixed and tested using USBCV Ch.9 tests.

      Signed-off-by: Minas Harutyunyan <hminas@synopsys.com>
      Fixes: fa389a6d7726 ("usb: dwc2: gadget: Add remote_wakeup_allowed flag")
      Tested-by: Jack Mitchell <ml@embed.me.uk>
      Cc: stable@vger.kernel.org
      Signed-off-by: Felipe Balbi <balbi@kernel.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit ee8a6e02d4ec5d56b3f9ff579368d1fea7e60f31
  Author: Hardik Gajjar <hgajjar@de.adit-jv.com>
  Date:   Thu Feb 6 12:49:23 2020 +0100

      USB: hub: Fix the broken detection of USB3 device in SMSC hub

      commit 1208f9e1d758c991b0a46a1bd60c616b906bbe27 upstream.

      Renesas R-Car H3ULCB + Kingfisher Infotainment Board is either not able
      to detect the USB3.0 mass storage devices or is detecting those as
      USB2.0 high speed devices.

      The explanation given by Renesas is that, due to a HW issue, the XHCI
      driver does not wake up after going to sleep on connecting a USB3.0
      device.

      In order to mitigate that, disable the auto-suspend feature
      specifically for SMSC hubs from hub_probe() function, as a quirk.

      Renesas Kingfisher Infotainment Board has two USB3.0 ports (CN2) which
      are connected via USB5534B 4-port SuperSpeed/Hi-Speed, low-power,
      configurable hub controller.

      [1] SanDisk USB 3.0 device detected as USB-2.0 before the patch
       [   74.036390] usb 5-1.1: new high-speed USB device number 4 using xhci-hcd
       [   74.061598] usb 5-1.1: New USB device found, idVendor=0781, idProduct=5581,
  bcdDevice= 1.00
       [   74.069976] usb 5-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
       [   74.077303] usb 5-1.1: Product: Ultra
       [   74.080980] usb 5-1.1: Manufacturer: SanDisk
       [   74.085263] usb 5-1.1: SerialNumber: 4C530001110208116550

      [2] SanDisk USB 3.0 device detected as USB-3.0 after the patch
       [   34.565078] usb 6-1.1: new SuperSpeed Gen 1 USB device number 3 using xhci-hcd
       [   34.588719] usb 6-1.1: New USB device found, idVendor=0781, idProduct=5581,
  bcdDevice= 1.00
       [   34.597098] usb 6-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
       [   34.604430] usb 6-1.1: Product: Ultra
       [   34.608110] usb 6-1.1: Manufacturer: SanDisk
       [   34.612397] usb 6-1.1: SerialNumber: 4C530001110208116550

      Suggested-by: Alan Stern <stern@rowland.harvard.edu>
      Signed-off-by: Hardik Gajjar <hgajjar@de.adit-jv.com>
      Acked-by: Alan Stern <stern@rowland.harvard.edu>
      Tested-by: Eugeniu Rosca <erosca@de.adit-jv.com>
      Cc: stable <stable@vger.kernel.org>
      Link: https://lore.kernel.org/r/1580989763-32291-1-git-send-email-hgajjar@de.adit-jv.com
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 11e485d8f83fbe465d480dd80951cde9aebff619
```

```
Author: Alan Stern <stern@rowland.harvard.edu>
Date:   Fri Jan 31 10:39:26 2020 -0500

    USB: hub: Don't record a connect-change event during reset-resume

    commit 8099f58f1ecddf4f374f4828a3dff8397c7cbd74 upstream.

    Paul Zimmerman reports that his USB Bluetooth adapter sometimes
    crashes following system resume, when it receives a
    Get-Device-Descriptor request while it is busy doing something else.

    Such a request was added by commit a4f55d8b8c14 ("usb: hub: Check
    device descriptor before resusciation").  It gets sent when the hub
    driver's work thread checks whether a connect-change event on an
    enabled port really indicates a new device has been connected, as
    opposed to an old device momentarily disconnecting and then
    reconnecting (which can happen with xHCI host controllers, since they
    automatically enable connected ports).

    The same kind of thing occurs when a port's power session is lost
    during system suspend.  When the system wakes up it sees a
    connect-change event on the port, and if the child device's
    persist_enabled flag was set then hub_activate() sets the device's
    reset_resume flag as well as the port's bit in hub->change_bits.  The
    reset-resume code then takes responsibility for checking that the same
    device is still attached to the port, and it does this as part of the
    device's resume pathway.  By the time the hub driver's work thread
    starts up again, the device has already been fully reinitialized and
    is busy doing its own thing.  There's no need for the work thread to
    do the same check a second time, and in fact this unnecessary check is
    what caused the problem that Paul observed.

    Note that performing the unnecessary check is not actually a bug.
    Devices are supposed to be able to send descriptors back to the host
    even when they are busy doing something else.  The underlying cause of
    Paul's problem lies in his Bluetooth adapter.  Nevertheless, we
    shouldn't perform the same check twice in a row -- and as a nice side
    benefit, removing the extra check allows the Bluetooth adapter to work
    more reliably.

    The work thread performs its check when it sees that the port's bit is
    set in hub->change_bits.  In this situation that bit is interpreted as
    though a connect-change event had occurred on the port _after_ the
    reset-resume, which is not what actually happened.

    One possible fix would be to make the reset-resume code clear the
    port's bit in hub->change_bits.  But it seems simpler to just avoid
    setting the bit during hub_activate() in the first place.  That's what
    this patch does.

    (Proving that the patch is correct when CONFIG_PM is disabled requires
    a little thought.  In that setting hub_activate() will be called only
    for initialization and resets, since there won't be any resumes or
    reset-resumes.  During initialization and hub resets the hub doesn't
    have any child devices, and so this code path never gets executed.)

    Reported-and-tested-by: Paul Zimmerman <pauldzim@gmail.com>
    Signed-off-by: Alan Stern <stern@rowland.harvard.edu>
    Link: https://marc.info/?t=157949360700001&r=1&w=2
    CC: David Heinzelmann <heinzelmann.david@gmail.com>
    CC: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/Pine.LNX.4.44L0.2001311037460.1577-
100000@iolanthe.rowland.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 8cce7f36c39e84554707aeeeb8289b274c1a9b06
Author: Richard Dodd <richard.o.dodd@gmail.com>
Date:   Wed Feb 12 14:22:18 2020 +0000

    USB: Fix novation SourceControl XL after suspend

    commit b692056db8ecc7f452b934f016c17348282b7699 upstream.

    Currently, the SourceControl will stay in power-down mode after resuming
    from suspend. This patch resets the device after suspend to power it up.

    Signed-off-by: Richard Dodd <richard.o.dodd@gmail.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200212142220.36892-1-richard.o.dodd@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aaeddd61d8ca3d5703e206bdd101c8ea611e5a2e
Author: EJ Hsu <ejh@nvidia.com>
Date:   Thu Jan 30 01:25:06 2020 -0800

    usb: uas: fix a plug & unplug racing

    commit 3e99862c05a9caa5a27969f41566b428696f5a9a upstream.

    When a uas disk is plugged into an external hub, uas_probe()
    will be called by the hub thread to do the probe. It will
    first create a SCSI host and then do the scan for this host.
    During the scan, it will probe the LUN using SCSI INQUERY command
    which will be packed in the URB and submitted to uas disk.

    There might be a chance that this external hub with uas disk
    attached is unplugged during the scan. In this case, uas driver
    will fail to submit the URB (due to the NOTATTACHED state of uas
    device) and try to put this SCSI command back to request queue
    waiting for next chance to run.

    In normal case, this cycle will terminate when hub thread gets
    disconnection event and calls into uas_disconnect() accordingly.
    But in this case, uas_disconnect() will not be called because
    hub thread of external hub gets stuck waiting for the completion
    of this SCSI command. A deadlock happened.

    In this fix, uas will call scsi_scan_host() asynchronously to
    avoid the blocking of hub thread.

    Signed-off-by: EJ Hsu <ejh@nvidia.com>
    Acked-by: Oliver Neukum <oneukum@suse.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200130092506.102760-1-ejh@nvidia.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ac2c127174c544cf9475a116d8aadaa45755bff3
Author: Johan Hovold <johan@kernel.org>
Date:   Mon Feb 3 16:38:29 2020 +0100

    USB: quirks: blacklist duplicate ep on Sound Devices USBPre2

    commit bdd1b147b8026df0e4260b387026b251d888ed01 upstream.

    This device has a broken vendor-specific altsetting for interface 1,
    where endpoint 0x85 is declared as an isochronous endpoint despite being
    used by interface 2 for audio capture.

    Device Descriptor:
      bLength                18
```

```
        bDescriptorType         1
        bcdUSB               2.00
        bDeviceClass          239 Miscellaneous Device
        bDeviceSubClass         2
        bDeviceProtocol         1 Interface Association
        bMaxPacketSize0        64
        idVendor           0x0926
        idProduct          0x0202
        bcdDevice            1.00
        iManufacturer           1 Sound Devices
        iProduct                2 USBPre2
        iSerial                 3 [...]
        bNumConfigurations      1

      [...]

          Interface Descriptor:
            bLength                 9
            bDescriptorType         4
            bInterfaceNumber        1
            bAlternateSetting       3
            bNumEndpoints           2
            bInterfaceClass       255 Vendor Specific Class
            bInterfaceSubClass      0
            bInterfaceProtocol      0
            iInterface              0
            Endpoint Descriptor:
              bLength                   7
              bDescriptorType           5
              bEndpointAddress     0x85  EP 5 IN
              bmAttributes              5
                Transfer Type             Isochronous
                Synch Type                Asynchronous
                Usage Type                Data
              wMaxPacketSize     0x0126  1x 294 bytes
              bInterval                 1

      [...]

          Interface Descriptor:
            bLength                 9
            bDescriptorType         4
            bInterfaceNumber        2
            bAlternateSetting       1
            bNumEndpoints           1
            bInterfaceClass         1 Audio
            bInterfaceSubClass      2 Streaming
            bInterfaceProtocol      0
            iInterface              0
            AudioStreaming Interface Descriptor:
              bLength                 7
              bDescriptorType        36
              bDescriptorSubtype      1 (AS_GENERAL)
              bTerminalLink           4
              bDelay                  1 frames
              wFormatTag         0x0001 PCM
            AudioStreaming Interface Descriptor:
              bLength                26
              bDescriptorType        36
              bDescriptorSubtype      2 (FORMAT_TYPE)
              bFormatType             1 (FORMAT_TYPE_I)
              bNrChannels             2
              bSubframeSize           2
              bBitResolution         16
              bSamFreqType            6 Discrete
```

```
              tSamFreq[ 0]         8000
              tSamFreq[ 1]        16000
              tSamFreq[ 2]        24000
              tSamFreq[ 3]        32000
              tSamFreq[ 4]        44100
              tSamFreq[ 5]        48000
            Endpoint Descriptor:
              bLength                 9
              bDescriptorType         5
              bEndpointAddress     0x85  EP 5 IN
              bmAttributes            5
                Transfer Type            Isochronous
                Synch Type               Asynchronous
                Usage Type               Data
              wMaxPacketSize     0x0126  1x 294 bytes
              bInterval               4
              bRefresh                0
              bSynchAddress           0
              AudioStreaming Endpoint Descriptor:
                bLength                 7
                bDescriptorType        37
                bDescriptorSubtype      1 (EP_GENERAL)
                bmAttributes         0x01
                  Sampling Frequency
                bLockDelayUnits         2 Decoded PCM samples
                wLockDelay         0x0000
```

    Since commit 3e4f8e21c4f2 ("USB: core: fix check for duplicate
    endpoints") USB core ignores any duplicate endpoints found during
    descriptor parsing, but in this case we need to ignore the first
    instance in order to avoid breaking the audio capture interface.

    Fixes: 3e4f8e21c4f2 ("USB: core: fix check for duplicate endpoints")
    Cc: stable <stable@vger.kernel.org>
    Reported-by: edes <edes@gmx.net>
    Tested-by: edes <edes@gmx.net>
    Link: https://lore.kernel.org/r/20200201105829.5682c887@acme7.acmenet
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Link: https://lore.kernel.org/r/20200203153830.26394-3-johan@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0e8fb84c5ca5b4b41242e8c230e84628da3f455f
Author: Johan Hovold <johan@kernel.org>
Date: Mon Feb 3 16:38:28 2020 +0100

    USB: core: add endpoint-blacklist quirk

    commit 73f8bda9b5dc1c69df2bc55c0cbb24461a6391a9 upstream.

    Add a new device quirk that can be used to blacklist endpoints.

    Since commit 3e4f8e21c4f2 ("USB: core: fix check for duplicate
    endpoints") USB core ignores any duplicate endpoints found during
    descriptor parsing.

    In order to handle devices where the first interfaces with duplicate
    endpoints are the ones that should have their endpoints ignored, we need
    to add a blacklist.

    Tested-by: edes <edes@gmx.net>
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Link: https://lore.kernel.org/r/20200203153830.26394-2-johan@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

```
commit e26de1426474637aff39e453956d9d09aa0ecbb5
Author: Mathias Nyman <mathias.nyman@linux.intel.com>
Date:   Tue Feb 11 17:01:58 2020 +0200

    xhci: Fix memory leak when caching protocol extended capability PSI tables - take 2

    commit cf0ee7c60c89641f6e4d1d3c7867fe32b9e30300 upstream.

    xhci driver assumed that xHC controllers have at most one custom
    supported speed table (PSI) for all usb 3.x ports.
    Memory was allocated for one PSI table under the xhci hub structure.

    Turns out this is not the case, some controllers have a separate
    "supported protocol capability" entry with a PSI table for each port.
    This means each usb3 roothub port can in theory support different custom
    speeds.

    To solve this, cache all supported protocol capabilities with their PSI
    tables in an array, and add pointers to the xhci port structure so that
    every port points to its capability entry in the array.

    When creating the SuperSpeedPlus USB Device Capability BOS descriptor
    for the xhci USB 3.1 roothub we for now will use only data from the
    first USB 3.1 capable protocol capability entry in the array.
    This could be improved later, this patch focuses resolving
    the memory leak.

    Reported-by: Paul Menzel <pmenzel@molgen.mpg.de>
    Reported-by: Sajja Venkateswara Rao <VenkateswaraRao.Sajja@amd.com>
    Fixes: 47189098f8be ("xhci: parse xhci protocol speed ID list for usb 3.1 usage")
    Cc: stable <stable@vger.kernel.org> # v4.4+
    Signed-off-by: Mathias Nyman <mathias.nyman@linux.intel.com>
    Tested-by: Marek Szyprowski <m.szyprowski@samsung.com>
    Link: https://lore.kernel.org/r/20200211150158.14475-1-mathias.nyman@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f16d0240575c0bace80e703ad8c897728bed89f5
Author: Mathias Nyman <mathias.nyman@linux.intel.com>
Date:   Mon Feb 10 15:45:53 2020 +0200

    xhci: apply XHCI_PME_STUCK_QUIRK to Intel Comet Lake platforms

    commit a3ae87dce3a5abe0b57c811bab02b2564b574106 upstream.

    Intel Comet Lake based platform require the XHCI_PME_STUCK_QUIRK
    quirk as well. Without this xHC can not enter D3 in runtime suspend.

    Cc: stable@vger.kernel.org
    Signed-off-by: Mathias Nyman <mathias.nyman@linux.intel.com>
    Link: https://lore.kernel.org/r/20200210134553.9144-5-mathias.nyman@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cb887dc599c0a524aed0e03a30b6ed325ed3b267
Author: Mathias Nyman <mathias.nyman@linux.intel.com>
Date:   Mon Feb 10 15:45:52 2020 +0200

    xhci: fix runtime pm enabling for quirky Intel hosts

    commit 024d411e9c5d49eb96c825af52a3ce2682895676 upstream.

    Intel hosts that need the XHCI_PME_STUCK_QUIRK flag should enable
    runtime pm by calling xhci_pme_acpi_rtd3_enable() before
    usb_hcd_pci_probe() calls pci_dev_run_wake().
    Otherwise usage count for the device won't be decreased, and runtime
    suspend is prevented.
```

```
    usb_hcd_pci_probe() only decreases the usage count if device can
    generate run-time wake-up events, i.e. when pci_dev_run_wake()
    returns true.

    This issue was exposed by pci_dev_run_wake() change in
    commit 8feaec33b986 ("PCI / PM: Always check PME wakeup capability for
    runtime wakeup support")
    and should be backported to kernels with that change

    Cc: <stable@vger.kernel.org> # 4.13+
    Signed-off-by: Mathias Nyman <mathias.nyman@linux.intel.com>
    Link: https://lore.kernel.org/r/20200210134553.9144-4-mathias.nyman@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cb5a92d20edfca7302941b88d49f7ff576054689
Author: Mathias Nyman <mathias.nyman@linux.intel.com>
Date:   Mon Feb 10 15:45:50 2020 +0200

    xhci: Force Maximum Packet size for Full-speed bulk devices to valid range.

    commit f148b9f402ef002b57bcff3964d45abc8ffb6c3f upstream.

    A Full-speed bulk USB audio device (DJ-Tech CTRL) with a invalid Maximum
    Packet Size of 4 causes a xHC "Parameter Error" at enumeration.

    This is because valid Maximum packet sizes for Full-speed bulk endpoints
    are 8, 16, 32 and 64 bytes. Hosts are not required to support other values
    than these. See usb 2 specs section 5.8.3 for details.

    The device starts working after forcing the maximum packet size to 8.
    This is most likely the case with other devices as well, so force the
    maximum packet size to a valid range.

    Cc: stable@vger.kernel.org
    Reported-by: Rene D Obermueller <cmdrrdo@gmail.com>
    Signed-off-by: Mathias Nyman <mathias.nyman@linux.intel.com>
    Link: https://lore.kernel.org/r/20200210134553.9144-2-mathias.nyman@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e9301ae44d39abeb3a3c6094c1ad71bc097e15f4
Author: Malcolm Priestley <tvboxspy@gmail.com>
Date:   Tue Feb 4 19:34:02 2020 +0000

    staging: vt6656: fix sign of rx_dbm to bb_pre_ed_rssi.

    commit 93134df520f23f4e9998c425b8987edca7016817 upstream.

    bb_pre_ed_rssi is an u8 rx_dm always returns negative signed
    values add minus operator to always yield positive.

    fixes issue where rx sensitivity is always set to maximum because
    the unsigned numbers were always greater then 100.

    Fixes: 63b9907f58f1 ("staging: vt6656: mac80211 conversion: create rx function.")
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Malcolm Priestley <tvboxspy@gmail.com>
    Link: https://lore.kernel.org/r/aceac98c-6e69-3ce1-dfec-2bf27b980221@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aa5122eaa89d875bec5f27d3be1fecd5504ca507
Author: Suren Baghdasaryan <surenb@google.com>
Date:   Mon Jan 27 15:56:16 2020 -0800

    staging: android: ashmem: Disallow ashmem memory from being remapped
```

```
    commit 6d67b0290b4b84c477e6a2fc6e005e174d3c7786 upstream.

    When ashmem file is mmapped, the resulting vma->vm_file points to the
    backing shmem file with the generic fops that do not check ashmem
    permissions like fops of ashmem do. If an mremap is done on the ashmem
    region, then the permission checks will be skipped. Fix that by disallowing
    mapping operation on the backing shmem file.

    Reported-by: Jann Horn <jannh@google.com>
    Signed-off-by: Suren Baghdasaryan <surenb@google.com>
    Cc: stable <stable@vger.kernel.org> # 4.4,4.9,4.14,4.18,5.4
    Signed-off-by: Todd Kjos <tkjos@google.com>
    Reviewed-by: Joel Fernandes (Google) <joel@joelfernandes.org>
    Link: https://lore.kernel.org/r/20200127235616.48920-1-tkjos@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit be61d458e9204e54030820a685649219a9b5c6a5
Author: Eric Dumazet <edumazet@google.com>
Date:   Mon Feb 10 11:07:21 2020 -0800

    vt: vt_ioctl: fix race in VT_RESIZEX

    commit 6cd1ed50efd88261298577cd92a14f2768eddeeb upstream.

    We need to make sure vc_cons[i].d is not NULL after grabbing
    console_lock(), or risk a crash.

    general protection fault, probably for non-canonical address 0xdfffc0000000068: 0000
[#1] PREEMPT SMP KASAN
    KASAN: null-ptr-deref in range [0x0000000000000340-0x0000000000000347]
    CPU: 1 PID: 19462 Comm: syz-executor.5 Not tainted 5.5.0-syzkaller #0
    Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google
01/01/2011
    RIP: 0010:vt_ioctl+0x1f96/0x26d0 drivers/tty/vt/vt_ioctl.c:883
    Code: 74 41 e8 bd a6 84 fd 48 89 d8 48 c1 e8 03 42 80 3c 28 00 0f 85 e4 04 00 00 48 8b
03 48 8d b8 40 03 00 00 48 89 fa 48 c1 ea 03 <42> 0f b6 14 2a 84 d2 74 09 80 fa 03 0f 8e b1
05 00 00 44 89 b8 40
    RSP: 0018:ffffc900086d7bb0 EFLAGS: 00010202
    RAX: 0000000000000000 RBX: ffffffff8c34ee88 RCX: ffffc9001415c000
    RDX: 0000000000000068 RSI: ffffffff83f0e6e3 RDI: 0000000000000340
    RBP: ffffc900086d7cd0 R08: ffff888054ce0100 R09: fffffbfff16a2f6d
    R10: ffff888054ce0998 R11: ffff888054ce0100 R12: 000000000000001d
    R13: dfffc0000000000 R14: 1ffff920010daf79 R15: 00000000000ff7f
    FS:  00007f7d13c12700(0000) GS:ffff8880ae900000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007ffd477e3c38 CR3: 0000000095d0a000 CR4: 00000000001406e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
    Call Trace:
     tty_ioctl+0xa37/0x14f0 drivers/tty/tty_io.c:2660
     vfs_ioctl fs/ioctl.c:47 [inline]
     ksys_ioctl+0x123/0x180 fs/ioctl.c:763
     __do_sys_ioctl fs/ioctl.c:772 [inline]
     __se_sys_ioctl fs/ioctl.c:770 [inline]
     __x64_sys_ioctl+0x73/0xb0 fs/ioctl.c:770
     do_syscall_64+0xfa/0x790 arch/x86/entry/common.c:294
     entry_SYSCALL_64_after_hwframe+0x49/0xbe
    RIP: 0033:0x45b399
    Code: ad b6 fb ff c3 66 2e 0f 1f 84 00 00 00 00 00 66 90 48 89 f8 48 89 f7 48 89 d6 48
89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 0f 83 7b b6 fb ff c3 66 2e
0f 1f 84 00 00 00 00
    RSP: 002b:00007f7d13c11c78 EFLAGS: 00000246 ORIG_RAX: 0000000000000010
    RAX: ffffffffffffffda RBX: 00007f7d13c126d4 RCX: 000000000045b399
    RDX: 0000000020000080 RSI: 000000000000560a RDI: 0000000000000003
```

```
    RBP: 000000000075bf20 R08: 0000000000000000 R09: 0000000000000000
    R10: 0000000000000000 R11: 0000000000000246 R12: 00000000ffffffff
    R13: 0000000000000666 R14: 00000000004c7f04 R15: 000000000075bf2c
    Modules linked in:
    ---[ end trace 80970faf7a67eb77 ]---
    RIP: 0010:vt_ioctl+0x1f96/0x26d0 drivers/tty/vt/vt_ioctl.c:883
    Code: 74 41 e8 bd a6 84 fd 48 89 d8 48 c1 e8 03 42 80 3c 28 00 0f 85 e4 04 00 00 48 8b
03 48 8d b8 40 03 00 00 48 89 fa 48 c1 ea 03 <42> 0f b6 14 2a 84 d2 74 09 80 fa 03 0f 8e b1
05 00 00 44 89 b8 40
    RSP: 0018:ffffc900086d7bb0 EFLAGS: 00010202
    RAX: 0000000000000000 RBX: ffffffff8c34ee88 RCX: ffffc9001415c000
    RDX: 0000000000000068 RSI: ffffffff83f0e6e3 RDI: 0000000000000340
    RBP: ffffc900086d7cd0 R08: ffff888054ce0100 R09: fffffbfff16a2f6d
    R10: ffff888054ce0998 R11: ffff888054ce0100 R12: 000000000000001d
    R13: dfffffc0000000000 R14: 1ffff920010daf79 R15: 000000000000ff7f
    FS:  00007f7d13c12700(0000) GS:ffff8880ae900000(0000) knlGS:0000000000000000
    CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
    CR2: 00007ffd477e3c38 CR3: 0000000095d0a000 CR4: 00000000001406e0
    DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
    DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400

    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Signed-off-by: Eric Dumazet <edumazet@google.com>
    Cc: stable <stable@vger.kernel.org>
    Reported-by: syzbot <syzkaller@googlegroups.com>
    Link: https://lore.kernel.org/r/20200210190721.200418-1-edumazet@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3812fcce4f40b9b74ee359cea17e9a37a9fc5c1b
Author: Jiri Slaby <jslaby@suse.cz>
Date:   Mon Feb 10 09:11:30 2020 +0100

    vt: selection, handle pending signals in paste_selection

    commit 687bff0cd08f790d540cfb7b2349f0d876cdddec upstream.

    When pasting a selection to a vt, the task is set as INTERRUPTIBLE while
    waiting for a tty to unthrottle. But signals are not handled at all.
    Normally, this is not a problem as tty_ldisc_receive_buf receives all
    the goods and a user has no reason to interrupt the task.

    There are two scenarios where this matters:
    1) when the tty is throttled and a signal is sent to the process, it
       spins on a CPU until the tty is unthrottled. schedule() does not
       really echedule, but returns immediately, of course.
    2) when the sel_buffer becomes invalid, KASAN prevents any reads from it
       and the loop simply does not proceed and spins forever (causing the
       tty to throttle, but the code never sleeps, the same as above). This
       sometimes happens as there is a race in the sel_buffer handling code.

    So add signal handling to this ioctl (TIOCL_PASTESEL) and return -EINTR
    in case a signal is pending.

    Signed-off-by: Jiri Slaby <jslaby@suse.cz>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200210081131.23572-1-jslaby@suse.cz
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4816ab6f0f196eba5fc9b375b1324c890ace9390
Author: Nicolas Pitre <nico@fluxnic.net>
Date:   Tue Jan 28 12:50:33 2020 -0500

    vt: fix scrollback flushing on background consoles

    commit 3f4ef485be9d54040b695f32ec76d0f1ea50bbf3 upstream.
```

```
    Commit a6dbe4427559 ("vt: perform safe console erase in the right
    order") provided fixes to an earlier commit by gathering all console
    scrollback flushing operations in a function of its own. This includes
    the invocation of vc_sw->con_switch() as previously done through a
    update_screen() call. That commit failed to carry over the
    con_is_visible() conditional though, as well as cursor handling, which
    caused problems when "\e[3J" was written to a background console.

    One could argue for preserving the call to update_screen(). However
    this does far more than we need, and it is best to remove scrollback
    assumptions from it. Instead let's gather the minimum needed to actually
    perform scrollback flushing properly in that one place.

    While at it, let's document the vc_sw->con_switch() side effect being
    relied upon.

    Signed-off-by: Nicolas Pitre <nico@fluxnic.net>
    Reported-and-tested-by: Lukas Wunner <lukas@wunner.de>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/nycvar.YSQ.7.76.2001281205560.1655@knanqh.ubzr
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7e4567568586e4efea3bd5ab3b3ab0488f14b5c9
Author: Johan Hovold <johan@kernel.org>
Date:   Thu Feb 6 12:18:19 2020 +0100

    USB: serial: ch341: fix receiver regression

    commit 7c3d02285ad558691f27fde760bcd841baa27eab upstream.

    While assumed not to make a difference, not using the factor-2 prescaler
    makes the receiver more susceptible to errors.

    Specifically, there have been reports of problems with devices that
    cannot generate a 115200 rate with a smaller error than 2.1% (e.g.
    117647 bps). But this can also be reproduced with a low-speed RS232
    tranceiver at 115200 when the input rate matches the nominal rate.

    So whenever possible, enable the factor-2 prescaler and halve the
    divisor in order to use settings closer to that of the previous
    algorithm.

    Fixes: 35714565089e ("USB: serial: ch341: reimplement line-speed handling")
    Cc: stable <stable@vger.kernel.org>     # 5.5
    Reported-by: Jakub Nantl <jn@forever.cz>
    Tested-by: Jakub Nantl <jn@forever.cz>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3604f164a31e6acb38ef2fa7588db18bbcd644fe
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Fri Feb 21 12:43:35 2020 -0800

    floppy: check FDC index for errors before assigning it

    commit 2e90ca68b0d2f5548804f22f0dd61145516171e3 upstream.

    Jordy Zomer reported a KASAN out-of-bounds read in the floppy driver in
    wait_til_ready().

    Which on the face of it can't happen, since as Willy Tarreau points out,
    the function does no particular memory access.  Except through the FDCS
    macro, which just indexes a static allocation through teh current fdc,
```

```
    which is always checked against N_FDC.

    Except the checking happens after we've already assigned the value.

    The floppy driver is a disgrace (a lot of it going back to my original
    horrd "design"), and has no real maintainer.  Nobody has the hardware,
    and nobody really cares.  But it still gets used in virtual environment
    because it's one of those things that everybody supports.

    The whole thing should be re-written, or at least parts of it should be
    seriously cleaned up.  The 'current fdc' index, which is used by the
    FDCS macro, and which is often shadowed by a local 'fdc' variable, is a
    prime example of how not to write code.

    But because nobody has the hardware or the motivation, let's just fix up
    the immediate problem with a nasty band-aid: test the fdc index before
    actually assigning it to the static 'fdc' variable.

    Reported-by: Jordy Zomer <jordy@simplyhacker.com>
    Cc: Willy Tarreau <w@1wt.eu>
    Cc: Dan Carpenter <dan.carpenter@oracle.com>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a54118bf4efd3d0bfaa18bec71d4f498f7cb5607
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Feb 14 08:11:48 2020 -0800

    USB: misc: iowarrior: add support for the 100 device

    commit bab5417f5f0118ce914bc5b2f8381e959e891155 upstream.

    Add a new device id for the 100 devie.  It has 4 interfaces like the 28
    and 28L devices but a larger endpoint so more I/O pins.

    Cc: Christoph Jung <jung@codemercs.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200214161148.GA3963518@kroah.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1b3425389eb0bd21a6825a29413becce781d659b
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Tue Feb 11 20:04:22 2020 -0800

    USB: misc: iowarrior: add support for the 28 and 28L devices

    commit 5f6f8da2d7b5a431d3f391d0d73ace8edfb42af7 upstream.

    Add new device ids for the 28 and 28L devices.  These have 4 interfaces
    instead of 2, but the driver binds the same, so the driver changes are
    minimal.

    Cc: Christoph Jung <jung@codemercs.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200212040422.2991-2-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 36e30081694351f6e6b988f0f76a81be41a85956
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Tue Feb 11 20:04:21 2020 -0800

    USB: misc: iowarrior: add support for 2 OEMed devices

    commit 461d8deb26a7d70254bc0391feb4fd8a95e674e8 upstream.
```

```
    Add support for two OEM devices that are identical to existing
    IO-Warrior devices, except for the USB device id.

    Cc: Christoph Jung <jung@codemercs.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200212040422.2991-1-gregkh@linuxfoundation.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c9301092829f7a6a60665acd3fa15c418833c877
Author: Mika Westerberg <mika.westerberg@linux.intel.com>
Date:   Thu Feb 13 12:56:04 2020 +0300

    thunderbolt: Prevent crash if non-active NVMem file is read

    commit 03cd45d2e219301880cabc357e3cf478a500080f upstream.

    The driver does not populate .reg_read callback for the non-active NVMem
    because the file is supposed to be write-only. However, it turns out
    NVMem subsystem does not yet support this and expects that the .reg_read
    callback is provided. If user reads the binary attribute it triggers
    NULL pointer dereference like this one:

      BUG: kernel NULL pointer dereference, address: 0000000000000000
      ...
      Call Trace:
       bin_attr_nvmem_read+0x64/0x80
       kernfs_fop_read+0xa7/0x180
       vfs_read+0xbd/0x170
       ksys_read+0x5a/0xd0
       do_syscall_64+0x43/0x150
       entry_SYSCALL_64_after_hwframe+0x44/0xa9

    Fix this in the driver by providing .reg_read callback that always
    returns an error.

    Reported-by: Nicholas Johnson <nicholas.johnson-opensource@outlook.com.au>
    Fixes: e6b245ccd524 ("thunderbolt: Add support for host and device NVM firmware
upgrade")
    Signed-off-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20200213095604.1074-1-mika.westerberg@linux.intel.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c22c7e71e6194ab6135ac8351b0fd823b905bc4d
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Thu Feb 13 10:47:30 2020 -0500

    btrfs: handle logged extent failure properly

    commit bd727173e4432fe6cb70ba108dc1f3602c5409d7 upstream.

    If we're allocating a logged extent we attempt to insert an extent
    record for the file extent directly.  We increase
    space_info->bytes_reserved, because the extent entry addition will call
    btrfs_update_block_group(), which will convert the ->bytes_reserved to
    ->bytes_used.  However if we fail at any point while inserting the
    extent entry we will bail and leave space on ->bytes_reserved, which
    will trigger a WARN_ON() on umount.  Fix this by pinning the space if we
    fail to insert, which is what happens in every other failure case that
    involves adding the extent entry.

    CC: stable@vger.kernel.org # 5.4+
    Reviewed-by: Johannes Thumshirn <johannes.thumshirn@wdc.com>
    Reviewed-by: Nikolay Borisov <nborisov@suse.com>
```

```
    Reviewed-by: Qu Wenruo <wqu@suse.com>
    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 350014475f6f977ad8f7c75f442ed80fe18495a3
Author: Josef Bacik <josef@toxicpanda.com>
Date:   Fri Jan 17 09:02:20 2020 -0500

    btrfs: don't set path->leave_spinning for truncate

    commit 52e29e331070cd7d52a64cbf1b0958212a340e28 upstream.

    The only time we actually leave the path spinning is if we're truncating
    a small amount and don't actually free an extent, which is not a common
    occurrence.  We have to set the path blocking in order to add the
    delayed ref anyway, so the first extent we find we set the path to
    blocking and stay blocking for the duration of the operation.  With the
    upcoming file extent map stuff there will be another case that we have
    to have the path blocking, so just swap to blocking always.

    Note: this patch also fixes a warning after 28553fa992cb ("Btrfs: fix
    race between shrinking truncate and fiemap") got merged that inserts
    extent locks around truncation so the path must not leave spinning locks
    after btrfs_search_slot.

      [70.794783] BUG: sleeping function called from invalid context at mm/slab.h:565
      [70.794834] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1141, name: rsync
      [70.794863] 5 locks held by rsync/1141:
      [70.794876]  #0: ffff888417b9c408 (sb_writers#17){.+.+}, at: mnt_want_write+0x20/0x50
      [70.795030]  #1: ffff888428de28e8 (&type->i_mutex_dir_key#13/1){+.+.}, at:
lock_rename+0xf1/0x100
      [70.795051]  #2: ffff888417b9c608 (sb_internal#2){.+.+}, at:
start_transaction+0x394/0x560
      [70.795124]  #3: ffff888403081768 (btrfs-fs-01){++++}, at:
btrfs_try_tree_write_lock+0x2f/0x160
      [70.795203]  #4: ffff888403086568 (btrfs-fs-00){++++}, at:
btrfs_try_tree_write_lock+0x2f/0x160
      [70.795222] CPU: 5 PID: 1141 Comm: rsync Not tainted 5.6.0-rc2-backup+ #2
      [70.795362] Call Trace:
      [70.795374]  dump_stack+0x71/0xa0
      [70.795445]  ___might_sleep.part.96.cold.106+0xa6/0xb6
      [70.795459]  kmem_cache_alloc+0x1d3/0x290
      [70.795471]  alloc_extent_state+0x22/0x1c0
      [70.795544]  __clear_extent_bit+0x3ba/0x580
      [70.795557]  ? _raw_spin_unlock_irq+0x24/0x30
      [70.795569]  btrfs_truncate_inode_items+0x339/0xe50
      [70.795647]  btrfs_evict_inode+0x269/0x540
      [70.795659]  ? dput.part.38+0x29/0x460
      [70.795671]  evict+0xcd/0x190
      [70.795682]  __dentry_kill+0xd6/0x180
      [70.795754]  dput.part.38+0x2ad/0x460
      [70.795765]  do_renameat2+0x3cb/0x540
      [70.795777]  __x64_sys_rename+0x1c/0x20

    Reported-by: Dave Jones <davej@codemonkey.org.uk>
    Fixes: 28553fa992cb ("Btrfs: fix race between shrinking truncate and fiemap")
    CC: stable@vger.kernel.org # 4.4+
    Reviewed-by: Filipe Manana <fdmanana@suse.com>
    Signed-off-by: Josef Bacik <josef@toxicpanda.com>
    Reviewed-by: David Sterba <dsterba@suse.com>
    [ add note ]
    Signed-off-by: David Sterba <dsterba@suse.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit d53be3908fbe38e9888c20db8b7e3cc4e6a0dfd3
Author: Filipe Manana <fdmanana@suse.com>
Date:   Fri Feb 7 12:23:09 2020 +0000

    Btrfs: fix race between shrinking truncate and fiemap

    commit 28553fa992cb28be6a65566681aac6cafabb4f2d upstream.

    When there is a fiemap executing in parallel with a shrinking truncate
    we can end up in a situation where we have extent maps for which we no
    longer have corresponding file extent items. This is generally harmless
    and at the moment the only consequences are missing file extent items
    representing holes after we expand the file size again after the
    truncate operation removed the prealloc extent items, and stale
    information for future fiemap calls (reporting extents that no longer
    exist or may have been reallocated to other files for example).

    Consider the following example:

    1) Our inode has a size of 128KiB, one 128KiB extent at file offset 0
       and a 1MiB prealloc extent at file offset 128KiB;

    2) Task A starts doing a shrinking truncate of our inode to reduce it to
       a size of 64KiB. Before it searches the subvolume tree for file
       extent items to delete, it drops all the extent maps in the range
       from 64KiB to (u64)-1 by calling btrfs_drop_extent_cache();

    3) Task B starts doing a fiemap against our inode. When looking up for
       the inode's extent maps in the range from 128KiB to (u64)-1, it
       doesn't find any in the inode's extent map tree, since they were
       removed by task A.  Because it didn't find any in the extent map
       tree, it scans the inode's subvolume tree for file extent items, and
       it finds the 1MiB prealloc extent at file offset 128KiB, then it
       creates an extent map based on that file extent item and adds it to
       inode's extent map tree (this ends up being done by
       btrfs_get_extent() <- btrfs_get_extent_fiemap() <-
       get_extent_skip_holes());

    4) Task A then drops the prealloc extent at file offset 128KiB and
       shrinks the 128KiB extent file offset 0 to a length of 64KiB. The
       truncation operation finishes and we end up with an extent map
       representing a 1MiB prealloc extent at file offset 128KiB, despite we
       don't have any more that extent;

    After this the two types of problems we have are:

    1) Future calls to fiemap always report that a 1MiB prealloc extent
       exists at file offset 128KiB. This is stale information, no longer
       correct;

    2) If the size of the file is increased, by a truncate operation that
       increases the file size or by a write into a file offset > 64KiB for
       example, we end up not inserting file extent items to represent holes
       for any range between 128KiB and 128KiB + 1MiB, since the hole
       expansion function, btrfs_cont_expand() will skip hole insertion for
       any range for which an extent map exists that represents a prealloc
       extent. This causes fsck to complain about missing file extent items
       when not using the NO_HOLES feature.

    The second issue could be often triggered by test case generic/561 from
    fstests, which runs fsstress and duperemove in parallel, and duperemove
    does frequent fiemap calls.

    Essentially the problems happens because fiemap does not acquire the
```

```
    inode's lock while truncate does, and fiemap locks the file range in the
    inode's iotree while truncate does not. So fix the issue by making
    btrfs_truncate_inode_items() lock the file range from the new file size
    to (u64)-1, so that it serializes with fiemap.

    CC: stable@vger.kernel.org # 4.4+
    Reviewed-by: Josef Bacik <josef@toxicpanda.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 3165e2ec1040d08ded110b5146d02cafbcb34aed
Author: Wenwen Wang <wenwen@cs.uga.edu>
Date:   Tue Aug 20 00:33:54 2019 -0500

    ecryptfs: fix a memory leak bug in ecryptfs_init_messaging()

    commit b4a81b87a4cfe2bb26a4a943b748d96a43ef20e8 upstream.

    In ecryptfs_init_messaging(), if the allocation for 'ecryptfs_msg_ctx_arr'
    fails, the previously allocated 'ecryptfs_daemon_hash' is not deallocated,
    leading to a memory leak bug. To fix this issue, free
    'ecryptfs_daemon_hash' before returning the error.

    Cc: stable@vger.kernel.org
    Fixes: 88b4a07e6610 ("[PATCH] eCryptfs: Public key transport mechanism")
    Signed-off-by: Wenwen Wang <wenwen@cs.uga.edu>
    Signed-off-by: Tyler Hicks <tyhicks@canonical.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5306c45f5b9178147a307166b1cf58681af41f24
Author: Wenwen Wang <wenwen@cs.uga.edu>
Date:   Tue Aug 20 00:16:40 2019 -0500

    ecryptfs: fix a memory leak bug in parse_tag_1_packet()

    commit fe2e082f5da5b4a0a92ae32978f81507ef37ec66 upstream.

    In parse_tag_1_packet(), if tag 1 packet contains a key larger than
    ECRYPTFS_MAX_ENCRYPTED_KEY_BYTES, no cleanup is executed, leading to a
    memory leak on the allocated 'auth_tok_list_item'. To fix this issue, go to
    the label 'out_free' to perform the cleanup work.

    Cc: stable@vger.kernel.org
    Fixes: dddfa461fc89 ("[PATCH] eCryptfs: Public key; packet management")
    Signed-off-by: Wenwen Wang <wenwen@cs.uga.edu>
    Signed-off-by: Tyler Hicks <tyhicks@canonical.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 30a57e3974cf6c955fd6b356b23c9d897fc00f4e
Author: Roberto Sassu <roberto.sassu@huawei.com>
Date:   Mon Feb 10 11:00:41 2020 +0100

    tpm: Initialize crypto_id of allocated_banks to HASH_ALGO__LAST

    commit dc10e4181c05a2315ddc375e963b7c763b5ee0df upstream.

    chip->allocated_banks, an array of tpm_bank_info structures, contains the
    list of TPM algorithm IDs of allocated PCR banks. It also contains the
    corresponding ID of the crypto subsystem, so that users of the TPM driver
    can calculate a digest for a PCR extend operation.

    However, if there is no mapping between TPM algorithm ID and crypto ID, the
    crypto_id field of tpm_bank_info remains set to zero (the array is
```

```
    allocated and initialized with kcalloc() in tpm2_get_pcr_allocation()).
    Zero should not be used as value for unknown mappings, as it is a valid
    crypto ID (HASH_ALGO_MD4).

    Thus, initialize crypto_id to HASH_ALGO__LAST.

    Cc: stable@vger.kernel.org # 5.1.x
    Fixes: 879b589210a9 ("tpm: retrieve digest size of unknown algorithms with PCR read")
    Signed-off-by: Roberto Sassu <roberto.sassu@huawei.com>
    Reviewed-by: Petr Vorel <pvorel@suse.cz>
    Reviewed-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Signed-off-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 311402804bc0871b18321f6e3c39e9d24930959d
Author: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
Date:   Tue Feb 4 14:16:27 2020 +0200

    tpm: Revert tpm_tis_spi_mod.ko to tpm_tis_spi.ko.

    commit faaca0a0d48e7b122f6e7e2521f4f6fc487d0451 upstream.

    Revert tpm_tis_spi_mod.ko back to tpm_tis_spi.ko as the rename could
    break user space scripts. This can be achieved by renaming tpm_tis_spi.c
    as tpm_tis_spi_main.c. Then tpm_tis_spi-y can be used inside the
    makefile.

    Cc: Andrey Pronin <apronin@chromium.org>
    Cc: Stephen Boyd <swboyd@chromium.org>
    Cc: stable@vger.kernel.org # 5.5.x
    Fixes: 797c0113c9a4 ("tpm: tpm_tis_spi: Support cr50 devices")
    Reported-by: Alexander Steffen <Alexander.Steffen@infineon.com>
    Tested-by: Alexander Steffen <Alexander.Steffen@infineon.com>
    Reviewed-by: Stephen Boyd <swboyd@chromium.org>
    Signed-off-by: Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 873ef039adf8617d31fc2e7d03476ed050246c59
Author: Samuel Holland <samuel@sholland.org>
Date:   Mon Feb 17 00:42:22 2020 -0600

    ASoC: sun8i-codec: Fix setting DAI data format

    commit 96781fd941b39e1f78098009344ebcd7af861c67 upstream.

    Use the correct mask for this two-bit field. This fixes setting the DAI
    data format to RIGHT_J or DSP_A.

    Fixes: 36c684936fae ("ASoC: Add sun8i digital audio codec")
    Signed-off-by: Samuel Holland <samuel@sholland.org>
    Acked-by: Chen-Yu Tsai <wens@csie.org>
    Cc: stable@kernel.org
    Link: https://lore.kernel.org/r/20200217064250.15516-7-samuel@sholland.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0ec27a09456b3f05a19832991c177e5544b6b80c
Author: Samuel Holland <samuel@sholland.org>
Date:   Thu Feb 13 00:11:44 2020 -0600

    ASoC: codec2codec: avoid invalid/double-free of pcm runtime

    commit b6570fdb96edf45bcf71884bd2644bd73d348d1a upstream.

    The PCM runtime was freed during PMU in the case that the event hook
```

```
    encountered an error. However, it is also unconditionally freed during
    PMD. Avoid a double-free by dropping the call to kfree in the PMU hook.

    Fixes: a72706ed8208 ("ASoC: codec2codec: remove ephemeral variables")
    Cc: stable@vger.kernel.org
    Signed-off-by: Samuel Holland <samuel@sholland.org>
    Link: https://lore.kernel.org/r/20200213061147.29386-2-samuel@sholland.org
    Signed-off-by: Mark Brown <broonie@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f44bddef1365ee224bfb1c8a62c19b495c0136e7
Author: Takashi Iwai <tiwai@suse.de>
Date:   Tue Feb 18 09:09:15 2020 +0100

    ALSA: hda/realtek - Apply quirk for yet another MSI laptop

    commit cc5049ae4d457194796f854eb2e38b9727ad8c2d upstream.

    MSI GP65 laptop with SSID 1462:1293 requires the same quirk as other
    MSI models.

    BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=204159
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200218080915.3433-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4f8fb52ba53ae676f889fa28711d9e00b1e5ee3e
Author: Takashi Iwai <tiwai@suse.de>
Date:   Mon Feb 17 16:19:47 2020 +0100

    ALSA: hda/realtek - Apply quirk for MSI GP63, too

    commit a655e2b107d463ce2745188ce050d07daed09a71 upstream.

    The same quirk that was applied to MSI GL73 is needed for MSI GP63,
    too.  Adding the entry with the SSID 1462:1228.

    BugLink: https://bugzilla.kernel.org/show_bug.cgi?id=206503
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200217151947.17528-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f60e065dbdb1e3600194d6f9952c3973f5badd61
Author: Takashi Iwai <tiwai@suse.de>
Date:   Tue Feb 18 10:14:09 2020 +0100

    ALSA: hda: Use scnprintf() for printing texts for sysfs/procfs

    commit 44eeb081b8630bb3ad3cd381d1ae1831463e48bb upstream.

    Some code in HD-audio driver calls snprintf() in a loop and still
    expects that the return value were actually written size, while
    snprintf() returns the expected would-be length instead.  When the
    given buffer limit were small, this leads to a buffer overflow.

    Use scnprintf() for addressing those issues.  It returns the actually
    written size unlike snprintf().

    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200218091409.27162-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 66a17bf4756d6b4f30c969293ec23cf89995c7cf
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 17 17:29:55 2020 +0100

    iommu/vt-d: Simplify check in identity_mapping()

    commit 1ddb32da4a629fa7f87873d0b6836c2e1feb7518 upstream.

    The function only has one call-site and there it is never called with
    dummy or deferred devices. Simplify the check in the function to
    account for that.

    Fixes: 1ee0186b9a12 ("iommu/vt-d: Refactor find_domain() helper")
    Cc: stable@vger.kernel.org # v5.5
    Reviewed-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 248e99f80d08dc17b935bd5d0bb9837c98390483
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 17 17:27:44 2020 +0100

    iommu/vt-d: Remove deferred_attach_domain()

    commit 96d170f3b1a607612caf3618c534d5c64fc2d61b upstream.

    The function is now only a wrapper around find_domain(). Remove the
    function and call find_domain() directly at the call-sites.

    Fixes: 1ee0186b9a12 ("iommu/vt-d: Refactor find_domain() helper")
    Cc: stable@vger.kernel.org # v5.5
    Reviewed-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 271b0526cb62fa6ba7d4a3771742bef46baad3f3
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 17 17:20:59 2020 +0100

    iommu/vt-d: Do deferred attachment in iommu_need_mapping()

    commit a11bfde9c77df1fd350ea27169ab921f511bf5d0 upstream.

    The attachment of deferred devices needs to happen before the check
    whether the device is identity mapped or not. Otherwise the check will
    return wrong results, cause warnings boot failures in kdump kernels, like

            WARNING: CPU: 0 PID: 318 at ../drivers/iommu/intel-iommu.c:592
domain_get_iommu+0x61/0x70

            [...]

             Call Trace:
              __intel_map_single+0x55/0x190
              intel_alloc_coherent+0xac/0x110
              dmam_alloc_attrs+0x50/0xa0
              ahci_port_start+0xfb/0x1f0 [libahci]
              ata_host_start.part.39+0x104/0x1e0 [libata]

    With the earlier check the kdump boot succeeds and a crashdump is written.

    Fixes: 1ee0186b9a12 ("iommu/vt-d: Refactor find_domain() helper")
    Cc: stable@vger.kernel.org # v5.5
```

```
    Reviewed-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c8995b96489944e406746838ecf5e8417f307cdd
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 17 17:16:19 2020 +0100

    iommu/vt-d: Move deferred device attachment into helper function

    commit 034d98cc0cdcde2415c6f598fa9125e3eaa02569 upstream.

    Move the code that does the deferred device attachment into a separate
    helper function.

    Fixes: 1ee0186b9a12 ("iommu/vt-d: Refactor find_domain() helper")
    Cc: stable@vger.kernel.org # v5.5
    Reviewed-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 68a57fda867e2085cfd3403f03fd2ac7802d6c6d
Author: Joerg Roedel <jroedel@suse.de>
Date:   Mon Feb 17 17:12:37 2020 +0100

    iommu/vt-d: Add attach_deferred() helper

    commit 1d4615978f525b769990a4a4ef22fb1b9a04cdf1 upstream.

    Implement a helper function to check whether a device's attach process
    is deferred.

    Fixes: 1ee0186b9a12 ("iommu/vt-d: Refactor find_domain() helper")
    Cc: stable@vger.kernel.org # v5.5
    Reviewed-by: Jerry Snitselaar <jsnitsel@redhat.com>
    Acked-by: Lu Baolu <baolu.lu@linux.intel.com>
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f2d52fb1253d702c485a861f9e0e15829c3342bc
Author: Robin Murphy <robin.murphy@arm.com>
Date:   Tue Feb 18 18:12:41 2020 +0000

    iommu/qcom: Fix bogus detach logic

    commit faf305c51aeabd1ea2d7131e798ef5f55f4a7750 upstream.

    Currently, the implementation of qcom_iommu_domain_free() is guaranteed
    to do one of two things: WARN() and leak everything, or dereference NULL
    and crash. That alone is terrible, but in fact the whole idea of trying
    to track the liveness of a domain via the qcom_domain->iommu pointer as
    a sanity check is full of fundamentally flawed assumptions. Make things
    robust and actually functional by not trying to be quite so clever.

    Reported-by: Brian Masney <masneyb@onstation.org>
    Tested-by: Brian Masney <masneyb@onstation.org>
    Reported-by: Naresh Kamboju <naresh.kamboju@linaro.org>
    Fixes: 0ae349a0f33f ("iommu/qcom: Add qcom_iommu")
    Signed-off-by: Robin Murphy <robin.murphy@arm.com>
    Tested-by: Stephan Gerhold <stephan@gerhold.net>
    Cc: stable@vger.kernel.org # v4.14+
    Signed-off-by: Joerg Roedel <jroedel@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```