```
commit 38f3ee12661fdc2805e06942e4e3d604e03cd9cf
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Sat Jan 14 10:33:46 2023 +0100

    Linux 6.1.6

    Link: https://lore.kernel.org/r/20230112135326.981869724@linuxfoundation.org
    Tested-by: Conor Dooley <conor.dooley@microchip.com>
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Kelsey Steele <kelseysteele@linux.microsoft.com>
    Tested-by: Takeshi Ogasawara <takeshi.ogasawara@futuring-girl.com>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Ron Economos <re@w6rz.net>
    Tested-by: Sudip Mukherjee <sudip.mukherjee@codethink.co.uk>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Allen Pais <apais@linux.microsoft.com>
    Tested-by: Rudi Heitbaum <rudi@heitbaum.com>
    Tested-by: Salvatore Bonaccorso <carnil@debian.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cb9b3951d6366c30f0085c9830381b1334cdcf62
Author: Takashi Iwai <tiwai@suse.de>
Date:   Thu Jan 5 10:35:31 2023 +0100

    ALSA: hda: cs35l41: Check runtime suspend capability at runtime_idle

    commit ae50e2ab122cef68f46b7799fb9deffe3334f5e2 upstream.

    The runtime PM core checks with runtime_idle callback whether it can
    goes to the runtime suspend or not, and we can put the boost type
    check there instead of runtime_suspend and _resume calls.  This will
    reduce the unnecessary runtime_suspend() calls.

    Fixes: 1873ebd30cc8 ("ALSA: hda: cs35l41: Support Hibernation during Suspend")
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20230105093531.16960-2-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a6d58eff4f8367af588c49cbceed28147b1030c7
Author: Chris Chiu <chris.chiu@canonical.com>
Date:   Tue Jan 3 17:53:32 2023 +0800

    ALSA: hda - Enable headset mic on another Dell laptop with ALC3254

    commit a5751933a7f6abbdad90d98f25a25bb4b133a9e6 upstream.

    There is another Dell Latitude laptop (1028:0c03) with Realtek
    codec ALC3254 which needs the ALC269_FIXUP_DELL4_MIC_NO_PRESENCE
    instead of the default matched ALC269_FIXUP_DELL1_MIC_NO_PRESENCE.
    Apply correct fixup for this particular model to enable headset mic.

    Signed-off-by: Chris Chiu <chris.chiu@canonical.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20230103095332.730677-1-chris.chiu@canonical.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8107550bfe1189d7459292a66aab30a74c349cda
Author: Takashi Iwai <tiwai@suse.de>
Date:   Thu Jan 5 10:35:30 2023 +0100
```

```
    ALSA: hda: cs35l41: Don't return -EINVAL from system suspend/resume

    commit 15a59cb0a3d6ddf2cb79f8dc3081b3130aad3767 upstream.

    The recent commit to support the system suspend for CS35L41 caused a
    regression on the models with CS35L41_EXT_BOOST_NO_VSPK_SWITC boost
    type, as the suspend/resume callbacks just return -EINVAL.  This is
    eventually handled as a fatal error and blocks the whole system
    suspend/resume.

    For avoiding the problem, this patch corrects the return code from
    cs35l41_system_suspend() and _resume() to 0, and replace dev_err()
    with dev_err_once() for stop spamming too much.

    Fixes: 88672826e2a4 ("ALSA: hda: cs35l41: Support System Suspend")
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/all/e6751ac2-34f3-d13f-13db-8174fade8308@pm.me
    Link: https://lore.kernel.org/r/20230105093531.16960-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 57f6bea5218c37426ccf68b37fa66a0f476ef869
Author: Jeremy Szu <jeremy.szu@canonical.com>
Date:   Thu Jan 5 12:41:53 2023 +0800

    ALSA: hda/realtek: fix mute/micmute LEDs don't work for a HP platform

    commit 9c694fbfe6f36017b060ad74c7565cb379852e40 upstream.

    There is a HP platform uses ALC236 codec which using GPIO2 to control
    mute LED and GPIO1 to control micmute LED.
    Thus, add a quirk to make them work.

    Signed-off-by: Jeremy Szu <jeremy.szu@canonical.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20230105044154.8242-1-jeremy.szu@canonical.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0e19738afc84fc2f8ed1ad1baa0373fcb8ebca36
Author: Adrian Chan <adchan@google.com>
Date:   Mon Jan 9 16:05:20 2023 -0500

    ALSA: hda/hdmi: Add a HP device 0x8715 to force connect list

    commit de1ccb9e61728dd941fe0e955a7a129418657267 upstream.

    Add the 'HP Engage Flex Mini' device to the force connect list to
    enable audio through HDMI.

    Signed-off-by: Adrian Chan <adchan@google.com>
    Cc: <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20230109210520.16060-1-adchan@google.com
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d6ad4bd1d896ae1daffd7628cd50f124280fb8b1
Author: Clement Lecigne <clecigne@google.com>
Date:   Fri Jan 13 13:07:45 2023 +0100

    ALSA: pcm: Move rwsem lock inside snd_ctl_elem_read to prevent UAF

    commit 56b88b50565cd8b946a2d00b0c83927b7ebb055e upstream.
```

```
    Takes rwsem lock inside snd_ctl_elem_read instead of snd_ctl_elem_read_user
    like it was done for write in commit 1fa4445f9adf1 ("ALSA: control - introduce
    snd_ctl_notify_one() helper"). Doing this way we are also fixing the following
    locking issue happening in the compat path which can be easily triggered and
    turned into an use-after-free.

    64-bits:
    snd_ctl_ioctl
      snd_ctl_elem_read_user
        [takes controls_rwsem]
        snd_ctl_elem_read [lock properly held, all good]
        [drops controls_rwsem]

    32-bits:
    snd_ctl_ioctl_compat
      snd_ctl_elem_write_read_compat
        ctl_elem_write_read
          snd_ctl_elem_read [missing lock, not good]

    CVE-2023-0266 was assigned for this issue.

    Cc: stable@kernel.org # 5.13+
    Signed-off-by: Clement Lecigne <clecigne@google.com>
    Reviewed-by: Jaroslav Kysela <perex@perex.cz>
    Link: https://lore.kernel.org/r/20230113120745.25464-1-tiwai@suse.de
    Signed-off-by: Takashi Iwai <tiwai@suse.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e8988e878af693ac13b0fa80ba2e72d22d68f2dd
Author: Frederick Lawler <fred@cloudflare.com>
Date:   Mon Jan 9 10:39:06 2023 -0600

    net: sched: disallow noqueue for qdisc classes

    commit 96398560f26aa07e8f2969d73c8197e6a6d10407 upstream.

    While experimenting with applying noqueue to a classful queue discipline,
    we discovered a NULL pointer dereference in the __dev_queue_xmit()
    path that generates a kernel OOPS:

        # dev=enp0s5
        # tc qdisc replace dev $dev root handle 1: htb default 1
        # tc class add dev $dev parent 1: classid 1:1 htb rate 10mbit
        # tc qdisc add dev $dev parent 1:1 handle 10: noqueue
        # ping -I $dev -w 1 -c 1 1.1.1.1

    [    2.172856] BUG: kernel NULL pointer dereference, address: 0000000000000000
    [    2.173217] #PF: supervisor instruction fetch in kernel mode
    ...
    [    2.178451] Call Trace:
    [    2.178577]  <TASK>
    [    2.178686]  htb_enqueue+0x1c8/0x370
    [    2.178880]  dev_qdisc_enqueue+0x15/0x90
    [    2.179093]  __dev_queue_xmit+0x798/0xd00
    [    2.179305]  ? _raw_write_lock_bh+0xe/0x30
    [    2.179522]  ? __local_bh_enable_ip+0x32/0x70
    [    2.179759]  ? ___neigh_create+0x610/0x840
    [    2.179968]  ? eth_header+0x21/0xc0
    [    2.180144]  ip_finish_output2+0x15e/0x4f0
    [    2.180348]  ? dst_output+0x30/0x30
    [    2.180525]  ip_push_pending_frames+0x9d/0xb0
    [    2.180739]  raw_sendmsg+0x601/0xcb0
    [    2.180916]  ? _raw_spin_trylock+0xe/0x50
    [    2.181112]  ? _raw_spin_unlock_irqrestore+0x16/0x30
    [    2.181354]  ? get_page_from_freelist+0xcd6/0xdf0
```

```
[    2.181594]  ? sock_sendmsg+0x56/0x60
[    2.181781]  sock_sendmsg+0x56/0x60
[    2.181958]  __sys_sendto+0xf7/0x160
[    2.182139]  ? handle_mm_fault+0x6e/0x1d0
[    2.182366]  ? do_user_addr_fault+0x1e1/0x660
[    2.182627]  __x64_sys_sendto+0x1b/0x30
[    2.182881]  do_syscall_64+0x38/0x90
[    2.183085]  entry_SYSCALL_64_after_hwframe+0x63/0xcd
...
[    2.187402]  </TASK>
```

Previously in commit d66d6c3152e8 ("net: sched: register noqueue qdisc"), NULL was set for the noqueue discipline on noqueue init so that __dev_queue_xmit() falls through for the noqueue case. This also sets a bypass of the enqueue NULL check in the register_qdisc() function for the struct noqueue_disc_ops.

Classful queue disciplines make it past the NULL check in __dev_queue_xmit() because the discipline is set to htb (in this case), and then in the call to __dev_xmit_skb(), it calls into htb_enqueue() which grabs a leaf node for a class and then calls qdisc_enqueue() by passing in a queue discipline which assumes ->enqueue() is not set to NULL.

Fix this by not allowing classes to be assigned to the noqueue discipline. Linux TC Notes states that classes cannot be set to the noqueue discipline. [1] Let's enforce that here.

Links:
1. https://linux-tc-notes.sourceforge.net/tc/doc/sch_noqueue.txt

Fixes: d66d6c3152e8 ("net: sched: register noqueue qdisc")
Cc: stable@vger.kernel.org
Signed-off-by: Frederick Lawler <fred@cloudflare.com>
Reviewed-by: Jakub Sitnicki <jakub@cloudflare.com>
Link: https://lore.kernel.org/r/20230109163906.706000-1-fred@cloudflare.com
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit cd2e80a3ac7005f964680ae63946cdde4e970e0a
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date: Mon Jan 9 17:04:49 2023 -0600

gcc: disable -Warray-bounds for gcc-11 too

commit 5a41237ad1d4b62008f93163af1d9b1da90729d8 upstream.

We had already disabled this warning for gcc-12 due to bugs in the value range analysis, but it turns out we end up having some similar problems with gcc-11.3 too, so let's disable it there too.

Older gcc versions end up being increasingly less relevant, and hopefully clang and newer version of gcc (ie gcc-13) end up working reliably enough that we still get the build coverage even when we disable this for some versions.

Link: https://lore.kernel.org/all/20221227002941.GA2691687@roeck-us.net/
Link: https://lore.kernel.org/all/D8BDBF66-E44C-45D4-9758-BAAA4F0C1998@kernel.org/
Cc: Kees Cook <kees@kernel.org>
Cc: Vlastimil Babka <vbabka@suse.cz>
Cc: Guenter Roeck <linux@roeck-us.net>
Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8cc0e63ba86c9f4e732de9f42642231c8e3a31d2
Author: Chuck Lever <chuck.lever@oracle.com>

```
Date:   Fri Jan 6 12:43:37 2023 -0500

    Revert "SUNRPC: Use RMW bitops in single-threaded hot paths"

    commit 7827c81f0248e3c2f40d438b020f3d222f002171 upstream.

    The premise that "Once an svc thread is scheduled and executing an
    RPC, no other processes will touch svc_rqst::rq_flags" is false.
    svc_xprt_enqueue() examines the RQ_BUSY flag in scheduled nfsd
    threads when determining which thread to wake up next.

    Found via KCSAN.

    Fixes: 28df0988815f ("SUNRPC: Use RMW bitops in single-threaded hot paths")
    Reviewed-by: Jeff Layton <jlayton@kernel.org>
    Signed-off-by: Chuck Lever <chuck.lever@oracle.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 29fbaa434c223f29917e8de054cc48a96e6c0ed4
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:32 2022 -0800

    selftests/vm/pkeys: Add a regression test for setting PKRU through ptrace

    commit 6ea25770b043c7997ab21d1ce95ba5de4d3d85d9 upstream.

    This tests PTRACE_SETREGSET with NT_X86_XSTATE modifying PKRU directly and
    removing the PKRU bit from XSTATE_BV.

    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-7-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d2602da3da87a1832f708c8639b0afbfb5704cd4
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:31 2022 -0800

    x86/fpu: Emulate XRSTOR's behavior if the xfeatures PKRU bit is not set

    commit d7e5aceace514a2b1b3ca3dc44f93f1704766ca7 upstream.

    The hardware XRSTOR instruction resets the PKRU register to its hardware
    init value (namely 0) if the PKRU bit is not set in the xfeatures mask.
    Emulating that here restores the pre-5.14 behavior for PTRACE_SET_REGSET
    with NT_X86_XSTATE, and makes sigreturn (which still uses XRSTOR) and
    ptrace behave identically. KVM has never used XRSTOR and never had this
    behavior, so KVM opts-out of this emulation by passing a NULL pkru pointer
    to copy_uabi_to_xstate().

    Fixes: e84ba47e313d ("x86/fpu: Hook up PKRU into ptrace()")
    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-6-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 587478470b0a632c706090a8b0cbf59de0e98b48
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:30 2022 -0800

    x86/fpu: Allow PKRU to be (once again) written by ptrace.

    commit 4a804c4f8356393d6b5eff7600f07615d7869c13 upstream.

    Move KVM's PKRU handling code in fpu_copy_uabi_to_guest_fpstate() to
```

```
    copy_uabi_to_xstate() so that it is shared with other APIs that write the
    XSTATE such as PTRACE_SETREGSET with NT_X86_XSTATE.

    This restores the pre-5.14 behavior of ptrace. The regression can be seen
    by running gdb and executing `p $pkru`, `set $pkru = 42`, and `p $pkru`.
    On affected kernels (5.14+) the write to the PKRU register (which gdb
    performs through ptrace) is ignored.

    [ dhansen: removed stable@ tag for now.  The ABI was broken for long
               enough that this is not urgent material.  Let's let it stew
               in tip for a few weeks before it's submitted to stable
               because there are so many ABIs potentially affected. ]

    Fixes: e84ba47e313d ("x86/fpu: Hook up PKRU into ptrace()")
    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-5-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bfa72faf69e82f7d64fefe0ca162f803d98e6679
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:29 2022 -0800

    x86/fpu: Add a pkru argument to copy_uabi_to_xstate()

    commit 2c87767c35ee9744f666ccec869d5fe742c3de0a upstream.

    In preparation for moving PKRU handling code out of
    fpu_copy_uabi_to_guest_fpstate() and into copy_uabi_to_xstate(), add an
    argument that copy_uabi_from_kernel_to_xstate() can use to pass the
    canonical location of the PKRU value. For
    copy_sigframe_from_user_to_xstate() the kernel will actually restore the
    PKRU value from the fpstate, but pass in the thread_struct's pkru location
    anyways for consistency.

    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-4-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a442736b704d4194ae68f75f4ee2e64cf2b8a142
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:28 2022 -0800

    x86/fpu: Add a pkru argument to copy_uabi_from_kernel_to_xstate().

    commit 1c813ce0305571e1b2e4cc4acca451da9e6ad18f upstream.

    Both KVM (through KVM_SET_XSTATE) and ptrace (through PTRACE_SETREGSET
    with NT_X86_XSTATE) ultimately call copy_uabi_from_kernel_to_xstate(),
    but the canonical locations for the current PKRU value for KVM guests
    and processes in a ptrace stop are different (in the kvm_vcpu_arch and
    the thread_state structs respectively).

    In preparation for eventually handling PKRU in
    copy_uabi_to_xstate, pass in a pointer to the PKRU location.

    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-3-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f28418b06f6b1e83c81369ef1705ee53da106ae6
Author: Kyle Huey <me@kylehuey.com>
Date:   Tue Nov 15 15:09:27 2022 -0800
```

```
    x86/fpu: Take task_struct* in copy_sigframe_from_user_to_xstate()

    commit 6a877d2450ace4f27c012519e5a1ae818f931983 upstream.

    This will allow copy_sigframe_from_user_to_xstate() to grab the address of
    thread_struct's pkru value in a later patch.

    Signed-off-by: Kyle Huey <me@kylehuey.com>
    Signed-off-by: Dave Hansen <dave.hansen@linux.intel.com>
    Link: https://lore.kernel.org/all/20221115230932.7126-2-khuey%40kylehuey.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 763b92568726b55569ea4896fe36e3c174cc3b0d
Author: Helge Deller <deller@gmx.de>
Date:   Sun Dec 11 19:50:20 2022 +0100

    parisc: Align parisc MADV_XXX constants with all other architectures

    commit 71bdea6f798b425bc0003780b13e3fdecb16a010 upstream.

    Adjust some MADV_XXX constants to be in sync what their values are on
    all other platforms. There is currently no reason to have an own
    numbering on parisc, but it requires workarounds in many userspace
    sources (e.g. glibc, qemu, ...) - which are often forgotten and thus
    introduce bugs and different behaviour on parisc.

    A wrapper avoids an ABI breakage for existing userspace applications by
    translating any old values to the new ones, so this change allows us to
    move over all programs to the new ABI over time.

    Signed-off-by: Helge Deller <deller@gmx.de>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```