```
commit fbd56ddcecab5a3623a89c8e941fdbcc55b41045
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Oct 12 09:39:05 2022 +0200

    Linux 6.0.1

    Link: https://lore.kernel.org/r/20221010070330.159911806@linuxfoundation.org
    Tested-by: Luna Jernberg <droidbittin@gmail.com>
    Tested-by: Fenil Jain <fkjainco@gmail.com>
    Tested-by: Linux Kernel Functional Testing <lkft@linaro.org>
    Tested-by: Justin M. Forbes <jforbes@fedoraproject.org>
    Tested-by: Florian Fainelli <f.fainelli@gmail.com>
    Tested-by: Ronald Warsow <rwarsow@gmx.de>
    Tested-by: Shuah Khan <skhan@linuxfoundation.org>
    Tested-by: Zan Aziz <zanaziz313@gmail.com>
    Tested-by: Slade Watkins <srw@sladewatkins.net>
    Tested-by: Guenter Roeck <linux@roeck-us.net>
    Tested-by: Rudi Heitbaum <rudi@heitbaum.com>
    Tested-by: Bagas Sanjaya <bagasdotme@gmail.com>
    Tested-by: Jon Hunter <jonathanh@nvidia.com>
    Tested-by: Sudip Mukherjee <sudip.mukherjee@codethink.co.uk>
    Tested-by: Ron Economos <re@w6rz.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3c6b036fe5c8ed8b6c4cbdc03605929882907ef0
Author: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Date:   Fri Sep 2 20:23:48 2022 +0900

    Bluetooth: use hdev->workqueue when queuing hdev->{cmd,ncmd}_timer works

    commit deee93d13d385103205879a8a0915036ecd83261 upstream.

    syzbot is reporting attempt to schedule hdev->cmd_work work from system_wq
    WQ into hdev->workqueue WQ which is under draining operation [1], for
    commit c8efcc2589464ac7 ("workqueue: allow chained queueing during
    destruction") does not allow such operation.

    The check introduced by commit 877afadad2dce8aa ("Bluetooth: When HCI work
    queue is drained, only queue chained work") was incomplete.

    Use hdev->workqueue WQ when queuing hdev->{cmd,ncmd}_timer works because
    hci_{cmd,ncmd}_timeout() calls queue_work(hdev->workqueue). Also, protect
    the queuing operation with RCU read lock in order to avoid calling
    queue_delayed_work() after cancel_delayed_work() completed.

    Link: https://syzkaller.appspot.com/bug?extid=243b7d89777f90f7613b [1]
    Reported-by: syzbot <syzbot+243b7d89777f90f7613b@syzkaller.appspotmail.com>
    Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
    Fixes: 877afadad2dce8aa ("Bluetooth: When HCI work queue is drained, only queue chained
work")
    Signed-off-by: Luiz Augusto von Dentz <luiz.von.dentz@intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 390ba6a7755d6db79e8000d5501096c24f3b8bfd
Author: Jules Irenge <jbi.octave@gmail.com>
Date:   Wed Sep 7 16:24:20 2022 +0100

    bpf: Fix resetting logic for unreferenced kptrs

    commit 9fad7fe5b29803584c7f17a2abe6c2936fec6828 upstream.

    Sparse reported a warning at bpf_map_free_kptrs()
    "warning: Using plain integer as NULL pointer"
    During the process of fixing this warning, it was discovered that the current
```

```
    code erroneously writes to the pointer variable instead of deferencing and
    writing to the actual kptr. Hence, Sparse tool accidentally helped to uncover
    this problem. Fix this by doing WRITE_ONCE(*p, 0) instead of WRITE_ONCE(p, 0).

    Note that the effect of this bug is that unreferenced kptrs will not be cleared
    during check_and_free_fields. It is not a problem if the clearing is not done
    during map_free stage, as there is nothing to free for them.

    Fixes: 14a324f6a67e ("bpf: Wire up freeing of referenced kptr")
    Signed-off-by: Jules Irenge <jbi.octave@gmail.com>
    Link: https://lore.kernel.org/r/Yxi3pJaK6UDjVJSy@playground
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 448faed285b9a0dc4101ffb6a256aa261fcd0979
Author: Daniel Golle <daniel@makrotopia.org>
Date:   Fri Sep 30 01:56:53 2022 +0100

    net: ethernet: mtk_eth_soc: fix state in __mtk_foe_entry_clear

    commit ae3ed15da5889263de372ff9df2e83e16acca4cb upstream.

    Setting ib1 state to MTK_FOE_STATE_UNBIND in __mtk_foe_entry_clear
    routine as done by commit 0e80707d94e4c8 ("net: ethernet: mtk_eth_soc:
    fix typo in __mtk_foe_entry_clear") breaks flow offloading, at least
    on older MTK_NETSYS_V1 SoCs, OpenWrt users have confirmed the bug on
    MT7622 and MT7621 systems.
    Felix Fietkau suggested to use MTK_FOE_STATE_INVALID instead which
    works well on both, MTK_NETSYS_V1 and MTK_NETSYS_V2.

    Tested on MT7622 (Linksys E8450) and MT7986 (BananaPi BPI-R3).

    Suggested-by: Felix Fietkau <nbd@nbd.name>
    Fixes: 0e80707d94e4c8 ("net: ethernet: mtk_eth_soc: fix typo in __mtk_foe_entry_clear")
    Fixes: 33fc42de33278b ("net: ethernet: mtk_eth_soc: support creating mac address based
offload entries")
    Signed-off-by: Daniel Golle <daniel@makrotopia.org>
    Link: https://lore.kernel.org/r/YzY+1Yg0FBXcnrtc@makrotopia.org
    Signed-off-by: Jakub Kicinski <kuba@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 50fe01fa7640bc216bb1a47754a1f799c20eff8e
Author: Kumar Kartikeya Dwivedi <memxor@gmail.com>
Date:   Wed Sep 21 16:35:50 2022 +0200

    bpf: Gate dynptr API behind CAP_BPF

    commit 8addbfc7b308d591f8a5f2f6bb24d08d9d79dfbb upstream.

    This has been enabled for unprivileged programs for only one kernel
    release, hence the expected annoyances due to this move are low. Users
    using ringbuf can stick to non-dynptr APIs. The actual use cases dynptr
    is meant to serve may not make sense in unprivileged BPF programs.

    Hence, gate these helpers behind CAP_BPF and limit use to privileged
    BPF programs.

    Fixes: 263ae152e962 ("bpf: Add bpf_dynptr_from_mem for local dynptrs")
    Fixes: bc34dee65a65 ("bpf: Dynptr support for ring buffers")
    Fixes: 13bbbfbea759 ("bpf: Add bpf_dynptr_read and bpf_dynptr_write")
    Fixes: 34d4ef5775f7 ("bpf: Add dynptr data slices")
    Signed-off-by: Kumar Kartikeya Dwivedi <memxor@gmail.com>
    Link: https://lore.kernel.org/r/20220921143550.30247-1-memxor@gmail.com
    Acked-by: Andrii Nakryiko <andrii@kernel.org>
    Signed-off-by: Alexei Starovoitov <ast@kernel.org>
```

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

commit 64517c21fd0f28b40a75d0d3121276b31399278e
Author: Palmer Dabbelt <palmer@rivosinc.com>
Date:   Tue Sep 20 13:45:18 2022 -0700

    RISC-V: Print SSTC in canonical order

    commit 61a41d16ad20657f93613229a8b17766c51dc849 upstream.

    This got out of order during a merge conflict, fix it by putting the
    entries in the correct order.

    Fixes: 7ab52f75a9cf ("RISC-V: Add Sstc extension support")
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Reviewed-by: Conor Dooley <conor.dooley@microchip.com>
    Reviewed-by: Heiko Stuebner <heiko@sntech.de>
    Cc: stable@vger.kernel.org
    Link: https://lore.kernel.org/r/20220920204518.10988-1-palmer@rivosinc.com/
    Signed-off-by: Palmer Dabbelt <palmer@rivosinc.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit d3b5e30e87b52404765b25d9ee156a85c3d8caf6
Author: Mario Limonciello <mario.limonciello@amd.com>
Date:   Tue Aug 2 23:25:00 2022 -0500

    gpiolib: acpi: Add a quirk for Asus UM325UAZ

    commit 0ea76c401f9245ac209f1b1ce03a7e1fb9de36e5 upstream.

    Asus UM325UAZ has GPIO 18 programmed as both an interrupt and a wake
    source, but confirmed with internal team on this design this pin is
    floating and shouldn't have been programmed. This causes lots of
    spurious IRQs on the system and horrendous battery life.

    Add a quirk to ignore attempts to program this pin on this system.

    Reported-by: Pavel Krc <reg.krn@pkrc.net>
    Link: https://bugzilla.kernel.org/show_bug.cgi?id=216208
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Mario Limonciello <mario.limonciello@amd.com>
    Reviewed-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 359e1b7a7f79738e00a59f367dbe0d3a819d1754
Author: Mario Limonciello <mario.limonciello@amd.com>
Date:   Tue Aug 2 23:24:59 2022 -0500

    gpiolib: acpi: Add support to ignore programming an interrupt

    commit 6b6af7bd5718f4e45a9b930533aec1158387d552 upstream.

    gpiolib-acpi already had support for ignoring a pin for wakeup, but
    if an OEM configures a floating pin as an interrupt source then
    stopping it from being a wakeup won't do much good to stop the
    interrupt storm.

    Add support for a module parameter and quirk infrastructure to
    ignore interrupts as well.

    Signed-off-by: Mario Limonciello <mario.limonciello@amd.com>
    Reviewed-by: Hans de Goede <hdegoede@redhat.com>
    Reviewed-by: Mika Westerberg <mika.westerberg@linux.intel.com>
    Signed-off-by: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
```

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

commit 94e14d5193c5692faa589b733e9a360154d4696b
Author: Johan Hovold <johan@kernel.org>
Date:   Tue Sep 13 16:53:12 2022 +0200

    USB: serial: ftdi_sio: fix 300 bps rate for SIO

    commit 7bd7ad3c310cd6766f170927381eea0aa6f46c69 upstream.

    The 300 bps rate of SIO devices has been mapped to 9600 bps since
    2003... Let's fix the regression.

    Cc: stable@vger.kernel.org
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 08e2c70e549b77f5f3af9c76da00779d5756f997
Author: Tadeusz Struk <tadeusz.struk@linaro.org>
Date:   Mon Sep 19 14:59:57 2022 -0700

    usb: mon: make mmapped memory read only

    commit a659daf63d16aa883be42f3f34ff84235c302198 upstream.

    Syzbot found an issue in usbmon module, where the user space client can
    corrupt the monitor's internal memory, causing the usbmon module to
    crash the kernel with segfault, UAF, etc.

    The reproducer mmaps the /dev/usbmon memory to user space, and
    overwrites it with arbitrary data, which causes all kinds of issues.

    Return an -EPERM error from mon_bin_mmap() if the flag VM_WRTIE is set.
    Also clear VM_MAYWRITE to make it impossible to change it to writable
    later.

    Cc: "Dmitry Vyukov" <dvyukov@google.com>
    Cc: stable <stable@kernel.org>
    Fixes: 6f23ee1fefdc ("USB: add binary API to usbmon")
    Suggested-by: PaX Team <pageexec@freemail.hu>   # for the VM_MAYRITE portion
    Link: https://syzkaller.appspot.com/bug?id=2eb1f35d6525fa4a74d75b4244971e5b1411c95a
    Reported-by: syzbot+23f57c5ae902429285d7@syzkaller.appspotmail.com
    Signed-off-by: Tadeusz Struk <tadeusz.struk@linaro.org>
    Link: https://lore.kernel.org/r/20220919215957.205681-1-tadeusz.struk@linaro.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 3a93bbb581f85367b3227a685aa24a051230229c
Author: Aleksa Savic <savicaleksa83@gmail.com>
Date:   Wed Sep 14 13:43:27 2022 +0200

    hwmon: (aquacomputer_d5next) Fix Quadro fan speed offsets

    commit b7f3e9650f12d1e06b94a0257bcb90279f691bf5 upstream.

    The offsets for setting speeds of fans connected to Quadro are off by one.
    Set them to their correct values.

    The offsets as shown point to registers for setting the fan control mode,
    which will be explored in future patches, but slipped in here. When
    setting fan speeds, the resulting values were overlapping, which made the
    fans still run in my initial testing.

    Fixes: cdbe34da01e3 ("hwmon: (aquacomputer_d5next) Add support for Aquacomputer Quadro
fan controller")
    Signed-off-by: Aleksa Savic <savicaleksa83@gmail.com>
```

```
    Link: https://lore.kernel.org/r/20220914114327.6941-1-savicaleksa83@gmail.com
    Cc: stable@vger.kenrel.org
    Signed-off-by: Guenter Roeck <linux@roeck-us.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4ec318d06084a807df8eb212e321120236915fd6
Author: Shuah Khan <skhan@linuxfoundation.org>
Date:   Thu Sep 1 15:23:19 2022 -0600

    docs: update mediator information in CoC docs

    commit 8bfdfa0d6b929ede7b6189e0e546ceb6a124d05d upstream.

    Update mediator information in the CoC interpretation document.

    Signed-off-by: Shuah Khan <skhan@linuxfoundation.org>
    Link: https://lore.kernel.org/r/20220901212319.56644-1-skhan@linuxfoundation.org
    Cc: stable@vger.kernel.org
    Signed-off-by: Jonathan Corbet <corbet@lwn.net>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 49a49d65e21085c088d2a96e583282d59769926b
Author: Kees Cook <keescook@chromium.org>
Date:   Thu Sep 29 22:57:43 2022 -0700

    hardening: Remove Clang's enable flag for -ftrivial-auto-var-init=zero

    commit 607e57c6c62c00965ae276902c166834ce73014a upstream.

    Now that Clang's -enable-trivial-auto-var-init-zero-knowing-it-will-be-removed-from-
clang
    option is no longer required, remove it from the command line. Clang 16
    and later will warn when it is used, which will cause Kconfig to think
    it can't use -ftrivial-auto-var-init=zero at all. Check for whether it
    is required and only use it when so.

    Cc: Nathan Chancellor <nathan@kernel.org>
    Cc: Masahiro Yamada <masahiroy@kernel.org>
    Cc: Nick Desaulniers <ndesaulniers@google.com>
    Cc: linux-kbuild@vger.kernel.org
    Cc: llvm@lists.linux.dev
    Cc: stable@vger.kernel.org
    Fixes: f02003c860d9 ("hardening: Avoid harmless Clang option under
CONFIG_INIT_STACK_ALL_ZERO")
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a3c72efe3d6143b0964642435d8d8bf42e4fbf75
Author: Sami Tolvanen <samitolvanen@google.com>
Date:   Fri Sep 30 20:33:10 2022 +0000

    Makefile.extrawarn: Move -Wcast-function-type-strict to W=1

    commit 2120635108b35ecad9c59c8b44f6cbdf4f98214e upstream.

    We enable -Wcast-function-type globally in the kernel to warn about
    mismatching types in function pointer casts. Compilers currently
    warn only about ABI incompability with this flag, but Clang 16 will
    enable a stricter version of the check by default that checks for an
    exact type match. This will be very noisy in the kernel, so disable
    -Wcast-function-type-strict without W=1 until the new warnings have
    been addressed.

    Cc: stable@vger.kernel.org
    Link: https://reviews.llvm.org/D134831
```

```
    Link: https://github.com/ClangBuiltLinux/linux/issues/1724
    Suggested-by: Nathan Chancellor <nathan@kernel.org>
    Signed-off-by: Sami Tolvanen <samitolvanen@google.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/20220930203310.4010564-1-samitolvanen@google.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit df3d15181e00f1e63d778c29d252a25e04f6b046
Author: Bart Van Assche <bvanassche@acm.org>
Date:   Tue Aug 30 13:58:42 2022 -0700

    sparc: Unbreak the build

    commit 17006e86a7641fa3c50324cfb602f0e74dac8527 upstream.

    Fix the following build errors:

    arch/sparc/mm/srmmu.c: In function ‘smp_flush_page_for_dma’:
    arch/sparc/mm/srmmu.c:1639:13: error: cast between incompatible function types from
‘void (*)(long unsigned int)’ to ‘void (*)(long unsigned int,  long unsigned int,  long
unsigned int,  long unsigned int,  long unsigned int)’ [-Werror=cast-function-type]
     1639 |         xc1((smpfunc_t) local_ops->page_for_dma, page);
          |             ^
    arch/sparc/mm/srmmu.c: In function ‘smp_flush_cache_mm’:
    arch/sparc/mm/srmmu.c:1662:29: error: cast between incompatible function types from
‘void (*)(struct mm_struct *)’ to ‘void (*)(long unsigned int,  long unsigned int,  long
unsigned int,  long unsigned int,  long unsigned int)’ [-Werror=cast-function-type]
     1662 |                         xc1((smpfunc_t) local_ops->cache_mm, (unsigned long)
mm);
          |
    [ ... ]

    Compile-tested only.

    Fixes: 552a23a0e5d0 ("Makefile: Enable -Wcast-function-type")
    Cc: stable@vger.kernel.org
    Signed-off-by: Bart Van Assche <bvanassche@acm.org>
    Tested-by: Andreas Larsson <andreas@gaisler.com>
    Signed-off-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/20220830205854.1918026-1-bvanassche@acm.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 79bc9ee841abf982affb75c8417abe9091e64304
Author: Al Viro <viro@zeniv.linux.org.uk>
Date:   Mon Oct 3 20:26:08 2022 -0400

    fix coredump breakage

    commit 4f526fef91b24197d489ff86789744c67f475bb4 upstream.

    Let me count the ways in which I'd screwed up:

    * when emitting a page, handling of gaps in coredump should happen
    before fetching the current file position.
    * fix for a problem that occurs on rather uncommon setups (and hadn't
    been observed in the wild) had been sent very late in the cycle.
    * ... with badly insufficient testing, introducing an easily
    reproducible breakage.  Without giving it time to soak in -next.

    Fucked-up-by: Al Viro <viro@zeniv.linux.org.uk>
    Reported-by: "J. R. Okajima" <hooanon05g@gmail.com>
    Tested-by: "J. R. Okajima" <hooanon05g@gmail.com>
    Fixes: 06bbaa6dc53c "[coredump] don't use __kernel_write() on kmap_local_page()"
    Cc: stable@kernel.org   # v6.0-only
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
```

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

commit 2a96b532098284ecf8e4849b8b9e5fc7a28bdee9
Author: Dongliang Mu <mudongliangabcd@gmail.com>
Date:   Tue Aug 16 12:08:58 2022 +0800

    fs: fix UAF/GPF bug in nilfs_mdt_destroy

    commit 2e488f13755ffbb60f307e991b27024716a33b29 upstream.

    In alloc_inode, inode_init_always() could return -ENOMEM if
    security_inode_alloc() fails, which causes inode->i_private
    uninitialized. Then nilfs_is_metadata_file_inode() returns
    true and nilfs_free_inode() wrongly calls nilfs_mdt_destroy(),
    which frees the uninitialized inode->i_private
    and leads to crashes(e.g., UAF/GPF).

    Fix this by moving security_inode_alloc just prior to
    this_cpu_inc(nr_inodes)


Link: https://lkml.kernel.org/r/CAFcO6XOcf1Jj2SeGt=jJV59wmhESeSKpfR0omdFRq+J9nD1vfQ@mail.gma
il.com
    Reported-by: butt3rflyh4ck <butterflyhuangxx@gmail.com>
    Reported-by: Hao Sun <sunhao.th@gmail.com>
    Reported-by: Jiacheng Xu <stitch@zju.edu.cn>
    Reviewed-by: Christian Brauner (Microsoft) <brauner@kernel.org>
    Signed-off-by: Dongliang Mu <mudongliangabcd@gmail.com>
    Cc: Al Viro <viro@zeniv.linux.org.uk>
    Cc: stable@vger.kernel.org
    Signed-off-by: Al Viro <viro@zeniv.linux.org.uk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a53f14b0a934de4c0ee6935c7ee224e979460a7b
Author: Jalal Mostafa <jalal.a.mostapha@gmail.com>
Date:   Wed Sep 21 13:57:01 2022 +0000

    xsk: Inherit need_wakeup flag for shared sockets

    commit 60240bc26114543fcbfcd8a28466e67e77b20388 upstream.

    The flag for need_wakeup is not set for xsks with `XDP_SHARED_UMEM`
    flag and of different queue ids and/or devices. They should inherit
    the flag from the first socket buffer pool since no flags can be
    specified once `XDP_SHARED_UMEM` is specified.

    Fixes: b5aea28dca134 ("xsk: Add shared umem support between queue ids")
    Signed-off-by: Jalal Mostafa <jalal.a.mostapha@gmail.com>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: Magnus Karlsson <magnus.karlsson@intel.com>
    Link: https://lore.kernel.org/bpf/20220921135701.10199-1-jalal.a.mostapha@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```