```
commit 7b86b8d18e494763ae4b5153a1596da4c2ff6b3c
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Wed Apr 1 10:58:19 2020 +0200

    Linux 5.6.1

commit 8e06cfc5af747644fe0af2034034180208377c74
Author: Dafna Hirschfeld <dafna.hirschfeld@collabora.com>
Date:   Wed Feb 19 16:25:54 2020 +0100

    media: v4l2-core: fix a use-after-free bug of sd->devnode

    commit 6990570f7e0a6078e11b9c5dc13f4b6e3f49a398 upstream.

    sd->devnode is released after calling
    v4l2_subdev_release. Therefore it should be set
    to NULL so that the subdev won't hold a pointer
    to a released object. This fixes a reference
    after free bug in function
    v4l2_device_unregister_subdev

    Fixes: 0e43734d4c46e ("media: v4l2-subdev: add release() internal op")

    Cc: stable@vger.kernel.org
    Signed-off-by: Dafna Hirschfeld <dafna.hirschfeld@collabora.com>
    Reviewed-by: Ezequiel Garcia <ezequiel@collabora.com>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c62dc65b5e8d0e5be21158ec2678ef4f2f0c739c
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 3 17:35:11 2020 +0100

    media: xirlink_cit: add missing descriptor sanity checks

    commit a246b4d547708f33ff4d4b9a7a5dbac741dc89d8 upstream.

    Make sure to check that we have two alternate settings and at least one
    endpoint before accessing the second altsetting structure and
    dereferencing the endpoint arrays.

    This specifically avoids dereferencing NULL-pointers or corrupting
    memory when a device does not have the expected descriptors.

    Note that the sanity check in cit_get_packet_size() is not redundant as
    the driver is mixing looking up altsettings by index and by number,
    which may not coincide.

    Fixes: 659fefa0eb17 ("V4L/DVB: gspca_xirlink_cit: Add support for camera with a bcd
version of 0.01")
    Fixes: 59f8b0bf3c12 ("V4L/DVB: gspca_xirlink_cit: support bandwidth changing for devices
with 1 alt setting")
    Cc: stable <stable@vger.kernel.org>     # 2.6.37
    Cc: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 57d103d5f219ac3fe842e4bbfa3f70e8d8be9352
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 3 17:35:10 2020 +0100
```

```
    media: stv06xx: add missing descriptor sanity checks

    commit 485b06aadb933190f4bc44e006076bc27a23f205 upstream.

    Make sure to check that we have two alternate settings and at least one
    endpoint before accessing the second altsetting structure and
    dereferencing the endpoint arrays.

    This specifically avoids dereferencing NULL-pointers or corrupting
    memory when a device does not have the expected descriptors.

    Note that the sanity checks in stv06xx_start() and pb0100_start() are
    not redundant as the driver is mixing looking up altsettings by index
    and by number, which may not coincide.

    Fixes: 8668d504d72c ("V4L/DVB (12082): gspca_stv06xx: Add support for st6422 bridge and
sensor")
    Fixes: c0b33bdc5b8d ("[media] gspca-stv06xx: support bandwidth changing")
    Cc: stable <stable@vger.kernel.org>     # 2.6.31
    Cc: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1d6daf10fd0822a34558a60528620315336321f7
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 3 17:35:12 2020 +0100

    media: dib0700: fix rc endpoint lookup

    commit f52981019ad8d6718de79b425a574c6bddf81f7c upstream.

    Make sure to use the current alternate setting when verifying the
    interface descriptors to avoid submitting an URB to an invalid endpoint.

    Failing to do so could cause the driver to misbehave or trigger a WARN()
    in usb_submit_urb() that kernels with panic_on_warn set would choke on.

    Fixes: c4018fa2e4c0 ("[media] dib0700: fix RC support on Hauppauge Nova-TD")
    Cc: stable <stable@vger.kernel.org>     # 3.16
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0d16043ecbc6b0e6b972d215b9003e4f51d45db1
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 3 17:35:09 2020 +0100

    media: ov519: add missing endpoint sanity checks

    commit 998912346c0da53a6dbb71fab3a138586b596b30 upstream.

    Make sure to check that we have at least one endpoint before accessing
    the endpoint array to avoid dereferencing a NULL-pointer on stream
    start.

    Note that these sanity checks are not redundant as the driver is mixing
    looking up altsettings by index and by number, which need not coincide.

    Fixes: 1876bb923c98 ("V4L/DVB (12079): gspca_ov519: add support for the ov511 bridge")
    Fixes: b282d87332f5 ("V4L/DVB (12080): gspca_ov519: Fix ov518+ with OV7620AE (Trust
spacecam 320)")
    Cc: stable <stable@vger.kernel.org>     # 2.6.31
```

```
    Cc: Hans de Goede <hdegoede@redhat.com>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bd02d910a22f90103f165f91a3e7304101acc3c4
Author: Eric Biggers <ebiggers@google.com>
Date:   Sat Mar 7 18:38:49 2020 -0800

    libfs: fix infoleak in simple_attr_read()

    commit a65cab7d7f05c2061a3e2490257d3086ff3202c6 upstream.

    Reading from a debugfs file at a nonzero position, without first reading
    at position 0, leaks uninitialized memory to userspace.

    It's a bit tricky to do this, since lseek() and pread() aren't allowed
    on these files, and write() doesn't update the position on them.  But
    writing to them with splice() *does* update the position:

            #define _GNU_SOURCE 1
            #include <fcntl.h>
            #include <stdio.h>
            #include <unistd.h>
            int main()
            {
                    int pipes[2], fd, n, i;
                    char buf[32];

                    pipe(pipes);
                    write(pipes[1], "0", 1);
                    fd = open("/sys/kernel/debug/fault_around_bytes", O_RDWR);
                    splice(pipes[0], NULL, fd, NULL, 1, 0);
                    n = read(fd, buf, sizeof(buf));
                    for (i = 0; i < n; i++)
                            printf("%02x", buf[i]);
                    printf("\n");
            }

    Output:
            5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a5a30

    Fix the infoleak by making simple_attr_read() always fill
    simple_attr::get_buf if it hasn't been filled yet.

    Reported-by: syzbot+fcab69d1ada3e8d6f06b@syzkaller.appspotmail.com
    Reported-by: Alexander Potapenko <glider@google.com>
    Fixes: acaefc25d21f ("[PATCH] libfs: add simple attribute files")
    Cc: stable@vger.kernel.org
    Signed-off-by: Eric Biggers <ebiggers@google.com>
    Acked-by: Kees Cook <keescook@chromium.org>
    Link: https://lore.kernel.org/r/20200308023849.988264-1-ebiggers@kernel.org
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ecab121e9cee44e9323130949b7a3ef994a80c8c
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Thu Feb 27 20:28:22 2020 +0800

    ahci: Add Intel Comet Lake H RAID PCI ID

    commit 32d2545462c6cede998267b86e57cda5d1dc2225 upstream.

    Add the PCI ID to the driver list to support this new device.
```

```
    Cc: stable@vger.kernel.org
    Signed-off-by: Kai-Heng Feng <kai.heng.feng@canonical.com>
    Signed-off-by: Jens Axboe <axboe@kernel.dk>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9c70d8d263b51aa83c2d71726f9b4e0c91436dcb
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Tue Feb 11 11:35:01 2020 +0100

    staging: wfx: annotate nested gc_list vs tx queue locking

    commit e2525a95cc0887c7dc0549cb5d0ac3e796e1d54c upstream.

    Lockdep is complaining about recursive locking, because it can't make
    a difference between locked skb_queues. Annotate nested locks and avoid
    double bh_disable/enable.

    [...]
    insmod/815 is trying to acquire lock:
    cb7d6418 (&(&list->lock)->rlock){+...}, at: wfx_tx_queues_clear+0xfc/0x198 [wfx]

    but task is already holding lock:
    cb7d61f4 (&(&list->lock)->rlock){+...}, at: wfx_tx_queues_clear+0xa0/0x198 [wfx]

    [...]
    Possible unsafe locking scenario:

          CPU0
          ----
     lock(&(&list->lock)->rlock);
     lock(&(&list->lock)->rlock);

    Cc: stable@vger.kernel.org
    Fixes: 9bca45f3d692 ("staging: wfx: allow to send 802.11 frames")
    Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Link:
https://lore.kernel.org/r/5e30397af95854b4a7deea073b730c00229f42ba.1581416843.git.mirq-
linux@rere.qmqm.pl
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 9f0927d5c226a57b98bdfc6e8d663c26905cdae6
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Tue Feb 11 11:35:01 2020 +0100

    staging: wfx: fix init/remove vs IRQ race

    commit 4033714d6cbe04893aa0708d1fcaa45dd8eb3f53 upstream.

    Current code races in init/exit with interrupt handlers. This is noticed
    by the warning below. Fix it by using devres for ordering allocations and
    IRQ de/registration.

    WARNING: CPU: 0 PID: 827 at drivers/staging/wfx/bus_spi.c:142
wfx_spi_irq_handler+0x5c/0x64 [wfx]
    race condition in driver init/deinit

    Cc: stable@vger.kernel.org
    Fixes: 0096214a59a7 ("staging: wfx: add support for I/O access")
    Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Reviewed-by: Jérôme Pouiller <jerome.pouiller@silabs.com>
    Link:
https://lore.kernel.org/r/f0c66cbb3110c2736cd4357c753fba8c14ee3aee.1581416843.git.mirq-
linux@rere.qmqm.pl
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

```
commit 59ebb78aa7ebb0f8c1bc3079ee5f7a21937f3bb8
Author: Michał Mirosław <mirq-linux@rere.qmqm.pl>
Date:   Tue Feb 11 11:35:02 2020 +0100

    staging: wfx: add proper "compatible" string

    commit eec6e3ee636ec3adaa85ebe4b4acaacfcf06277e upstream.

    Add "compatible" string matching "vendor,chip" template and proper
    GPIO flags handling. Keep support for old name and reset polarity
    for older devicetrees.

    Cc: stable@vger.kernel.org   # d3a5bcb4a17f ("gpio: add gpiod_toggle_active_low()")
    Cc: stable@vger.kernel.org
    Fixes: 0096214a59a7 ("staging: wfx: add support for I/O access")
    Signed-off-by: Michał Mirosław <mirq-linux@rere.qmqm.pl>
    Link:
https://lore.kernel.org/r/0e6dda06f145676861860f073a53dc95987c7ab5.1581416843.git.mirq-
linux@rere.qmqm.pl
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5b8f88b78aef413039c22b93e0687267f3f79780
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Thu Mar 26 21:18:50 2020 +0800

    staging: wlan-ng: fix use-after-free Read in hfa384x_usbin_callback

    commit 1165dd73e811a07d947aee218510571f516081f6 upstream.

    We can't handle the case length > WLAN_DATA_MAXLEN.
    Because the size of rxfrm->data is WLAN_DATA_MAXLEN(2312), and we can't
    read more than that.

    Thanks-to: Hillf Danton <hdanton@sina.com>
    Reported-and-tested-by: syzbot+7d42d68643a35f71ac8a@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200326131850.17711-1-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dcbad6dd3aea65c16055c0617ec48bae1f0db7e5
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Wed Mar 25 15:06:46 2020 +0800

    staging: wlan-ng: fix ODEBUG bug in prism2sta_disconnect_usb

    commit a1f165a6b738f0c9d744bad4af7a53909278f5fc upstream.

    We should cancel hw->usb_work before kfree(hw).

    Reported-by: syzbot+6d2e7f6fa90e27be9d62@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/1585120006-30042-1-git-send-email-hqjagain@gmail.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8df70f84e08045628921a4f32dca7a1dd902f343
Author: Larry Finger <Larry.Finger@lwfinger.net>
Date:   Sat Mar 21 13:00:11 2020 -0500

    staging: rtl8188eu: Add ASUS USB-N10 Nano B1 to device table

    commit 38ef48f7d4b7342f145a1b4f96023bde99aeb245 upstream.

    The ASUS USB-N10 Nano B1 has been reported as a new RTL8188EU device.
```

```
    Add it to the device tables.

    Signed-off-by: Larry Finger <Larry.Finger@lwfinger.net>
    Reported-by: kovi <zraetn@gmail.com>
    Cc: Stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200321180011.26153-1-Larry.Finger@lwfinger.net
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 5bec92c1a178c9ab48acf2e0a6218e900c1b4721
Author: Dan Carpenter <dan.carpenter@oracle.com>
Date:   Mon Feb 24 13:33:25 2020 +0300

    staging: kpc2000: prevent underflow in cpld_reconfigure()

    commit 72db61d7d17a475d3cc9de1a7c871d518fcd82f0 upstream.

    This function should not allow negative values of "wr_val".  If
    negatives are allowed then capping the upper bound at 7 is
    meaningless.  Let's make it unsigned.

    Fixes: 7dc7967fc39a ("staging: kpc2000: add initial set of Daktronics drivers")
    Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
    Cc: stable <stable@vger.kernel.org>
    Link: https://lore.kernel.org/r/20200224103325.hrxdnaeqsthplu42@kili.mountain
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 58d9cb2e8263e98ea5f2c27ea5264e080030284a
Author: Johan Hovold <johan@kernel.org>
Date:   Mon Jan 13 18:18:18 2020 +0100

    media: usbtv: fix control-message timeouts

    commit 536f561d871c5781bc33d26d415685211b94032e upstream.

    The driver was issuing synchronous uninterruptible control requests
    without using a timeout. This could lead to the driver hanging on
    various user requests due to a malfunctioning (or malicious) device
    until the device is physically disconnected.

    The USB upper limit of five seconds per request should be more than
    enough.

    Fixes: f3d27f34fdd7 ("[media] usbtv: Add driver for Fushicai USBTV007 video frame
grabber")
    Fixes: c53a846c48f2 ("[media] usbtv: add video controls")
    Cc: stable <stable@vger.kernel.org>     # 3.11
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Acked-by: Lubomir Rintel <lkundrak@v3.sk>
    Reviewed-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Hans Verkuil <hverkuil-cisco@xs4all.nl>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 7601f24730efc88d97669445159b3ac25fa165cb
Author: Johan Hovold <johan@kernel.org>
Date:   Fri Jan 3 17:35:08 2020 +0100

    media: flexcop-usb: fix endpoint sanity check

    commit bca243b1ce0e46be26f7c63b5591dfbb41f558e5 upstream.

    commit 1b976fc6d684 ("media: b2c2-flexcop-usb: add sanity checking") added
    an endpoint sanity check to address a NULL-pointer dereference on probe.
    Unfortunately the check was done on the current altsetting which was later
    changed.
```

```
    Fix this by moving the sanity check to after the altsetting is changed.

    Fixes: 1b976fc6d684 ("media: b2c2-flexcop-usb: add sanity checking")
    Cc: Oliver Neukum <oneukum@suse.com>
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Sean Young <sean@mess.org>
    Signed-off-by: Mauro Carvalho Chehab <mchehab+huawei@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b699abf84d42ad1901a949e95fd2eab7ab8c2484
Author: Mans Rullgard <mans@mansr.com>
Date:   Mon Mar 16 16:11:35 2020 -0500

    usb: musb: fix crash with highmen PIO and usbmon

    commit 52974d94a206ce428d9d9b6eaa208238024be82a upstream.

    When handling a PIO bulk transfer with highmem buffer, a temporary
    mapping is assigned to urb->transfer_buffer.  After the transfer is
    complete, an invalid address is left behind in this pointer.  This is
    not ordinarily a problem since nothing touches that buffer before the
    urb is released.  However, when usbmon is active, usbmon_urb_complete()
    calls (indirectly) mon_bin_get_data() which does access the transfer
    buffer if it is set.  To prevent an invalid memory access here, reset
    urb->transfer_buffer to NULL when finished (musb_host_rx()), or do not
    set it at all (musb_host_tx()).

    Fixes: 8e8a55165469 ("usb: musb: host: Handle highmem in PIO mode")
    Signed-off-by: Mans Rullgard <mans@mansr.com>
    Cc: stable@vger.kernel.org
    Signed-off-by: Bin Liu <b-liu@ti.com>
    Link: https://lore.kernel.org/r/20200316211136.2274-8-b-liu@ti.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 052849e8698348b9f01ba10714ce5debe767b781
Author: Qiujun Huang <hqjagain@gmail.com>
Date:   Wed Mar 25 15:52:37 2020 +0800

    USB: serial: io_edgeport: fix slab-out-of-bounds read in edge_interrupt_callback

    commit 57aa9f294b09463492f604feaa5cc719beaace32 upstream.

    Fix slab-out-of-bounds read in the interrupt-URB completion handler.

    The boundary condition should be (length - 1) as we access
    data[position + 1].

    Reported-and-tested-by: syzbot+37ba33391ad5f3935bbd@syzkaller.appspotmail.com
    Signed-off-by: Qiujun Huang <hqjagain@gmail.com>
    Fixes: 1da177e4c3f4 ("Linux-2.6.12-rc2")
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 785cd31f6f67f250fed8d7a4e9f4df81b673649f
Author: Matthias Reichl <hias@horus.com>
Date:   Fri Mar 27 16:03:50 2020 +0100

    USB: cdc-acm: restore capability check order

    commit 62d65bdd9d05158aa2547f8ef72375535f3bc6e3 upstream.

    commit b401f8c4f492c ("USB: cdc-acm: fix rounding error in TIOCSSERIAL")
```

```
    introduced a regression by changing the order of capability and close
    settings change checks. When running with CAP_SYS_ADMIN setting the
    close settings to the values already set resulted in -EOPNOTSUPP.

    Fix this by changing the check order back to how it was before.

    Fixes: b401f8c4f492c ("USB: cdc-acm: fix rounding error in TIOCSSERIAL")
    Cc: Anthony Mallet <anthony.mallet@laas.fr>
    Cc: stable <stable@vger.kernel.org>
    Cc: Oliver Neukum <oneukum@suse.com>
    Signed-off-by: Matthias Reichl <hias@horus.com>
    Link: https://lore.kernel.org/r/20200327150350.3657-1-hias@horus.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e52936edccc95a54c477118e91364b53a2ceeed9
Author: Pawel Dembicki <paweldembicki@gmail.com>
Date:   Wed Mar 25 06:44:19 2020 +0100

    USB: serial: option: add Wistron Neweb D19Q1

    commit dfee7e2f478346b12ea651d5c28b069f6a4af563 upstream.

    This modem is embedded on dlink dwr-960 router.
    The oem configuration states:

    T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 2 Spd=480 MxCh= 0
    D: Ver= 2.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
    P: Vendor=1435 ProdID=d191 Rev=ff.ff
    S: Manufacturer=Android
    S: Product=Android
    S: SerialNumber=0123456789ABCDEF
    C:* #Ifs= 6 Cfg#= 1 Atr=80 MxPwr=500mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
    E: Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=42 Prot=01 Driver=(none)
    E: Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=(none)
    E: Ad=84(I) Atr=03(Int.) MxPS= 10 Ivl=32ms
    E: Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=(none)
    E: Ad=86(I) Atr=03(Int.) MxPS= 10 Ivl=32ms
    E: Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=qmi_wwan
    E: Ad=88(I) Atr=03(Int.) MxPS= 8 Ivl=32ms
    E: Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 5 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=(none)
    E: Ad=89(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E: Ad=06(O) Atr=02(Bulk) MxPS= 512 Ivl=125us

    Tested on openwrt distribution

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

commit 8264e939338e6532d79bbc484780c4a9e841c01c
Author: Pawel Dembicki <paweldembicki@gmail.com>
Date:   Wed Mar 25 06:44:18 2020 +0100
```

```
    USB: serial: option: add BroadMobi BM806U

    commit 6cb2669cb97fc4fdf526127159ac59caae052247 upstream.

    BroadMobi BM806U is an Qualcomm MDM9225 based 3G/4G modem.
    Tested hardware BM806U is mounted on D-Link DWR-921-C3 router.

    T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#=  2 Spd=480  MxCh= 0
    D:  Ver= 2.01 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=2020 ProdID=2033 Rev= 2.28
    S:  Manufacturer=Mobile Connect
    S:  Product=Mobile Connect
    S:  SerialNumber=f842866cfd5a
    C:* #Ifs= 5 Cfg#= 1 Atr=80 MxPwr=500mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
    E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=83(I) Atr=03(Int.) MxPS=  10 Ivl=32ms
    E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=85(I) Atr=03(Int.) MxPS=  10 Ivl=32ms
    E:  Ad=84(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
    E:  Ad=87(I) Atr=03(Int.) MxPS=  10 Ivl=32ms
    E:  Ad=86(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=qmi_wwan
    E:  Ad=89(I) Atr=03(Int.) MxPS=   8 Ivl=32ms
    E:  Ad=88(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms

    Co-developed-by: Cezary Jackiewicz <cezary@eko.one.pl>
    Signed-off-by: Cezary Jackiewicz <cezary@eko.one.pl>
    Signed-off-by: Pawel Dembicki <paweldembicki@gmail.com>
    Cc: stable <stable@vger.kernel.org>
    Signed-off-by: Johan Hovold <johan@kernel.org>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c68a7be319eea030976be6a2382af04fa9a353b2
Author: Pawel Dembicki <paweldembicki@gmail.com>
Date:   Wed Mar 25 06:44:17 2020 +0100

    USB: serial: option: add support for ASKEY WWHC050

    commit 007d20dca2376a751b1dad03442f118438b7e65e upstream.

    ASKEY WWHC050 is a mcie LTE modem.
    The oem configuration states:

    T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=480  MxCh= 0
    D:  Ver= 2.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
    P:  Vendor=1690 ProdID=7588 Rev=ff.ff
    S:  Manufacturer=Android
    S:  Product=Android
    S:  SerialNumber=813f0eef6e6e
    C:* #Ifs= 6 Cfg#= 1 Atr=80 MxPwr=500mA
    I:* If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
    E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=01(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    I:* If#= 1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=42 Prot=01 Driver=(none)
    E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
    E:  Ad=82(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
```

```
      I:* If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=84(I) Atr=03(Int.) MxPS=  10 Ivl=32ms
      E:  Ad=83(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=03(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
      E:  Ad=86(I) Atr=03(Int.) MxPS=  10 Ivl=32ms
      E:  Ad=85(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=04(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=qmi_wwan
      E:  Ad=88(I) Atr=03(Int.) MxPS=   8 Ivl=32ms
      E:  Ad=87(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=05(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      I:* If#= 5 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=(none)
      E:  Ad=89(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
      E:  Ad=06(O) Atr=02(Bulk) MxPS= 512 Ivl=125us

      Tested on openwrt distribution.

      Co-developed-by: Cezary Jackiewicz <cezary@eko.one.pl>
      Signed-off-by: Cezary Jackiewicz <cezary@eko.one.pl>
      Signed-off-by: Pawel Dembicki <paweldembicki@gmail.com>
      Cc: stable <stable@vger.kernel.org>
      Signed-off-by: Johan Hovold <johan@kernel.org>
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

  commit 6797143df51c8ae259aa4bfe4e99c832b20bde8a
  Author: Daniel Borkmann <daniel@iogearbox.net>
  Date:   Mon Mar 30 18:03:22 2020 +0200

      bpf: Undo incorrect __reg_bound_offset32 handling

      commit f2d67fec0b43edce8c416101cdc52e71145b5fef upstream.

      Anatoly has been fuzzing with kBdysch harness and reported a hang in
      one of the outcomes:

        0: (b7) r0 = 808464432
        1: (7f) r0 >>= r0
        2: (14) w0 -= 808464432
        3: (07) r0 += 808464432
        4: (b7) r1 = 808464432
        5: (de) if w1 s<= w0 goto pc+0
         R0_w=invP(id=0,umin_value=808464432,umax_value=5103431727,var_off=
  (0x30303020;0x10000001f)) R1_w=invP808464432 R10=fp0
        6: (07) r0 += -2144337872
        7: (14) w0 -= -1607454672
        8: (25) if r0 > 0x30303030 goto pc+0
         R0_w=invP(id=0,umin_value=271581184,umax_value=271581311,var_off=(0x10300000;0x7f))
  R1_w=invP808464432 R10=fp0
        9: (76) if w0 s>= 0x303030 goto pc+2
        12: (95) exit

        from 8 to 9: safe

        from 5 to 6: R0_w=invP(id=0,umin_value=808464432,umax_value=5103431727,var_off=
  (0x30303020;0x10000001f)) R1_w=invP808464432 R10=fp0
        6: (07) r0 += -2144337872
        7: (14) w0 -= -1607454672
        8: (25) if r0 > 0x30303030 goto pc+0
         R0_w=invP(id=0,umin_value=271581184,umax_value=271581311,var_off=(0x10300000;0x7f))
  R1_w=invP808464432 R10=fp0
        9: safe

        from 8 to 9: safe
        verification time 589 usec
```

```
       stack depth 0
       processed 17 insns (limit 1000000) [...]
```

The underlying program was xlated as follows:

```
       # bpftool p d x i 9
        0: (b7) r0 = 808464432
        1: (7f) r0 >>= r0
        2: (14) w0 -= 808464432
        3: (07) r0 += 808464432
        4: (b7) r1 = 808464432
        5: (de) if w1 s<= w0 goto pc+0
        6: (07) r0 += -2144337872
        7: (14) w0 -= -1607454672
        8: (25) if r0 > 0x30303030 goto pc+0
        9: (76) if w0 s>= 0x303030 goto pc+2
       10: (05) goto pc-1
       11: (05) goto pc-1
       12: (95) exit
```

The verifier rewrote original instructions it recognized as dead code with 'goto pc-1', but reality differs from verifier simulation in that we're actually able to trigger a hang due to hitting the 'goto pc-1' instructions.

Taking different examples to make the issue more obvious: in this example we're probing bounds on a completely unknown scalar variable in r1:

```
       [...]
       5: R0_w=inv1 R1_w=inv(id=0) R10=fp0
       5: (18) r2 = 0x4000000000
       7: R0_w=inv1 R1_w=inv(id=0) R2_w=inv274877906944 R10=fp0
       7: (18) r3 = 0x2000000000
       9: R0_w=inv1 R1_w=inv(id=0) R2_w=inv274877906944 R3_w=inv137438953472 R10=fp0
       9: (18) r4 = 0x400
       11: R0_w=inv1 R1_w=inv(id=0) R2_w=inv274877906944 R3_w=inv137438953472 R4_w=inv1024
R10=fp0
       11: (18) r5 = 0x200
       13: R0_w=inv1 R1_w=inv(id=0) R2_w=inv274877906944 R3_w=inv137438953472 R4_w=inv1024
R5_w=inv512 R10=fp0
       13: (2d) if r1 > r2 goto pc+4
        R0_w=inv1 R1_w=inv(id=0,umax_value=274877906944,var_off=(0x0; 0x7fffffffff))
R2_w=inv274877906944 R3_w=inv137438953472 R4_w=inv1024 R5_w=inv512 R10=fp0
       14: R0_w=inv1 R1_w=inv(id=0,umax_value=274877906944,var_off=(0x0; 0x7fffffffff))
R2_w=inv274877906944 R3_w=inv137438953472 R4_w=inv1024 R5_w=inv512 R10=fp0
       14: (ad) if r1 < r3 goto pc+3
        R0_w=inv1 R1_w=inv(id=0,umin_value=137438953472,umax_value=274877906944,var_off=(0x0;
0x7fffffffff)) R2_w=inv274877906944 R3_w=inv137438953472 R4_w=inv1024 R5_w=inv512 R10=fp0
       15: R0=inv1 R1=inv(id=0,umin_value=137438953472,umax_value=274877906944,var_off=(0x0;
0x7fffffffff)) R2=inv274877906944 R3=inv137438953472 R4=inv1024 R5=inv512 R10=fp0
       15: (2e) if w1 > w4 goto pc+2
        R0=inv1 R1=inv(id=0,umin_value=137438953472,umax_value=274877906944,var_off=(0x0;
0x7f00000000)) R2=inv274877906944 R3=inv137438953472 R4=inv1024 R5=inv512 R10=fp0
       16: R0=inv1 R1=inv(id=0,umin_value=137438953472,umax_value=274877906944,var_off=(0x0;
0x7f00000000)) R2=inv274877906944 R3=inv137438953472 R4=inv1024 R5=inv512 R10=fp0
       16: (ae) if w1 < w5 goto pc+1
        R0=inv1 R1=inv(id=0,umin_value=137438953472,umax_value=274877906944,var_off=(0x0;
0x7f00000000)) R2=inv274877906944 R3=inv137438953472 R4=inv1024 R5=inv512 R10=fp0
       [...]
```

We're first probing lower/upper bounds via jmp64, later we do a similar check via jmp32 and examine the resulting var_off there. After fall-through in insn 14, we get the following bounded r1 with 0x7fffffffff unknown marked bits in the variable section.

Thus, after knowing r1 <= 0x4000000000 and r1 >= 0x2000000000:

```
  max: 0b1000000000000000000000000000000000 / 0x4000000000
  var: 0b0111111111111111111111111111111111 / 0x7fffffffff
  min: 0b0100000000000000000000000000000000 / 0x2000000000
```

Now, in insn 15 and 16, we perform a similar probe with lower/upper bounds in jmp32.

Thus, after knowing r1 <= 0x4000000000 and r1 >= 0x2000000000 and w1 <= 0x400 and w1 >= 0x200:

```
  max: 0b1000000000000000000000000000000000 / 0x4000000000
  var: 0b0111111100000000000000000000000000 / 0x7f00000000
  min: 0b0100000000000000000000000000000000 / 0x2000000000
```

The lower/upper bounds haven't changed since they have high bits set in u64 space and the jmp32 tests can only refine bounds in the low bits.

However, for the var part the expectation would have been 0x7f000007ff or something less precise up to 0x7fffffffff. A outcome of 0x7f00000000 is not correct since it would contradict the earlier probed bounds where we know that the result should have been in [0x200,0x400] in u32 space. Therefore, tests with such info will lead to wrong verifier assumptions later on like falsely predicting conditional jumps to be always taken, etc.

The issue here is that __reg_bound_offset32()'s implementation from commit 581738a681b6 ("bpf: Provide better register bounds after jmp32 instructions") makes an incorrect range assumption:

```
  static void __reg_bound_offset32(struct bpf_reg_state *reg)
  {
        u64 mask = 0xffffFFFF;
        struct tnum range = tnum_range(reg->umin_value & mask,
                                       reg->umax_value & mask);
        struct tnum lo32 = tnum_cast(reg->var_off, 4);
        struct tnum hi32 = tnum_lshift(tnum_rshift(reg->var_off, 32), 32);

        reg->var_off = tnum_or(hi32, tnum_intersect(lo32, range));
  }
```

In the above walk-through example, __reg_bound_offset32() as-is chose a range after masking with 0xffffffff of [0x0,0x0] since umin:0x2000000000 and umax:0x4000000000 and therefore the lo32 part was clamped to 0x0 as well. However, in the umin:0x2000000000 and umax:0x4000000000 range above we'd end up with an actual possible interval of [0x0,0xffffffff] for u32 space instead.

In case of the original reproducer, the situation looked as follows at insn 5 for r0:

```
  [...]
  5: R0_w=invP(id=0,umin_value=808464432,umax_value=5103431727,var_off=(0x0;
0x1fffffffff)) R1_w=invP808464432 R10=fp0
                              0x30303030           0x13030302f
  5: (de) if w1 s<= w0 goto pc+0
   R0_w=invP(id=0,umin_value=808464432,umax_value=5103431727,var_off=(0x30303020;
0x10000001f)) R1_w=invP808464432 R10=fp0
                             0x30303030           0x13030302f
  [...]
```

After the fall-through, we similarly forced the var_off result into the wrong range [0x30303030,0x3030302f] suggesting later on that fixed bits must only be of 0x30303020 with 0x10000001f unknowns whereas such assumption can only be made when both bounds in hi32 range match.

```
    Originally, I was thinking to fix this by moving reg into a temp reg and
    use proper coerce_reg_to_size() helper on the temp reg where we can then
    based on that define the range tnum for later intersection:

      static void __reg_bound_offset32(struct bpf_reg_state *reg)
      {
            struct bpf_reg_state tmp = *reg;
            struct tnum lo32, hi32, range;

            coerce_reg_to_size(&tmp, 4);
            range = tnum_range(tmp.umin_value, tmp.umax_value);
            lo32 = tnum_cast(reg->var_off, 4);
            hi32 = tnum_lshift(tnum_rshift(reg->var_off, 32), 32);
            reg->var_off = tnum_or(hi32, tnum_intersect(lo32, range));
      }

    In the case of the concrete example, this gives us a more conservative unknown
    section. Thus, after knowing r1 <= 0x4000000000 and r1 >= 0x2000000000 and
                                w1 <= 0x400        and w1 >= 0x200:

      max: 0b1000000000000000000000000000000000000000 / 0x4000000000
      var: 0b1111111111111111111111111111111111111111 / 0x7fffffffff
      min: 0b0100000000000000000000000000000000000000 / 0x2000000000

    However, above new __reg_bound_offset32() has no effect on refining the
    knowledge of the register contents. Meaning, if the bounds in hi32 range
    mismatch we'll get the identity function given the range reg spans
    [0x0,0xffffffff] and we cast var_off into lo32 only to later on binary
    or it again with the hi32.

    Likewise, if the bounds in hi32 range match, then we mask both bounds
    with 0xffffffff, use the resulting umin/umax for the range to later
    intersect the lo32 with it. However, _prior_ called __reg_bound_offset()
    did already such intersection on the full reg and we therefore would only
    repeat the same operation on the lo32 part twice.

    Given this has no effect and the original commit had false assumptions,
    this patch reverts the code entirely which is also more straight forward
    for stable trees: apparently 581738a681b6 got auto-selected by Sasha's
    ML system and misclassified as a fix, so it got sucked into v5.4 where
    it should never have landed. A revert is low-risk also from a user PoV
    since it requires a recent kernel and llc to opt-into -mcpu=v3 BPF CPU
    to generate jmp32 instructions. A proper bounds refinement would need a
    significantly more complex approach which is currently being worked, but
    no stable material [0]. Hence revert is best option for stable. After the
    revert, the original reported program gets rejected as follows:

      1: (7f) r0 >>= r0
      2: (14) w0 -= 808464432
      3: (07) r0 += 808464432
      4: (b7) r1 = 808464432
      5: (de) if w1 s<= w0 goto pc+0
       R0_w=invP(id=0,umin_value=808464432,umax_value=5103431727,var_off=(0x0; 0x1fffffff))
R1_w=invP808464432 R10=fp0
      6: (07) r0 += -2144337872
      7: (14) w0 -= -1607454672
      8: (25) if r0 > 0x30303030 goto pc+0
       R0_w=invP(id=0,umax_value=808464432,var_off=(0x0; 0x3fffffff)) R1_w=invP808464432
R10=fp0
      9: (76) if w0 s>= 0x303030 goto pc+2
       R0=invP(id=0,umax_value=3158063,var_off=(0x0; 0x3fffff)) R1=invP808464432 R10=fp0
      10: (30) r0 = *(u8 *)skb[808464432]
      BPF_LD_[ABS|IND] uses reserved fields
      processed 11 insns (limit 1000000) [...]
```

```
       [0] https://lore.kernel.org/bpf/158507130343.15666.8018068546764556975.stgit@john-
  Precision-5820-Tower/T/

      Fixes: 581738a681b6 ("bpf: Provide better register bounds after jmp32 instructions")
      Reported-by: Anatoly Trosinenko <anatoly.trosinenko@gmail.com>
      Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
      Signed-off-by: Alexei Starovoitov <ast@kernel.org>
      Link: https://lore.kernel.org/bpf/20200330160324.15259-2-daniel@iogearbox.net
      Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```