

linux-media.vger.kernel.org archive mirror

search help / color / mirror / Atom feed

From: imv4bel@gmail.com  
To: mchehab@kernel.org  
Cc: Hyunwoo Kim <imv4bel@gmail.com>, kernel@tuxforce.de, linux-media@vger.kernel.org, linux-usb@vger.kernel.org, cai.huqing@linux.dev, tiwai@suse.de  
Subject: [PATCH 0/4] Fix multiple race condition vulnerabilities in dvb-core and device driver  
Date: Tue, 15 Nov 2022 05:18:18 -0800 [thread overview]  
Message-ID: <20221115131822.6640-1-imv4bel@gmail.com> (raw)

From: Hyunwoo Kim <imv4bel@gmail.com>

Dear,

This patch set is a security patch for various race condition vulnerabilities that occur in 'dvb-core' and 'tusb\_dec', a dvb-based device driver.

# 1. media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_frontend  
This is a security patch for a race condition that occurs in the dvb\_frontend system of dvb-core.

The race condition that occurs here will occur with \_any\_ device driver using dvb\_frontend.

The race conditions that occur in dvb\_frontend are as follows  
(Description is based on drivers/media/usb/as102/as102\_drv.c using dvb\_frontend):

```
...
        cpu0                                cpu1
1. open()
   dvb_frontend_open()
   dvb_frontend_get() // kref : 3

2. as102_usb_disconnect()
   as102_dvb_unregister()
   dvb_unregister_frontend()
   dvb_frontend_put() // kref : 2
   dvb_frontend_detach()
   dvb_frontend_put() // kref : 1

3. close()
   fput()
   dvb_frontend_release()
   dvb_frontend_put() // kref : 0
   dvb_frontend_free()
   dvb_frontend_free()
   dvb_free_device()
   kfree (dvbdev->fops);
   ...
   fops_put(file->f_op); // UAF!!
...
```

UAF occurs in the following order: '.probe -> open() -> .disconnect -> close()'.

The root cause of this is that wake\_up() for dvbdev->wait\_queue is implemented in the dvb\_frontend\_release() function, but wait\_event() is not implemented in the dvb\_frontend\_stop() function.

The KASAN log caused by this is as follows:

```
...
[ 60.754938] =====
[ 60.754942] BUG: KASAN: use-after-free in __fput+0xa55/0xaf0
[ 60.754945] Read of size 8 at addr ffff88b134ddf000 by task as102_test/2139

[ 60.754949] CPU: 3 PID: 2139 Comm: as102_test Not tainted 6.1.0-rc2+ #16
[ 60.754951] Hardware name: Gigabyte Technology Co., Ltd. B460MDS3H/B460M DS3H, BIOS F3 05/27/2020
[ 60.754953] Call Trace:
[ 60.754954] <TASK>
[ 60.754956] dump_stack_lvl+0x49/0x63
[ 60.754958] print_report+0x177/0x46e
[ 60.754962] ? kasan_complete_mode_report_info+0x7c/0x210
[ 60.754965] ? __fput+0xa55/0xaf0
[ 60.754970] kasan_report+0xb0/0x140
[ 60.754976] ? __fput+0xa55/0xaf0
[ 60.754979] ? asan_report_load0_noabort+0x14/0x20
[ 60.754982] __fput+0xa55/0xaf0
[ 60.754985] __fput+0xe/0x20
[ 60.754987] task_work_run+0x153/0x240
[ 60.754991] ? task_work_cancel+0x20/0x20
[ 60.754994] ? fput+0xab/0x140
[ 60.754997] exit_to_user_mode_prepare+0x18f/0x1a0
[ 60.754999] syscall_exit_to_user_mode+0x26/0x50
[ 60.755003] do_syscall_64+0x69/0x90
[ 60.755005] ? do_syscall_64+0x69/0x90
[ 60.755008] ? debug_smp_processor_id+0x17/0x20
[ 60.755010] ? fpregs_assert_state_consistent+0x52/0xc0
[ 60.755013] ? exit_to_user_mode_prepare+0x49/0x1a0
[ 60.755015] ? irqentry_exit_to_user_mode+0x9/0x20
[ 60.755018] ? irqentry_exit+0x3b/0x50
[ 60.755021] ? sysvec_aplic_timer_interrupt+0x57/0xc0
[ 60.755024] entry_SYSCALL_64_after_hwframe+0x63/0xcd
[ 60.755027] RIP: 0033:0x4537eb
[ 60.755029] Code: 03 00 00 00 0f 05 48 3d 00 f0 ff ff 77 41 c3 48 83 ec 18 89 7c 24 0c e8 c3 a8 02 00 8b 7c 24 0c 41 89 c0 b8 03 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 35 44 89 c7 89 44
24 0c e8 11 a9 02 00 8b 44
[ 60.755031] RSP: 002b:00007ff8c1c001a0 EFLAGS: 00000293 ORIG_RAX: 0000000000000003
[ 60.755034] RAX: 0000000000000000 RBX: 00007ff8c1c00640 RCX: 000000000004537eb
[ 60.755036] RDX: 0000000000000000 RSI: 00007ff8bc000b70 RDI: 0000000000000003
[ 60.755038] RBP: 00007ff8c1c001a0 R08: 0000000000000000 R09: 0000000000000000
[ 60.755040] R10: 000000000000000a R11: 0000000000000293 R12: 00007ff8c1c00640
[ 60.755042] R13: 0000000000000000 R14: 000000000041b290 R15: 00007ff8c1400000
[ 60.755044] </TASK>

[ 60.755047] Allocated by task 2114:
[ 60.755049] kasan_save_stack+0x26/0x50
[ 60.755052] kasan_set_track+0x25/0x40
[ 60.755054] kasan_save_alloc_info+0x1e/0x30
[ 60.755056] kasan_kmalloc+0xb4/0xc0
[ 60.755058] __kmalloc_node_track_caller+0x66/0x160
[ 60.755061] kmemdup+0x23/0x50
[ 60.755063] dvb_register_device+0x1cd/0x15c0 [dvb_core]
[ 60.755070] dvb_register_frontend+0x3cb/0x630 [dvb_core]
[ 60.755078] as102_dvb_register+0x335/0x4d0 [dvb_as102]
[ 60.755083] as102_usb_probe.cold+0x680/0x6eb [dvb_as102]
[ 60.755087] usb_probe_interface+0x266/0x740
[ 60.755089] really_probe+0x1fa/0xa80
[ 60.755092] driver_probe_device+0x2cb/0x490
[ 60.755094] driver_probe_device+0x4e/0x140
[ 60.755096] driver_attach+0x1a3/0x520
[ 60.755098] bus_for_each_dev+0x11e/0x1c0
[ 60.755100] driver_attach+0x3d/0x60
[ 60.755102] bus_add_driver+0x449/0x5a0
[ 60.755103] driver_register+0x219/0x390
[ 60.755105] usb_register_driver+0x228/0x400
[ 60.755107] 0xfffffff0c04c8023
[ 60.755110] do_one_initcall+0x97/0x310
[ 60.755113] do_init_module+0x19a/0x630
[ 60.755115] load_module+0x6ca4/0x7d90
[ 60.755117] __do_sys_finit_module+0x134/0x1d0
[ 60.755119] __x64_sys_finit_module+0x72/0xb0
[ 60.755121] do_syscall_64+0x59/0x90
[ 60.755123] entry_SYSCALL_64_after_hwframe+0x63/0xcd

[ 60.755126] Freed by task 2139:
[ 60.755128] kasan_save_stack+0x26/0x50
[ 60.755130] kasan_set_track+0x25/0x40
[ 60.755132] kasan_save_free_info+0x2e/0x50
[ 60.755134] kasan_slab_free+0x174/0x1e0
[ 60.755136] kasan_slab_free+0x12/0x20
[ 60.755138] slab_free_freelist_hook+0xd0/0x1a0
[ 60.755140] kmem_cache_free+0x193/0x2c0
[ 60.755143] kfree+0x79/0x120
```

```
[ 60.755145] dvb_free_device+0x39/0x60 [dvb_core]
[ 60.755151] dvb_frontend_put.cold+0xa6/0x15a [dvb_core]
[ 60.755160] dvb_frontend_release.cold+0xc7/0xf6 [dvb_core]
[ 60.755167] __fput+0x2ce/0xaf0
[ 60.755169] __fput+0xe/0x20
[ 60.755171] task_work_run+0x153/0x240
[ 60.755173] exit_to_user_mode_prepare+0x18f/0x1a0
[ 60.755175] syscall_exit_to_user_mode+0x26/0x50
[ 60.755177] do_syscall_64+0x69/0x90
[ 60.755179] entry_SYSCALL_64_after_hwframe+0x63/0xcd
...
```

Also, UAF can occur for driver-specific structures (such as 'struct XXX\_dev'):

```
...
    cpu0                                cpu1
1. open()
   dvb_frontend_open()

2. as102_usb_disconnect()
   kref_put(&as102_dev->kref, as102_usb_release); // kref : 0
   as102_usb_release()
   kfree(as102_dev);

3. close()
   dvb_frontend_release()
   mutex_lock(&fe->dvb->mdev_lock); // UAF
...
```

The KASAN log caused by this is as follows:

```
...
[ 82.144178] =====
[ 82.144182] BUG: KASAN: use-after-free in mutex_lock+0x81/0xe0
[ 82.144189] Write of size 8 at addr ffff888121b6a168 by task as102_test/2356

[ 82.144193] CPU: 12 PID: 2356 Comm: as102_test Not tainted 6.1.0-rc2+ #16
[ 82.144196] Hardware name: Gigabyte Technology Co., Ltd. B460MDS3H/B460M DS3H, BIOS F3 05/27/2020
[ 82.144198] Call Trace:
[ 82.144200] <TASK>
[ 82.144201] dump_stack_lvl+0x49/0x63
[ 82.144205] print_report+0x177/0x46e
[ 82.144208] ? kasan_complete_mode_report_info+0x7c/0x210
[ 82.144212] ? mutex_lock+0x81/0xe0
[ 82.144215] kasan_report+0xb0/0x140
[ 82.144218] ? mutex_lock+0x81/0xe0
[ 82.144222] kasan_check_range+0x3a/0x1d0
[ 82.144224] __kasan_check_write+0x14/0x20
[ 82.144226] mutex_lock+0x81/0xe0
[ 82.144229] ? __mutex_lock_slowpath+0x20/0x20
[ 82.144234] dvb_frontend_release.cold+0x178/0x4d2 [dvb_core]
[ 82.144246] __fput+0x2ce/0xaf0
[ 82.144250] __fput+0xe/0x20
[ 82.144253] task_work_run+0x153/0x240
[ 82.144257] ? task_work_cancel+0x20/0x20
[ 82.144260] ? fput+0xab/0x140
[ 82.144263] exit_to_user_mode_prepare+0x18f/0x1a0
[ 82.144266] syscall_exit_to_user_mode+0x26/0x50
[ 82.144270] do_syscall_64+0x69/0x90
[ 82.144273] ? debug_smp_processor_id+0x17/0x20
[ 82.144275] ? fpregs_assert_state_consistent+0x52/0xc0
[ 82.144279] ? exit_to_user_mode_prepare+0x49/0x1a0
[ 82.144281] ? irqentry_exit_to_user_mode+0x9/0x20
[ 82.144284] ? irqentry_exit+0x3b/0x50
[ 82.144287] ? sysvec_apic_timer_interrupt+0x57/0xc0
[ 82.144290] entry_SYSCALL_64_after_hwframe+0x63/0xcd
[ 82.144293] RIP: 0033:0x4537eb
[ 82.144296] Code: 03 00 00 00 0f 05 48 3d 00 f0 ff ff 77 41 c3 48 83 ec 18 89 7c 24 0c e8 c3 a8 02 00 8b 7c 24 0c 41 89 c0 b8 03 00 00 0f 05 <48> 3d 00 f0 ff ff 77 35 44 89 c7 89 44
24 0c e8 11 a9 02 00 8b 44
[ 82.144298] RSP: 002b:00007fc7600001a0 EFLAGS: 00000293 ORIG_RAX: 0000000000000003
[ 82.144302] RAX: 0000000000000000 RBX: 00007fc760000640 RCX: 000000000004537eb
[ 82.144304] RDX: 0000000000000000 RSI: 00007fc758000b70 RDI: 0000000000000003
[ 82.144306] RBP: 00007fc7600001d0 R08: 0000000000000000 R09: 0000000000000000
[ 82.144308] R10: 000000000000000a R11: 0000000000000293 R12: 00007fc760000640
[ 82.144310] R13: 0000000000000000 R14: 000000000041b290 R15: 00007fc758000000
[ 82.144313] </TASK>

[ 82.144315] Allocated by task 2225:
[ 82.144317] kasan_save_stack+0x26/0x50
[ 82.144320] kasan_set_track+0x25/0x40
[ 82.144322] kasan_save_alloc_info+0x1e/0x30
[ 82.144325] __kasan_kmalloc+0xb4/0xc0
[ 82.144327] kmalloc_trace+0x4a/0xb0
[ 82.144329] as102_usb_probe.cold+0x58/0x6eb [dvb_as102]
[ 82.144335] usb_probe_interface+0x266/0x740
[ 82.144338] really_probe+0x1fa/0xa80
[ 82.144341] __driver_probe_device+0x2cb/0x490
[ 82.144343] driver_probe_device+0x4e/0x140
[ 82.144345] driver_attach+0x1a3/0x520
[ 82.144347] bus_for_each_dev+0x11e/0x1c0
[ 82.144348] driver_attach+0x3d/0x60
[ 82.144350] bus_add_driver+0x449/0x5a0
[ 82.144352] driver_register+0x219/0x390
[ 82.144354] usb_register_driver+0x228/0x400
[ 82.144356] 0xfffffffefc0655023
[ 82.144358] do_one_initcall+0x97/0x310
[ 82.144361] do_init_module+0x19a/0x630
[ 82.144363] load_module+0x6ca4/0x7d90
[ 82.144365] __do_sys_finit_module+0x134/0x1d0
[ 82.144367] __x64_sys_finit_module+0x72/0xb0
[ 82.144369] do_syscall_64+0x59/0x90
[ 82.144371] entry_SYSCALL_64_after_hwframe+0x63/0xcd

[ 82.144374] Freed by task 158:
[ 82.144376] kasan_save_stack+0x26/0x50
[ 82.144378] kasan_set_track+0x25/0x40
[ 82.144380] kasan_save_free_info+0x2e/0x50
[ 82.144382] __kasan_slab_free+0x174/0x1e0
[ 82.144384] __kasan_slab_free+0x12/0x20
[ 82.144386] slab_free_freelist_hook+0xd0/0x1a0
[ 82.144388] kmem_cache_free+0x193/0x2c0
[ 82.144391] kfree+0x79/0x120
[ 82.144393] as102_usb_release+0x5d/0x75 [dvb_as102]
[ 82.144397] as102_usb_disconnect+0x125/0x176 [dvb_as102]
[ 82.144400] usb_unbind_interface+0x187/0x7c0
[ 82.144402] device_remove+0x117/0x170
[ 82.144404] device_release_driver_internal+0x418/0x660
[ 82.144407] device_release_driver+0x12/0x20
[ 82.144409] bus_remove_device+0x28f/0x540
[ 82.144410] device_del+0x501/0xc30
[ 82.144413] usb_disable_device+0x2a5/0x660
[ 82.144415] usb_disconnect.cold+0x1f9/0x620
[ 82.144417] hub_event+0x16d3/0x3d20
[ 82.144420] process_one_work+0x778/0x11c0
[ 82.144422] worker_thread+0x544/0x1180
[ 82.144424] kthread+0x280/0x320
[ 82.144426] ret_from_fork+0x1f/0x30
...
```

# 2. media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_net  
This is a security patch for a race condition that occurs in the dvb\_net system of dvb-core.

The race condition that occurs here will occur with \_any\_ device driver using dvb\_net.

The race condition that occurs in dvb\_net is:

```
...
    cpu0                                cpu1
1. .disconnect()
   dvb_net_release()
   dvbnet->exit = 1;
   if (dvbnet->dvbdev->users < 1) // improper reference counting
2. open()
   dvb_device_open()
...
```

```
3. dvb_unregister_device()
   dvb_remove_device()
   down_write(&minor_rwlock);
   dvb_minors[dvbdev->minor] = NULL;
   up_write(&minor_rwlock);
   dvb_free_device()
   kfree(dvbdev->fops);
```

...

The root cause of this is that you use the `dvb_device_open()` function, which does not implement a conditional statement that checks `'dvbnet->exit'`.

The KASAN log caused by this is as follows:

```
[ 952.372690] CPU: 3 PID: 2522 Comm: dvb_net_test Not tainted 6.1.0-rc2+ #16
[ 952.372707] Hardware name: Gigabyte Technology Co., Ltd. B460MDS3H/B460M DS3H, BIOS F3 05/27/2020
[ 952.372718] Call Trace:
```

```

952.373707] Freed by task 592:
952.373717]   kasan_save_stack+0x26/0x50
952.373731]   kasan_set_track+0x20/0x40
952.373744]   kasan_save_info+0x2e/0x50
952.373754]   __kasan_slab_free+0x12/0x1e0
952.373767]   __kasan_slab_free+0x12/0x20
952.373780]   slab_free freelist hook+0x20/0x1a0
952.373793]   kmem_cache_free+0x193/0x2c0
952.373805]   kfree+0x79/0x120
952.373817]   dvb_free_device.part.0+0x33/0x70 [dvb_core]
952.373830]   dvb_unregister_device+0x40/0x54 [dvb_core]
952.373905]   dvb_net_release+0x264/0x316 [dvb_core]
952.373952]   ttusb_dec_disconnect+0x391/0x4e1 [ttusb_dec]
952.373973]   usb_unbind_interface+0x187/0x7c0
952.373986]   device_remove+0x117/0x170
952.373997]   device_release_driver_internal+0x418/0x660
952.374010]   device_remove+0x117/0x170
952.374022]   bus_remove_device+0x28/0x540
952.374032]   device_del+0x501/0xc30
952.374047]   usb_disable_device+0x2a5/0x660
952.374058]   usb_disconnect.cold+0x1f9/0xf60
952.374070]   hub_event+0x1d3/0x3d20
952.374084]   process_one_work+0x73d/0x1c0
952.374096]   worker_thread+0x54/0x180
952.374107]   kthread+0x280/0x320
952.374117]   ret from fork+0x1f/0x30

```

This race condition can occur anywhere the `dvb_register_device()` function is called: `dvb demux`, `dvb dvr`, `dvb frontend`, `dvb net`, etc.

The race condition flow is as follows (dvb\_net is used as an example):

[illegible]

```

2. close()
   fput()
   dvb_net_close()

3. .disconnect()
   dvb_net_release()
   dvb_unregister_device()
   dvb_free_device()
   kfree (dvbdev->fops);

4. ...
   fops_put(file->f_op); // UAF!!
...
```

UAF occurs in '.probe -> open() -> close() -> .disconnect' flow.

The root cause of this is that fops used as an argument of replace\_fops() in dvb\_device\_open() are kfree()d in the .disconnect flow.  
It's not common for fops used in replace\_fops() to be dynamically allocated and kfree()d like this.

The KASAN log caused by this is as follows:

```

[ 67.857811] =====
[ 67.857830] BUG: KASAN: use-after-free in fput+0xa55/0xaf0
[ 67.857855] Read of size 8 at addr ffff88810f7dfc00 by task dvb_net_fput/2152

[ 67.857879] CPU: 15 PID: 2152 Comm: dvb_net_fput Not tainted 6.1.0-rc2+ #17
[ 67.857896] Hardware name: Gigabyte Technology Co., Ltd. B460MDS3H/B460M DS3H, BIOS F3 05/27/2020
[ 67.857907] Call Trace:
[ 67.857917] <TASK>
[ 67.857928] dump_stack_lvl+0x49/0x63
[ 67.857947] print_report+0x177/0x46e
[ 67.857967] ? kasan_complete_mode_report_info+0x7c/0x210
[ 67.857984] ? __fput+0xa55/0xaf0
[ 67.858001] kasan_report+0xb0/0x140
[ 67.858021] ? __fput+0xa55/0xaf0
[ 67.858039] ? __asan_report_load8_noabort+0x14/0x20
[ 67.858053] __fput+0xa55/0xaf0
[ 67.858072] __fput+0xe/0x20
[ 67.858087] task_work_run+0x153/0x240
[ 67.858108] ? task_work_cancel+0x20/0x20
[ 67.858127] ? fput+0xab/0x140
[ 67.858144] exit_to_user_mode_prepare+0x18f/0x1a0
[ 67.858160] syscall_exit_to_user_mode+0x26/0x50
[ 67.858179] do_syscall_64+0x69/0x90
[ 67.858194] ? exit_to_user_mode_prepare+0x49/0x1a0
[ 67.858208] ? irqentry_exit_to_user_mode+0x9/0x20
[ 67.858226] ? irqentry_exit+0x3b/0x50
[ 67.858243] ? exc_page_fault+0x72/0xf0
[ 67.858262] entry_SYSCALL_64_after_hwframe+0x63/0xcd
[ 67.858278] RIP: 0033:0x4537eb
[ 67.858294] Code: 03 00 00 0f 05 48 3d 00 f0 ff ff 77 41 c3 48 83 ec 18 89 7c 24 0c e8 c3 a8 02 00 8b 7c 24 0c 41 89 c0 b8 03 00 00 0f 05 <48> 3d 00 f0 ff ff 77 35 44 89 c7 89 44
24 0c e8 11 a9 02 00 8b 7c 24 0c 41 89 c0 b8 03 00 00 0f 05 <48> 3d 00 f0 ff ff 77 35 44 89 c7 89 44
[ 67.858309] RSP: 002b:00007f607be001a0 EFLAGS: 00000293 ORIG_RAX: 0000000000000003
[ 67.858330] RAX: 0000000000000000 RBX: 00007f607be00640 RCX: 000000000004537eb
[ 67.858343] RDX: 0000000000000000 RSI: 00007f6074000b70 RDI: 0000000000000003
[ 67.858353] RBP: 00007f607be001d0 R08: 0000000000000000 R09: 0000000000000000
[ 67.858364] R10: 00007f607be00180 R11: 0000000000000293 R12: 00007f607be00640
[ 67.858375] R13: 0000000000000000 R14: 000000000041b2a0 R15: 00007f607b600000
[ 67.858392] </TASK>

[ 67.858407] Allocated by task 2125:
[ 67.858417] kasan_save_stack+0x26/0x50
[ 67.858433] kasan_set_track+0x25/0x40
[ 67.858447] kasan_save_alloc_info+0x1e/0x30
[ 67.858456] __kasan_kmalloc+0xb4/0xc0
[ 67.858469] __kmalloc_node_track_caller+0x66/0x160
[ 67.858483] kmemdup+0x23/0x20
[ 67.858495] dvb_register_device+0x1cd/0x15c0 [dvb_core]
[ 67.858543] dvb_net_init+0xe1/0x120 [dvb_core]
[ 67.858591] ttusb_dec_probe.cold+0x14de/0x1f1e [ttusb_dec]
[ 67.858615] usb_probe_interface+0x266/0x740
[ 67.858629] really_probe+0x1fa/0xa80
[ 67.858642] __driver_probe_device+0x2cb/0x490
[ 67.858654] driver_probe_device+0x4e/0x140
[ 67.858666] driver_attach+0x1a3/0x520
[ 67.858677] bus_for_each_dev+0x11e/0x1c0
[ 67.858688] driver_attach+0x3d/0x60
[ 67.858699] bus_add_driver+0x449/0x5a0
[ 67.858710] driver_register+0x219/0x390
[ 67.858722] usb_register_driver+0x228/0x400
[ 67.858733] 0xffffffffffc0506023
[ 67.858744] do_one_initcall+0x97/0x310
[ 67.858758] do_init_module+0x19a/0x630
[ 67.858770] load_module+0x5ca4/0x7d90
[ 67.858781] __do_sys_finit_module+0x134/0x1d0
[ 67.858792] __x64_sys_finit_module+0x72/0xb0
[ 67.858803] do_syscall_64+0x59/0x90
[ 67.858815] entry_SYSCALL_64_after_hwframe+0x63/0xcd

[ 67.858834] Freed by task 666:
[ 67.858843] kasan_save_stack+0x26/0x50
[ 67.858857] kasan_set_track+0x25/0x40
[ 67.858870] kasan_save_free_info+0x2e/0x50
[ 67.858880] __kasan_slab_free+0x174/0x1e0
[ 67.858893] __kasan_slab_free+0x12/0x20
[ 67.858907] slab_free_freelist_hook+0xd0/0x1a0
[ 67.858919] kmem_cache_free+0x193/0x2c0
[ 67.858932] kfree+0x79/0x120
[ 67.858944] dvb_free_device.part.0+0x33/0x70 [dvb_core]
[ 67.858984] dvb_unregister_device+0x40/0x54 [dvb_core]
[ 67.859032] dvb_net_release+0x267/0x319 [dvb_core]
[ 67.859080] ttusb_dec_disconnect+0x391/0x4e1 [ttusb_dec]
[ 67.859102] usb_unbind_interface+0x187/0x7c0
[ 67.859115] device_remove+0x117/0x170
[ 67.859126] device_release_driver_internal+0x418/0x660
[ 67.859139] device_release_driver+0x12/0x20
[ 67.859150] bus_remove_device+0x28f/0x540
[ 67.859161] device_del+0x501/0xc30
[ 67.859176] usb_disable_device+0x2a5/0x660
[ 67.859187] usb_disconnect.cold+0x1f9/0x620
[ 67.859200] hub_event+0x16d3/0x3d20
[ 67.859215] process_one_work+0x778/0x11c0
[ 67.859228] worker_thread+0x544/0x1180
[ 67.859239] kthread+0x280/0x320
[ 67.859249] ret_from_fork+0x1f/0x30
...
```

# 4. media: ttusb-dec: Fix memory leak in ttusb\_dec\_exit\_dvb()  
This is a patch for a memory leak that occurs in the ttusb\_dec\_exit\_dvb() function.

Because ttusb\_dec\_exit\_dvb() does not call dvb\_frontend\_detach(), several fe related structures are not kfree()d.

Users can trigger a memory leak just by repeating connecting and disconnecting the ttusb\_dec device.

Finally, most of these patches are similar to this one, the security patch for CVE-2022-41218 that I reported:  
<https://lore.kernel.org/linux-media/20221031100245.23702-1-tiwai@suse.de/>

Regards,  
Hyunwoo Kim

Hyunwoo Kim (4):  
media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_frontend  
media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_net

```
media: dvb-core: Fix use-after-free due to race condition occurring in
dvb_register_device()
media: ttusb-dec: Fix memory leak in ttusb_dec_exit_dvb()

drivers/media/dvb-core/dvb_frontend.c | 39 ++++++--
drivers/media/dvb-core/dvb_net.c      | 37 ++++++--
drivers/media/dvb-core/dvbdev.c       | 83 ++++++-----
drivers/media/usb/ttusb-dec/ttusb_dec.c | 3 +-
include/media/dvb_frontend.h          | 6 +-
include/media/dvb_net.h               | 4 ++
include/media/dvbdev.h                 | 15 ++++
7 files changed, 155 insertions(+), 32 deletions(-)

--
2.25.1
```

---

next            reply   other threads:[~2022-11-15 13:19 UTC|newest]

**Thread overview:** 6+ messages / expand[flat|nested]   mbox.gz   Atom feed   top  
2022-11-15 13:18   **imv4bel**   **[this message]**  
2022-11-15 13:18   ` [PATCH 1/4] media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_frontend imv4bel  
2022-11-15 13:18   ` [PATCH 2/4] media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_net imv4bel  
2022-11-15 13:18   ` [PATCH 3/4] media: dvb-core: Fix use-after-free due to race condition occurring in dvb\_register\_device() imv4bel  
2022-11-17  4:16   ` Dan Carpenter  
2022-11-15 13:18   ` [PATCH 4/4] media: ttusb-dec: Fix memory leak in ttusb\_dec\_exit\_dvb() imv4bel

---

**Reply instructions:**

You may reply publicly to [this message](#) via plain-text email  
using any one of the following methods:

- \* Save the following mbox file, import it into your mail client,  
and reply-to-all from there: [mbox](#)

Avoid top-posting and favor interleaved quoting:  
[https://en.wikipedia.org/wiki/Posting\\_style#Interleaved\\_style](https://en.wikipedia.org/wiki/Posting_style#Interleaved_style)

- \* Reply using the **--to**, **--cc**, and **--in-reply-to**  
switches of git-send-email(1):

```
git send-email \
--in-reply-to=20221115131822.6640-1-imv4bel@gmail.com \
--to=imv4bel@gmail.com \
--cc=cai.huqing@linux.dev \
--cc=kernel@tuxforce.de \
--cc=linux-media@vger.kernel.org \
--cc=linux-usb@vger.kernel.org \
--cc=mchehab@kernel.org \
--cc=tiwai@suse.de \
/path/to/YOUR_REPLY
```

<https://kernel.org/pub/software/scm/git/docs/git-send-email.html>

- \* If your mail client supports setting the **In-Reply-To** header  
via mailto: links, try the [mailto: link](#)

Be sure your reply has a **Subject:** header at the top and a blank line before the message body.

---

This is a public inbox, see [mirroring instructions](#)  
for how to clone and mirror all data and code used for this inbox;  
as well as URLs for NNTP newsgroup(s).