[<prev] [next>] [day] [month] [year] [list]

```
Date: Sat, 2 Apr 2022 15:50:56 +0800 (GMT+08:00)
From: 周多明 <duoming@....edu.cn>
To: oss-security@...ts.openwall.com
Subject: CVE-2022-1198 kernel: use-after-free in
 drivers/net/hamradio/6pack.c
```

Hello there,

There are use-after-free vulnerabilities in drivers/net/hamradio/6pack.c
of linux that allow attacker to crash linux kernel by simulating ax25 device
using 6pack driver from user space.

=*=*=*=*=*=*=*=*= Bug Details =*=*=*=*=*=*=*=*=

When a 6pack device is detaching, the sixpack_close() will act to cleanup
necessary resources. Although del_timer_sync() in sixpack_close()
won't return if there is an active timer, one could use mod_timer() in
sp_xmit_on_air() to wake up timer again by calling userspace syscall such
as ax25_sendmsg(), ax25_connect() and ax25_ioctl().

This unexpected waked handler, sp_xmit_on_air(), realizes nothing about
the undergoing cleanup and may still call pty_write() to use driver layer
resources that have already been released.

The race condition is shown below:

```
      (USE)                       |      (FREE)
ax25_sendmsg()                    |
 ax25_queue_xmit()                |
  ...                             |
  sp_xmit()                       |
   sp_encaps()                    | sixpack_close()
    sp_xmit_on_air()              |  del_timer_sync(&sp->tx_t)
     mod_timer(&sp->tx_t,...)     |  ...
                                  |  unregister_netdev()
                                  |  ...
     (wait a while)               | tty_release()
                                  |  tty_release_struct()
                                  |   release_tty()
    sp_xmit_on_air()              |    tty_kref_put(tty_struct) //FREE
     pty_write(tty_struct) //USE  |    ...
```

=*=*=*=*=*=*=*=*= Bug Effects =*=*=*=*=*=*=*=*=

We can successfully trigger the vulnerabilities to crash the linux kernel.

```
[  196.518578] BUG: KASAN: use-after-free in __run_timers.part.0+0x170/0x470
[  196.518578] Write of size 8 at addr ffff88800a652ab8 by task swapper/2/0
[  196.518578] Call Trace:
[  196.518578]  <IRQ>
[  196.518578]  dump_stack+0x7d/0xa3
[  196.518578]  print_address_description.constprop.0+0x18/0x130
[  196.518578]  ? __run_timers.part.0+0x170/0x470
[  196.518578]  ? __run_timers.part.0+0x170/0x470
[  196.518578]  kasan_report.cold+0x7f/0x10e
[  196.518578]  ? __run_timers.part.0+0x170/0x470
[  196.518578]  __run_timers.part.0+0x170/0x470
[  196.518578]  ? call_timer_fn+0x150/0x150
[  196.518578]  ? lapic_timer_shutdown.part.0+0x7c/0x90
[  196.518578]  ? lapic_timer_shutdown+0x18/0x20
[  196.531225]  ? clockevents_switch_state+0xa1/0x160
[  196.531225]  ? tick_program_event+0x5f/0x80
[  196.531225]  ? hrtimer_interrupt+0x418/0x440
[  196.531225]  run_timer_softirq+0x3b/0x80
[  196.531225]  __do_softirq+0xf1/0x380
[  196.531225]  asm_call_irq_on_stack+0x12/0x20
[  196.531225]  </IRQ>
[  196.531225]  do_softirq_own_stack+0x32/0x40
[  196.531225]  irq_exit_rcu+0xb3/0x100
[  196.531225]  sysvec_apic_timer_interrupt+0x2e/0x80
[  196.531225]  asm_sysvec_apic_timer_interrupt+0x12/0x20
[  196.531225] RIP: 0010:default_idle+0xe/0x10
[  196.531225] Code: 98 36 e8 fe f0 80 63 02 df 5b 41 5c c3 0f ae f0 0f ae 3b 0f ae f0 eb 8d 0f 1f 40 00 e9
07 00 00 00 0f 00 2d 64 f7 59 00 fb f4 <c3> cc 41 55 41 54 55 48 89 fd 53 485
[  196.531225] RSP: 0018:ffff888005237e80 EFLAGS: 00000202
[  196.531225] RAX: ffffffff82481af0 RBX: ffff888005212940 RCX: ffffffff82476f62
[  196.531225] RDX: 0000000000064ec2 RSI: 0000000000000004 RDI: ffff88806d532240
[  196.531225] RBP: 0000000000000002 R08: 0000000000000001 R09: ffff88806d532243
[  196.531225] R10: ffffed100daa6448 R11: 0000000000000001 R12: 0000000000000002
[  196.531225] R13: 0000000000000000 R14: 0000000000000000 R15: 1ffff11000a46fd6
[  196.531225]  ? mwait_idle+0xc0/0xc0
[  196.531225]  ? rcu_eqs_enter.constprop.0+0x92/0xb0
[  196.531225]  default_idle_call+0x56/0x140
[  196.531225]  do_idle+0x30a/0x3b0
[  196.531225]  ? arch_cpu_idle_exit+0x30/0x30
[  196.531225]  ? schedule_idle+0x41/0x50
[  196.531225]  cpu_startup_entry+0x14/0x20
[  196.531225]  secondary_startup_64_no_verify+0xc2/0xcb

[  196.531225] Allocated by task 135:
[  196.531225]  kasan_save_stack+0x1b/0x40
[  196.531225]  ____kasan_kmalloc.constprop.0+0x84/0xa0
[  196.552309]  alloc_netdev_mqs+0x5a/0x630
[  196.552309]  sixpack_open+0xbf/0x4e0
[  196.552309]  tty_ldisc_open+0x55/0x90
[  196.552309]  tty_set_ldisc+0x187/0x2d0
[  196.552309]  tty_ioctl+0x43f/0xce0
[  196.552309]  __x64_sys_ioctl+0xb4/0xf0
[  196.552309]  do_syscall_64+0x33/0x40
[  196.552309]  entry_SYSCALL_64_after_hwframe+0x44/0xa9

[  196.552309] Freed by task 3414:
[  196.552309]  kasan_save_stack+0x1b/0x40
[  196.552309]  kasan_set_track+0x1c/0x30
[  196.552309]  kasan_set_free_info+0x20/0x30
[  196.552309]  ____kasan_slab_free+0xec/0x120
[  196.552309]  kfree+0x8f/0x210
[  196.560946]  device_release+0x54/0xe0
[  196.560946]  kobject_put+0xa5/0x120
[  196.560946]  tty_ldisc_hangup+0x1ab/0x2d0
[  196.560946]  __tty_hangup.part.0+0x306/0x510
[  196.560946]  tty_release+0x200/0x670
[  196.563997]  __fput+0x104/0x3b0
[  196.563997]  task_work_run+0x8f/0xd0
[  196.563997]  exit_to_user_mode_prepare+0x114/0x120
[  196.563997]  syscall_exit_to_user_mode+0x1d/0x40
[  196.563997]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
```

=*=*=*=*=*=*=*= Bug Reproduce =*=*=*=*=*=*=*=

We could use pseudoterminal-based device emulation to simulate ax25 device from user space and create a socket for it. Then, we create four threads: the first thread is used to initialize and start ax25 device, the second thread is used to close the pseudoterminal-based device, the third thread is used to execute bind and sendmsg syscalls, the last thread is used to close the socket. Let these four threads to interleave, we could reproduce the bug.

=*=*=*=*=*=*=*= Bug Fix =*=*=*=*=*=*=*=

The patch that have been applied to mainline Linux kernel is shown below.
https://github.com/torvalds/linux/commit/efe4186e6a1b54bf38b9e05450d43b0da1fd7739

=*=*=*=*=*=*=*= Timeline =*=*=*=*=*=*=*=

2022-02-18: commit efe4186e6a1b accepted to mainline kernel
2022-04-01: CVE-2022-1198 is assigned

=*=*=*=*=*=*=*= Credit =*=*=*=*=*=*=*=

Duoming Zhou <duoming@....edu.cn>

Best Regards,
Duoming Zhou