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

```
Date: Sun, 5 Jun 2022 21:10:06 +0200
From: Salvatore Bonaccorso <carnil@...ian.org>
To: oss-security@...ts.openwall.com
Cc: solar@...nwall.com
Subject: Re: Linux kernel: UAF, null-ptr-deref and double-free
 vulnerabilities in nfcmrvl module

Hi,

On Sun, Jun 05, 2022 at 09:14:00PM +0800, duoming@....edu.cn wrote:
> Hello there,
>
> There are double-free, use-after-free(write,read), null-ptr-deref vulnerabilities
> in drivers/nfc/nfcmrvl of linux that allow attacker to crash linux kernel by simulating
> nfc device from user-space.
>
> =*=*=*=*=*=*=*=*=  Bug Details  =*=*=*=*=*=*=*=*=
>
> There are destructive operations such as nfcmrvl_fw_dnld_abort and
> gpio_free in nfcmrvl_nci_unregister_dev. The resources such as firmware,
> gpio and so on could be destructed while the upper layer functions such as
> nfcmrvl_fw_dnld_start and nfcmrvl_nci_recv_frame is executing, which leads
> to double-free, use-after-free and null-ptr-deref bugs.
>
> There are three situations that could lead to double-free bugs.
>
> The first situation is shown below:
>
>    (Thread 1)                 |      (Thread 2)
> nfcmrvl_fw_dnld_start         |
>  ...                          |  nfcmrvl_nci_unregister_dev
>  release_firmware()           |   nfcmrvl_fw_dnld_abort
>   kfree(fw) //(1)             |    fw_dnld_over
>                               |     release_firmware
>   ...                         |      kfree(fw) //(2)
>                               |     ...
>
> The second situation is shown below:
>
>    (Thread 1)                 |      (Thread 2)
> nfcmrvl_fw_dnld_start         |
>  ...                          |
>  mod_timer                    |
>  (wait a time)                |
>  fw_dnld_timeout              |  nfcmrvl_nci_unregister_dev
>    fw_dnld_over               |   nfcmrvl_fw_dnld_abort
>     release_firmware          |    fw_dnld_over
>      kfree(fw) //(1)          |     release_firmware
>      ...                      |      kfree(fw) //(2)
>
> The third situation is shown below:
>
>        (Thread 1)               |       (Thread 2)
> nfcmrvl_nci_recv_frame          |
>  if(..->fw_download_in_progress)|
>   nfcmrvl_fw_dnld_recv_frame    |
>    queue_work                   |
>                                 |
> fw_dnld_rx_work                 | nfcmrvl_nci_unregister_dev
>  fw_dnld_over                   |  nfcmrvl_fw_dnld_abort
>   release_firmware              |   fw_dnld_over
>    kfree(fw) //(1)              |    release_firmware
>                                 |     kfree(fw) //(2)
>
> The firmware struct is deallocated in position (1) and deallocated
> in position (2) again.
>
> What's more, there are also use-after-free and null-ptr-deref bugs
> in nfcmrvl_fw_dnld_start.
>
> One of the use-after-free bugs about firmware is shown below:
>
>    (Use Thread)               |      (Free Thread )
> nfcmrvl_fw_dnld_start         |
>                               |  nfcmrvl_nci_unregister_dev
>                               |   nfcmrvl_fw_dnld_abort
>   ...                         |    fw_dnld_over
>                               |     release_firmware
>                               |      kfree(fw) //(1)
>   priv->fw_dnld.fw->data;//(2)|     ...
>
> One of the use-after-free bugs about gpio is shown below:
>
>    (Use Thread)               |      (Free Thread )
> nfcmrvl_fw_dnld_start         |
>                               |  nfcmrvl_nci_unregister_dev
>   ...                         |   ...
>                               |   gpio_free //(1)
>   nfcmrvl_chip_reset          |     ...
>    gpio_set_value //(2)       |
>
> One of the null-ptr-deref bugs about firmware is shown below:
>
>    (Use Thread)               |      (Free Thread )
> nfcmrvl_fw_dnld_start         |
>                               |  nfcmrvl_nci_unregister_dev
>                               |   nfcmrvl_fw_dnld_abort
>   ...                         |    fw_dnld_over
>                               |     priv->fw_dnld.fw = NULL;//(1)
>                               |
>   priv->fw_dnld.fw->data;//(2)|     ...
>
> If we deallocate firmware struct, gpio or set null to the members of priv->fw_dnld
> in position(1), then, we dereference firmware, gpio or the members of priv->fw_dnld
> in position(2), the UAF or NPD bugs will happen.
>
> =*=*=*=*=*=*=*=*=  Bug Effects  =*=*=*=*=*=*=*=*=
>
> We can successfully trigger the vulnerabilities to crash the linux kernel.
>
> (1) One of the backtraces caused by use-after-free(write) bug is shown below.
>
> [  138.280382] BUG: KASAN: use-after-free in _request_firmware+0x52/0x690
> [  138.280382] Write of size 8 at addr ffff88800c114850 by task download/11174
> [  138.280382] Call Trace:
> [  138.280382]  <TASK>
> [  138.280382]  dump_stack_lvl+0x57/0x7d
> [  138.280382]  print_report.cold+0x5e/0x5db
> [  138.280382]  ? _request_firmware+0x52/0x690
> [  138.280382]  kasan_report+0xbe/0x1c0
> [  138.280382]  ? _request_firmware+0x52/0x690
> [  138.280382]  _request_firmware+0x52/0x690
> [  138.280382]  request_firmware+0x2d/0x50
> [  138.280382]  nfcmrvl_fw_dnld_start+0x7a/0xb0
> [  138.280382]  nfc_fw_download+0x92/0xe0
> [  138.280382]  nfc_genl_fw_download+0x10b/0x170
> [  138.280382]  ? nfc_genl_enable_se+0xa0/0xa0
> [  138.280382]  ? __kasan_slab_alloc+0x2c/0x80
> [  138.280382]  ? __nla_parse+0x22/0x30
> [  138.280382]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0xd3/0x130
> [  138.280382]  genl_family_rcv_msg_doit+0x17a/0x200
> [  138.280382]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0x130/0x130
> [  138.280382]  ? mutex_lock_io_nested+0xb63/0xbd0
> [  138.280382]  ? security_capable+0x48/0x60
> [  138.280382]  genl_rcv_msg+0x18d/0x2c0
> [  138.280382]  ? genl_get_cmd+0x1b0/0x1b0
> [  138.280382]  ? rcu_read_lock_sched_held+0xd/0x70
> [  138.280382]  ? nfc_genl_enable_se+0xa0/0xa0
> [  138.280382]  ? rcu_read_lock_sched_held+0xd/0x70
> [  138.280382]  ? lock_acquire+0xce/0x410
> [  138.280382]  netlink_rcv_skb+0xc4/0x1f0
> [  138.280382]  ? genl_get_cmd+0x1b0/0x1b0
> [  138.280382]  ? netlink_ack+0x4d0/0x4d0
> [  138.280382]  ? netlink_deliver_tap+0xf7/0x5a0
> [  138.280382]  genl_rcv+0x1f/0x30
> [  138.280382]  netlink_unicast+0x2d8/0x420
> [  138.280382]  ? netlink_attachskb+0x430/0x430
> [  138.280382]  netlink_sendmsg+0x3a9/0x6e0
> [  138.280382]  ? netlink_unicast+0x420/0x420
> [  138.280382]  ? netlink_unicast+0x420/0x420
> [  138.280382]  sock_sendmsg+0x91/0xa0
> [  138.280382]  __sys_sendto+0x168/0x200
> [  138.280382]  ? __ia32_sys_getpeername+0x40/0x40
> [  138.280382]  ? preempt_count_sub+0xf/0xb0
> [  138.280382]  ? fd_install+0xfb/0x340
> [  138.280382]  ? __sys_socket+0xf0/0x160
> [  138.280382]  ? __x64_sys_clock_nanosleep+0x195/0x220
> [  138.280382]  ? compat_sock_ioctl+0x410/0x410
> [  138.280382]  __x64_sys_sendto+0x6f/0x80
> [  138.280382]  do_syscall_64+0x3b/0x90
> [  138.280382]  entry_SYSCALL_64_after_hwframe+0x44/0xae
> [  138.280382] RIP: 0033:0x7ff12ac0602c
> [  138.280382] Code: 0a f8 ff ff 44 8b 4c 24 2c 4c 8b 44 24 20 89 c5 44 8b 54 2b
> [  138.280382] RSP: 002b:00007ff12aa1ee00 EFLAGS: 00000293 ORIG_RAX: 000000000000000c
> [  138.280382] RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007ff12ac0602c
> [  138.280382] RDX: 000000000000002c RSI: 000055eab88030b0 RDI: 0000000000000000
> [  138.280382] RBP: 0000000000000000 R08: 00007ff12aa1ee7c R09: 000000000000000c
> [  138.280382] R10: 0000000000000000 R11: 0000000000000293 R12: 00007ffca74ba00e
> [  138.280382] R13: 00007ffca74ba00f R14: 00007ff12aa1efc0 R15: 00007ff12aa1f700
>
> (2) One of the backtraces caused by use-after-free(read) bug is shown below.
>
> [   65.835462] BUG: KASAN: use-after-free in nci_fw_download+0x26/0x60
> [   65.840236] Read of size 8 at addr ffff88800c2f5008 by task download/160
> [   65.845755] Call Trace:
> [   65.845755]  <TASK>
> [   65.845755]  dump_stack_lvl+0x57/0x7d
> [   65.845755]  print_report.cold+0x5e/0x5db
> [   65.845755]  ? nci_fw_download+0x26/0x60
> [   65.845755]  kasan_report+0xbe/0x1c0
> [   65.856061]  ? nfc_driver_failure+0x90/0xa0
> [   65.856235]  ? nci_fw_download+0x26/0x60
> [   65.856235]  nci_fw_download+0x26/0x60
> [   65.856235]  nfc_fw_download+0x99/0xe0
> [   65.856235]  nfc_genl_fw_download+0x10b/0x170
> [   65.861189]  ? nfc_genl_enable_se+0xa0/0xa0
> [   65.861189]  ? __kasan_slab_alloc+0x2c/0x80
> [   65.861189]  ? __nla_parse+0x22/0x30
> [   65.865988]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0xd3/0x130
> [   65.865988]  genl_family_rcv_msg_doit+0x17a/0x200
> [   65.865988]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0x130/0x130
> [   65.870892]  ? asm_spurious_interrupt+0x3/0x30
> [   65.870892]  ? security_capable+0x48/0x60
> [   65.870892]  genl_rcv_msg+0x18d/0x2c0
> [   65.870892]  ? genl_get_cmd+0x1b0/0x1b0
> [   65.870892]  ? rcu_read_lock_sched_held+0xd/0x70
> [   65.875946]  ? nfc_genl_enable_se+0xa0/0xa0
> [   65.875946]  ? rcu_read_lock_sched_held+0xd/0x70
> [   65.875946]  ? lock_acquire+0xce/0x410
> [   65.875946]  netlink_rcv_skb+0xc4/0x1f0
> [   65.880842]  ? genl_get_cmd+0x1b0/0x1b0
> [   65.881778]  ? netlink_ack+0x4d0/0x4d0
> [   65.881778]  ? netlink_deliver_tap+0xf7/0x5a0
> [   65.881778]  genl_rcv+0x1f/0x30
> [   65.881778]  netlink_unicast+0x2d8/0x420
> [   65.885734]  ? netlink_attachskb+0x430/0x430
> [   65.887472]  netlink_sendmsg+0x3a9/0x6e0
> [   65.887472]  ? netlink_unicast+0x420/0x420
> [   65.887472]  ? netlink_unicast+0x420/0x420
> [   65.887472]  sock_sendmsg+0x91/0xa0
> [   65.891949]  __sys_sendto+0x168/0x200
> [   65.893134]  ? __ia32_sys_getpeername+0x40/0x40
> [   65.893134]  ? lockdep_hardirqs_on_prepare+0xe/0x220
> [   65.893134]  ? __schedule+0x5c5/0x1180
> [   65.893134]  ? io_schedule_timeout+0xb0/0xb0
> [   65.897936]  ? clockevents_program_event+0xd3/0x130
> [   65.897936]  ? hrtimer_interrupt+0x332/0x350
> [   65.897936]  __x64_sys_sendto+0x6f/0x80
> [   65.897936]  do_syscall_64+0x3b/0x90
> [   65.897936]  entry_SYSCALL_64_after_hwframe+0x44/0xae
> [   65.902930] RIP: 0033:0x7f96173ec02c
> [   65.902930] Code: 0a f8 ff ff 44 8b 4c 24 2c 4c 8b 44 24 20 89 c5 44 8b 54 24 28 48 8b 54 24 18 b8 2c
00 00 00 48 8b 74 24 10 8b 7c 24 08 0f 05 <48> 3d 00 fb
> [   65.908959] RSP: 002b:00007f9617204df0 EFLAGS: 00000293 ORIG_RAX: 000000000000002c
> [   65.908959] RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007f96173ec02c
> [   65.908959] RDX: 0000000000000034 RSI: 0000556fa2a030b0 RDI: 0000000000000003
> [   65.908959] RBP: 0000000000000000 R08: 00007f9617204e6c R09: 000000000000000c
> [   65.915542] R10: 0000000000000000 R11: 0000000000000293 R12: 00007ffde78477ee
> [   65.916990] R13: 00007ffde78477ef R14: 00007f9617204fc0 R15: 00007f9617205700
>
> (3) One of the backtraces caused by double-free bug is shown below.
>
> [  122.640457] BUG: KASAN: double-free or invalid-free in fw_dnld_over+0x28/0xf0
> [  122.640457] Call Trace:
> [  122.640457]  <TASK>
> [  122.640457]  dump_stack_lvl+0x57/0x7d
> [  122.640457]  print_report.cold+0x5e/0x5db
> [  122.640457]  ? fw_dnld_over+0x28/0xf0
> [  122.640457]  ? fw_dnld_over+0x28/0xf0
> [  1re22.640457]  kasan_report_invalid_free+0x90/0x180
> [  122.640457]  ? refcount_warn_saturate+0x40/0x110
> [  122.640457]  ? fw_dnld_over+0x28/0xf0
> [  122.640457]  __kasan_slab_free+0x152/0x170
> [  122.640457]  ? fw_dnld_over+0x28/0xf0
> [  122.640457]  kfree+0xb0/0x330
> [  122.640457]  fw_dnld_over+0x28/0xf0
> [  122.640457]  nfcmrvl_nci_unregister_dev+0x61/0x70
> [  122.640457]  nci_uart_tty_close+0x87/0xd0
> [  122.640457]  tty_ldisc_kill+0x3e/0x80
> [  122.640457]  tty_ldisc_hangup+0x1b2/0x2c0
> [  122.640457]  __tty_hangup.part.0+0x316/0x520
> [  122.640457]  tty_release+0x200/0x670
> [  122.640457]  __fput+0x110/0x410
> [  122.640457]  ? _raw_spin_unlock_irq+0x1f/0x40
> [  122.640457]  task_work_run+0x86/0xd0
> [  122.640457]  exit_to_user_mode_prepare+0x1aa/0x1b0
> [  122.640457]  syscall_exit_to_user_mode+0x19/0x50
> [  122.640457]  do_syscall_64+0x48/0x90
> [  122.640457]  entry_SYSCALL_64_after_hwframe+0x44/0xae
> [  122.640457] RIP: 0033:0x7f68433f6beb
> [  122.640457] Code: 0f 05 48 3d 00 f0 ff ff 77 45 c3 0f 1f 40 00 48 83 ec 18 84
> [  122.640457] RSP: 002b:00007f684320fee0 EFLAGS: 00000293 ORIG_RAX: 00000000003
> [  122.640457] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 00007f68433f6beb
> [  122.640457] RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003
> [  122.640457] RBP: 00007f684320ff00 R08: 0000000000000000 R09: 00007f6843210700
> [  122.640457] R10: 0000000000000000 R11: 0000000000000293 R12: 00007ffd5d6f9fde
> [  122.640457] R13: 00007ffd5d6f9fdf R14: 00007f684320ffc0 R15: 00007f6843210700
>
> (4) One of the backtraces caused by null-ptr-deref bug is shown below.
>
> [   80.495478] BUG: KASAN: null-ptr-deref in nfcmrvl_fw_dnld_start.cold+0x19/0x276
> [   80.498745] Read of size 8 at addr 0000000000000008 by task download/161
> [   80.502308] Call Trace:
> [   80.502308]  <TASK>
> [   80.502308]  dump_stack_lvl+0x57/0x7d
> [   80.502308]  kasan_report+0xbe/0x1c0
> [   80.502308]  ? nfcmrvl_fw_dnld_start.cold+0x19/0x276
> [   80.502308]  nfcmrvl_fw_dnld_start.cold+0x19/0x276
> [   80.508210]  ? nfc_fw_download+0x79/0xe0
> [   80.508210]  nfc_fw_download+0x99/0xe0
> [   80.508210]  nfc_genl_fw_download+0x10b/0x170
> [   80.508210]  ? nfc_genl_enable_se+0xa0/0xa0
> [   80.508210]  ? __kasan_slab_alloc+0x2c/0x80
> [   80.508210]  ? __nla_parse+0x22/0x30
> [   80.508210]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0xd3/0x130
> [   80.508210]  genl_family_rcv_msg_doit+0x17a/0x200
> [   80.508210]  ? genl_family_rcv_msg_attrs_parse.constprop.0+0x130/0x130
> [   80.513085]  ? mutex_lock_io_nested+0xb43/0xbd0
> [   80.513085]  ? security_capable+0x48/0x60
> [   80.513085]  genl_rcv_msg+0x18d/0x2c0
> [   80.513085]  ? genl_get_cmd+0x1b0/0x1b0
> [   80.513085]  ? rcu_read_lock_sched_held+0xd/0x70
> [   80.513085]  ? nfc_genl_enable_se+0xa0/0xa0
> [   80.513085]  ? rcu_read_lock_sched_held+0xd/0x70
> [   80.513085]  ? lock_acquire+0xce/0x410
> [   80.513085]  netlink_rcv_skb+0xc4/0x1f0
> [   80.513085]  ? genl_get_cmd+0x1b0/0x1b0
> [   80.518420]  ? netlink_ack+0x4d0/0x4d0
> [   80.518420]  ? netlink_deliver_tap+0xf7/0x5a0
> [   80.518420]  genl_rcv+0x1f/0x30
> [   80.518420]  netlink_unicast+0x2d8/0x420
> [   80.518420]  ? netlink_attachskb+0x430/0x430
> [   80.518420]  netlink_sendmsg+0x3a9/0x6e0
> [   80.518420]  ? netlink_unicast+0x420/0x420
> [   80.518420]  ? netlink_unicast+0x420/0x420
> [   80.518420]  sock_sendmsg+0x91/0xa0
> [   80.518420]  __sys_sendto+0x168/0x200
> [   80.523005]  ? __ia32_sys_getpeername+0x40/0x40
> [   80.523005]  ? preempt_count_sub+0xf/0xb0
> [   80.523005]  ? fd_install+0xfb/0x340
> [   80.523005]  ? __sys_socket+0xf0/0x160
> [   80.523005]  ? compat_sock_ioctl+0x410/0x410
> [   80.523005]  __x64_sys_sendto+0x6f/0x80
> [   80.523005]  do_syscall_64+0x3b/0x90
> [   80.523005]  entry_SYSCALL_64_after_hwframe+0x44/0xae
> [   80.523005] RIP: 0033:0x7f30f54f402c
> [   80.523005] Code: 0a f8 ff ff 44 8b 4c 24 2c 4c 8b 44 24 20 89 c5 44 8b 54 24 28 48 8b 54 24 18 b8 2b
> [   80.528021] RSP: 002b:00007f30f530cdf0 EFLAGS: 00000293 ORIG_RAX: 000000000000002c
> [   80.528021] RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007f30f54f402c
> [   80.528021] RDX: 0000000000000034 RSI: 00005571766030b0 RDI: 0000000000000005
> [   80.533650] RBP: 0000000000000000 R08: 00007f30f530ce6c R09: 000000000000000c
> [   80.533650] R10: 0000000000000000 R11: 0000000000000293 R12: 00007ffd9c6c6cee
> [   80.533650] R13: 00007ffd9c6c6cef R14: 00007f30f530cfc0 R15: 00007f30f530d700
>
> =*=*=*=*=*=*=*=*=  Bug Fix  =*=*=*=*=*=*=*=*=
>
> The patch that have been applied to mainline Linux kernel is shown below.
> https://github.com/torvalds/linux/commit/d270453a0d9ec10bb8a802a142fb1b3601a83098
>
> =*=*=*=*=*=*=*=*=  Timeline  =*=*=*=*=*=*=*=*=
>
> 2022-05-01: commit d270453a0d9e accepted to mainline kernel
> 2022-06-05: send an email to secalert@...hat.com in order to request CVE number
>
> =*=*=*=*=*=*=*=*=  Credit  =*=*=*=*=*=*=*=*=
>
> Duoming Zhou <duoming@....edu.cn>
```

According to https://bugzilla.redhat.com/show_bug.cgi?id=2086766 this
should be CVE-2022-1734.

Regards,
Salvatore