



author Zheng Wang <zyytlz.wz@163.com> 2023-03-13 00:08:37 +0800
committer Jakub Kicinski <kuba@kernel.org> 2023-03-15 00:28:23 -0700
commit [5000fe6c27827a61d8250a7e4a1d26c3298ef4f6](#) (patch)
tree [8a8fa841617b149e8523b1f78caf4bbe1f1552d0](#)
parent [cf18d55e535b34fd4845e87e31e1c5609db6d0a3](#) (diff)
download [linux-5000fe6c27827a61d8250a7e4a1d26c3298ef4f6.tar.gz](#)

diff options

context:

3 ▼

space:

include ▼

mode:

unified ▼

nfc: st-nci: Fix use after free bug in ndlc_remove due to race condition

This bug influences both `st_nci_i2c_remove` and `st_nci_spi_remove`.
Take `st_nci_i2c_remove` as an example.

In `st_nci_i2c_probe`, it called `ndlc_probe` and bound `&ndlc->sm_work` with `llt_ndlc_sm_work`.

When it calls `ndlc_rcv` or timeout handler, it will finally call `schedule_work` to start the work.

When we call `st_nci_i2c_remove` to remove the driver, there may be a sequence as follows:

Fix it by finishing the work before cleanup in `ndlc_remove`

CPU0	CPU1
	<code>llt_ndlc_sm_work</code>
<code>st_nci_i2c_remove</code>	
<code>ndlc_remove</code>	
<code>st_nci_remove</code>	
<code>nci_free_device</code>	
<code>kfree(ndev)</code>	
<code>//free ndlc->ndev</code>	
	<code>llt_ndlc_rcv_queue</code>
	<code>nci_rcv_frame</code>
	<code>//use ndlc->ndev</code>

Fixes: [35630df68d60](#) ("NFC: st21nfc: Add driver for STMicroelectronics ST21NFCB NFC chip")

Signed-off-by: Zheng Wang <zyytlz.wz@163.com>

Reviewed-by: Krzysztof Kozłowski <krzysztof.kozłowski@linaro.org>

Link: <https://lore.kernel.org/r/20230312160837.2040857-1-zyytlz.wz@163.com>

Signed-off-by: Jakub Kicinski <kuba@kernel.org>

Diffstat

```
-rw-r--r-- drivers/nfc/st-nci/ndlc.c 6
```

1 files changed, 4 insertions, 2 deletions

diff --git a/drivers/nfc/st-nci/ndlc.c b/drivers/nfc/st-nci/ndlc.c

index 755460a73c0dce..d2aa9f766738ec 100644

--- a/drivers/nfc/st-nci/ndlc.c

+++ b/drivers/nfc/st-nci/ndlc.c

@@ -282,13 +282,15 @@ EXPORT_SYMBOL(ndlc_probe);

```
void ndlc_remove(struct llt_ndlc *ndlc)
{
-     st_nci_remove(ndlc->ndev);
-
    /* cancel timers */
    del_timer_sync(&ndlc->t1_timer);
    del_timer_sync(&ndlc->t2_timer);
    ndlc->t2_active = false;
    ndlc->t1_active = false;
+     /* cancel work */
+     cancel_work_sync(&ndlc->sm_work);
+
+     st_nci_remove(ndlc->ndev);

    skb_queue_purge(&ndlc->rcv_q);
    skb_queue_purge(&ndlc->send_q);
}
```