



author Zheng Wang <zyytlz.wz@163.com> 2023-03-13 00:08:37 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-22 13:26:14 +0100
commit [2156490c4b7cacda9a18ec99929940b8376dc0e3](#) (patch)
tree [678b30fd9d46dc29e6a07f5383c46b4ae46109a5](#)
parent [85f922f42399f779a440592944dd90fd4cf6ebcd](#) (diff)
download [linux-2156490c4b7cacda9a18ec99929940b8376dc0e3.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

[Upstream commit 5000fe6c27827a61d8250a7e4a1d26c3298ef4f6]

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_recv` 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_recv_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>

Signed-off-by: Sasha Levin <sashal@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 9477994cf97534..a3dfb3d1202103 100644
--- a/drivers/nfc/st-nci/ndlc.c
+++ b/drivers/nfc/st-nci/ndlc.c
@@ -302,13 +302,15 @@ 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);
}
```