

Linux-Kernel: [2.4] "Fix" introduced in 2.4.27pre2 for bluetooth hci_usb race causes kernel hang

[2.4] "Fix" introduced in 2.4.27pre2 for bluetooth hci_usb race causes kernel hang

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-04/2365.html>

From: Tomas Qgren?= (stric_at_acc.umu.se)

Date: 04/08/05

Date: Fri, 8 Apr 2005 21:56:33 +0200

To: linux-kernel@vger.kernel.org

Hello.

I have noticed a problem with a race condition fix introduced in 2.4.27-pre2 that causes the kernel to hang when disconnecting a Bluetooth USB dongle or doing 'hciconfig hci0 down'. No message is printed, the kernel just doesn't respond anymore.

Seen in Changelog:

Marcel Holtmann:

- o [Bluetooth] Fix race in RX complete routine of the USB drivers

Reversing the following patch to hci_usb_rx_complete() makes 2.4.27-pre2 up until 2.4.30 happy and does not hang when removing the dongle anymore. (bfusb.c has the same patch applied)

2.6.11.7 does not show the same problem, but has similar code to the "fixed" (that hangs) code in 2.4, so the real problem is probably somewhere else.

I have tested this on Dell Optiplex GX150, 260 and 280's which has Intel P3 and P4 with Intel UHCI USB chipset. I have tested both usb-uhci.o and uhci.o with the same results. Tested with USB Bluetooth dongles with both Broadcom and Cambridge Silicon Radio chipsets, same results.

modules loaded: l2cap, hci_usb, bluez, (usb-)uhci, usbcore

```
diff -ruN linux-2.4.27-pre1/drivers/bluetooth/hci_usb.c linux-2.4.27-pre2/drivers/bluetooth/hci_usb.c
--- linux-2.4.27-pre1/drivers/bluetooth/hci_usb.c 2004-04-14 15:05:29.000000000 +0200
+++ linux-2.4.27-pre2/drivers/bluetooth/hci_usb.c 2005-04-08 20:16:51.000000000 +0200
@@ -699,11 +699,11 @@
     BT_DBG("%s urb %p type %d status %d count %d flags %x", hdev->name, urb,
           _urb->type, urb->status, count, urb->transfer_flags);

- if (!test_bit(HCI_RUNNING, &hdev->flags))
- return;
```

Linux-Kernel: [2.4] "Fix" introduced in 2.4.27pre2 for bluetooth hci_usb race causes kernel hang

```
-
    read_lock(&husb->completion_lock);

+ if (!test_bit(HCI_RUNNING, &hdev->flags))
+ goto unlock;
+
+     if (urb->status || !count)
+         goto resubmit;

@@ -740,6 +740,8 @@
     BT_DBG("%s urb %p type %d resubmit status %d", hdev->name, urb,
            _urb->type, err);
    }
+
+unlock:
    read_unlock(&husb->completion_lock);
}
```

Please CC me for any responses, not on the list.

/Tomas

```
--
Tomas Ögren, stric@acc.umu.se, http://www.acc.umu.se/~stric/
|- Student at Computing Science, University of Umeå
`- Sysadmin at {cs,acc}.umu.se
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```