

# GDB remote debugging

## Setup

Ở phần trình bày này, kernel 4.4.35 được sử dụng. Driver ath5k. Máy target 32bit. IDA tools hỗ trợ:

<https://www.dropbox.com/s/klhyb93a1wmk26y/IDAPro6.6full.7z?dl=0>

### **\*Trên máy target:**

- Trước hết ta cần compile lại kernel và load theo hướng dẫn:

[https://courses.uit.edu.vn/pluginfile.php?file=%2F61892%2Fmod\\_resource%2Fcontent%2F3%2FLab5%20-%20Linux kernel ath9k htc.pdf](https://courses.uit.edu.vn/pluginfile.php?file=%2F61892%2Fmod_resource%2Fcontent%2F3%2FLab5%20-%20Linux%20kernel%20ath9k%20htc.pdf)

```
minhvo@minhvo-ubuntu:~$ uname -a
Linux minhvo-ubuntu 4.4.35 #1 SMP Wed Nov 30 20:18:13 ICT 2016 i686 i686 i686 GNU/Linux
```

- Kết nối ethernet với router.

- Cài đặt kgdboe để hỗ trợ remote debug over ethernet:

```
target$ tar -xzf kgdboe.tgz
```

```
target$ cd kgdboe
```

```
target$ make -C /lib/modules/$(uname -r)/build M=$(pwd)
```

```
target$ sudo insmod kgdboe.ko device_name=[device_name]
```

```
udp_port=[port]
```

```
minhvo@minhvo-ubuntu:~/Downloads/kgdboe$ sudo insmod kgdboe.ko device_name=enp4s0
[sudo] password for minhvo:
minhvo@minhvo-ubuntu:~/Downloads/kgdboe$ lsmod | grep kgdb
kgdboe                36864  0
minhvo@minhvo-ubuntu:~/Downloads/kgdboe$
```

### **\*Trên máy host:**

- Copy file "vmlinux", folder "net/mac80211/", folder

"driver/net/wireless/ath/ath5k" trong folder kernel source-code vừa build trên máy target vào máy host.

(Nên dùng secure-copy của ssh bằng lệnh "scp")

```
boe@boe:~/Desktop/HK5/he_thong_nhung/project/mydebug$ ls
ath5k.ko          drivers           peda-session-unknown.txt
ath5k_report.docx info_module.py   peda-session-vmlinux.txt
ath5k_rx.docx     mac80211.ko      vmlinux
ath.ko            net
```

## Debug step

- Khởi động gdb trên máy host.

```
Host$ sudo gdb vmlinux
```

```
gdb$ target remote udp:[target_ip]:[port]
```

```
gdb-peda$ target remote udp:192.168.1.6:31337
warning: The remote protocol may be unreliable over UDP.
Some events may be lost, rendering further debugging impossible.
Remote debugging using udp:192.168.1.6:31337
Warning: not running or target is remote
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072      kernel/debug/debug_core.c: No such file or directory.
gdb-peda$
```

(31337 is default port of kgdboe)

(Nếu thành công máy target sẽ bị đơ)

```
gdb$ c
```

- Lấy thông tin địa chỉ được load của module ath5k trên target.  
(code đính kèm info.py)

```
target$ sudo python info.py
```

```
minhvo@minhvo-ubuntu:~/Desktop$ sudo python info.py
[sudo] password for minhvo:
cat: ../: Is a directory
cat: ../: Is a directory
0xf87bb000 -s .bss 0xf87d9380 -s .data 0xf87d9000 -s .data.unlikely 0xf87d9160 -
s .exit.text 0xf87d01f6 -s .gnu.linkonce.this_module 0xf87d9180 -s .init.text 0x
f87de000 -s .note.gnu.build-id 0xf87d1000 -s .parainstructions 0xf87d8d34 -s .ro
data 0xf87d1040 -s .rodata.str1.1 0xf87d8094 -s .rodata.str1.4 0xf87d8688 -s .sm
p_locks 0xf87d8dd8 -s .strtab 0xf87e1080 -s .symtab 0xf87df000
minhvo@minhvo-ubuntu:~/Desktop$
```

- Quay trở lại máy host và load module vào gdb.

```
gdb$ Ctrl+C
```

```
gdb$ add-symbol-file drivers/net/wireless/ath/ath5k/ath5k.o
[output của info.py]
```

```
boe@boe: ~/project/mydebug
minhvo@minhv... x boe@boe: ~/proj... x boe@boe: ~/Des... x boe@boe: ~/Des... x
^C[New Thread 2553]
[New Thread 2591]

Thread 1 received signal SIGTRAP, Trace/breakpoint trap.
Warning: not running or target is remote
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072      in kernel/debug/debug_core.c
gdb-peda$ add-symbol-file drivers/net/wireless/ath/ath5k/ath5k.o 0xf87bb000 -s .
bss 0xf87d9380 -s .data 0xf87d9000 -s .data.unlikely 0xf87d9160 -s .exit.text 0x
f87d01f6 -s .gnu.linkonce.this_module 0xf87d9180 -s .init.text 0xf87de000 -s .no
te.gnu.build-id 0xf87d1000 -s .parainstructions 0xf87d8d34 -s .rodata 0xf87d1040
-s .rodata.str1.1 0xf87d8094 -s .rodata.str1.4 0xf87d8688 -s .smp_locks 0xf87d8
dd8 -s .strtab 0xf87e1080 -s .syntab 0xf87df000
add symbol table from file "drivers/net/wireless/ath/ath5k/ath5k.o" at
    .text_addr = 0xf87bb000
    .bss_addr = 0xf87d9380
    .data_addr = 0xf87d9000
    .data.unlikely_addr = 0xf87d9160
    .exit.text_addr = 0xf87d01f6
    .gnu.linkonce.this_module_addr = 0xf87d9180
    .init.text_addr = 0xf87de000
    .note.gnu.build-id_addr = 0xf87d1000
    .parainstructions_addr = 0xf87d8d34
    .rodata_addr = 0xf87d1040
    .rodata.str1.1_addr = 0xf87d8094
    .rodata.str1.4_addr = 0xf87d8688
    .smp_locks_addr = 0xf87d8dd8
    .strtab_addr = 0xf87e1080
    .syntab_addr = 0xf87df000
Reading symbols from drivers/net/wireless/ath/ath5k/ath5k.o...warning: section .
gnu.linkonce.this_module not found in /home/boe/Desktop/HK5/he_thong_nhung/proje
ct/mydebug/drivers/net/wireless/ath/ath5k/ath5k.o
warning: section .note.gnu.build-id not found in /home/boe/Desktop/HK5/he_thong_
nhung/project/mydebug/drivers/net/wireless/ath/ath5k/ath5k.o
warning: section .strtab not found in /home/boe/Desktop/HK5/he_thong_nhung/proje
ct/mydebug/drivers/net/wireless/ath/ath5k/ath5k.o
warning: section .syntab not found in /home/boe/Desktop/HK5/he_thong_nhung/proje
ct/mydebug/drivers/net/wireless/ath/ath5k/ath5k.o
done.
gdb-peda$
```

- Ở máy host, load file ath5k.o vào IDA để đọc code và assembly.

# Debug result

## Interrupt handling

**brkp1 (break point):** Nhận được interrupt.

```
gdb$ b ath5k_intr
```

```
gdb$ c
```

```
Hit brkp1
```

**brkp2:** Xem giá trị biến status và behavior của driver.

[source code ath5k\_intr() IDA]

```
19 do
20 {
21     ath5k_hw_get_isr((ath5k_hw *)v4, &status);
22     v7 = status;
23     if ( status & 0x80000 )
24         goto LABEL_38;
25     if ( status & 0x20 )
26     {
27         ++(_DWORD *) (v4 + 19680);
28         if ( *(_DWORD *) (v4 + 19936) > 0x4Fu )
29         {
30             *(_BYTE *) (v4 + 18316) = 1;
31             if ( !_interlockedbittestandset((volatile signed __int32 *) (v4 + 18384), 0) )
32                 _tasklet_schedule();
33         }
34     }
```

[source code ath5k\_intr()]

```
/** Main loop */
do {
    ath5k_hw_get_isr(ah, &status); /* NB: clears IRQ too */

    ATH5K_DBG(ah, ATH5K_DEBUG_INTR, "status 0x%x/0x%x\n",
              status, ah->imask);
```

[assembly code]

```
text:080106F2      lea     edx, [ebp+status] ; interrupt_mask
text:080106F5      mov     eax, dev_id      ; ah_0
text:080106F7      call   ath5k_hw_get_isr
```

```
gdb$ b*ath5k_intr+167
```

```
gdb$ c
```

```
Hit brkp2
```

```
gdb-peda$ b*4170426103
Breakpoint 2 at 0xf893a6f7: file drivers/net/wireless/ath/ath5k/base.c, line 228
7.
gdb-peda$ c
Continuing.
Warning: not running or target is remote

Thread 1 hit Breakpoint 2, 0xf893a6f7 in ath5k_intr (irq=<optimized out>,
dev_id=0xea828da0) at drivers/net/wireless/ath/ath5k/base.c:2287
2287      ath5k_hw_get_isr(ah, &status); /* NB: clears IRQ too */
gdb-peda$
```

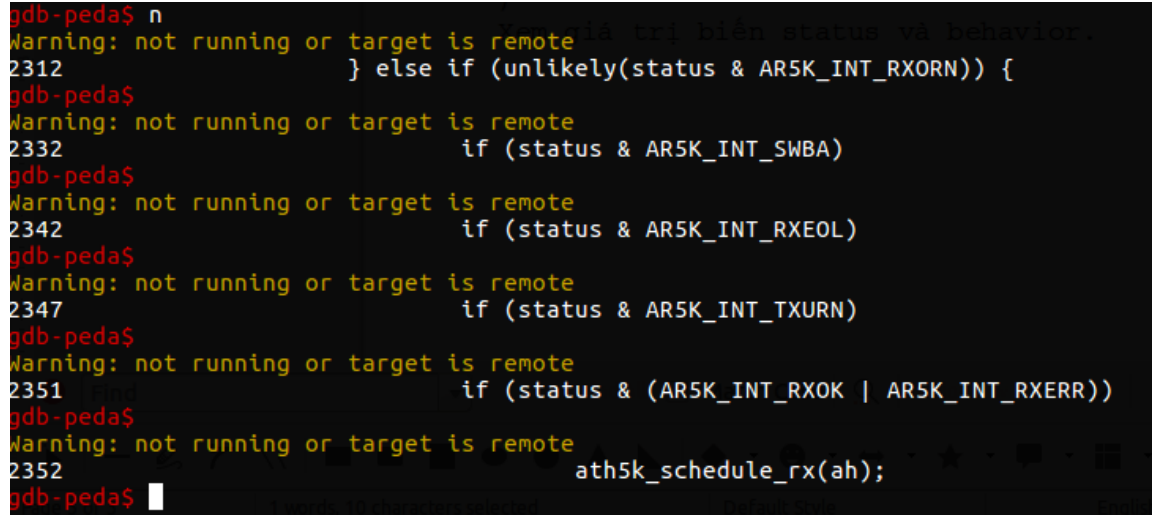
```
gdb$ ni
```

```
gdb$ p status
```

```
gdb-peda$ p status
$1 = 4101
gdb-peda$
```

gdb\$ n

(vài lần đến khi hit `ath5k_schedule_rx` hoặc `ath5k_schedule_tx`)

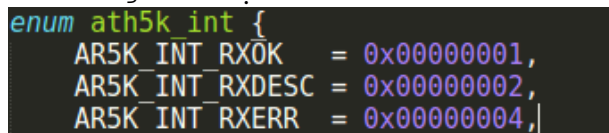


```
gdb-peda$ n
Warning: not running or target is remote
2312         } else if (unlikely(status & AR5K_INT_RXORN)) {
gdb-peda$
Warning: not running or target is remote
2332         if (status & AR5K_INT_SWBA)
gdb-peda$
Warning: not running or target is remote
2342         if (status & AR5K_INT_RXEOL)
gdb-peda$
Warning: not running or target is remote
2347         if (status & AR5K_INT_TXURN)
gdb-peda$
Warning: not running or target is remote
2351         if (status & (AR5K_INT_RXOK | AR5K_INT_RXERR))
gdb-peda$
Warning: not running or target is remote
2352         ath5k_schedule_rx(ah);
gdb-peda$
```

\*Giải thích flow:

`status = 4101 = 0x1005`

Ta có giá trị từ file `ath5k.h`:



```
enum ath5k_int {
    AR5K_INT_RXOK      = 0x00000001,
    AR5K_INT_RXDESC    = 0x00000002,
    AR5K_INT_RXERR     = 0x00000004,
```

`=> status & (AR5K_INT_RXOK | AR5K_INT_RXERR)`

`= 0x1005 & (0x4 | 0x1) = 5 → true`

`=> Gọi hàm ath5k_schedule_rx(ah)`

Thật ra hàm `ath5k_schedule_rx()` không tồn tại, mà code của nó được nhúng thẳng vào hàm `ath5k_intr()`. Phía dưới sẽ giải thích rõ hơn.

\*Giải thích cách đặt breakpoint `ath5k_intr+167`

`167 = addr_brkp - addr_ath5k_intr`

`= 0x080106f7 - 0x08010650` (Địa chỉ xem trên assembly code của IDA)

**brkp3&4:** Xem function sẽ được gọi khi tasklet\_schedule() thực thi.

[source code in IDA]

```
58 | if ( v7 & 5 )
59 | {
60 |     *(_BYTE *) (v4 + 18316) = 1;
61 |     if ( !_interlockedbittestandset((volatile signed __int32 *) (v4 + 0x47D0), 0) )
62 |         _tasklet_schedule();
63 |     v7 = status;
64 | }
65 | if ( v7 & 0x5C0 )
66 | {
67 |     *(_BYTE *) (v4 + 18317) = 1;
68 |     if ( !_interlockedbittestandset((volatile signed __int32 *) (v4 + 19864), 0) )
69 |         _tasklet_schedule();
70 |     v7 = status;
71 | }
```

[source code ath5k\_intr()]

```
/* RX -> Schedule rx tasklet */
if (status & (AR5K_INT_RXOK | AR5K_INT_RXERR))
    ath5k_schedule_rx(ah);

/* TX -> Schedule tx tasklet */
if (status & (AR5K_INT_TXOK
             | AR5K_INT_TXDESC
             | AR5K_INT_TXERR
             | AR5K_INT_TXEOL))
    ath5k_schedule_tx(ah);
```

[source code ath5k\_schedule\_rx()]

```
static void
ath5k_schedule_rx(struct ath5k_hw *ah)
{
    ah->rx_pending = true;
    tasklet_schedule(&ah->rxtq);
}
```

[source code ath5k\_schedule\_tx()]

```
static void
ath5k_schedule_tx(struct ath5k_hw *ah)
{
    ah->tx_pending = true;
    tasklet_schedule(&ah->txtq);
}
```

\*Nhận xét:

- Khi compile, một số hàm sẽ bị lược bớt và nhúng thẳng vào callee.

\*Lý giải số 0x5c0 trong IDA code.

Ta có các giá trị trong ath5k.h:

AR5K\_INT\_TXOK = 0x40

AR5K\_INT\_TXDESC = 0x80

AR5K\_INT\_TXERR = 0x100

AR5K\_INT\_TXEOL = 0x400

0x40 | 0x80 | 0x100 | 0x400 = 0x5c0

\*Nếu là **rx path** sẽ hit brkp này.

[assembly code]

```
.text:00010810      mov     byte ptr [dev_id+478Ch], 1
.text:00010817      lock bts dword ptr [dev_id+47D0h], 0
.text:00010820      jb     short loc_801082D
.text:00010822      lea     eax, [dev_id+47CCh]
.text:00010824      call    __tasklet_schedule
```

gdb\$ b\*ath5k\_intr+472

\*Nếu là **tx path** sẽ hit brkp này.

[assembly code]

```
.text:000107E8      mov     byte ptr [dev_id+478Dh], 1
.text:000107EF      lock bts dword ptr [dev_id+4A78h], 0
.text:000107F8      jb     short loc_8010802
.text:000107FA      mov     eax, [ebp+var_18]
.text:000107FD      call    __tasklet_schedule
```

gdb\$ b\*ath5k\_intr+429

gdb\$ c

Hit brkp3 (**rx\_path**)

gdb\$ p \*((struct tasklet\_struct \*) \$eax)->func

```
Thread 539 hit Breakpoint 3, ath5k_schedule_rx (ah=<optimized out>)
at drivers/net/wireless/ath/ath5k/base.c:2252
2252      tasklet_schedule(&ah->rxtq);
gdb-peda$ p *((struct tasklet_struct *) $eax)->func
$13 = {void (unsigned long)} 0xf893aec0 <ath5k_tasklet_rx>
```

- Theo code assembly ta thấy arguments pass vào hàm  
\_\_tasklet\_schedule() là thanh ghi eax. Nó là một  
struct tasklet\_struct {

```
...;
void (*func) (unsigned long);
```

}

- Khi đó struct của ta sẽ vào queue đợi đến lượt. Khi đến lượt,  
trường func chứa con trỏ hàm sẽ được gọi.

- Ở trường hợp này chính là hàm ath5k\_tasklet\_rx().

=> ath5k\_intr() → ath5k\_tasklet\_rx()



Hit brkp4 (**tx\_path**)

`gdb$ p *((struct tasklet_struct *) $eax)->func`

```
gdb-peda$ b*ath5k_intr+429
Breakpoint 5 at 0xf893a7fd: file include/linux/interrupt.h, line 536.
gdb-peda$ c
Continuing.
[New Thread 2313]
[Switching to Thread 1867]
Warning: not running or target is remote
Thread 541 hit Breakpoint 4, ath5k_tasklet_rx (data=0xea828da0)
  at drivers/net/wireless/ath/ath5k/base.c:1541
1541 {
gdb-peda$ c
Continuing.
[Switching to Thread 2300]
Warning: not running or target is remote
Thread 635 hit Breakpoint 5, 0xf893a7fd in tasklet_schedule (t=<optimized out>)
  at include/linux/interrupt.h:536
536 include/linux/interrupt.h: No such file or directory.
gdb-peda$ b* *((struct tasklet_struct *) $eax)->func
Breakpoint 6 at 0xf893aab0: file drivers/net/wireless/ath/ath5k/base.c, line 178
9.
gdb-peda$ p *((struct tasklet_struct *) $eax)->func
$15 = {void (unsigned long)} 0xf893aab0 <ath5k_tasklet_tx>
gdb-peda$
```

Tương tự như trên.

=> `ath5k_intr()` → `ath5k_tasklet_tx()`

**brkp5:** Tiếp tục flow

`gdb$ b* *((struct tasklet_struct *) $eax)->func`

(Nếu rx path → `ath5k_tasklet_rx()`)

tx path → `ath5k_tasklet_tx()` )

`gdb$ c`

**Kết luận:** interrupt handling

- Nhận interrupt và kiểm tra status từ hardware để thực thi các hàm tương ứng.

- ah là một struct `ath5k_hw` {

...;

struct tasklet\_struct rxtq; /\* rx intr tasklet\*/

struct tasklet\_struct txtq; /\* tx intr tasklet\*/

}

- Thực ra là các hàm tương ứng quyết định bởi `ah→rxtq` hay `ah→txtq` được pass vào `__tasklet_schedule(struct tasklet_struct *t)` mà sau đó `t->func` sẽ được gọi.



## Rx path

**brkp5:** `ath5k_tasklet_rx(unsigned long data), xem arguments.`  
Hit brkp5

```
Thread 539 hit Breakpoint 3, ath5k_schedule_rx (ah=<optimized out>)
  at drivers/net/wireless/ath/ath5k/base.c:2252
2252      tasklet_schedule(&ah->rxtq);
gdb-peda$ p *((struct tasklet_struct *) $eax)->func
$13 = {void (unsigned long)} 0xf893aec0 <ath5k_tasklet_rx>
```

- `sk_buff` của hardware pass vào.

```
gdb$ p *((struct ath5k_buf *) ((struct ath5k_hw *) $eax)->rxbuf)->skb
(result in file struct/sk_buf_ath5k_tasklet_rx)
```

**brkp6:** Behavior `ath5k_receive_frame_ok()` phụ thuộc vào biến `rs`.  
[source code]

<http://lxr.free->

[electronics.com/source/drivers/net/wireless/ath/ath5k/base.c#L1449](http://electronics.com/source/drivers/net/wireless/ath/ath5k/base.c#L1449)

[Một phần source code `ath5k_receive_frame_ok()` IDA]

```
v7 = rs->rs_datalen; // if (ath5k_receive_frame_ok(ah, &rs)) {
++*(_DWORD *) (ath_hw + 19472);
*(_DWORD *) (ath_hw + 19488) += v7;
v8 = rs->rs_status;
if ( !rs->rs_status )
    goto LABEL_7;
if ( rs->rs_status & 1 )
    ++*(_DWORD *) (ath_hw + 19488);
if ( v8 & 4 )
    ++*(_DWORD *) (ath_hw + 19624);
```

```
gdb$ b* ath5k_tasklet_rx+157
```

```
gdb$ c
```

Hit brkp6. Biến `rs` được chiết xuất từ hardware.

```
gdb$ p rs
```

(struct, result, behavior in struct/ath5k\_rx\_status)

```
Thread 635 hit Breakpoint 7, ath5k_tasklet_rx (data=0xea828da0)
  at drivers/net/wireless/ath/ath5k/base.c:1575
1575      if (ath5k_receive_frame_ok(ah, &rs)) {
gdb-peda$ p rs
$24 = {
  rs_datalen = 0xef,
  rs_tstamp = 0x226e,
  rs_status = 0x0,
  rs_phyerr = 0x0,
  rs_rssi = 0x13,
  rs_keyix = 0xff,
  rs_rate = 0x1b,
  rs_antenna = 0x1,
  rs_more = 0x0
}
```

\*Lưu ý:

Hàm `ath5k_receive_frame_ok()` cũng không tồn tại mà được nhúng thẳng vào `ath5k_tasklet_rx()`, nên khi `trace-cmd` sẽ không gặp được.

**brkp7&8:** Set breakpoint để biết return của hàm  
ath5k\_recieve\_frame\_ok().  
[source code]

```

if (ath5k_receive_frame_ok(ah, &rs)) {
    next_skb = ath5k_rx_skb_alloc(ah, &next_skb_addr);

    /*
     * If we can't replace bf->skb with a new skb under
     * memory pressure, just skip this packet
     */
    if (!next_skb)
        goto next;

    dma_unmap_single(ah->dev, bf->skbaddr,
                     common->rx_bufsize,
                     DMA_FROM_DEVICE);

    skb_put(skb, rs.rs_datalen);

    ath5k_receive_frame(ah, skb, &rs);

    bf->skb = next_skb;
    bf->skbaddr = next_skb_addr;
}
next:
    list_move_tail(&bf->list, &ah->rxbuf);
} while (ath5k_rxbuf_setup(ah, bf) == 0);

```

#### \*Phân tích:

Nếu hàm ath5k\_receive\_frame\_ok() trả về true thì frame sẽ được pass lên cho mac80211. Ngược lại thì không. Tương ứng với [Một phần source code ath5k\_receive\_frame\_ok() IDA]

```

if ( (v30 & 0xF7) != 23 )
    goto next;
v8 = v8 & 0xFFFFFFF0 | 1;
rs.rs_status = v8;
}
if ( !(v8 & 8) || (++*(DWORD *) (ath_hw + 19628), rs.rs_keyix != -1) || v8 & 1 )
{
    if ( v8 & 0x10 )
    {
        ++*(DWORD *) (ath_hw + 19632);
        goto LABEL_8;
    }
    if ( ((*(DWORD *) (ath_hw + 18296) << 29 >> 31) - 9) & (unsigned __int8)v8 )
        goto next;
LABEL_7:
    if ( rs.rs_more )
    {
        ++*(DWORD *) (ath_hw + 19640);
        goto next;
    }
}

```

#### [Label8]

```

LABEL_8:
    ds_code = ath5k_rx_skb_alloc((ath5k_hw *)ath_hw, &next_skb_addr); // next_skb = ath5k_rx_skb_alloc(ah, &next_skb_addr);
    if ( !ds_code )
        goto next;

```

#### [next]

```

next:
    v26 = *(DWORD *)bf;
    v27 = *(DWORD *) (bf + 4);
    *(DWORD *) (v26 + 4) = v27;
    *(int v27; // eax@828)
    v28 = *(DWORD *) (ath_hw + 18368);
    *(DWORD *) (ath_hw + 18368) = bf;
    *(DWORD *)bf = ath_hw + 0x47BC;
    *(DWORD *) (bf + 4) = v28;
    *(DWORD *)v28 = bf;
    if ( ath5k_rxbuf_setup((ath5k_hw *)ath_hw, (ath5k_buf *)bf) )
        break;

```

### \*Nhận xét:

Từ code trên, ta suy ra được:

```
hit LABEL_8      → true
hit next         → false
```

```
gdb$ b*ath5k_tasklet_rx+196 (LABEL_8)
```

```
gdb$ b*ath5k_tasklet_rx+700 (next)
```

**brkp9:** Xem arguments pass to upper layer ieee80211\_rx(ah->hw,skb)

```
gdb$ b*ath5k_tasklet_rx+677
```

[Một phần source code ath5k\_receive\_frame()]

```
static void
ath5k_receive_frame(struct ath5k_hw *ah, struct sk_buff *skb,
                    struct ath5k_rx_status *rs)
{
    struct ieee80211_rx_status *rxs;
    struct ath_common *common = ath5k_hw_common(ah);

    ath5k_remove_padding(skb);

    rxs = IEEE80211_SKB_RXCB(skb);

    rxs->flag = 0;
    if (unlikely(rs->rs_status & AR5K_RXERR_MIC))
        rxs->flag |= RX_FLAG_MMIC_ERROR;
    if (unlikely(rs->rs_status & AR5K_RXERR_CRC))
        rxs->flag |= RX_FLAG_FAILED_FCS_CRC;

    if (ath_is_mybeacon(common, (struct ieee80211_hdr *)skb->data)) {
        ewma_beacon_rssi_add(&ah->ah_beacon_rssi_avg, rs->rs_rssi);

        /* check beacons in IBSS mode */
        if (ah->opmode == NL80211_IFTYPE_ADHOC)
            ath5k_check_ibss_tsf(ah, skb, rxs);
    }

    ieee80211_rx(ah->hw, skb);
}
```

[source code tương ứng trên IDA]

```
ath5k_remove_padding(skb); // ath5k_receive_frame(ah, skb, &rs);
v11 = rs.rs_status;
*(DWORD *)&skb->cb[16] = 0;
if (v11 & 0x10)
    *(DWORD *)&skb->cb[16] = 1;
if (v11 & 1)
    *(DWORD *)&skb->cb[16] |= 0x20u;
v12 = rs.rs_tstamp; // rxs->mactime = ath5k_extend_tsf(ah, rs->rs_tstamp);
mactime = ath5k_hw_get_tsf64((ath5k_hw *)ath_hw);
if ((mactime & 0x7FFF) < v12)
    mactime -= 0x8000LL;

if ( !ath5k_hw_check_beacon_timers((ath5k_hw *)ath_hw, *(DWORD *)&ath_hw + 19412) )
    ath5k_beacon_update_timers(v39, v40);
}
v23 = *(DWORD *)&ath_hw + 504;
ieee80211_rx_napi(0, skb);
```

### \*Lưu ý:

Hàm ath5k\_receive\_frame() không tồn tại, được nhúng thẳng vào ath5k\_tasklet\_rx().

[assembly code khi gọi ieee80211\_rx\_napi()]

```
.text:0801115A      mov     eax, [data+1F8h]
.text:08011160      mov     edx, [ebp+skb]
.text:08011163      xor     ecx, ecx
.text:08011165      call    ieee80211_rx_napi
```

[source code ieee80211\_rx()]

```
/**
 * ieee80211_rx - receive frame
 *
 * Use this function to hand received frames to mac80211. The receive
 * buffer in @skb must start with an IEEE 802.11 header. In case of a
 * paged @skb is used, the driver is recommended to put the ieee80211
 * header of the frame on the linear part of the @skb to avoid memory
 * allocation and/or memcpy by the stack.
 *
 * This function may not be called in IRQ context. Calls to this function
 * for a single hardware must be synchronized against each other. Calls to
 * this function, ieee80211_rx_ni() and ieee80211_rx_irqsafe() may not be
 * mixed for a single hardware. Must not run concurrently with
 * ieee80211_tx_status() or ieee80211_tx_status_ni().
 *
 * In process context use instead ieee80211_rx_ni().
 *
 * @hw: the hardware this frame came in on
 * @skb: the buffer to receive, owned by mac80211 after this call
 */
static inline void ieee80211_rx(struct ieee80211_hw *hw, struct sk_buff *skb)
{
    ieee80211_rx_napi(hw, NULL, skb, NULL);
}
```

#### \*Phân tích:

Theo source code và assembly code, ta suy ra được:

eax tương ứng với ah->hw là một struct ieee80211\_hw.

edx tương ứng với skb là một struct sk\_buff.

- Xem ah->hw (result in struct/ieee80211\_hw\_mac80211)

gdb\$ p \*((struct ieee80211\_hw \*) \$eax)

- Xem skb (result in struct/sk\_buff\_mac80211)

gdb\$ p \*((struct sk\_buff \*) \$edx)

#### **Kết luận:** rx path

ath5k\_intr() → [ath5k\_schedule\_rx()] → ath5k\_tasklet\_rx() →

[ath5k\_receive\_frame\_ok()] → [ath5k\_receive\_frame()] →

ieee80211\_rx()

(Các hàm để trong [] là hàm được nhúng vào hàm gọi nó chứ không thật sự tồn tại)

- Sau khi đợi trong queue đến lượt được thực thi,

ath5k\_tasklet\_rx() sẽ kiểm tra ta có muốn nhận frame này hay không

(ath5k\_receive\_frame\_ok()). Nếu có thực hiện một số thay đổi trên

biến ah và skb (ath5k\_receive\_frame()) rồi pass lên mac80211

(ieee80211\_rx()).