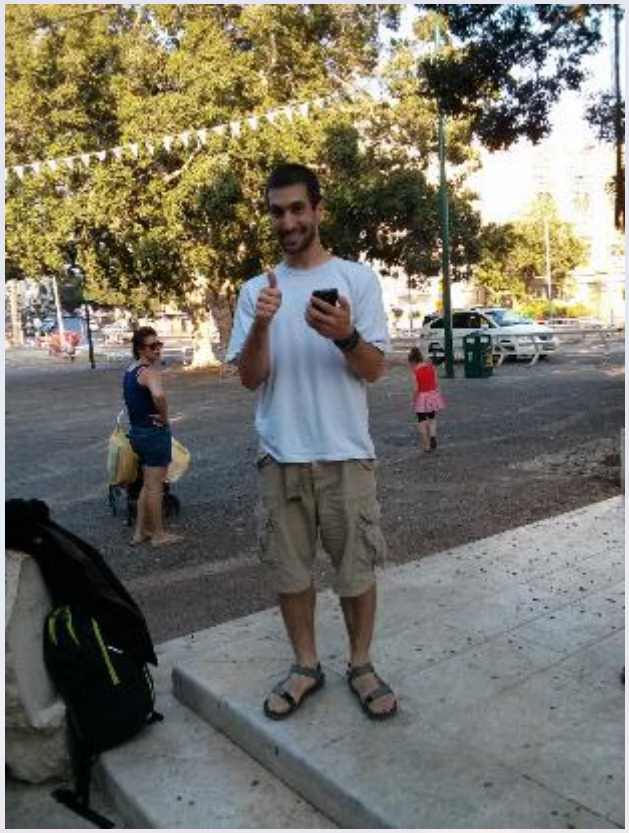
Wardriving from your pocket

Using Wireshark to Reverse Engineer Broadcom WiFi chipsets

Omri Ildis
Yuval Ofir
Ruby Feinstein



Motivation!

```
d:\> adb shell
# iwconfig

iwconfig

eth0      IEEE 802.11-DS  ESSID:"GeverNet"  Nickname:""
          Mode:Managed  Frequency:2.437 GHz  Access Point: 00:25:9C:34:D4:F6
          Bit Rate=11 Mb/s   Tx-Power:32 dBm
          Retry min limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Managementmode:All packets received
          Link Quality=3/5  Signal level=-68 dBm  Noise level=-84 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:4  Invalid misc:0   Missed beacon:0

# tcpdump -i eth0 -w /data/data/bcmon/test.cap

tcpdump -i eth0 -w /data/data/bcmon/test.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
```

802.11

MGMT

CONTROL

DATA

BEACON

CTS

DATA

ASS REQ

802.11

MGMT

CONTROL

DATA

ETHERNET

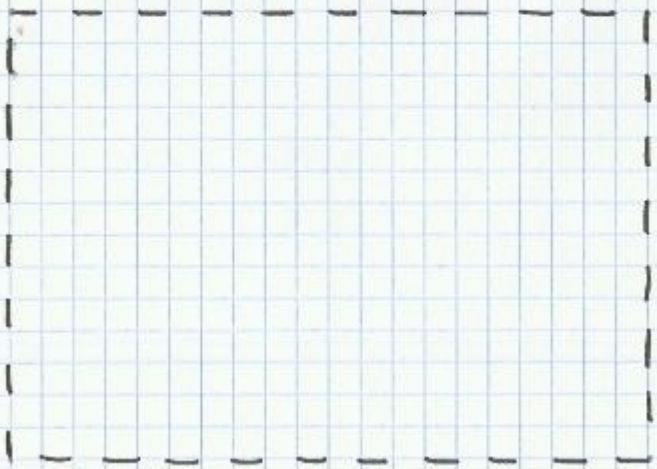
BEACON

CTS

DATA

ASS REQ

/dev/null



AIR
802.11

LINUX
ETHER
NET

```
/* Return true if there may be more frames to read */
static uint dhdsdio_readframes(dhd_bus_t *bus,
                               uint maxframes, bool *finished){
        .
        .
        .

#ifdef DHD_DEBUG
        if (DHD_BYTES_ON() && DHD_DATA_ON()) {
                printk(KERN_DEBUG "Rx Data:\n");
                print_hex_dump_bytes("", DUMP_PREFIX_OFFSET,
                                                rxbuf, len);
        }
#endif
        .
        .
        .
}
```

```
.
.
.
#ifdef DHD_DEBUG

#define DHD_ERROR(args)  do {if (dhd_msg_level & DHD_ERROR_VAL) printf args;} while (0)
#define DHD_TRACE(args)  do {if (dhd_msg_level & DHD_TRACE_VAL) printf args;} while (0)
#define DHD_INFO(args)   do {if (dhd_msg_level & DHD_INFO_VAL) printf args;} while (0)
#define DHD_DATA(args)   do {if (dhd_msg_level & DHD_DATA_VAL) printf args;} while (0)
#define DHD_CTL(args)    do {if (dhd_msg_level & DHD_CTL_VAL) printf args;} while (0)
#define DHD_TIMER(args)  do {if (dhd_msg_level & DHD_TIMER_VAL) printf args;} while (0)
#define DHD_HDRS(args)   do {if (dhd_msg_level & DHD_HDRS_VAL) printf args;} while (0)
#define DHD_BYTES(args)  do {if (dhd_msg_level & DHD_BYTES_VAL) printf args;} while (0)
#define DHD_INTR(args)   do {if (dhd_msg_level & DHD_INTR_VAL) printf args;} while (0)
#define DHD_GLOM(args)   do {if (dhd_msg_level & DHD_GLOM_VAL) printf args;} while (0)
#define DHD_EVENT(args)  do {if (dhd_msg_level & DHD_EVENT_VAL) printf args;} while (0)
#define DHD_BTA(args)    do {if (dhd_msg_level & DHD_BTA_VAL) printf args;} while (0)
#define DHD_ISCAN(args)  do {if (dhd_msg_level & DHD_ISCAN_VAL) printf args;} while (0)
.
.
.
```

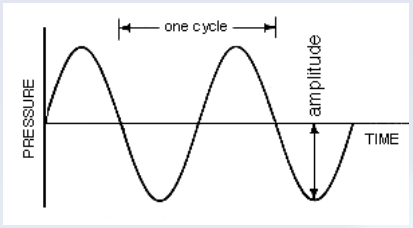
```
# dmesg
        .
        .
        .
dhdsdio_readframes: Enter
RxHdr:
 0000: 74 00 8b ff 1a 02 00 0e 00 12 00 00
Rx Data:
 0000: 74 00 8b ff 1a 02 00 0e 00 12 00 00 00 00 10 00
 0016: 00 ba 38 e7 d8 64 22 02 00 25 9c 34 d4 f4 08 00
 0032: 45 00 00 54 00 00 00 00 2b 01 0f b7 08 08 08 08
 0048: 0a 64 65 7f 00 00 4b cb 27 08 00 01 6f 36 eb 50
 0064: 45 a1 02 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
 0080: 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
 0096: 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33
 0112: 34 35 36 37
        .
        .
        .
```

```
# dmesg
        .
        .
        .
dhdsdio_readframes: Enter
RxHdr:
 0000: 74 00 8b ff 1a 02 00 0e 0
Rx Data:
 0000: 74 00 8b ff 1a 02 00 0e 00 12 0  0  00 00 10 00
 0016: 00 ba 38 e7 d8 64 22 02 00 25 9c 34 d4 f4 08 00
 0032: 45 00 00 54 00 00 00 00 2b 01 0f b7 08 08 08 08
 0048: 0a 64 65 7f 00 00 4b cb 27 08 00 01 6f 36 eb 50
 0064: 45 a1 02 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
 0080: 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
 0096: 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33
 0112: 34 35 36 37
        .
        .
        .
```

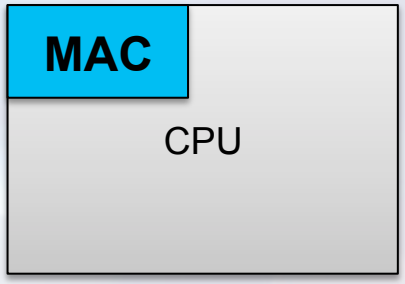
ETHERNET HEADER

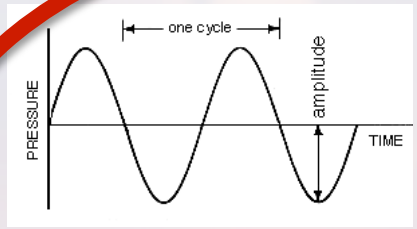
```
# dmesg
        .
        .
        .
dhdsdio_readframes: Enter
RxHdr:
 0000: 74 00 8b ff 1a 02 00
Rx Data:
 0000: 74 00 8b ff 1a 02 00 0e 00 12 00 00 00 00 10 00
 0016: 00 ba 38 e7 d8 64 22 02 00 25 9c 34 d4 f4 08 00
 0032: 45 00 00 54 00 00 00 00 2b 01 0f b7 08 08 08 08
 0048: 0a 64 65 7f 00 00 4b cb 27 08 00 01 6f 36 eb 50
 0064: 45 a1 02 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
 0080: 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
 0096: 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33
 0112: 34 35 36 37
        .
        .
        .
```

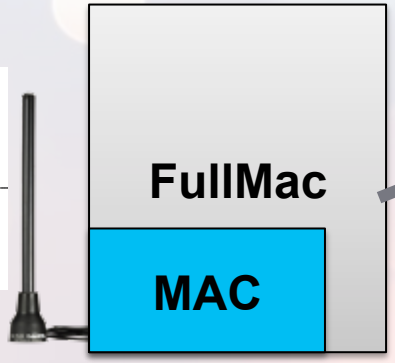
BROADCOM HEADER



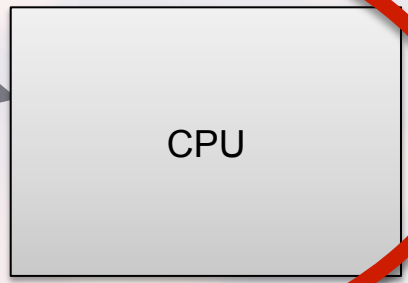
WiFi frames





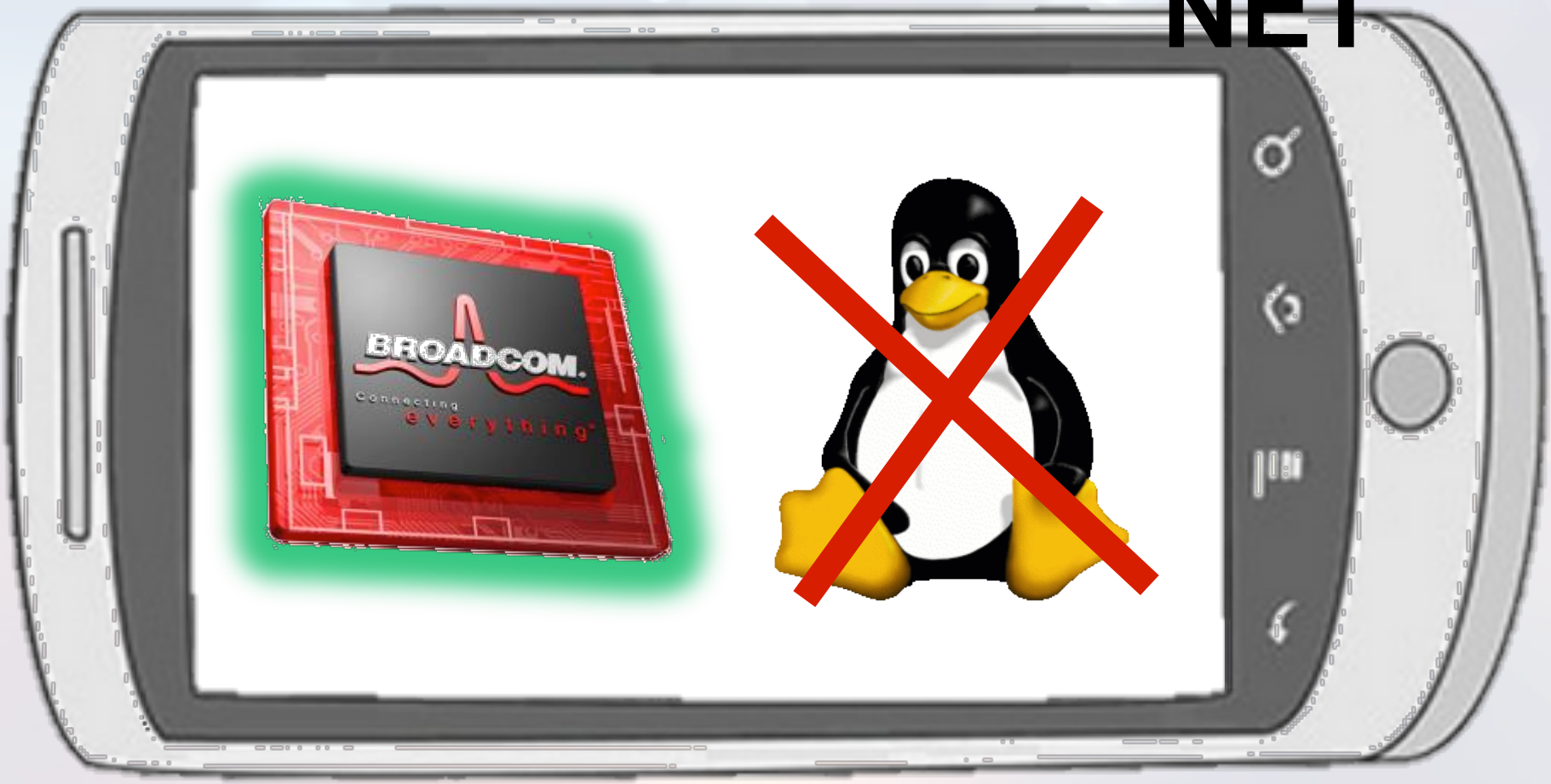


Ethernet frames



AIR
802.11

LINUX
ETHER
NET



WIKIPEDIA
The Free Encyclopedia

Monitor mode

From Wikipedia, the free encyclopedia

Monitor mode, or RFMON (Radio Frequency MONitor) mode, allows a computer with a wireless network interface controller (WNIC) to monitor all traffic received from the wireless network. Unlike promiscuous mode, which is also used for packet sniffing, monitor mode allows packets to be captured

tasks

- Easy solutions
  - ☒ Driver modification
  - ☒ Contact Broadcom
- Implement monitor mode ourselves
  - Extract Firmware ←
  - RE firmware
  - Patch firmware - add monitor mode

```
# dmesg
.
.
.
dhdsdio_download_code_file: download firmware /system/
vendor/firmware/fw_bcm4329.bin
dhdsdio_membytes: write 2048 bytes at offset 0x00000000
dhdsdio_membytes: write 2048 bytes at offset 0x00000800
dhdsdio_membytes: write 2048 bytes at offset 0x00001000
dhdsdio_membytes: write 2048 bytes at offset 0x00001800
dhdsdio_membytes: write 2048 bytes at offset 0x00002000
dhdsdio_membytes: write 2048 bytes at offset 0x00002800
dhdsdio_membytes: write 2048 bytes at offset 0x00003000
.
.
.
```

tasks

- [x] Easy solutions
- [ ] Implement monitor mode ourselves
  - [x] Extract Firmware ←
  - [ ] RE firmware ←
  - [ ] Patch firmware - add monitor mode

Hex Workshop - [E:\temp\play\fw_bcm4329.bin]

File Edit Disk Options Tools Window Help

BROADCOM®

Connecting everything®

BCM4330 PRODUCT Brief

SINGLE-CHIP IEEE 802.11n/BLUETOOTH/FM (Rx AND Tx)

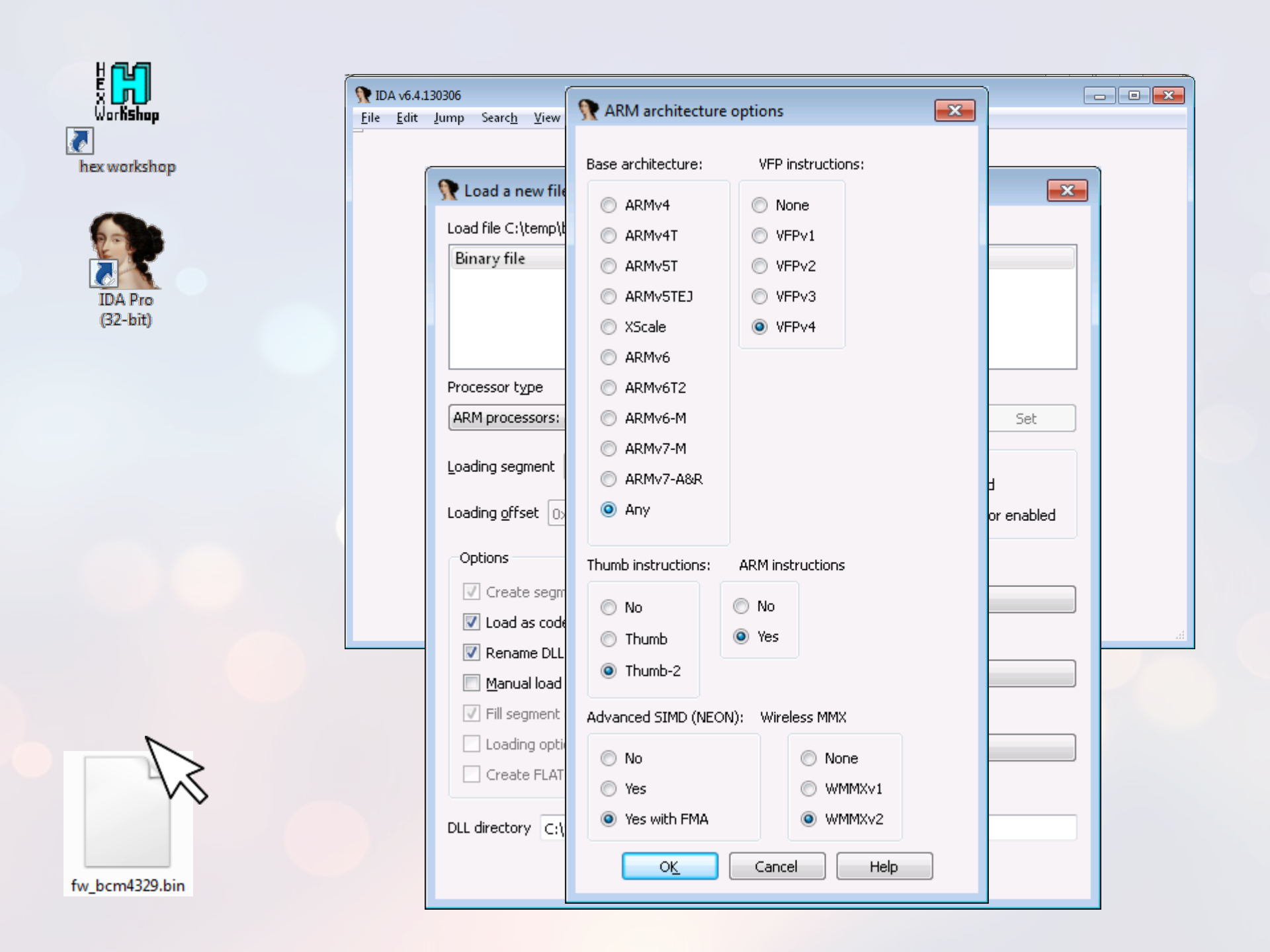
FEATURES

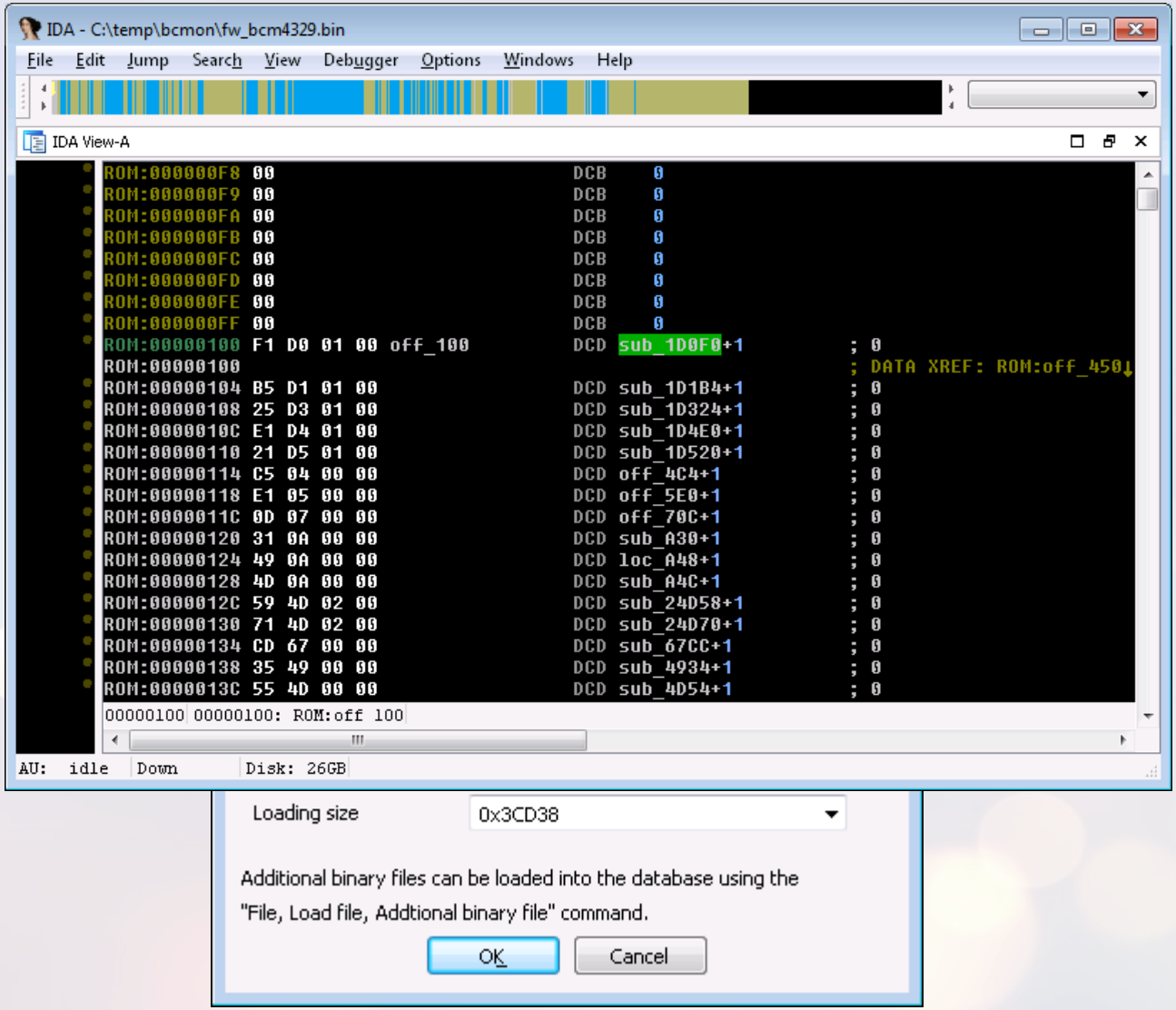
IEEE 802.11x Key Features

- Single-band 2.4 GHz IEEE 802.11 b/g/n or dual-band 2.4 GHz and 5 GHz 802.11 a/b/g/n (non-simultaneous).
- Single-stream IEEE 802.11n support for 20 MHz channels provides PHY layer rates up to MCS7 (72 Mbps) for typical upper-layer throughput in excess of 45 Mbps.
- Integrated Power Amplifiers, LNA, and PMU.
- Supports a single antenna shared between WLAN 5G, WLAN 2.4G, and Bluetooth® blocks.
- Supports optional external antenna diversity.
- Shared Bluetooth and WLAN receive signal path eliminates the need for an external power splitter while maintaining excellent sensitivity for both Bluetooth and WLAN.
- Internal fractional nPLL allows support for a wide range of reference clock frequencies.

- Supports IEEE 802.15.2 external three-wire coexistence scheme to support additional wireless technologies such as GPS, WiMax, or UWB.
- Supports standard SDIO v2.0 (50 MHz, 4-bit, and 1-bit), and gSPI (48 MHz) host interfaces.
- Alternative host interface supports HSIC (short distance USB device).
- Integrated ARM® Cortex™-M3 processor and on-chip memory for complete WLAN subsystem functionality, minimizing the need to wake up the application processor for standard WLAN functions. This allows for further minimization of power consumption, while maintaining the ability to field-upgrade with future features.
- OneDriver™ software architecture for easy migration from existing embedded WLAN and Bluetooth devices as well as future devices.

fw_bcm4329.bin


hex workshop
IDA Pro
(32-bit)
fw_bcm4329.bin
IDA v6.4.130306
File Edit Jump Search View
Load a new file
Load file C:\temp\b
Binary file
Processor type
ARM processors:
Loading segment
Loading offset
Options
Create segm
Load as code
Rename DLL
Manual load
Fill segment
Loading opti
Create FLAT
DLL directory
C:\
Set
or enabled
ARM architecture options
Base architecture:
ARMv4
ARMv4T
ARMv5T
ARMv5TEJ
XScale
ARMv6
ARMv6T2
ARMv6-M
ARMv7-M
ARMv7-A&R
Any
VFP instructions:
None
VFPv1
VFPv2
VFPv3
VFPv4
Thumb instructions:
No
Thumb
Thumb-2
ARM instructions
No
Yes
Advanced SIMD (NEON):
No
Yes
Yes with FMA
Wireless MMX
None
WMMXv1
WMMXv2
OK
Cancel
Help


IDA - C:\temp\bcmon\fw_bcm4329.bin
File Edit Jump Search View Debugger Options Windows Help
IDA View-A
ROM:000000F8 00 DCB 0
ROM:000000F9 00 DCB 0
ROM:000000FA 00 DCB 0
ROM:000000FB 00 DCB 0
ROM:000000FC 00 DCB 0
ROM:000000FD 00 DCB 0
ROM:000000FE 00 DCB 0
ROM:000000FF 00 DCB 0
ROM:00000100 F1 D0 01 00 off_100 DCD sub_1D0F0+1 ; 0
ROM:00000100 ; DATA XREF: ROM:off_450↓
ROM:00000104 B5 D1 01 00 DCD sub_1D1B4+1 ; 0
ROM:00000108 25 D3 01 00 DCD sub_1D324+1 ; 0
ROM:0000010C E1 D4 01 00 DCD sub_1D4E0+1 ; 0
ROM:00000110 21 D5 01 00 DCD sub_1D520+1 ; 0
ROM:00000114 C5 04 00 00 DCD off_4C4+1 ; 0
ROM:00000118 E1 05 00 00 DCD off_5E0+1 ; 0
ROM:0000011C 0D 07 00 00 DCD off_70C+1 ; 0
ROM:00000120 31 0A 00 00 DCD sub_A30+1 ; 0
ROM:00000124 49 0A 00 00 DCD loc_A48+1 ; 0
ROM:00000128 4D 0A 00 00 DCD sub_A4C+1 ; 0
ROM:0000012C 59 4D 02 00 DCD sub_24D58+1 ; 0
ROM:00000130 71 4D 02 00 DCD sub_24D70+1 ; 0
ROM:00000134 CD 67 00 00 DCD sub_67CC+1 ; 0
ROM:00000138 35 49 00 00 DCD sub_4934+1 ; 0
ROM:0000013C 55 4D 00 00 DCD sub_4D54+1 ; 0
00000100 00000100: ROM:off_100
AU: idle Down Disk: 26GB
Loading size
0x3CD38
Additional binary files can be loaded into the database using the
"File, Load file, Addtional binary file" command.
OK
Cancel

hex workshop
IDA Pro
(32-bit)
fw_bcm4329.bin

tasks

- [x] Easy solutions
- [ ] Implement monitor mode ourselves
  - [x] Extract Firmware
  - [x] Determine Architecture
  - [ ] RE firmware
  - [ ] Patch firmware - add monitor mode
    - [ ] Is it possible?

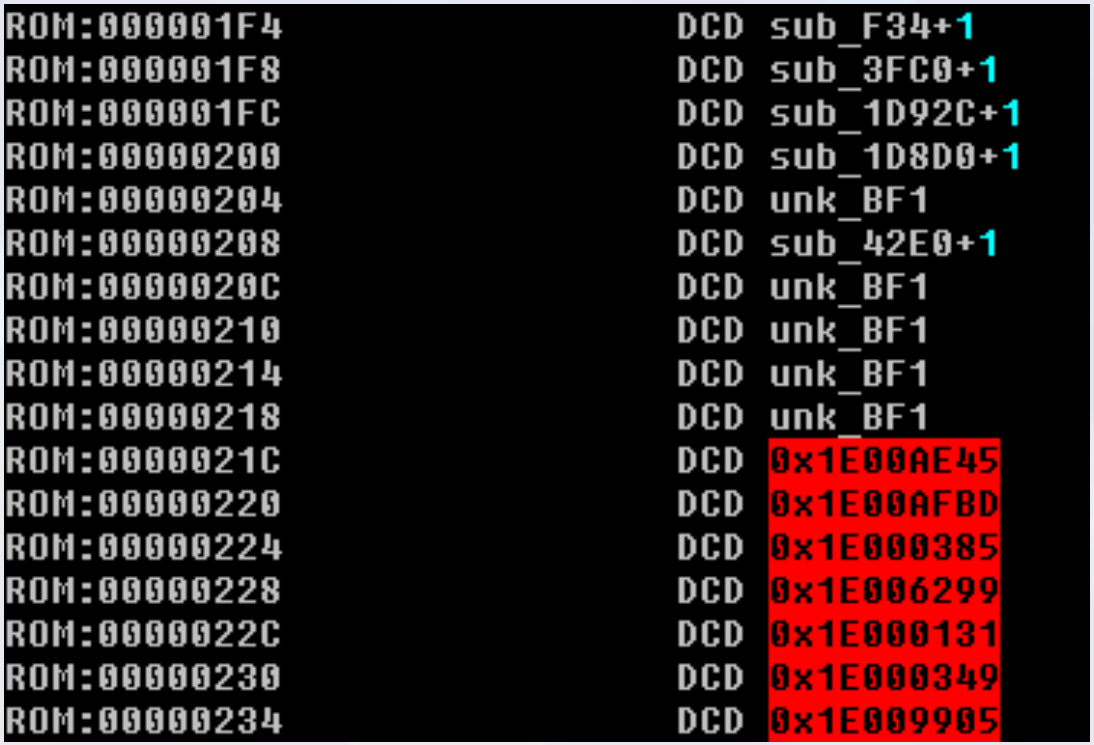
```
# dmesg

.
.
.
Firmware version = wl0: Dec 20 2010
19:21:39 version 4.218.248.18
.
.
.
```

Firmware version = wl0: Dec 20 2010 19:21:39 version 4.218.248.18

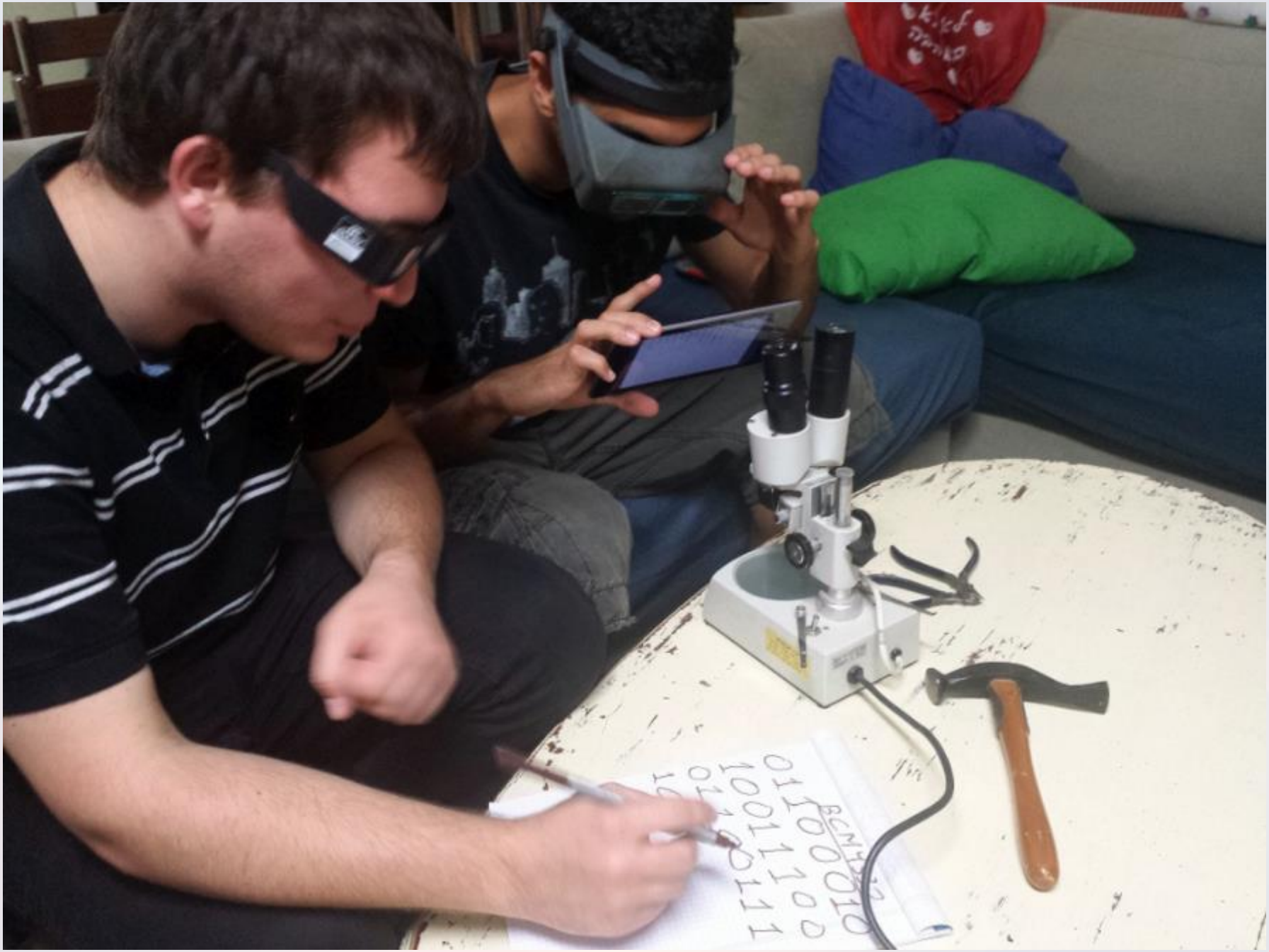
```
000284B0   00 34 2E 32 31 38 2E 32    .4.218.2
000284B8   34 38 2E 31 38 00 77 6C    48.18.wl
```

```
000284B0   00 62 63 6D 6F 6E 00 32    .bcmon.2
000284B8   34 38 2E 31 38 00 77 6C    48.18.wl
```

Firmware version = wl0: Dec 20 2010 19:21:39 version bcmon



~67% of the functions are



```
#
.
.
.
d
v
d
dh
dhdsdio_mem          write 2048 bytes at offset 0x00001000
dhdsdio_membytes: write 2048 bytes at offset 0x00001800
dhdsdio_membytes: write 2048 bytes at offset 0x00002000
dhdsdio_membytes: write 2048 bytes at offset 0x00002800
dhdsdio_membytes: write 2048 bytes at offset 0x00003000
.
.
.
```

```
dhdsdio_membytes(dhd_bus_t *bus,
                 bool write,
                 u32 address,
                 u8 *data,
                 uint size)
```

```
static int
dhdsdio_download_code_file(struct dhd_bus *bus, char *fw_path){
        .
        .
        .
 /* Download image */
     while ((len = dhd_os_get_image_block((char*)memptr, MEMBLOCK, image))) {
          bcmerror = dhdsdio_membytes(bus, TRUE, offset, memptr, len);
          if (bcmerror) {...}
          offset += MEMBLOCK;
     }
        .
        .
        .
}
```

```
static int
dhdsdio_download_code_file(struct dhd_bus *bus, char *fw_path){
        .
        .
        .
 /* Download image */
     while ((len = dhd_os_get_image_block((char*)memptr, MEMBLOCK, image))) {
          bcmerror = dhdsdio_membytes(bus, TRUE, offset, memptr, len);
          if (bcmerror) {...}
          offset += MEMBLOCK;
     }
     pasten_func(bus);
        .
        .
        .
}
```

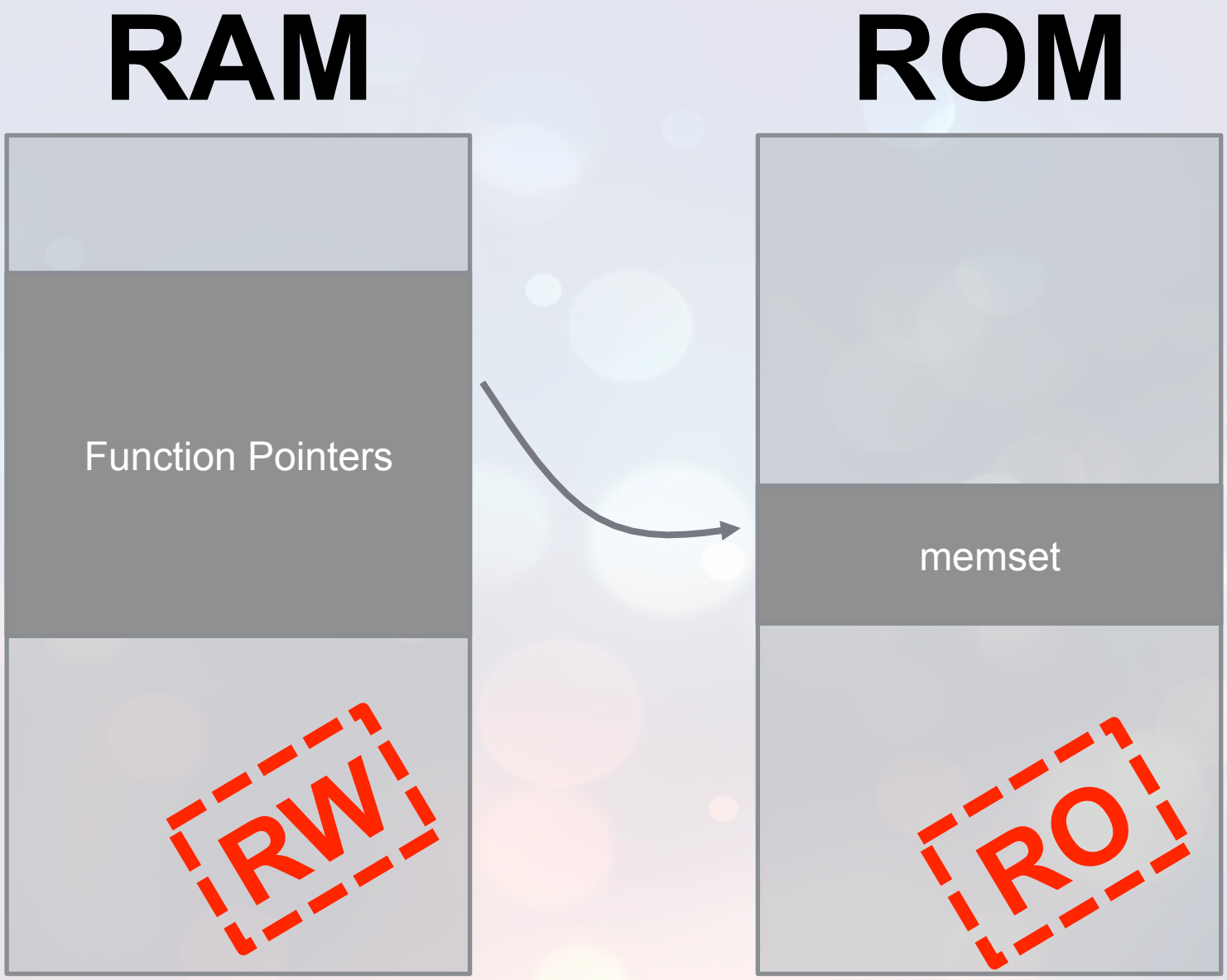
```
int pasten_func (struct dhd_bus *bus)
{
    bool write;
    uint32 address;
    uint8 data[0x100];
    uint32 chunk_sz = 0x100;
    printf("pasten_func: enter!!!\n");
    write = false;
    address = 0x1E000000;
    while(true)
    {
        dhdsdio_membytes(bus, write, address, data, chunk_sz);
        printf("address = 0x%X\n", address);
        hexdump("pasten OUT: ", data, chunk_sz);
        address += chunk_sz;
    }
    printf("pasten_func: return!!!\n");
    return 0;
}
```

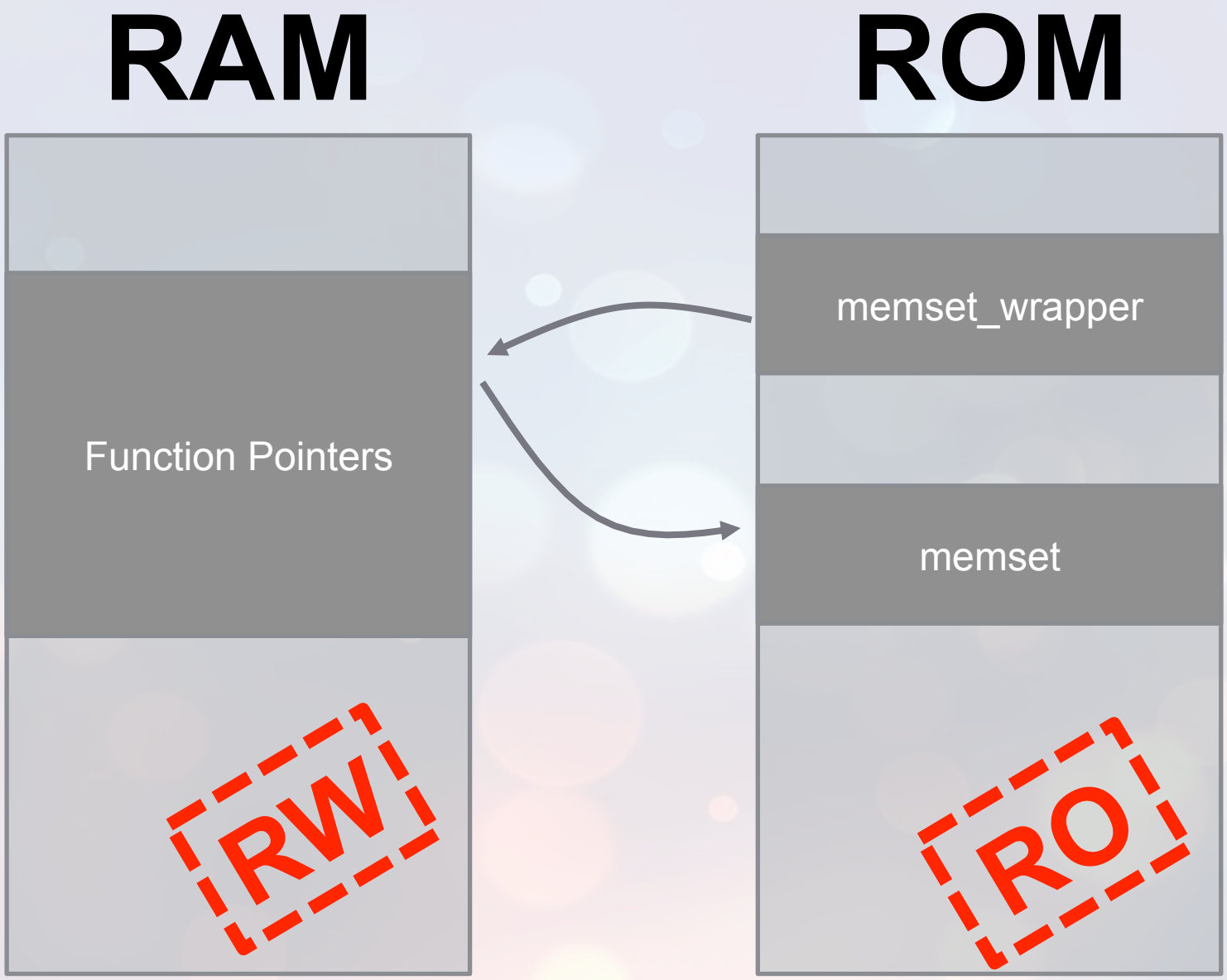
```
# dmesg

        .
        .
        .
address = 0x1E000000
pasten OUT: 13 b5 01 90 00 91 05 e0 07 4b 01 a9 1b 68 6a 46
pasten OUT: 98 47 38 b9 01 98 00 99 dd 22 0f f0 b1 fc 04 46
pasten OUT: 00 28 f1 d1 20 46 1c bd 5c 03 00 00 2d e9 f0 41
pasten OUT: 43 78 45 1c 04 2b 04 46 0f 46 16 46 07 d9 02 30
pasten OUT: 09 49 04 22 0f f0 a4 fc 08 b9 01 30 0a e0 2b 78
pasten OUT: 3a 68 a1 1c c9 18 33 68 c2 eb 01 02 9b 1a 33 60
pasten OUT: 39 60 00 20 bd e8 f0 81 08 fe 00 1e 10 b5 04 46
pasten OUT: 4f f0 00 0e 10 46 09 e0 1e f8 01 c0 1e f8 00 20
pasten OUT: 94 45 01 d0 34 b9 08 e0 0e f1 01 0e 9e 45 f3 d1
pasten OUT: 00 20 0b e0 62 45 08 d9 08 e0 10 f8 0e 20 11 f8
pasten OUT: 0e 30 9a 42 28 bf 08 46 00 e0 08 46 10 bd c0 46
        .
        .
        .
```

```
ROM:000001F4 off_1F4          DCD sub_F34+1
ROM:000001F8 off_1F8          DCD sub_3FC0+1
ROM:000001FC off_1FC          DCD sub_1D92C+1
ROM:00000200 off_200          DCD sub_1D8D0+1
ROM:00000204 off_204          DCD sub_BF0+1
ROM:00000208 off_208          DCD sub_42E0+1
ROM:0000020C off_20C          DCD sub_BF0+1
ROM:00000210 off_210          DCD sub_BF0+1
ROM:00000214 off_214          DCD sub_BF0+1
ROM:00000218 off_218          DCD sub_BF0+1
ROM:0000021C off_21C          DCD sub_1E00AE44+1
ROM:00000220 off_220          DCD sub_1E00AFBC+1
ROM:00000224 off_224          DCD sub_1E000384+1
ROM:00000228 off_228          DCD sub_1E006298+1
ROM:0000022C off_22C          DCD sub_1E000130+1
ROM:00000230 off_230          DCD sub_1E000348+1
ROM:00000234 off_234          DCD sub_1E009904+1
```

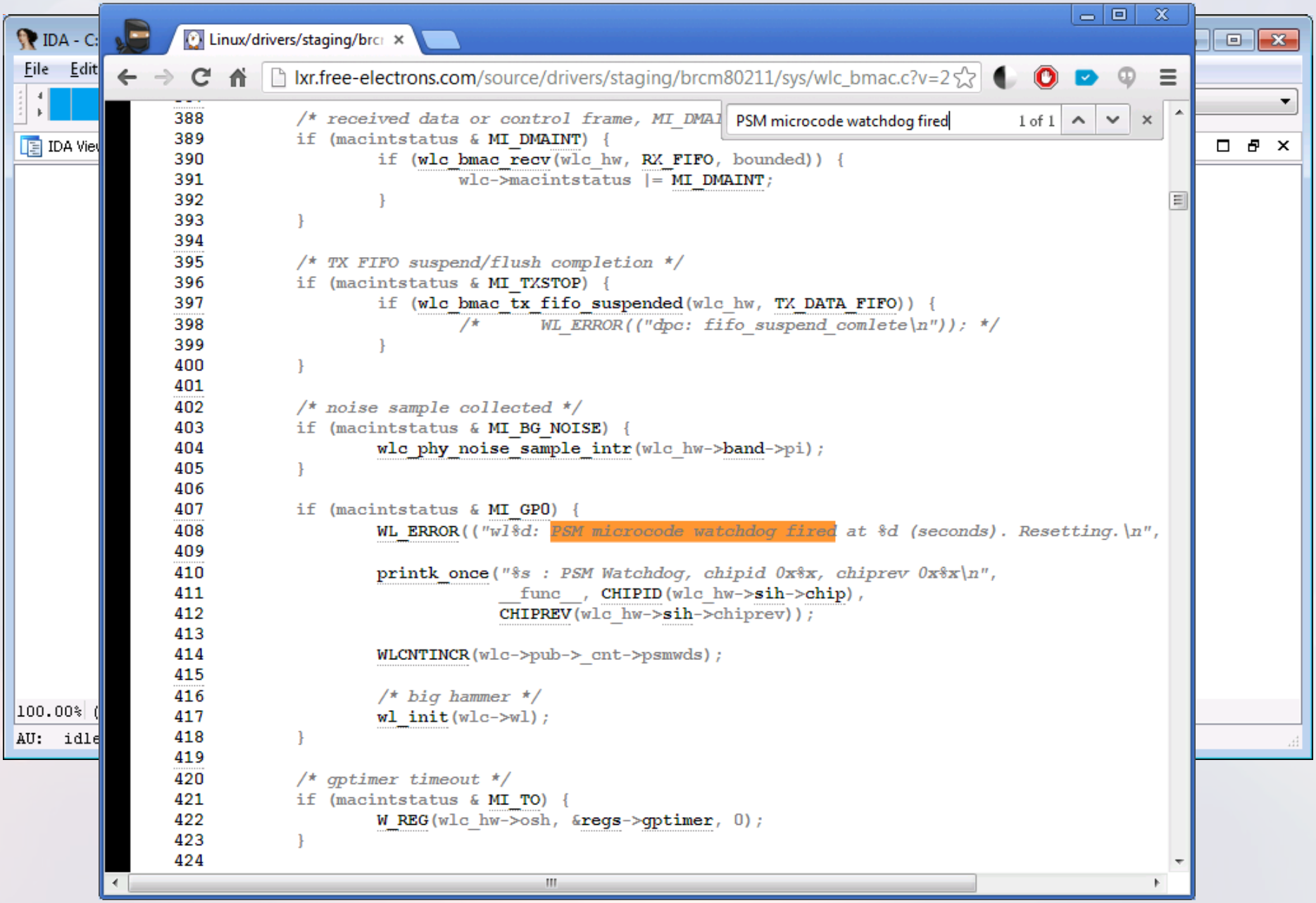
tasks

- [x] Easy solutions
- [ ] Implement monitor mode ourselves
  - [x] Extract Firmware ←
  - [ ] RE firmware ←
  - [ ] Patch firmware - add monitor mode
    - [x] Is it possible?

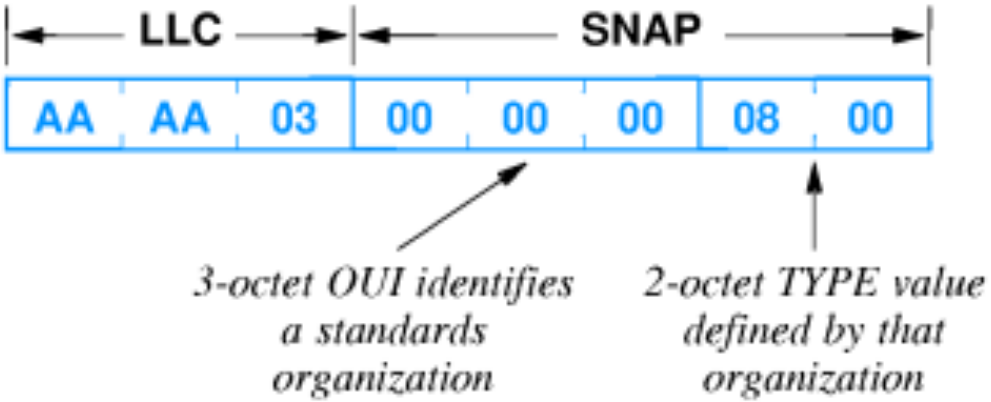

RAM
ROM
Function Pointers
memset
RW
RO

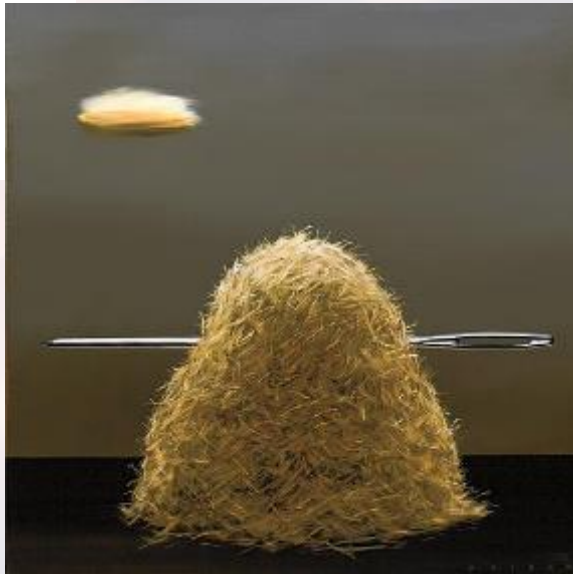

RAM
ROM
Function Pointers
memset_wrapper
memset
RW
RO

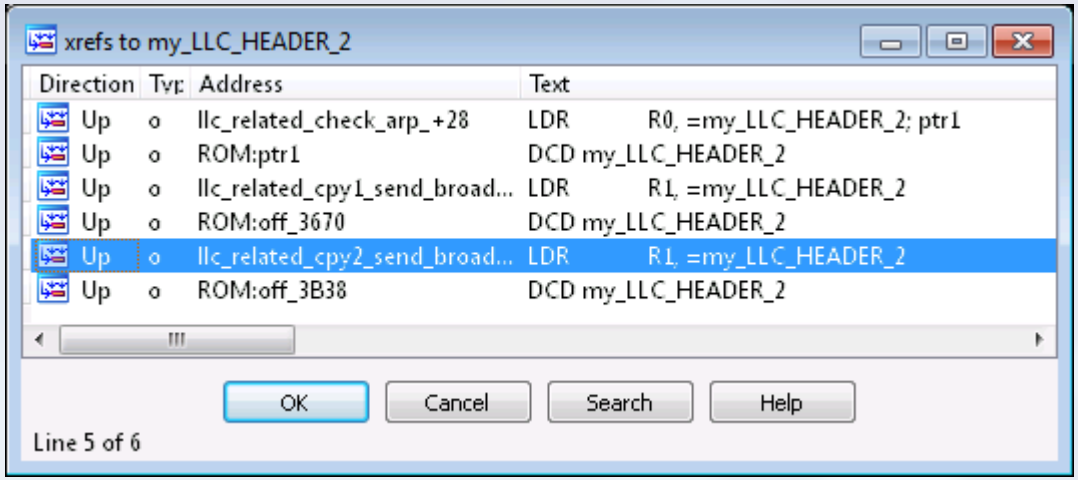

DEBUG
PRINTS

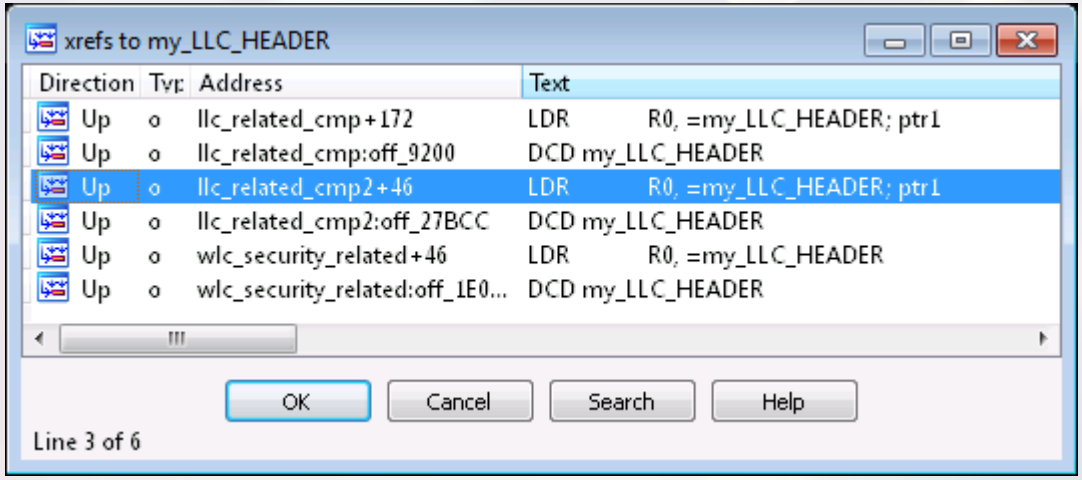

IDA - C:
File Edit
IDA View
100.00% (
AU: idle
Linux/drivers/staging/brc
lxr.free-electrons.com/source/drivers/staging/brcm80211/sys/wlc_bmac.c?v=2
PSM microcode watchdog fired
1 of 1
388 /* received data or control frame, MI_DMAI
389 if (macintstatus & MI_DMAINT) {
390 if (wlc_bmac_recv(wlc_hw, RX_FIFO, bounded)) {
391 wlc->macintstatus |= MI_DMAINT;
392 }
393 }
394
395 /* TX FIFO suspend/flush completion */
396 if (macintstatus & MI_TXSTOP) {
397 if (wlc_bmac_tx_fifo_suspended(wlc_hw, TX_DATA_FIFO)) {
398 /* WL_ERROR(("dpc: fifo_suspend_comlete\n")); */
399 }
400 }
401
402 /* noise sample collected */
403 if (macintstatus & MI_BG_NOISE) {
404 wlc_phy_noise_sample_intr(wlc_hw->band->pi);
405 }
406
407 if (macintstatus & MI_GP0) {
408 WL_ERROR(("wl%d: PSM microcode watchdog fired at %d (seconds). Resetting.\n",
409
410 printk_once("%s : PSM Watchdog, chipid 0x%x, chiprev 0x%x\n",
411 __func__, CHIPID(wlc_hw->sih->chip),
412 CHIPREV(wlc_hw->sih->chiprev));
413
414 WLCNTINCR(wlc->pub->_cnt->psmwds);
415
416 /* big hammer */
417 wl_init(wlc->wl);
418 }
419
420 /* gptimer timeout */
421 if (macintstatus & MI_TO) {
422 W_REG(wlc_hw->osh, &regs->gptimer, 0);
423 }
424

Why Ethernet??? WHY???



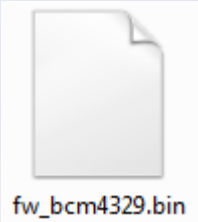



xrefs to my_LLC_HEADER_2
Direction Typ Address Text
Up o llc_related_check_arp_+28 LDR R0, =my_LLC_HEADER_2; ptr1
Up o ROM:ptr1 DCD my_LLC_HEADER_2
Up o llc_related_cpy1_send_broad... LDR R1, =my_LLC_HEADER_2
Up o ROM:off_3670 DCD my_LLC_HEADER_2
Up o llc_related_cpy2_send_broad... LDR R1, =my_LLC_HEADER_2
Up o ROM:off_3B38 DCD my_LLC_HEADER_2
OK
Cancel
Search
Help
Line 5 of 6


xrefs to my_LLC_HEADER
Direction Typ Address Text
Up o llc_related_cmp+172 LDR R0, =my_LLC_HEADER; ptr1
Up o llc_related_cmp:off_9200 DCD my_LLC_HEADER
Up o llc_related_cmp2+46 LDR R0, =my_LLC_HEADER; ptr1
Up o llc_related_cmp2:off_27BCC DCD my_LLC_HEADER
Up o wlc_security_related+46 LDR R0, =my_LLC_HEADER
Up o wlc_security_related:off_1E0... DCD my_LLC_HEADER
OK
Cancel
Search
Help
Line 3 of 6

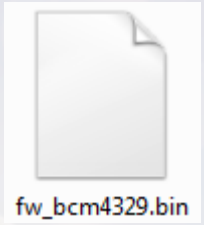
```
push    {r0-r11, lr}
; pre logic
bl      original_function
; post logic
pop     {r0-r11, pc}
```

```
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
```



```
push    {r0-r11, lr}
; pre logic
bl      original_function
;
; memcpy(0x46300, sp, 0x100)
;
pop     {r0-r11, pc}
```

```
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
6F 2D D7 27 03 9E 84 87 94 E1
```




```
int pasten_func (struct dhd_bus *bus)
{
        ...
        printf("pasten_func: enter!!!\n");
        dhdsdio_membytes(...);
        ...
}
```

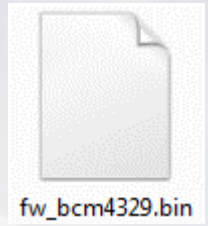
```
push    {r0-
; bug <-----
bl      orig
;
; memcpy(0x4
;
pop     {r0-
```

Stress Reduction Kit

Bang Head Here

Directions:
1. Place kit on FIRM surface.
2. Follow directions in circle of kit.
3. Repeat step 2 as necessary, or until unconscious.
4. If unconscious, cease stress reduction activity.

```
9E 84 87 94 E1
9E 84 87 94 E1
9E 84 87 94 E1
9E 84 87 94 E1
```




```
bus *bus)

enter!!!\n");
```

tasks

- ☒ Easy solutions
- ☐ Implement monitor mode ourselves
  - ☑ Extract Firmware
  - ☐ RE firmware ←
  - ☐ Patch firmware - add monitor mode
    - ☑ Is it possible?
- ☐ Save time
  - ☐ Research infrastructure ←

Developing a patch

- Develop in C
- Easily add illegal opcodes
- Detours
- Assembly patches

generic_patcher.py


```
detour1 = CPatch(code_address,
                 “detour.c”,
                 max_size=0x100)

patch_firmware("fw_bcm4329.bin",
               "patched.bin",
               [detour1])
```

Additional Patch Types

- Assembly

```
ArmPatch(0x00006E30, ["nop", "nop"])
```

- Detour relative jumps

```
BLPatch(bl_ptr, new_dst)
```

- PTR replacement

```
ReplacePatch(old_data, new_data)
```

- Trap

```
TrapPatch(offset)
```

generic_patcher.py


```
int pasten_func (struct dhd_bus *bus)
{
        ...
        printf("pasten_func: enter!!!\n");
        dhdsdio_membytes(...);
        ...
}
```

Usermode Interface

- Read/write chip memory
- Change kernel debug level

Usermode Interface

```
read_mem: address(0x1E000000), count(0x00000100):
  0000  13 b5 01 90 00 91 05 e0 07 4b 01 a9 1b 68 6a 46  .........K...hjF
  0010  98 47 38 b9 01 98 00 99 dd 22 0f f0 b1 fc 04 46  .G8......".....F
  0020  00 28 f1 d1 20 46 1c bd 5c 03 00 00 2d e9 f0 41  .(.. F..\...-..A
  0030  43 78 45 1c 04 2b 04 46 0f 46 16 46 07 d9 02 30  CxE..+.F.F.F...0
  0040  09 49 04 22 0f f0 a4 fc 08 b9 01 30 0a e0 2b 78  .I."......0..+x
  0050  3a 68 a1 1c c9 18 33 68 c2 eb 01 02 9b 1a 33 60  :h....3h......3`
  0060  39 60 00 20 bd e8 f0 81 08 fe 00 1e 10 b5 04 46  9`. ...........F
  0070  4f f0 00 0e 10 46 09 e0 1e f8 01 c0 1e f8 00 20  O....F.........
  0080  94 45 01 d0 34 b9 08 e0 0e f1 01 0e 9e 45 f3 d1  .E..4........E..
  0090  00 20 0b e0 62 45 08 d9 08 e0 10 f8 0e 20 11 f8  . ..bE....... ..
  00a0  0e 30 9a 42 28 bf 08 46 00 e0 08 46 10 bd c0 46  .0.B(..F...F...F
  00b0  2d e9 70 47 bb b0 0d f1 df 04 06 46 88 46 91 46  -.pG.......F.F.F
  00c0  19 49 09 22 20 46 9a 46 0f f0 72 fc 20 46 0f f0  .I." F.F..r. F..
  00d0  b5 f9 0d f1 0f 05 02 46 21 46 28 46 0f f0 68 fc  .......F!F(F..h.
  00e0  20 46 0f f0 ab f9 31 46 04 46 06 22 28 18 0f f0   F....1F.F."(...
  00f0  5f fc a0 1d 41 46 06 22 28 18 0f f0 59 fc 0d f1  _...AF."(...Y...
```

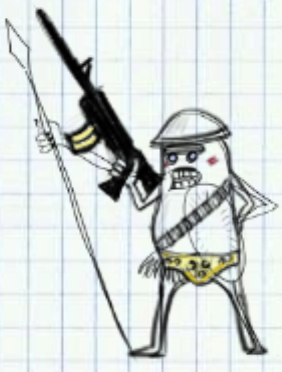
generic_patcher.py

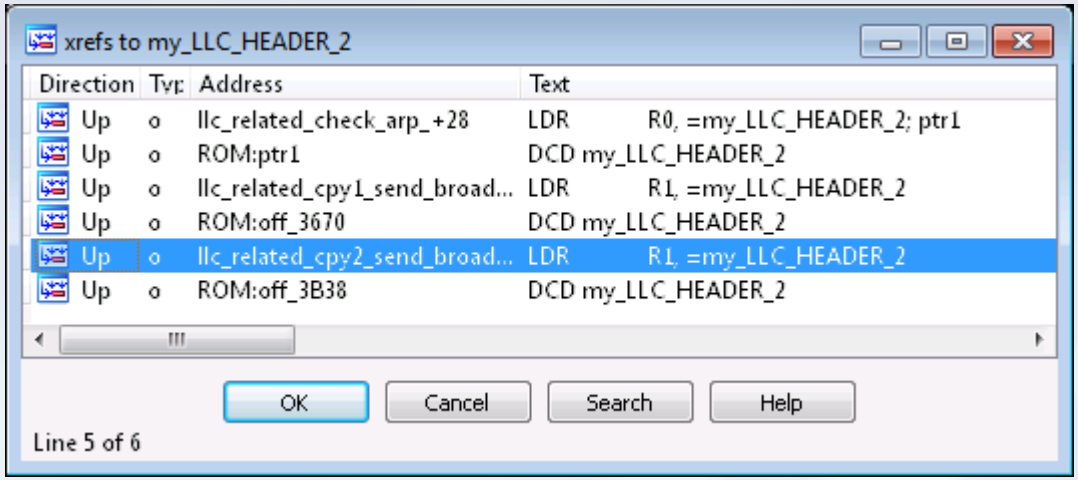
Usermode interface

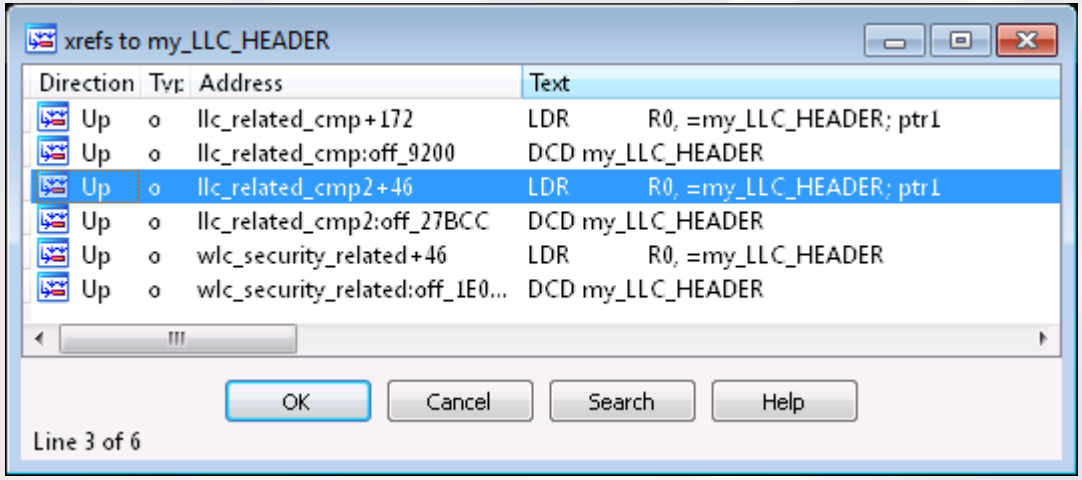
tasks

- ☒ Easy solutions
- ☐ Implement monitor mode ourselves
  - ☑ Extract Firmware
  - ☐ RE firmware
  - ☐ Patch firmware - add monitor mode
    - ☑ Is it possible?
- ☑ Save time
  - ☑ Research infrastructure

- ☐ Find a raw WiFi Packet




xrefs to my_LLC_HEADER_2
Direction Typ Address Text
Up o llc_related_check_arp_+28 LDR R0, =my_LLC_HEADER_2; ptr1
Up o ROM:ptr1 DCD my_LLC_HEADER_2
Up o llc_related_cpy1_send_broad... LDR R1, =my_LLC_HEADER_2
Up o ROM:off_3670 DCD my_LLC_HEADER_2
Up o llc_related_cpy2_send_broad... LDR R1, =my_LLC_HEADER_2
Up o ROM:off_3B38 DCD my_LLC_HEADER_2
OK
Cancel
Search
Help
Line 5 of 6


xrefs to my_LLC_HEADER
Direction Typ Address Text
Up o llc_related_cmp+172 LDR R0, =my_LLC_HEADER; ptr1
Up o llc_related_cmp:off_9200 DCD my_LLC_HEADER
Up o llc_related_cmp2+46 LDR R0, =my_LLC_HEADER; ptr1
Up o llc_related_cmp2:off_27BCC DCD my_LLC_HEADER
Up o wlc_security_related+46 LDR R0, =my_LLC_HEADER
Up o wlc_security_related:off_1E0... DCD my_LLC_HEADER
OK
Cancel
Search
Help
Line 3 of 6

Our first raw 802.11 packet!

wlc_bmac_recv

```
...
ROM:0000D35C 04 F1+              ADD.W           R0, R4, #0x10
ROM:0000D360    // got here with a probe request packet
ROM:0000D360 F4 F7+              BL              my_mac_compare_
ROM:0000D364 18 B9               CBNZ            R0, loc_D36E
ROM:0000D366 23 7C               LDRB            R3, [R4,#0x10]
ROM:0000D368 13 F0+              TST.W           R3, #1
```

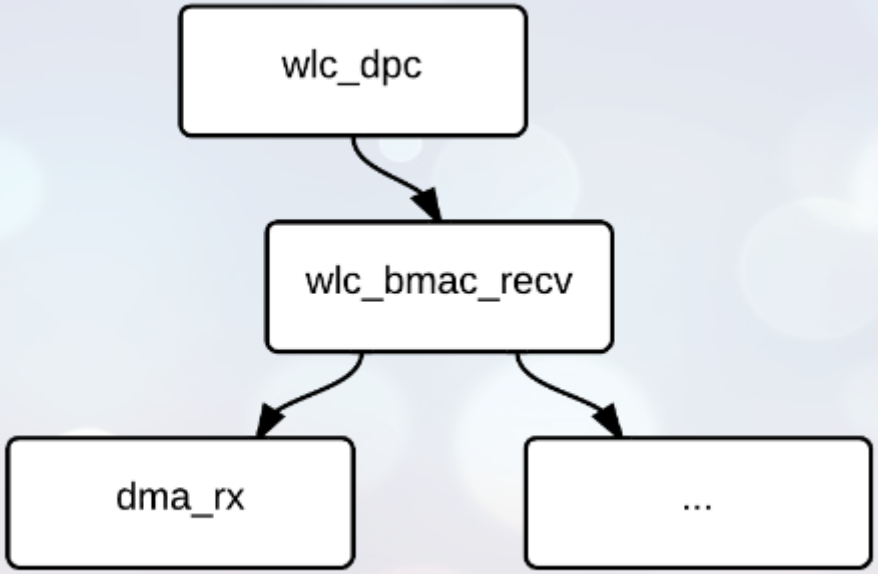
Probe request

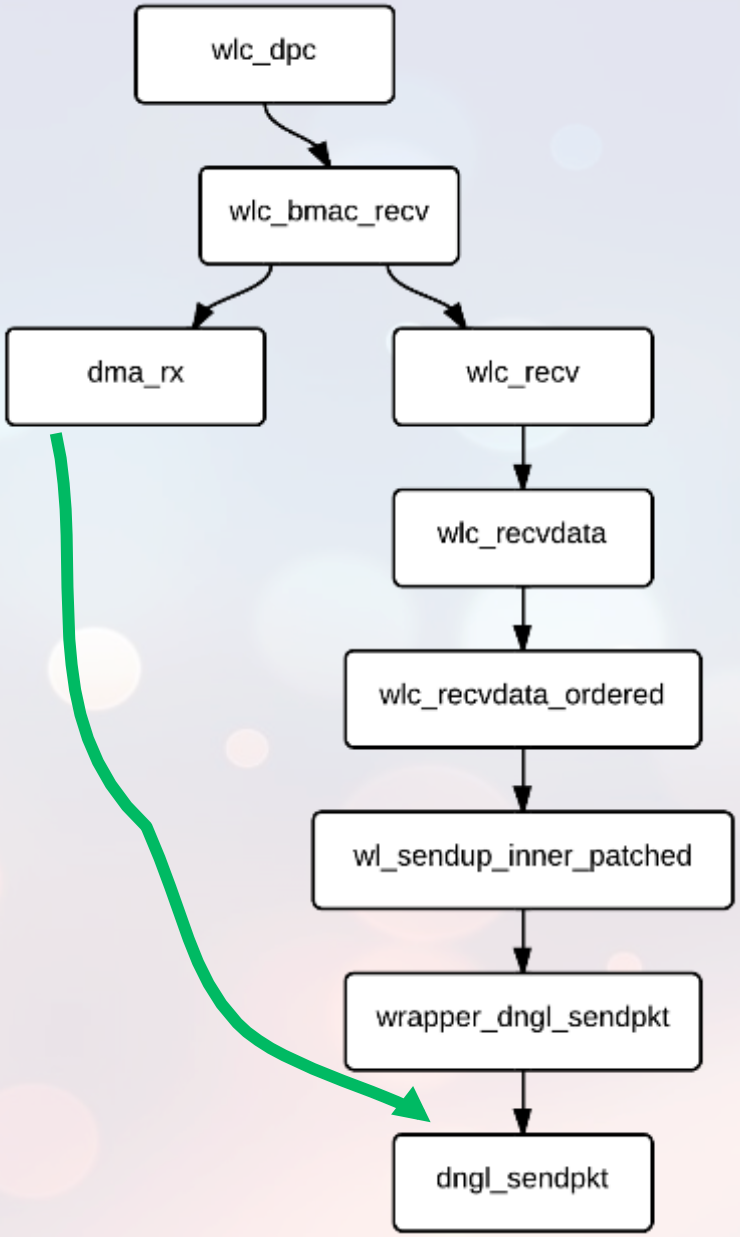
```
0003d510: 00 00 14 04 cc 00 f8 4d 40 00 00 00 ff ff ff ff
0003d520: ff ff 00 21 91 98 7d c3 ff ff ff ff ff ff b0 6b
0003d530: 00 05 4b 61 6c 69 72 01 08 82 84 8b 96 12 24 48
0003d540: 6c 32 04 0c 18 30 60 05 b8 34 27 00 00 00 00 00
```

tasks

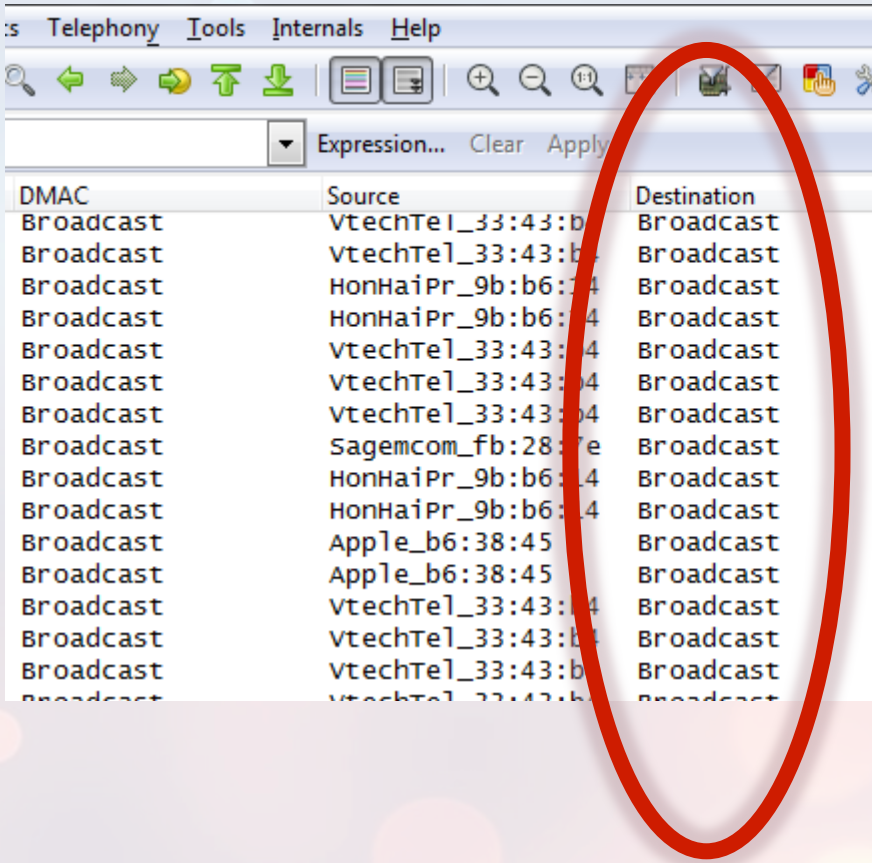
- [x] Easy solutions
- [ ] Implement monitor mode ourselves
  - [x] Extract Firmware
  - [ ] RE firmware
  - [ ] Patch firmware - add monitor mode
    - [x] Is it possible?
- [x] Save time
  - [x] Research infrastructure

- [x] Find a raw WiFi Packet
- [ ] Find a route home


wlc_dpc
wlc_bmac_recv
dma_rx
...


wlc_dpc
wlc_bmac_recv
dma_rx
wlc_recv
wlc_recvdata
wlc_recvdata_ordered
wl_sendup_inner_patched
wrapper_dngl_sendpkt
dngl_sendpkt

Raw packet sniffing

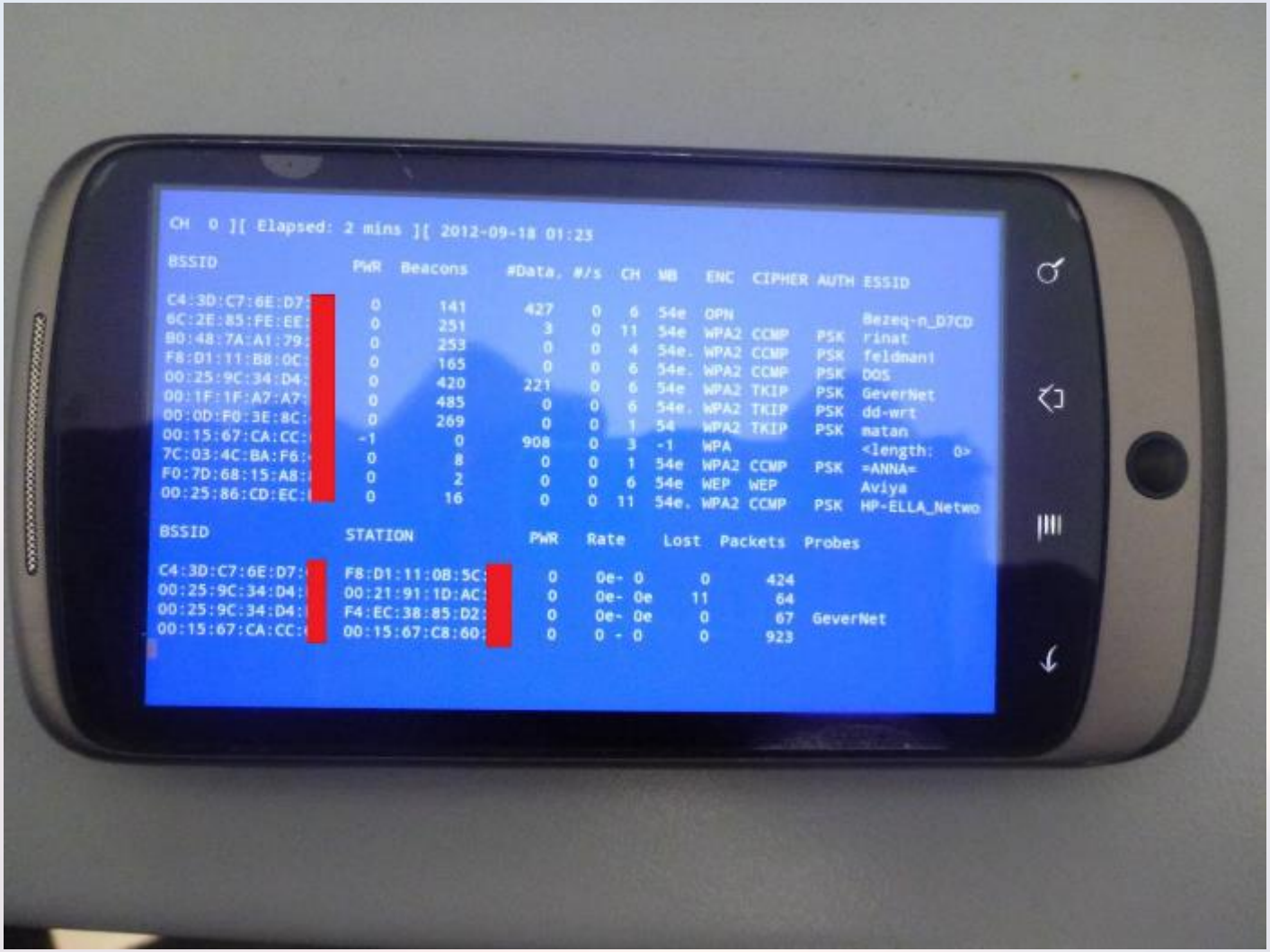


Capture the flag

```
/*== maccontrol register ==*/
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
```

```
void brcms_c_mac_promisc(struct brcms_c_info *wlc, uint filter_flags)
{
        u32 promisc_bits = 0;
        wlc->filter_flags = filter_flags;
        if (filter_flags & (FIF_PROMISC_IN_BSS | FIF_OTHER_BSS))
                promisc_bits |= MCTL_PROMISC;

        if (filter_flags & FIF_BCN_PRBRESP_PROMISC)
                promisc_bits |= MCTL_BCNS_PROMISC;

        if (filter_flags & FIF_FCSFAIL)
                promisc_bits |= MCTL_KEEPBADFCS;

        if (filter_flags & (FIF_CONTROL | FIF_PSPOLL))
                promisc_bits |= MCTL_KEEPCONTROL;

        brcms_b_mctrl(wlc->hw,
                MCTL_PROMISC | MCTL_BCNS_PROMISC |
                MCTL_KEEPCONTROL | MCTL_KEEPBADFCS,
                promisc_bits);
}
```

Result


CH 0][Elapsed: 2 mins][2012-09-18 01:25
BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
C4:3D:C7:6E:D7: 0 141 427 0 6 54e OPN Bezeq-n_D7CD
6C:2E:85:FE:EE: 0 251 3 0 11 54e WPA2 CCMP PSK rinat
B0:48:7A:A1:79: 0 253 0 0 4 54e. WPA2 CCMP PSK feldman1
F8:D1:11:B8:0C: 0 165 0 0 6 54e. WPA2 CCMP PSK DOS
00:25:9C:34:D4: 0 420 221 0 6 54e WPA2 TKIP PSK GeverNet
00:1F:1F:A7:A7: 0 485 0 0 6 54e. WPA2 TKIP PSK dd-wrt
00:0D:F0:3E:8C: 0 269 0 0 1 54 WPA2 TKIP PSK matan
00:15:67:CA:CC: -1 0 908 0 3 -1 WPA <length: 0>
7C:03:4C:BA:F6: 0 8 0 0 1 54e WPA2 CCMP PSK =ANNA=
F0:7D:68:15:A8: 0 2 0 0 6 54e WEP WEP Aviya
00:25:86:CD:EC: 0 16 0 0 11 54e. WPA2 CCMP PSK HP-ELLA_Netwo
BSSID STATION PWR Rate Lost Packets Probes
C4:3D:C7:6E:D7: F8:D1:11:0B:5C: 0 0e- 0 0 424
00:25:9C:34:D4: 00:21:91:1D:AC: 0 0e- 0e 11 64
00:25:9C:34:D4: F4:EC:38:85:D2: 0 0e- 0e 0 67 GeverNet
00:15:67:CA:CC: 00:15:67:C8:60: 0 0 - 0 0 923

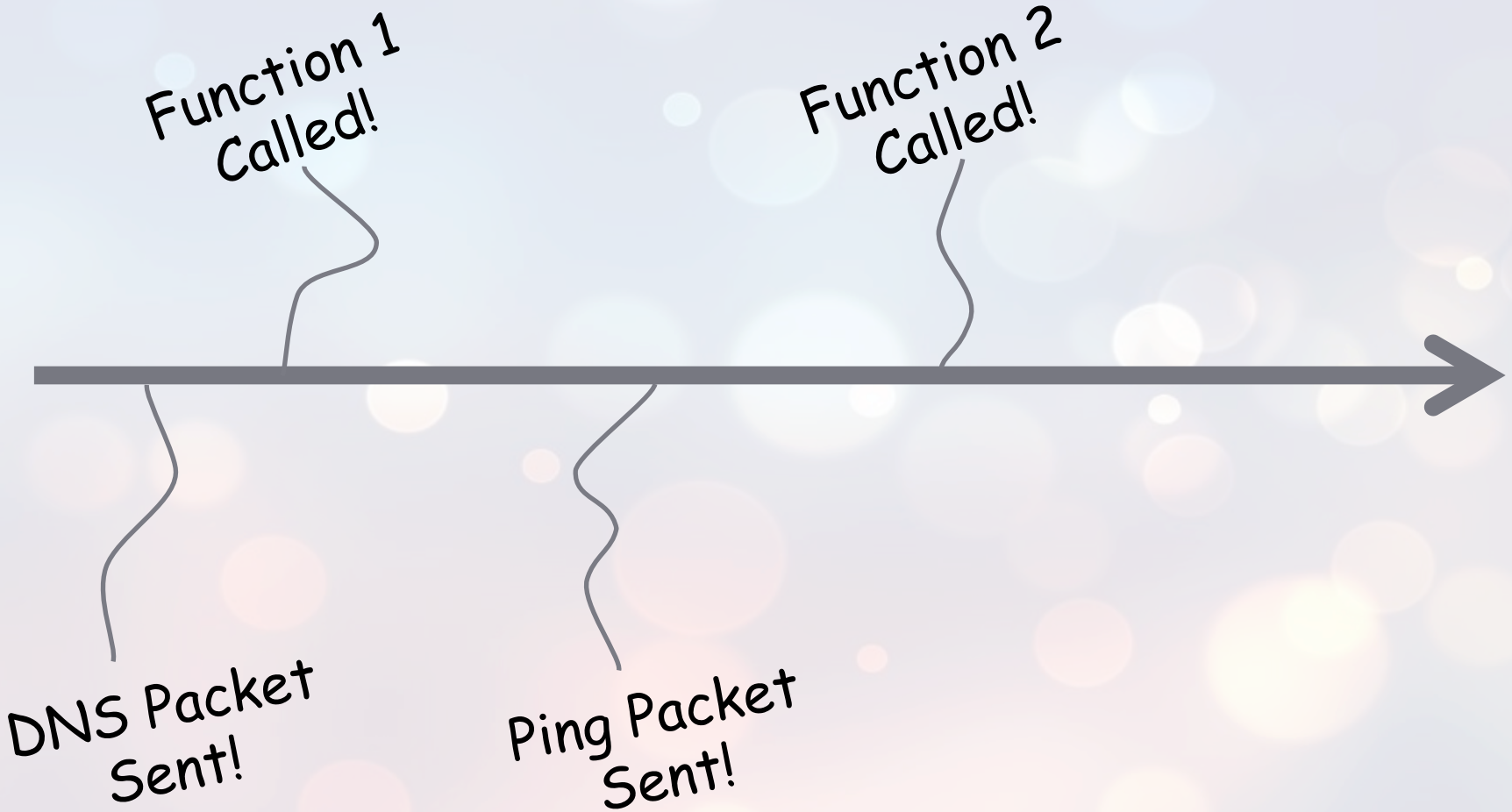
tasks

- [x] Implement monitor mode ourselves
- [x] Save time
  - [x] Research infrastructure

tasks

- [x] Implement monitor mode ourselves
- [ ] Packet injection support ←
- [x] Save time
  - [x] Research infrastructure ←

Tracing

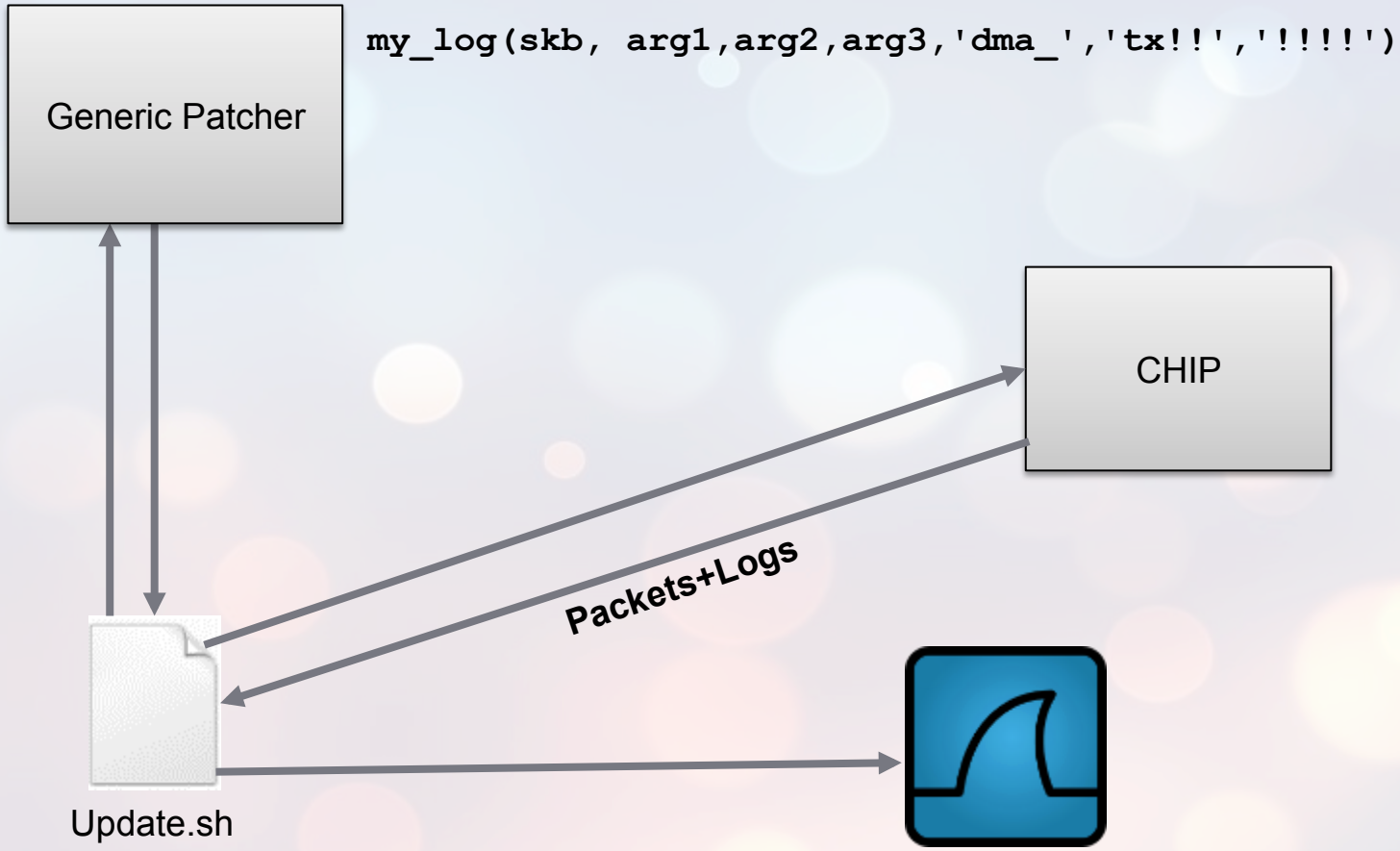

Function 1 Called!
Function 2 Called!
DNS Packet Sent!
Ping Packet Sent!

Demo !!!

Tracing

- Stack trace
- Real time packet buffers (sk_buff)
- Function arguments
- Return values

Upgrading our tracing system



tasks

- [x] Implement monitor mode ourselves
- [x] Packet injection support
- [x] Save time
  - [x] Research infrastructure

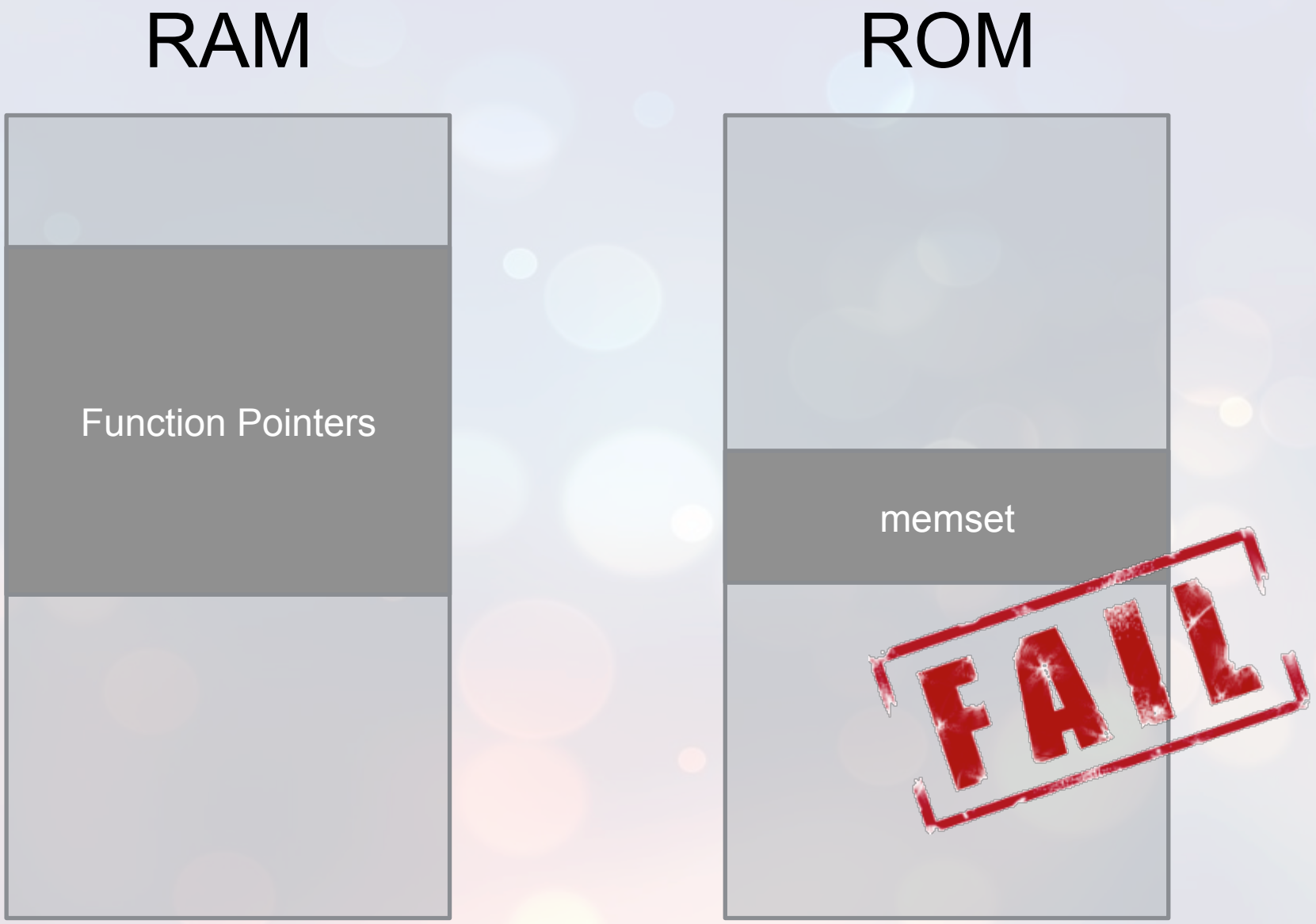
Nexus One
BCM4329

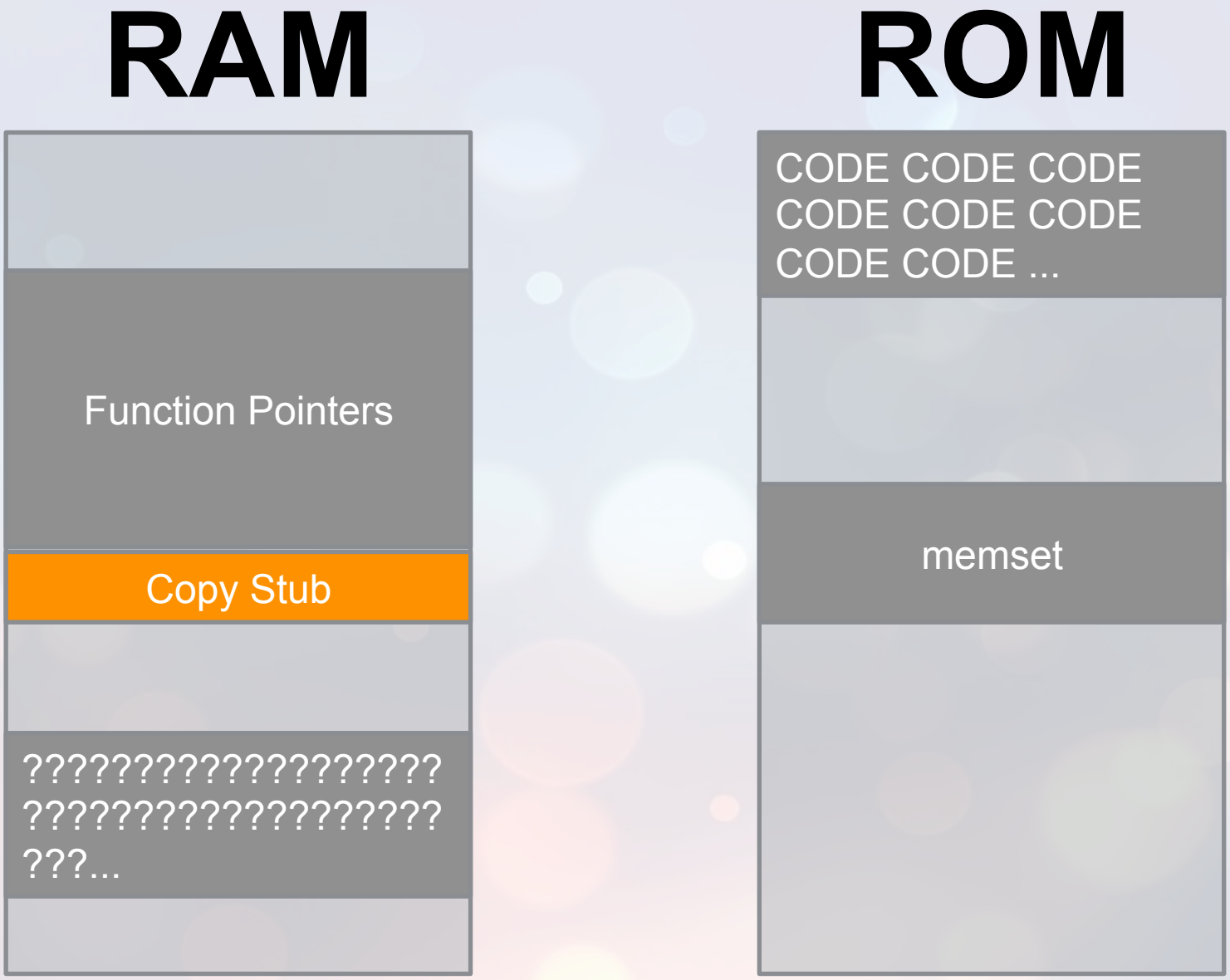
SGS II
BCM4330B1

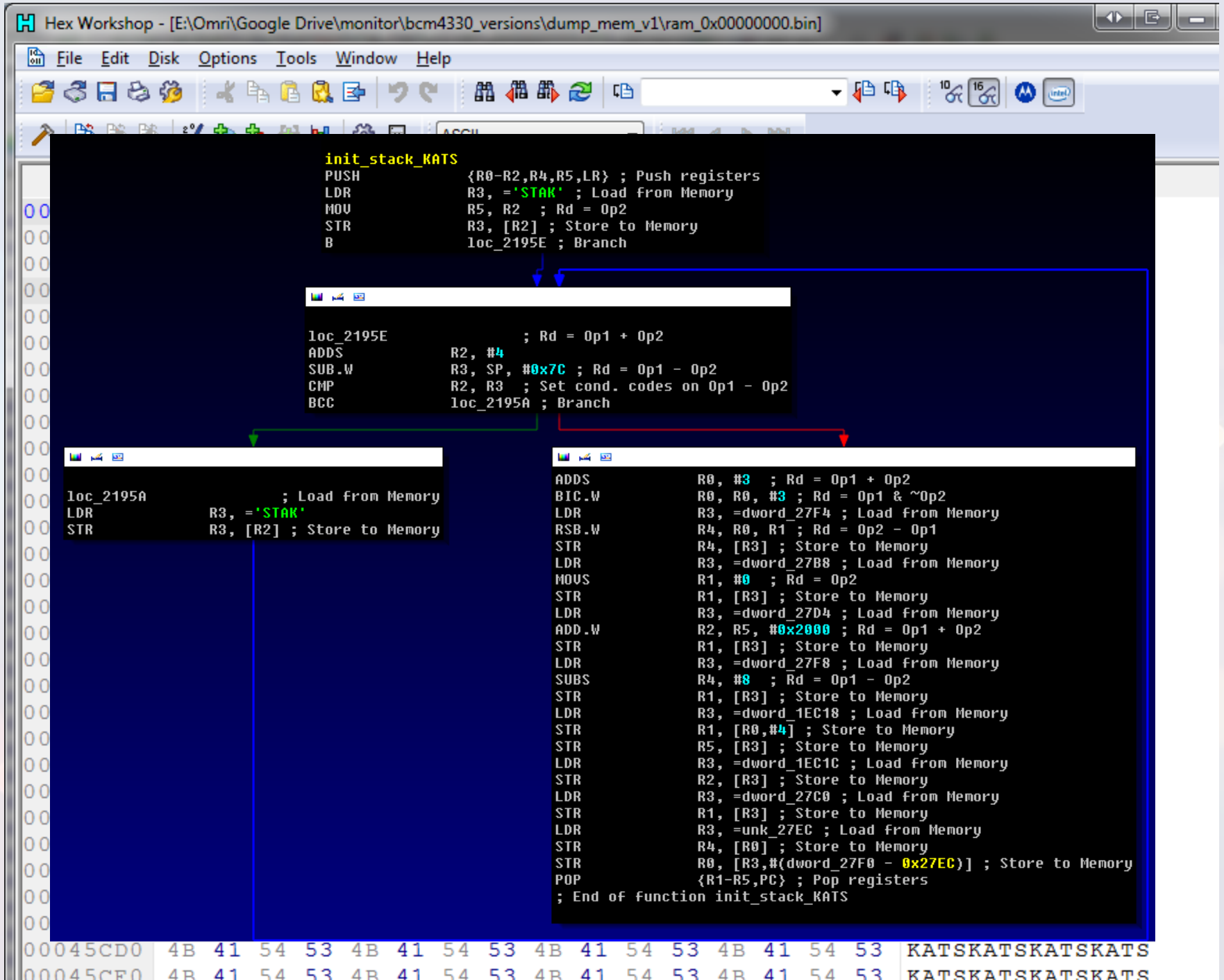
Nexus 7
BCM4330B2

SGS III
BCM4334

SGS IV
BCM4335


RAM
ROM
Function Pointers
memset
FAIL

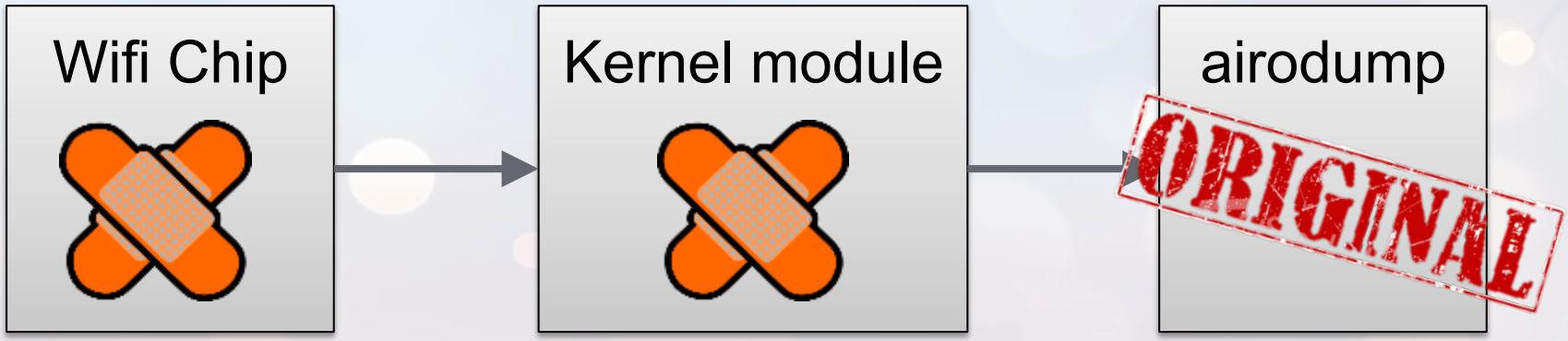

RAM
ROM
Function Pointers
Copy Stub
??????????????????
??????????????????
???...
CODE CODE CODE
CODE CODE CODE
CODE CODE ...
memset


Hex Workshop - [E:\Omri\Google Drive\monitor\bcm4330_versions\dump_mem_v1\ram_0x00000000.bin]
File Edit Disk Options Tools Window Help
init_stack_KATS
PUSH {R0-R2,R4,R5,LR} ; Push registers
LDR R3, ='STAK' ; Load from Memory
MOV R5, R2 ; Rd = Op2
STR R3, [R2] ; Store to Memory
B loc_2195E ; Branch
loc_2195E ; Rd = Op1 + Op2
ADDS R2, #4
SUB.W R3, SP, #0x7C ; Rd = Op1 - Op2
CMP R2, R3 ; Set cond. codes on Op1 - Op2
BCC loc_2195A ; Branch
loc_2195A ; Load from Memory
LDR R3, ='STAK'
STR R3, [R2] ; Store to Memory
ADDS R0, #3 ; Rd = Op1 + Op2
BIC.W R0, R0, #3 ; Rd = Op1 & ~Op2
LDR R3, =dword_27F4 ; Load from Memory
RSB.W R4, R0, R1 ; Rd = Op2 - Op1
STR R4, [R3] ; Store to Memory
LDR R3, =dword_27B8 ; Load from Memory
MOVS R1, #0 ; Rd = Op2
STR R1, [R3] ; Store to Memory
LDR R3, =dword_27D4 ; Load from Memory
ADD.W R2, R5, #0x2000 ; Rd = Op1 + Op2
STR R1, [R3] ; Store to Memory
LDR R3, =dword_27F8 ; Load from Memory
SUBS R4, #8 ; Rd = Op1 - Op2
STR R1, [R3] ; Store to Memory
LDR R3, =dword_1EC18 ; Load from Memory
STR R1, [R0,#4] ; Store to Memory
STR R5, [R3] ; Store to Memory
LDR R3, =dword_1EC1C ; Load from Memory
STR R2, [R3] ; Store to Memory
LDR R3, =dword_27C0 ; Load from Memory
STR R1, [R3] ; Store to Memory
LDR R3, =unk_27EC ; Load from Memory
STR R4, [R0] ; Store to Memory
STR R0, [R3,#(dword_27F0 - 0x27EC)] ; Store to Memory
POP {R1-R5,PC} ; Pop registers
; End of function init_stack_KATS
00045CD0 4B 41 54 53 4B 41 54 53 4B 41 54 53 4B 41 54 53 KATSKATSKATSKATS
00045CE0 4B 41 54 53 4B 41 54 53 4B 41 54 53 4B 41 54 53 KATSKATSKATSKATS

tasks

- [x] Implement monitor mode ourselves
- [x] Packet injection support
- [x] Save time
  - [x] Research infrastructure
- [x] Support other chipsets

Current Solution



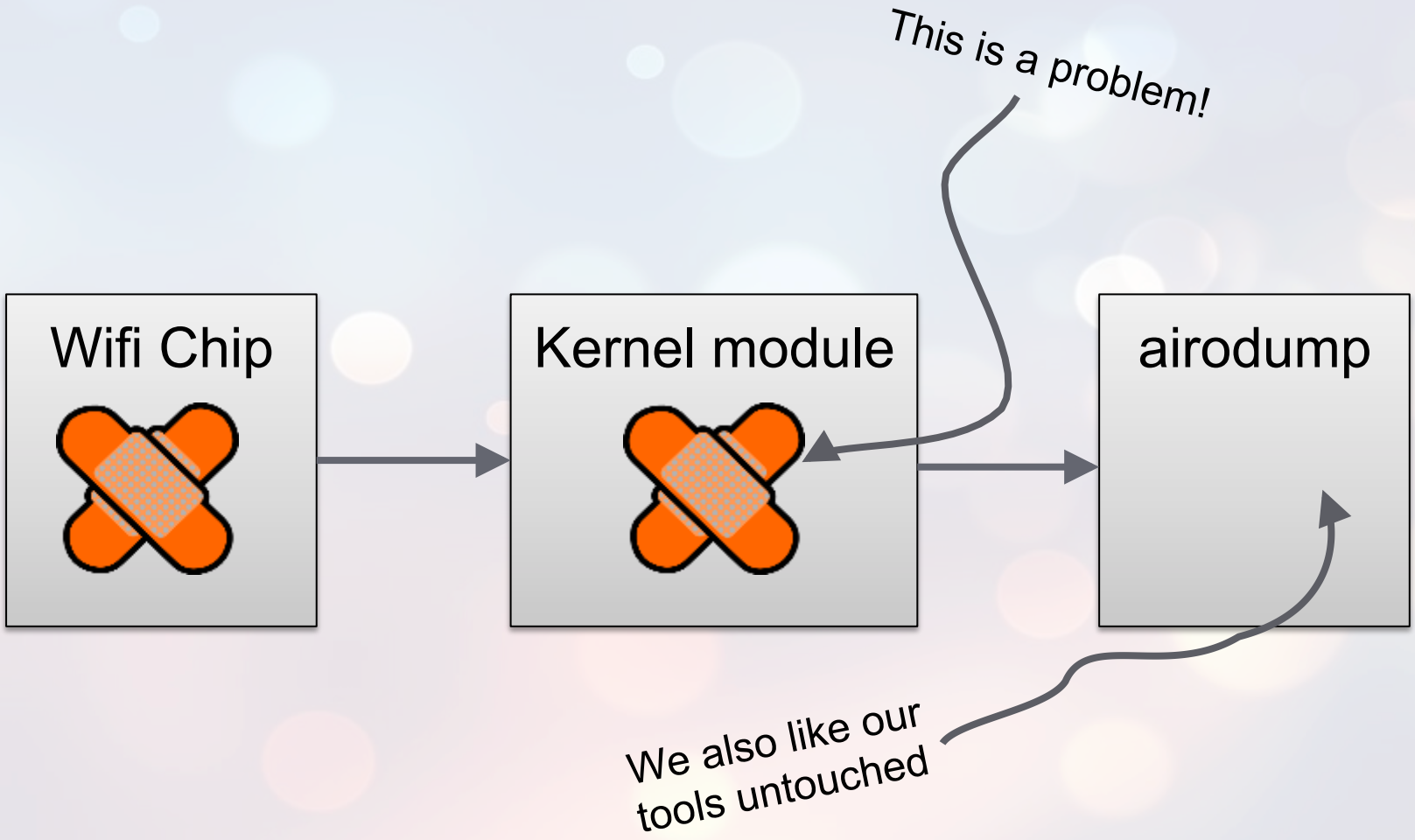
Compiling the kernel takes time...

We support 3 chips

~100 android devices

$$X \frac{100 \text{ devices}}{\#\text{ of kernels}} \propto \text{time}$$

Staying in userland



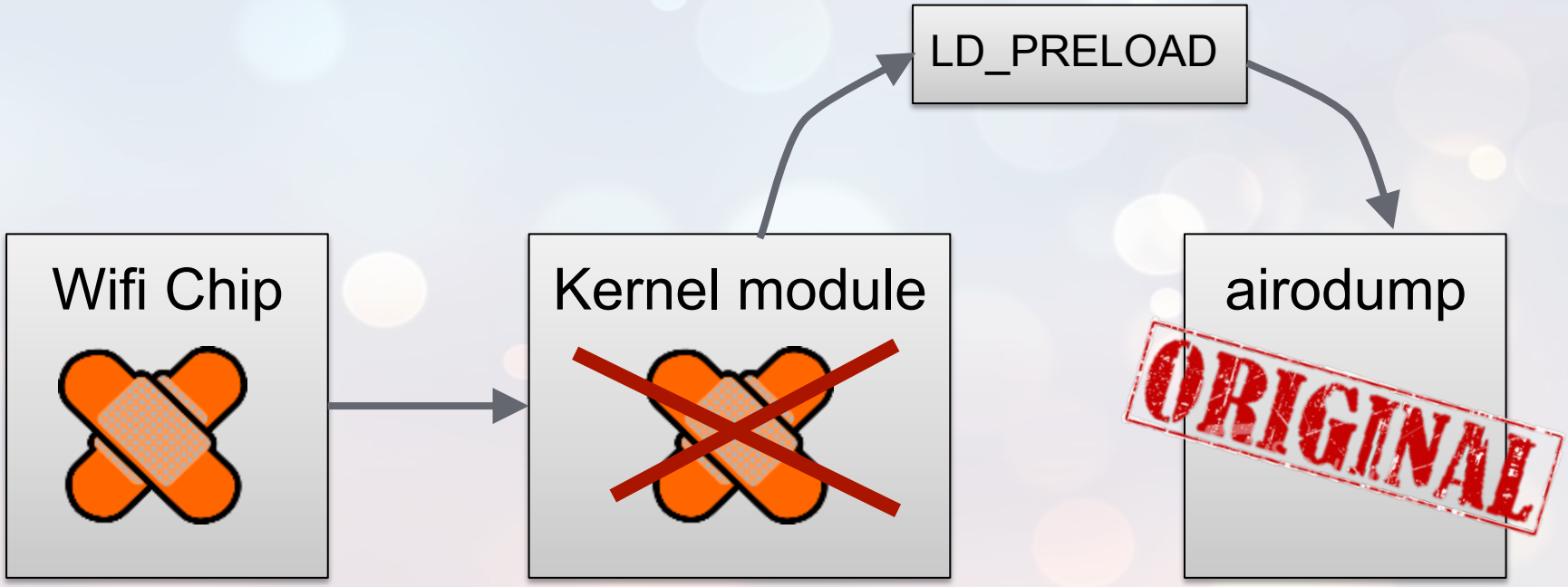
LD_PRELOAD

- Environment variable
- Shared object
- Loaded first
- Ideal for hooking

```
# LD_PRELOAD=`pwd`/libmytest.so ping 8.8.8.8
HELP!!! I'm trapped inside a foreign binary!!
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=43 time=75.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=43 time=75.5 ms
        .
        .
        .
```

```
int ioctl(int fd, int request, void *data)
    {
        ...
        switch (request)
        {
            case SIOCGIFHWADDR:
                debug_print("Handling HW Addr\n");
                struct ifreq* p_ifr;
                p_ifr = (struct ifreq*)data;
                p_ifr->ifr_hwaddr.sa_family = ARPHRD_IEEE80211_RADIOTAP;
                goto done;
            case SIOCGIWMODE:
                struct iwreq* p_wrq;
                p_wrq = (struct iwreq*)data;
                p_wrq->u.mode = IW_MODE_MONITOR;
                goto done;
        }
        ...
    }
```

Staying in userland



tasks

- [x] Implement monitor mode ourselves
- [x] Packet injection support
- [x] Save time
  - [x] Research infrastructure
- [x] Support other chipsets
- [x] Idiot proof

Demo !!!

Nexus One
BCM4329

SGS II
BCM4330B1

Nexus 7
BCM4330B2

SGS III
BCM4334

SGS IV
BCM4335

Lessons Learned

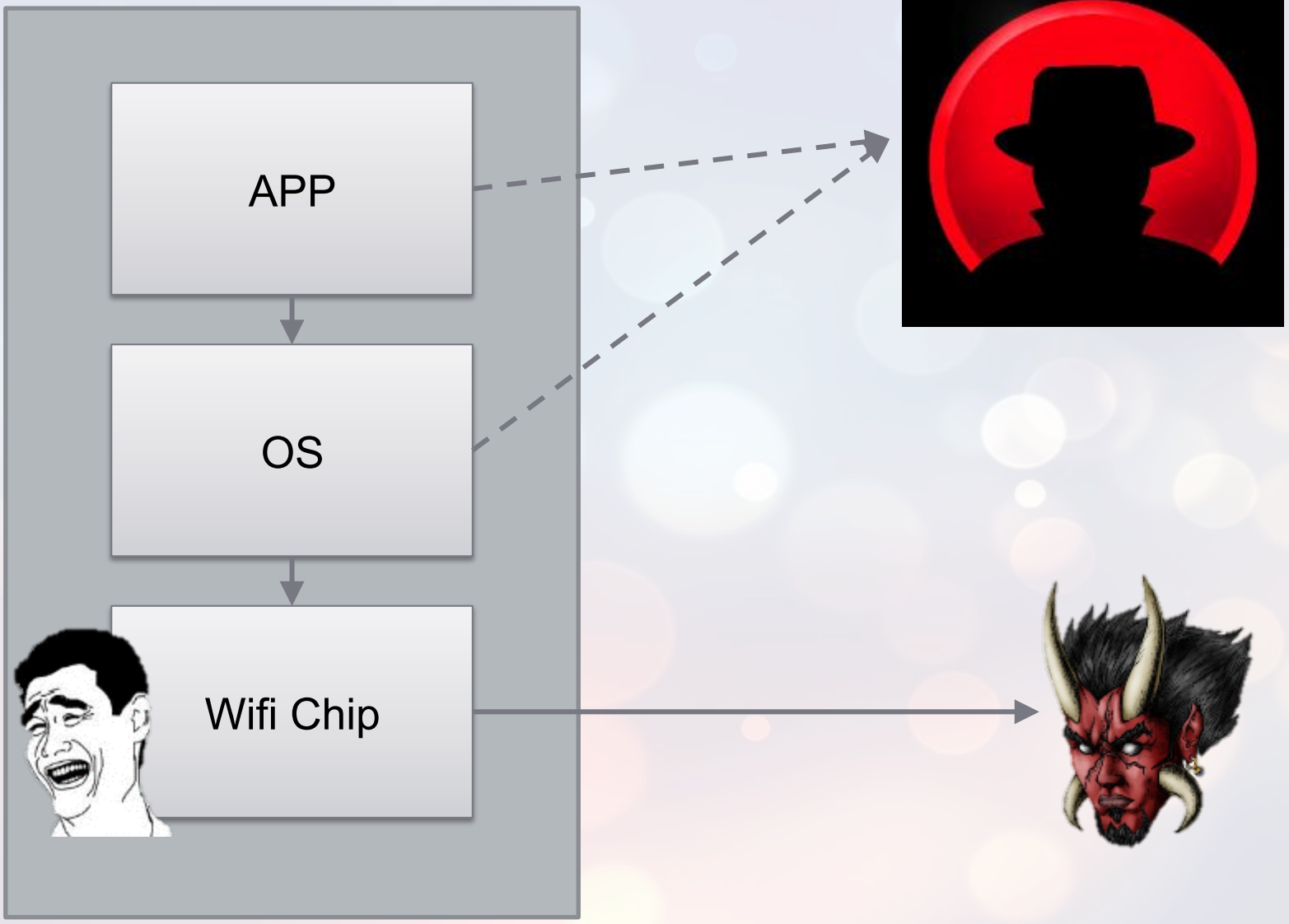
- Research Infrastructure
- BFS vs. DFS
- Google Decompiler

Related work

- monmob - Core Security
  - Andrés Blanco and Matías Eissler
  - Monitor Mode on iPhone (also Broadcom chips)

Something Completely Different

Phone



Demo !!!

Where can I get it?

- generic_patcher.py
  - https://github.com/shoote/generic_patcher
- Android App (currently for bcm4329/30)
  - https://bcmon.googlecode.com/files/bcmon.apk
- More updates:
  - bcmon.blogspot.com
  - contact.bcmon@gmail.com


Questions