black hat USA 2021

August 4-5, 2021

BRIEFINGS

QUALCOMM Wi-Fi： INFINITY WAR

Haikuo Xie of Singular Security Lab

About Me

Haikuo Xie

Senior security researcher at Singular Security Lab

focus on the field of protocol and short-distance communication

Agenda

Qualcomm Wi-Fi Security Actuality

Qualcomm Wi-Fi Architecture

Wi-Fi Driver Security Research

Wi-Fi Firmware Security Research

Conclusion

Qualcomm Wi-Fi Security Actuality

History

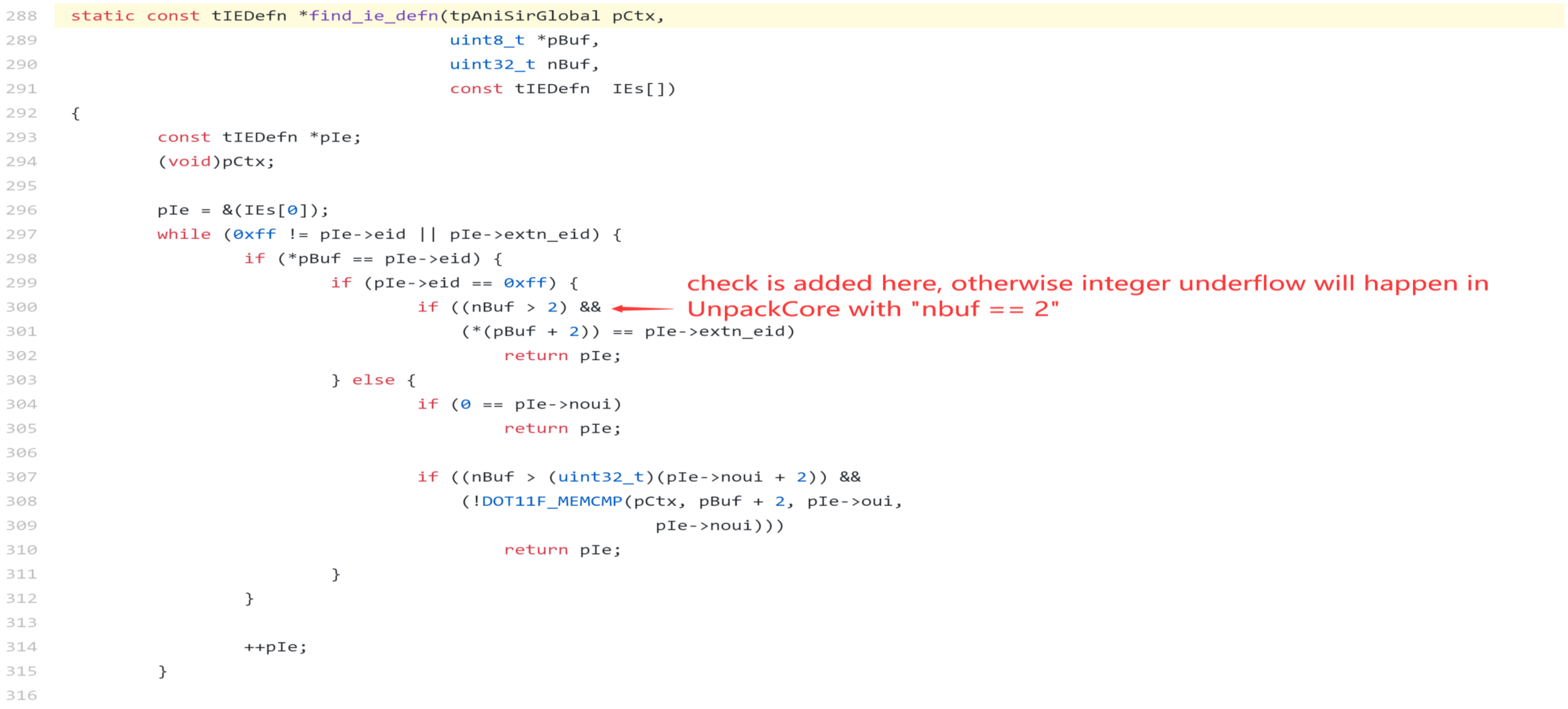
1.2017 Over The Air: Exploiting Broadcom’ s Wi-Fi Stack by Gal Beniamini

2.Blackhat USA 2019 Exploiting Qualcomm WLAN And Modem Over-The-Air

3.2020 An iOS zero-click radio proximity exploit odyssey by Ian Beer

Qualcomm Wi-Fi Security Actuality

Relatively good security Actuality



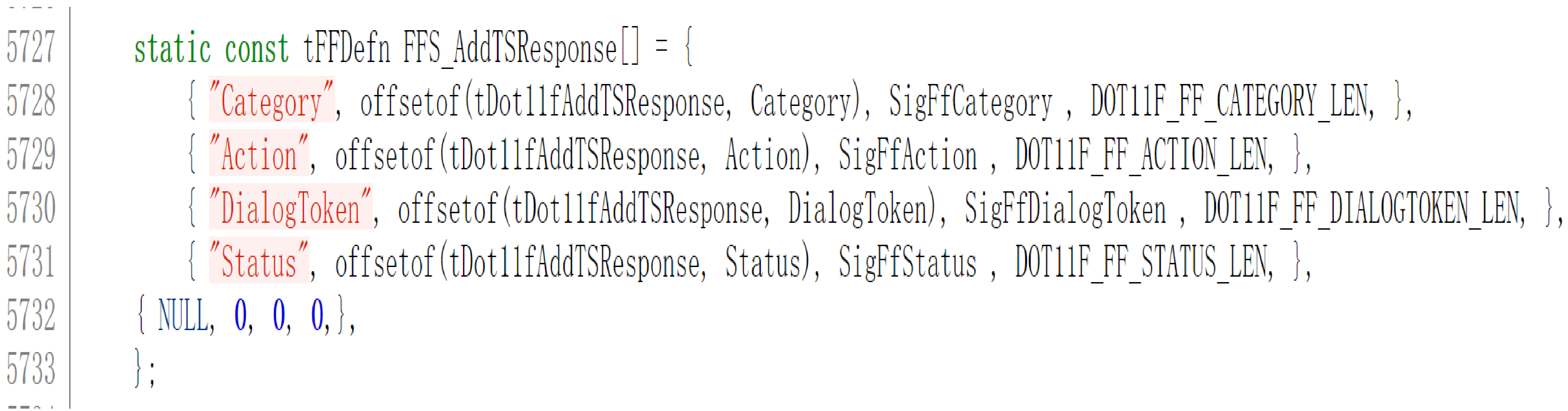
```
static const tIEDefn *find_ie_defn(tpAniSirGlobal pCtx,
                                   uint8_t *pBuf,
                                   uint32_t nBuf,
                                   const tIEDefn  IEs[])
{
        const tIEDefn *pIe;
        (void)pCtx;

        pIe = &(IEs[0]);
        while (0xff != pIe->eid || pIe->extn_eid) {
                if (*pBuf == pIe->eid) {
                        if (pIe->eid == 0xff) {
                                if ((nBuf > 2) &&
                                    (*(pBuf + 2)) == pIe->extn_eid)
                                        return pIe;
                        } else {
                                if (0 == pIe->noui)
                                        return pIe;

                                if ((nBuf > (uint32_t)(pIe->noui + 2)) &&
                                    (!DOT11F_MEMCMP(pCtx, pBuf + 2, pIe->oui,
                                                  pIe->noui)))
                                        return pIe;
                        }
                }

                ++pIe;
        }

```

check is added here, otherwise integer underflow will happen in UnpackCore with "nbuf == 2"

https://github.com/MiCode/vendor_qcom_opensource_wlan/blob/cas-q-oss/qcacld-3.0/core/mac/src/sys/legacy/src/utils/src/dot11f.c#L288

Qualcomm Wi-Fi Security Actuality

Relatively good security Actuality



```
static const tFFDefn FFS_AddTSResponse[] = {
    { "Category", offsetof(tDot11fAddTSResponse, Category), SigFfCategory , DOT11F_FF_CATEGORY_LEN, },
    { "Action", offsetof(tDot11fAddTSResponse, Action), SigFfAction , DOT11F_FF_ACTION_LEN, },
    { "DialogToken", offsetof(tDot11fAddTSResponse, DialogToken), SigFfDialogToken , DOT11F_FF_DIALOGTOKEN_LEN, },
    { "Status", offsetof(tDot11fAddTSResponse, Status), SigFfStatus , DOT11F_FF_STATUS_LEN, },
{ NULL, 0, 0, 0,},
};
```

https://source.codeaurora.org/quic/la/platform/vendor/qcom-opensource/wlan/qcacld-2.0/tree/CORE/SYS/legacy/src/utils/src/dot11f.c

Qualcomm Wi-Fi Security Actuality

There are still great security risks

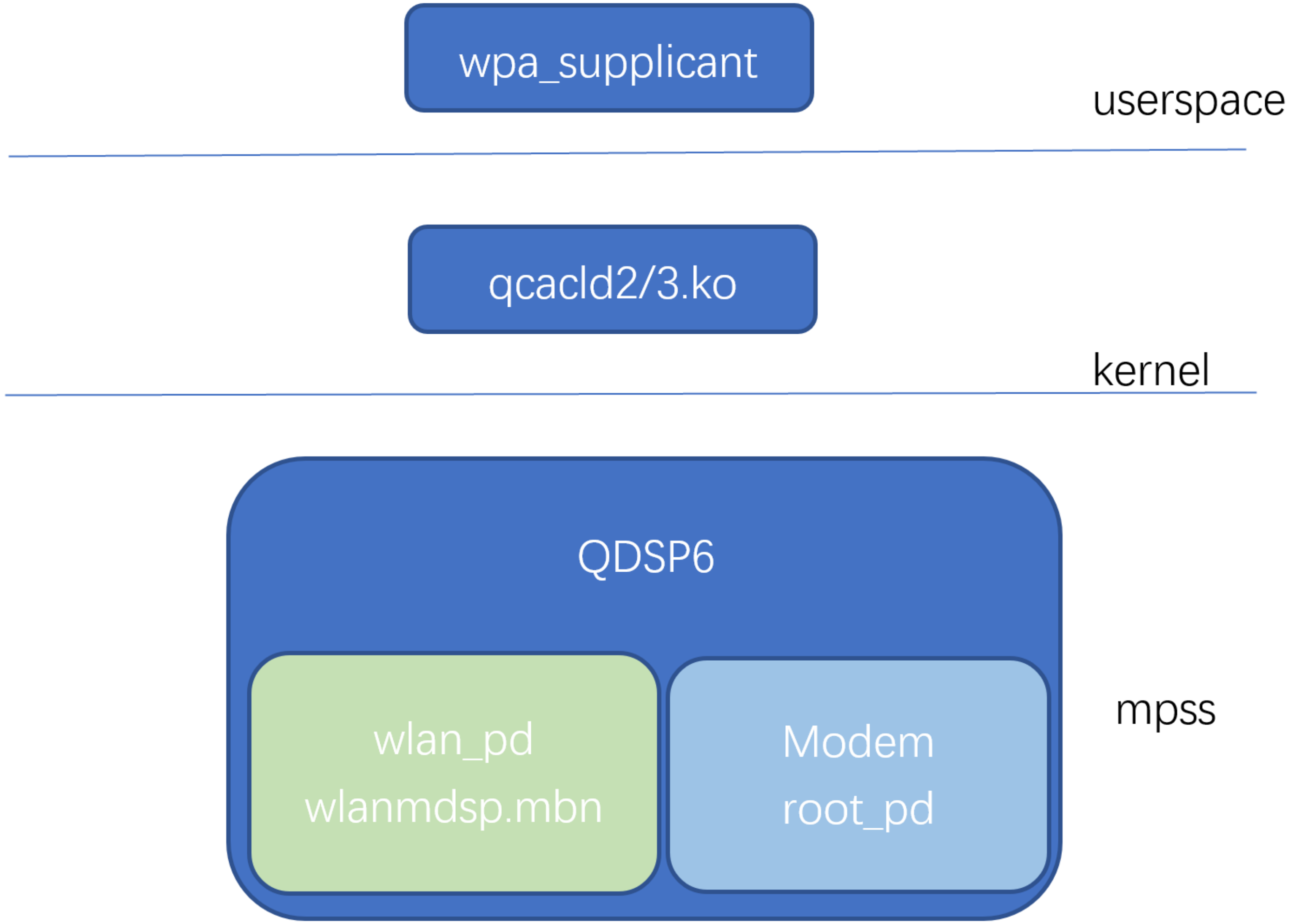
- 0-click
- adjacent network
- non-privileged

Qualcomm Wi-Fi Architecture

Snapdragon 845

Integrated Snapdragon X20 LTE modem to support latest air interfaces including 5x CA up to 1.2 Gbps

Wi-Fi 802.11ac and Bluetooth 5.0 with the Qualcomm® WCN3990 device

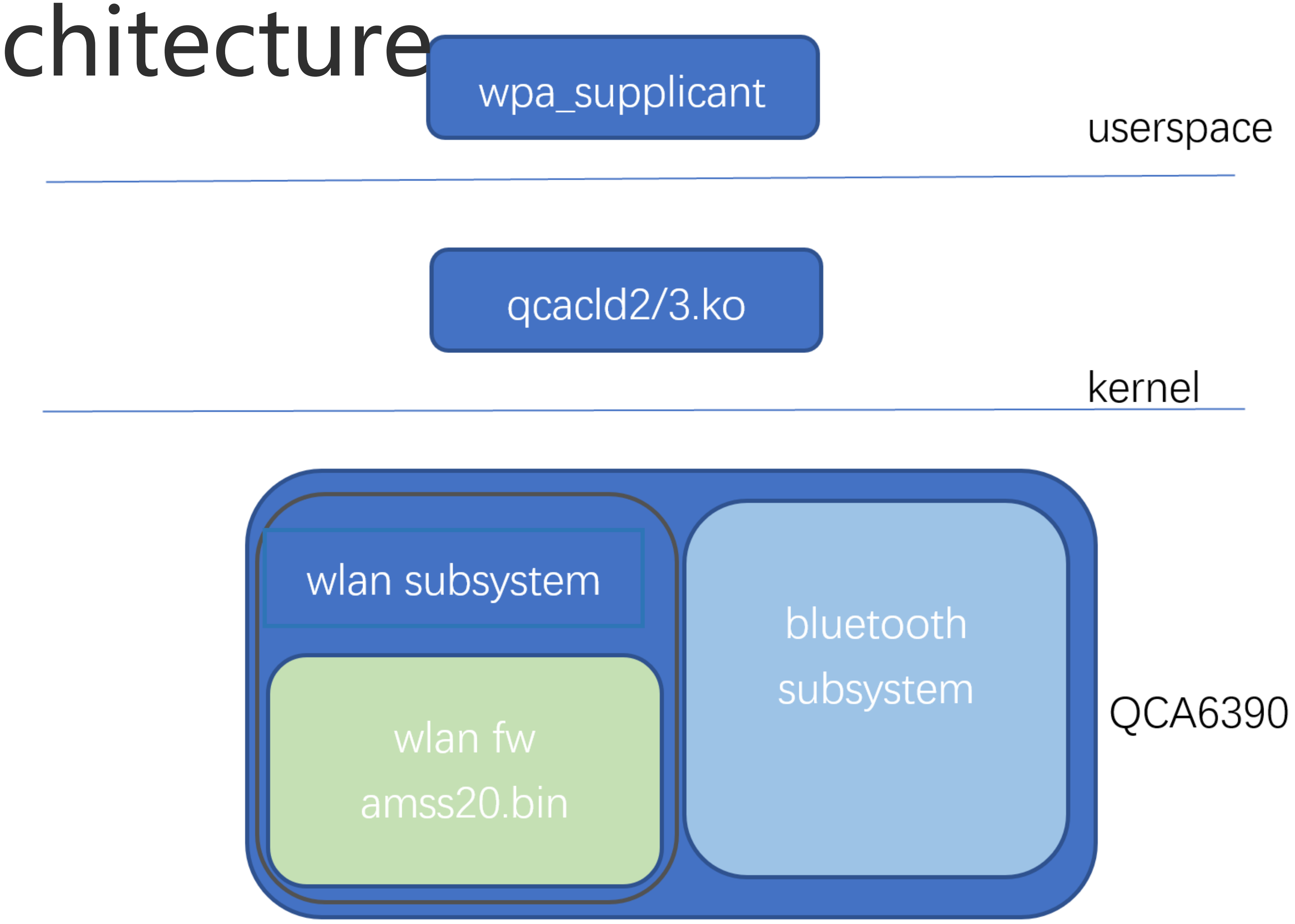


Qualcomm Wi-Fi Architecture

Snapdragon 865

Snapdragon X55 5G Modem-RF System

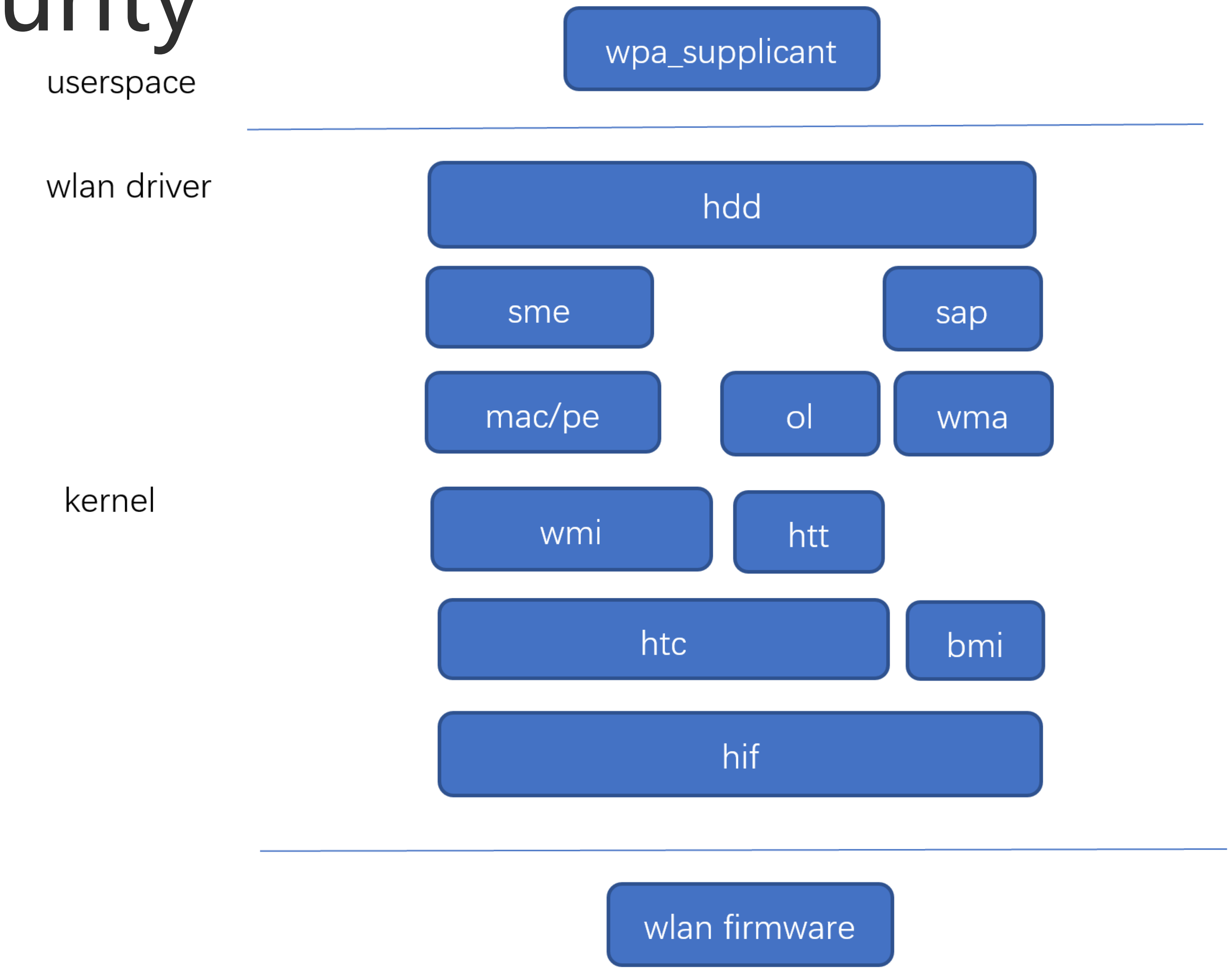
Qualcomm® FastConnect 6800 Subsystem Wi-Fi & Bluetooth with the Qualcomm qca6390 device



Wi-Fi Driver Security

Driver Architecture

This part belongs to the open source software of Qualcomm



userspace

wpa_supplicant

wlan driver

hdd

sme

sap

mac/pe

ol

wma

wmi

htt

htc

bmi

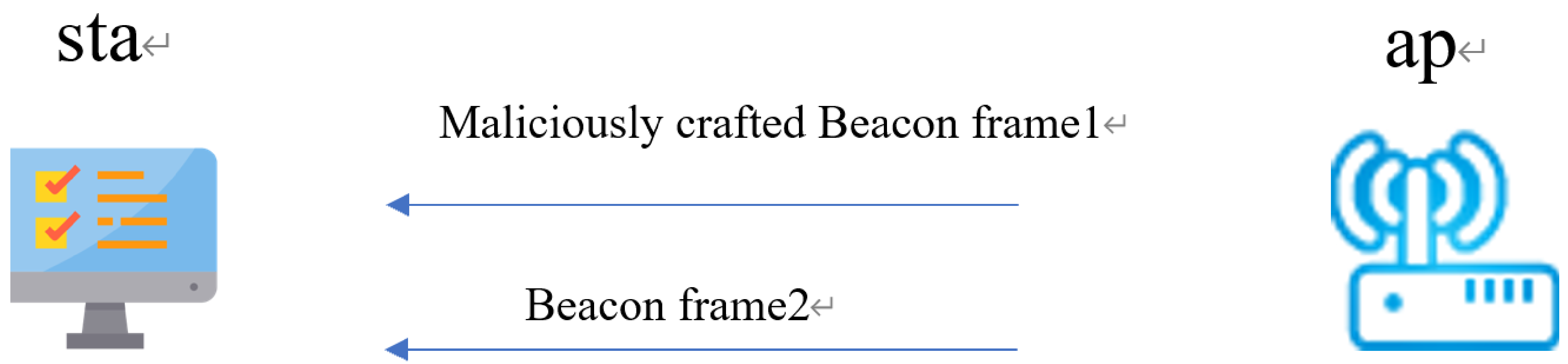
hif

kernel

wlan firmware

Wi-Fi Driver Security

CVE-2020-11225



sta↵

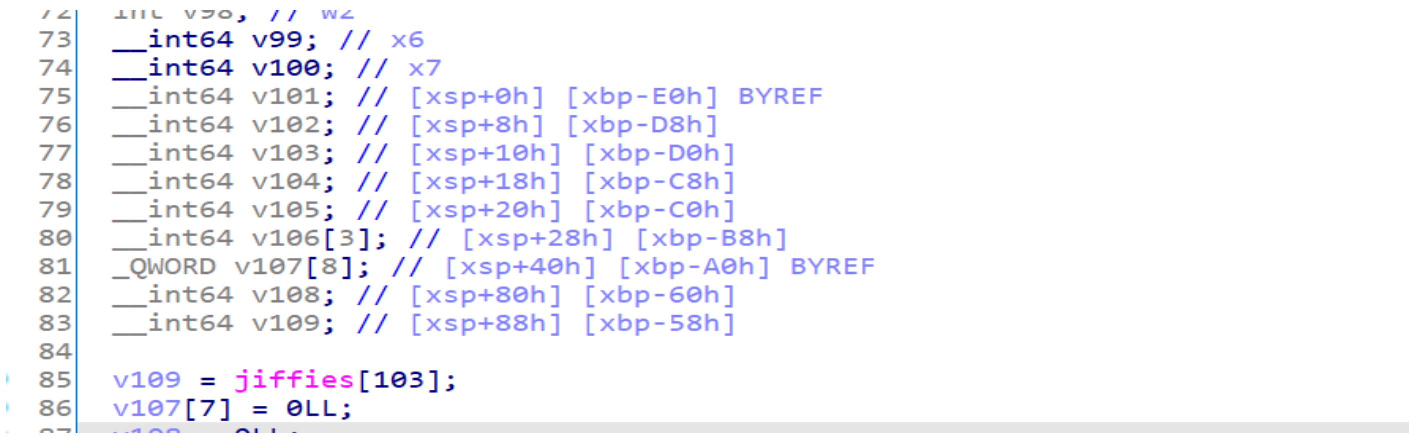
ap↵

Maliciously crafted Beacon frame1↵

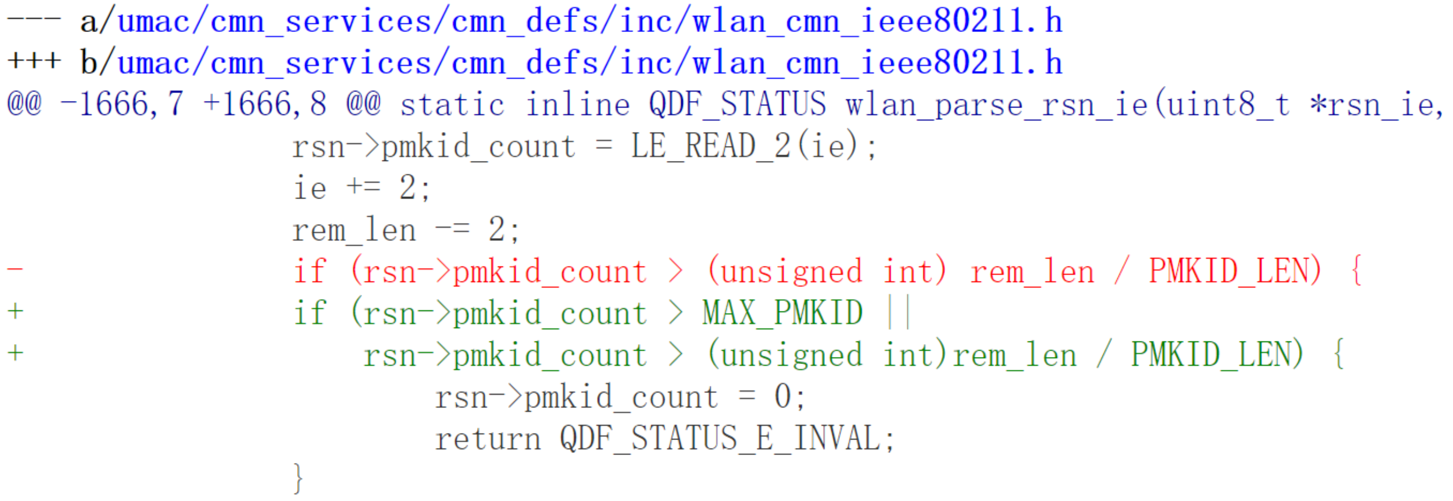
Beacon frame2↵

Wi-Fi Driver Security

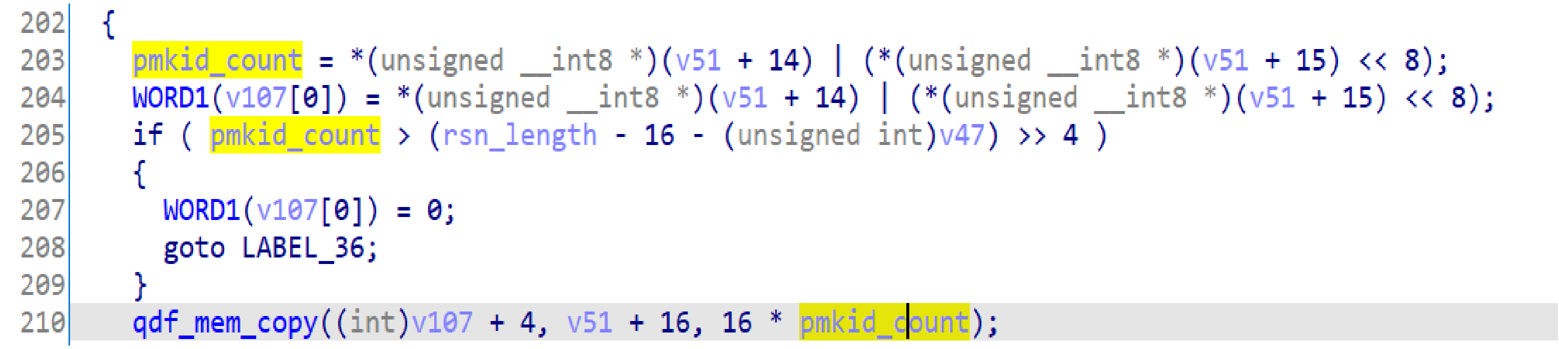
CVE-2020-11225



```
  int v98; // w2
  __int64 v99; // x6
  __int64 v100; // x7
  __int64 v101; // [xsp+0h] [xbp-E0h] BYREF
  __int64 v102; // [xsp+8h] [xbp-D8h]
  __int64 v103; // [xsp+10h] [xbp-D0h]
  __int64 v104; // [xsp+18h] [xbp-C8h]
  __int64 v105; // [xsp+20h] [xbp-C0h]
  __int64 v106[3]; // [xsp+28h] [xbp-B8h]
  _QWORD v107[8]; // [xsp+40h] [xbp-A0h] BYREF
  __int64 v108; // [xsp+80h] [xbp-60h]
  __int64 v109; // [xsp+88h] [xbp-58h]

  v109 = jiffies[103];
  v107[7] = 0LL;
  v108 = 0LL;
```



```
--- a/umac/cmn_services/cmn_defs/inc/wlan_cmn_ieee80211.h
+++ b/umac/cmn_services/cmn_defs/inc/wlan_cmn_ieee80211.h
@@ -1666,7 +1666,8 @@ static inline QDF_STATUS wlan_parse_rsn_ie(uint8_t *rsn_ie,
                rsn->pmkid_count = LE_READ_2(ie);
                ie += 2;
                rem_len -= 2;
-               if (rsn->pmkid_count > (unsigned int) rem_len / PMKID_LEN) {
+               if (rsn->pmkid_count > MAX_PMKID ||
+                   rsn->pmkid_count > (unsigned int)rem_len / PMKID_LEN) {
                        rsn->pmkid_count = 0;
                        return QDF_STATUS_E_INVAL;
                }
```



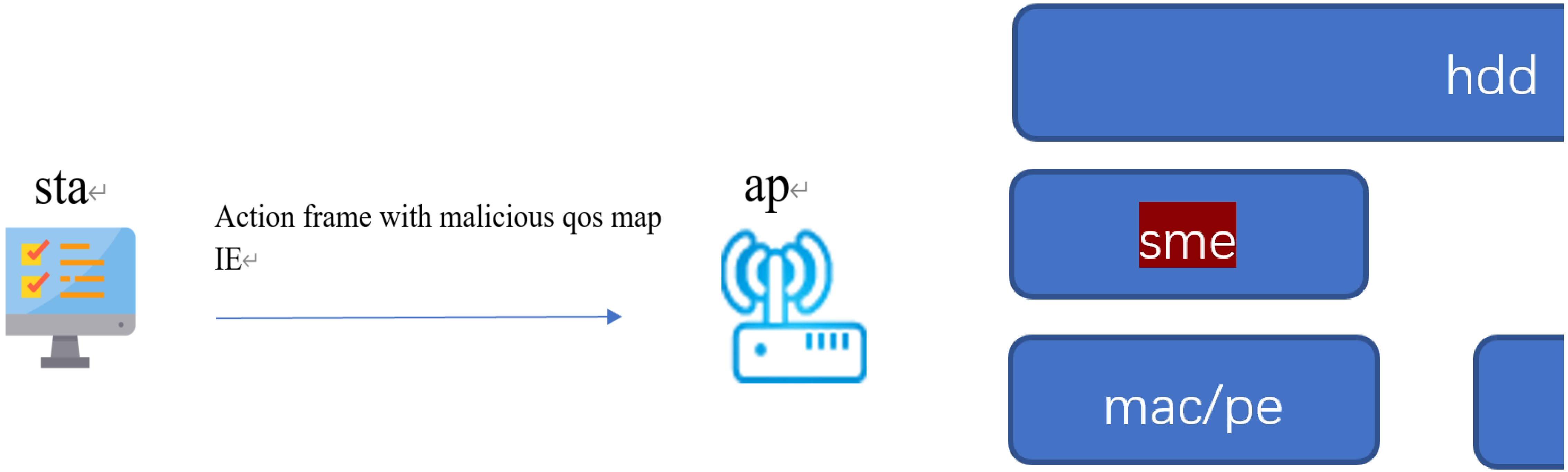
```
  {
    pmkid_count = *(unsigned __int8 *)(v51 + 14) | (*(unsigned __int8 *)(v51 + 15) << 8);
    WORD1(v107[0]) = *(unsigned __int8 *)(v51 + 14) | (*(unsigned __int8 *)(v51 + 15) << 8);
    if ( pmkid_count > (rsn_length - 16 - (unsigned int)v47) >> 4 )
    {
      WORD1(v107[0]) = 0;
      goto LABEL_36;
    }
    qdf_mem_copy((int)v107 + 4, v51 + 16, 16 * pmkid_count);
```

scm_is_rsn_security

https://source.codeaurora.org/quic/qsdk/platform/vendor/qcom-opensource/wlan/qca-Wi-Fi-host-cmn/commit/?id=fe1e85068c57d8c4e4557ed6b265ac6b9694c3a1

Wi-Fi Driver Security

CVE-2020-3698



sta↵

Action frame with malicious qos map IE↵

ap↵

hdd

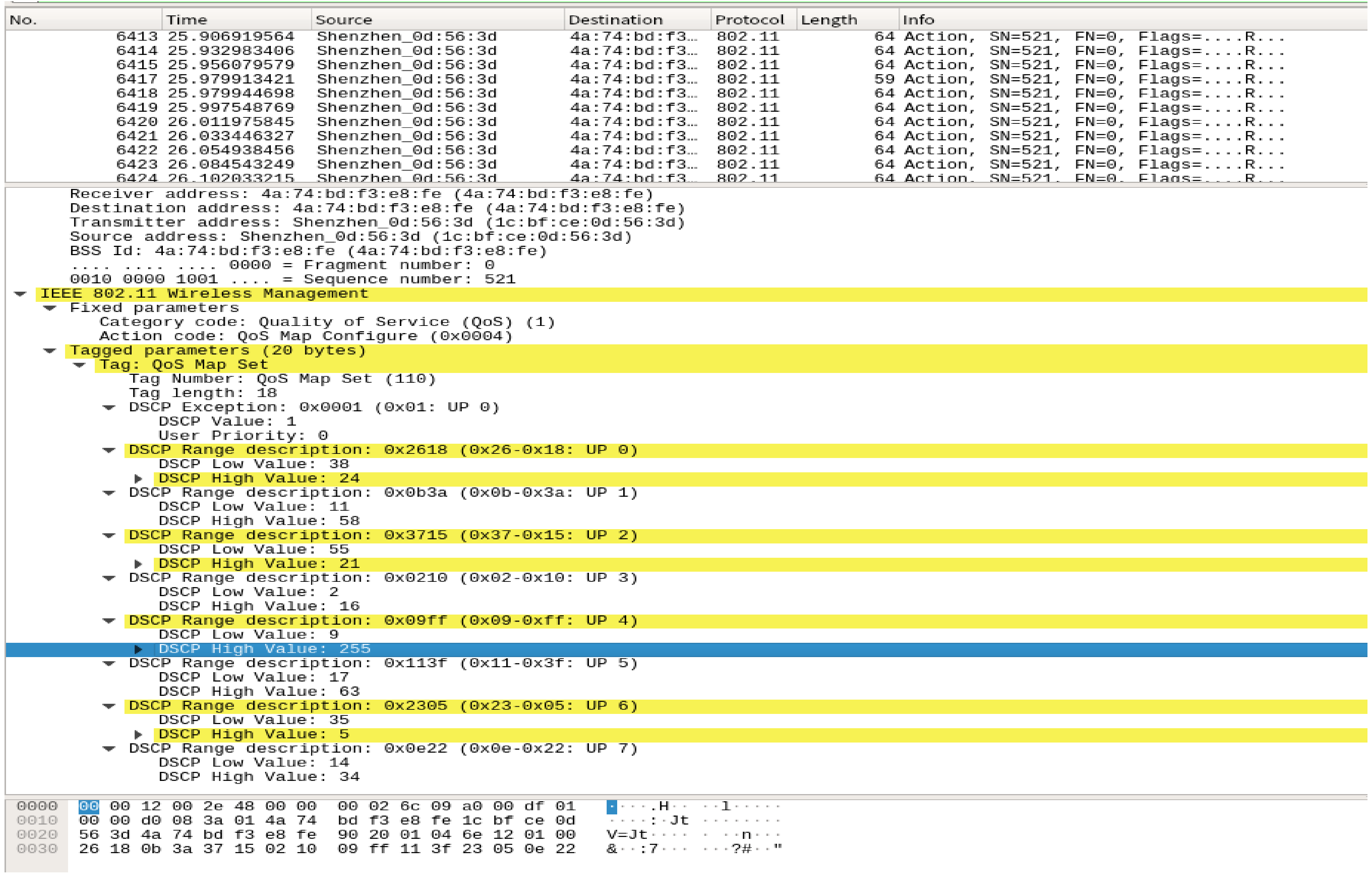
sme

mac/pe

Wi-Fi Driver Security

CVE-2020-3698

POC



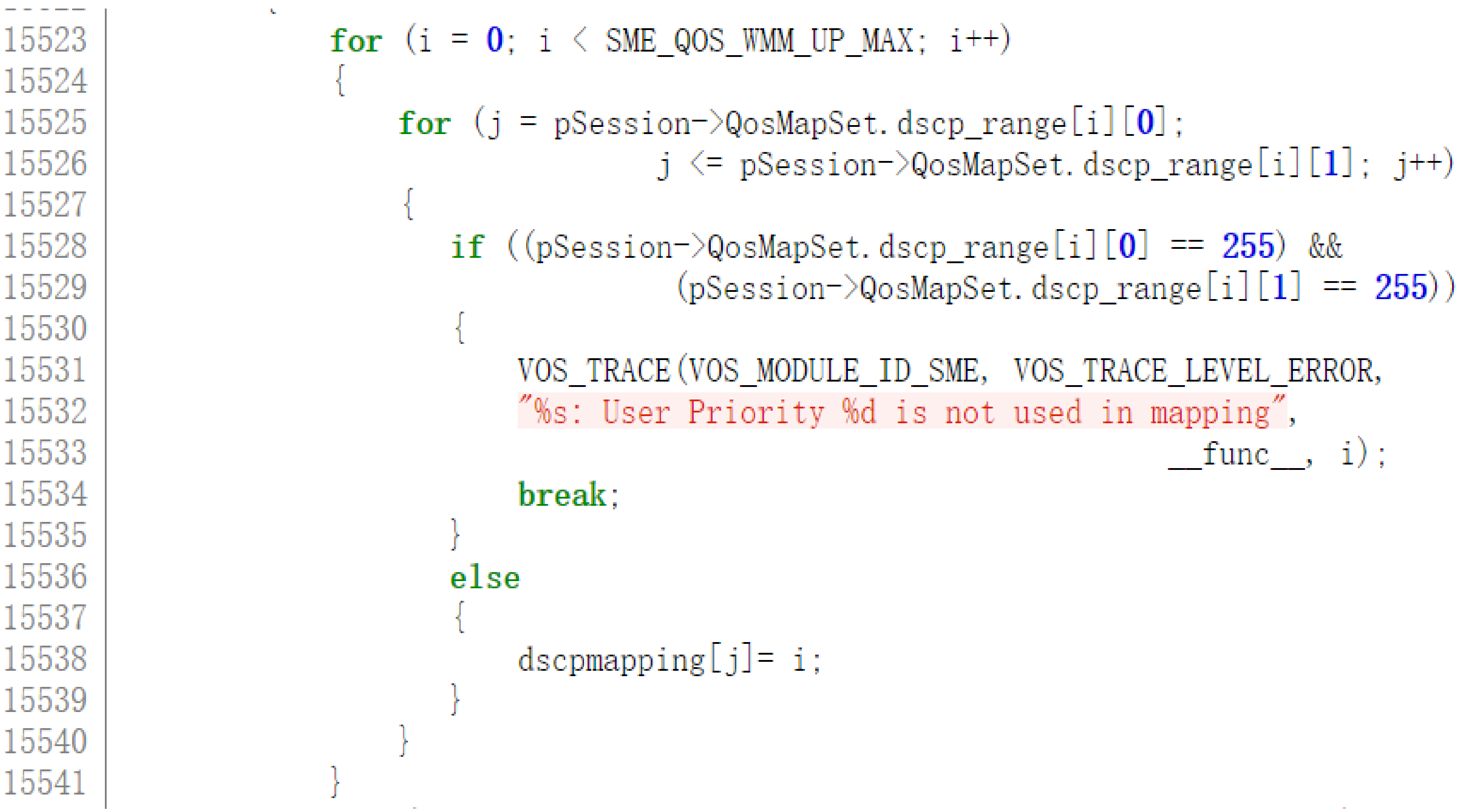
```
No.         Time                Source                Destination        Protocol  Length  Info
      6413 25.906919564   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6414 25.932983406   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6415 25.956079579   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6417 25.979913421   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          59 Action, SN=521, FN=0, Flags=....R...
      6418 25.979944698   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6419 25.997548769   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6420 26.011975845   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6421 26.033446327   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6422 26.054938456   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6423 26.084543249   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
      6424 26.102033215   Shenzhen_0d:56:3d     4a:74:bd:f3…       802.11          64 Action, SN=521, FN=0, Flags=....R...
```

```
    Receiver address: 4a:74:bd:f3:e8:fe (4a:74:bd:f3:e8:fe)
    Destination address: 4a:74:bd:f3:e8:fe (4a:74:bd:f3:e8:fe)
    Transmitter address: Shenzhen_0d:56:3d (1c:bf:ce:0d:56:3d)
    Source address: Shenzhen_0d:56:3d (1c:bf:ce:0d:56:3d)
    BSS Id: 4a:74:bd:f3:e8:fe (4a:74:bd:f3:e8:fe)
    .... .... .... 0000 = Fragment number: 0
    0010 0000 1001 .... = Sequence number: 521
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters
      Category code: Quality of Service (QoS) (1)
      Action code: QoS Map Configure (0x0004)
  ▾ Tagged parameters (20 bytes)
    ▾ Tag: QoS Map Set
        Tag Number: QoS Map Set (110)
        Tag length: 18
      ▾ DSCP Exception: 0x0001 (0x01: UP 0)
          DSCP Value: 1
          User Priority: 0
      ▾ DSCP Range description: 0x2618 (0x26-0x18: UP 0)
          DSCP Low Value: 38
        ▸ DSCP High Value: 24
      ▾ DSCP Range description: 0x0b3a (0x0b-0x3a: UP 1)
          DSCP Low Value: 11
          DSCP High Value: 58
      ▾ DSCP Range description: 0x3715 (0x37-0x15: UP 2)
          DSCP Low Value: 55
        ▸ DSCP High Value: 21
      ▾ DSCP Range description: 0x0210 (0x02-0x10: UP 3)
          DSCP Low Value: 2
          DSCP High Value: 16
      ▾ DSCP Range description: 0x09ff (0x09-0xff: UP 4)
          DSCP Low Value: 9
        ▸ DSCP High Value: 255
      ▾ DSCP Range description: 0x113f (0x11-0x3f: UP 5)
          DSCP Low Value: 17
          DSCP High Value: 63
      ▾ DSCP Range description: 0x2305 (0x23-0x05: UP 6)
          DSCP Low Value: 35
        ▸ DSCP High Value: 5
      ▾ DSCP Range description: 0x0e22 (0x0e-0x22: UP 7)
          DSCP Low Value: 14
          DSCP High Value: 34
```

```
0000  00 00 12 00 2e 48 00 00  00 02 6c 09 a0 00 df 01   ·····.H·· ··l·····
0010  00 00 d0 08 3a 01 4a 74  bd f3 e8 fe 1c bf ce 0d   ·····:·Jt ········
0020  56 3d 4a 74 bd f3 e8 fe  90 20 01 04 6e 12 01 00   V=Jt····  · ··n···
0030  26 18 0b 3a 37 15 02 10  09 ff 11 3f 23 05 0e 22   &··:7···  ···?#··"
```

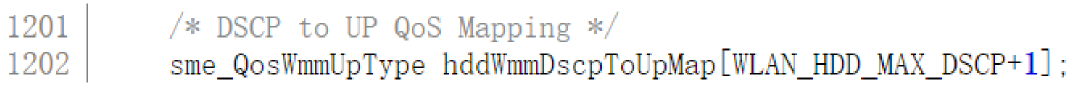
Wi-Fi Driver Security

CVE-2020-3698



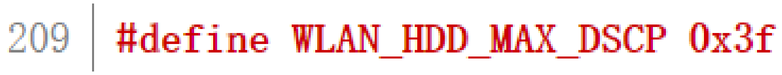
```
        for (i = 0; i < SME_QOS_WMM_UP_MAX; i++)
        {
            for (j = pSession->QosMapSet.dscp_range[i][0];
                            j <= pSession->QosMapSet.dscp_range[i][1]; j++)
            {
                if ((pSession->QosMapSet.dscp_range[i][0] == 255) &&
                             (pSession->QosMapSet.dscp_range[i][1] == 255))
                {
                    VOS_TRACE(VOS_MODULE_ID_SME, VOS_TRACE_LEVEL_ERROR,
                    "%s: User Priority %d is not used in mapping",
                                                   __func__, i);
                    break;
                }
                else
                {
                    dscpmapping[j]= i;
                }
            }
        }
```

the vulnerability code in sme_api.c



```
    /* DSCP to UP QoS Mapping */
    sme_QosWmmUpType hddWmmDscpToUpMap[WLAN_HDD_MAX_DSCP+1];
```

hddWmmDscpToUpMap in wlan_hdd_main.h

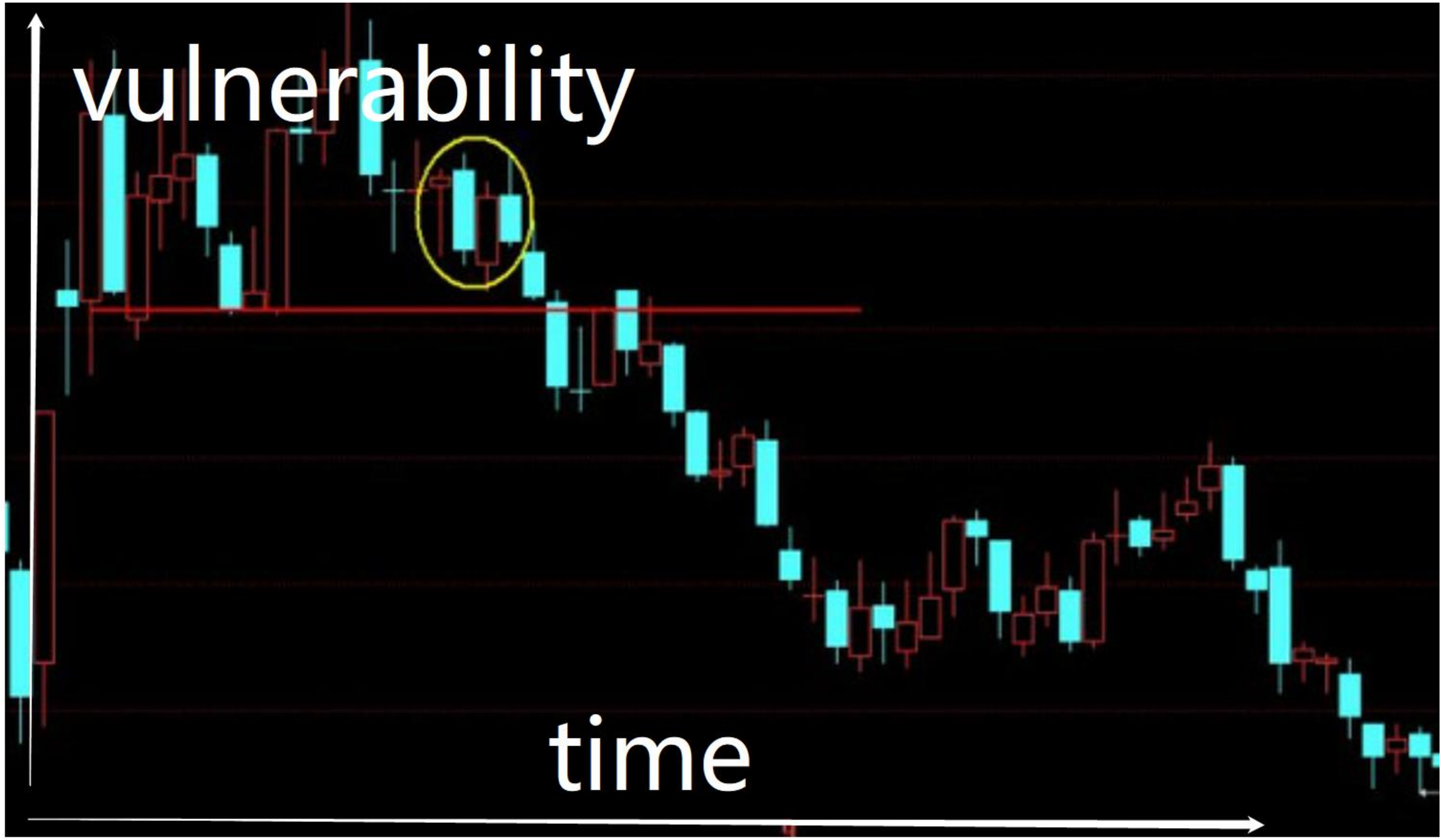


```
#define WLAN_HDD_MAX_DSCP 0x3f
```

Wi-Fi_HDD_MAX_DSCP in wlan_hdd_wmm.h

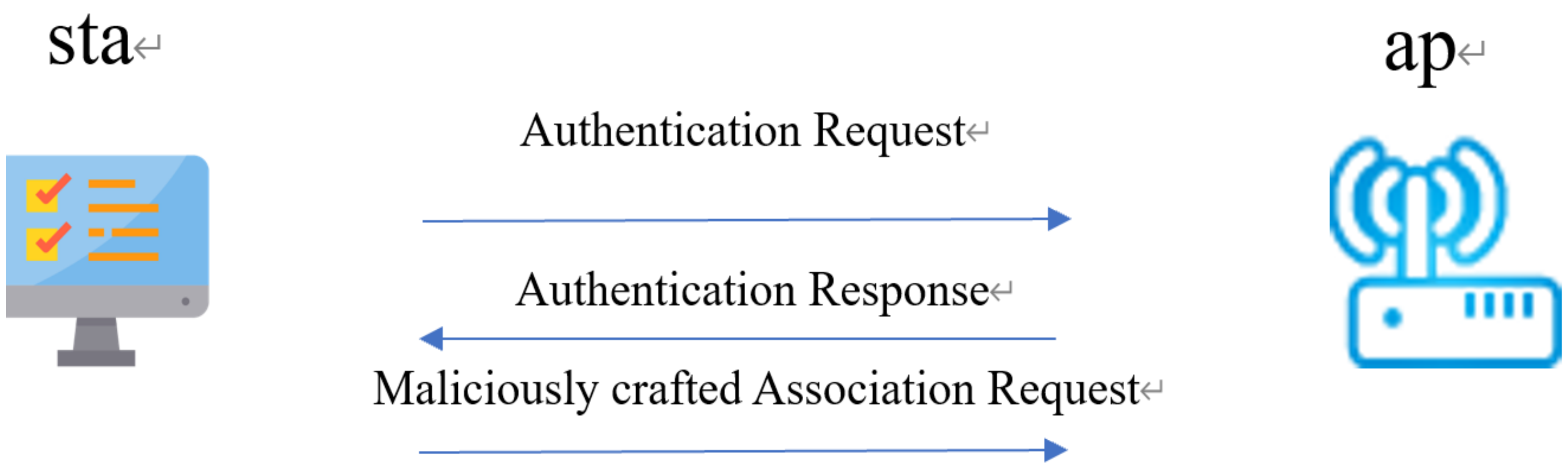
https://source.codeaurora.org/quic/la/platform/vendor/qcom-opensource/wlan/qcacld-3.0/commit/?id=df541cea94d83533ff8f34a9b8ae77964788b1c7

Wi-Fi Driver Security



Wi-Fi Driver Security

New born vulnerability CVE-2021-1955



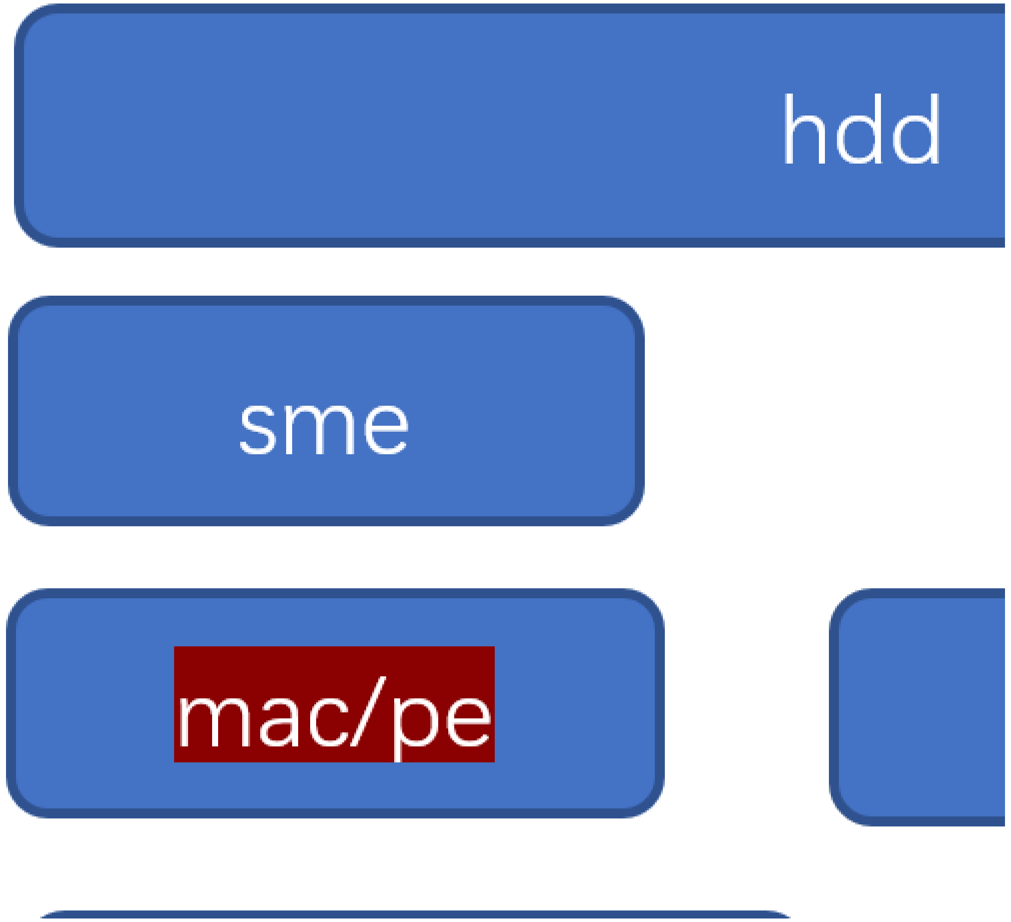
sta↵

ap↵

Authentication Request↵

Authentication Response↵

Maliciously crafted Association Request↵



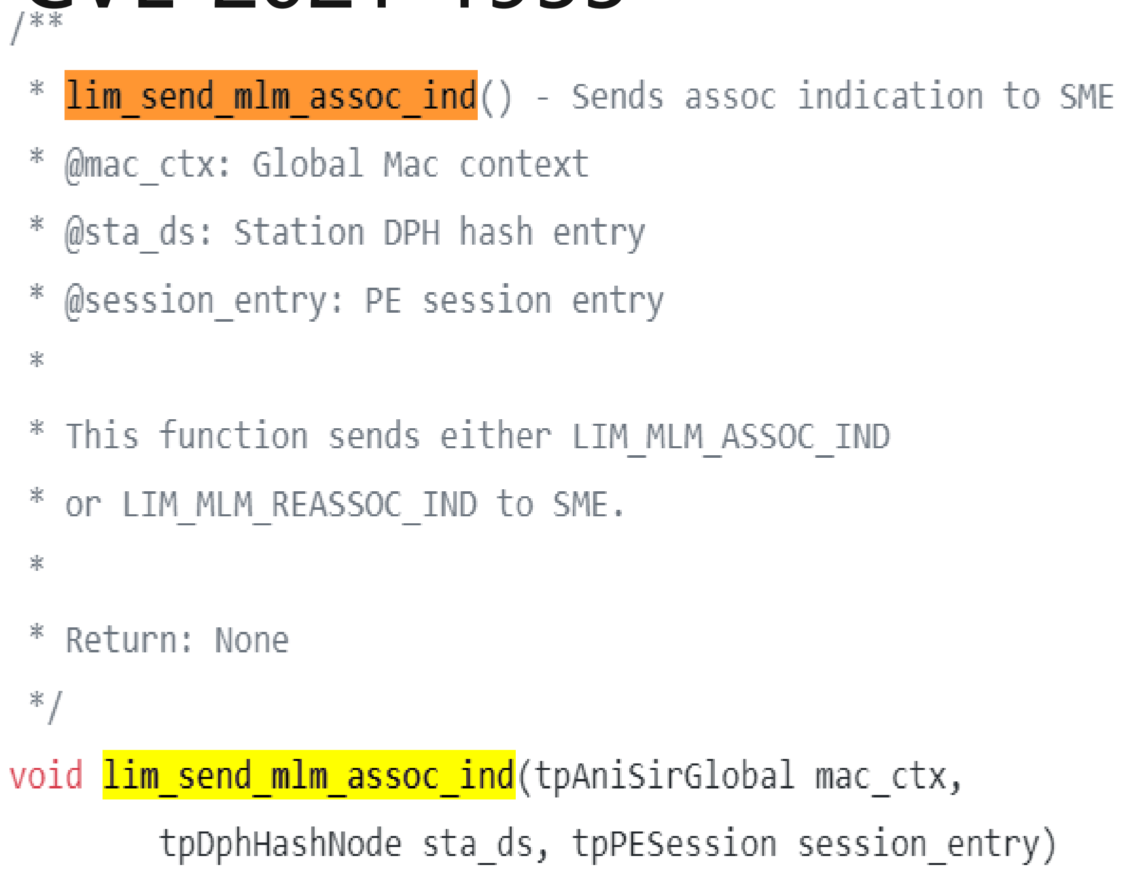
hdd

sme

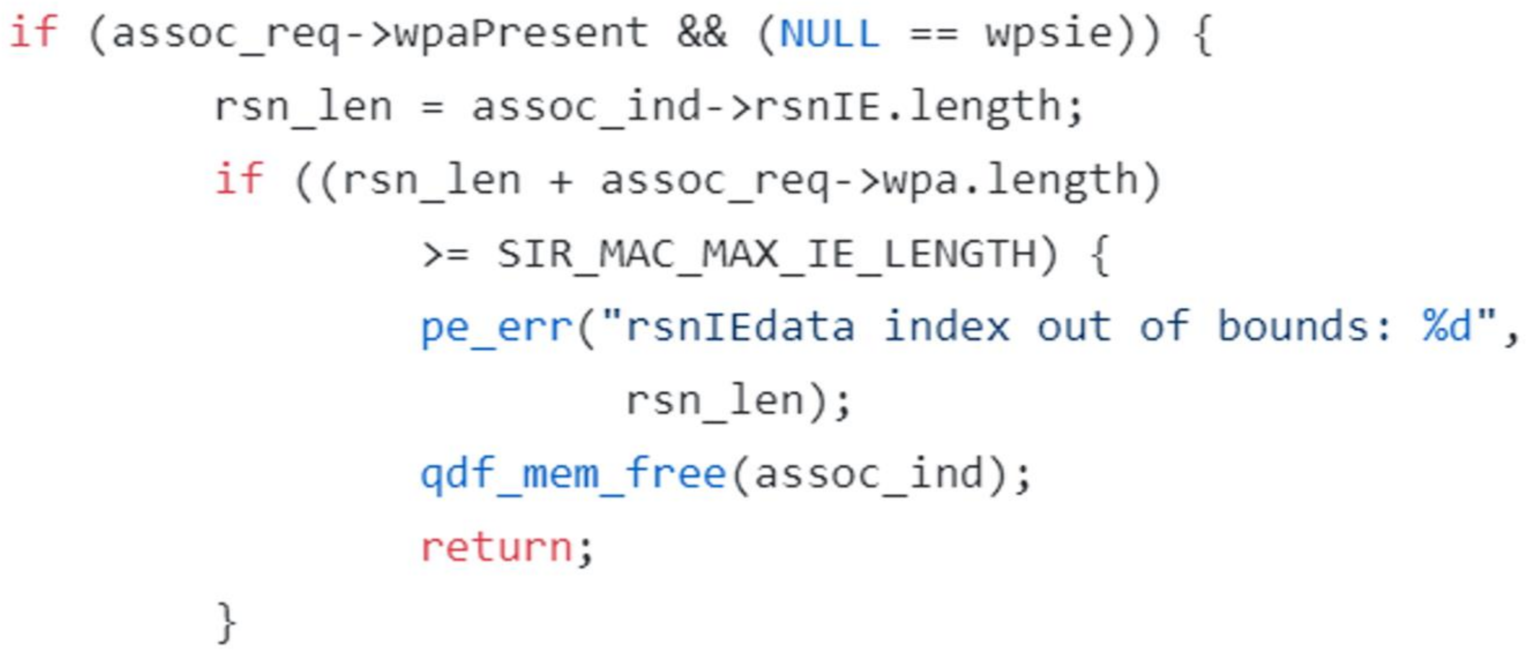
mac/pe

Wi-Fi Driver Security

CVE-2021-1955



```
/**
 * lim_send_mlm_assoc_ind() - Sends assoc indication to SME
 * @mac_ctx: Global Mac context
 * @sta_ds: Station DPH hash entry
 * @session_entry: PE session entry
 *
 * This function sends either LIM_MLM_ASSOC_IND
 * or LIM_MLM_REASSOC_IND to SME.
 *
 * Return: None
 */
void lim_send_mlm_assoc_ind(tpAniSirGlobal mac_ctx,
        tpDphHashNode sta_ds, tpPESession session_entry)
```



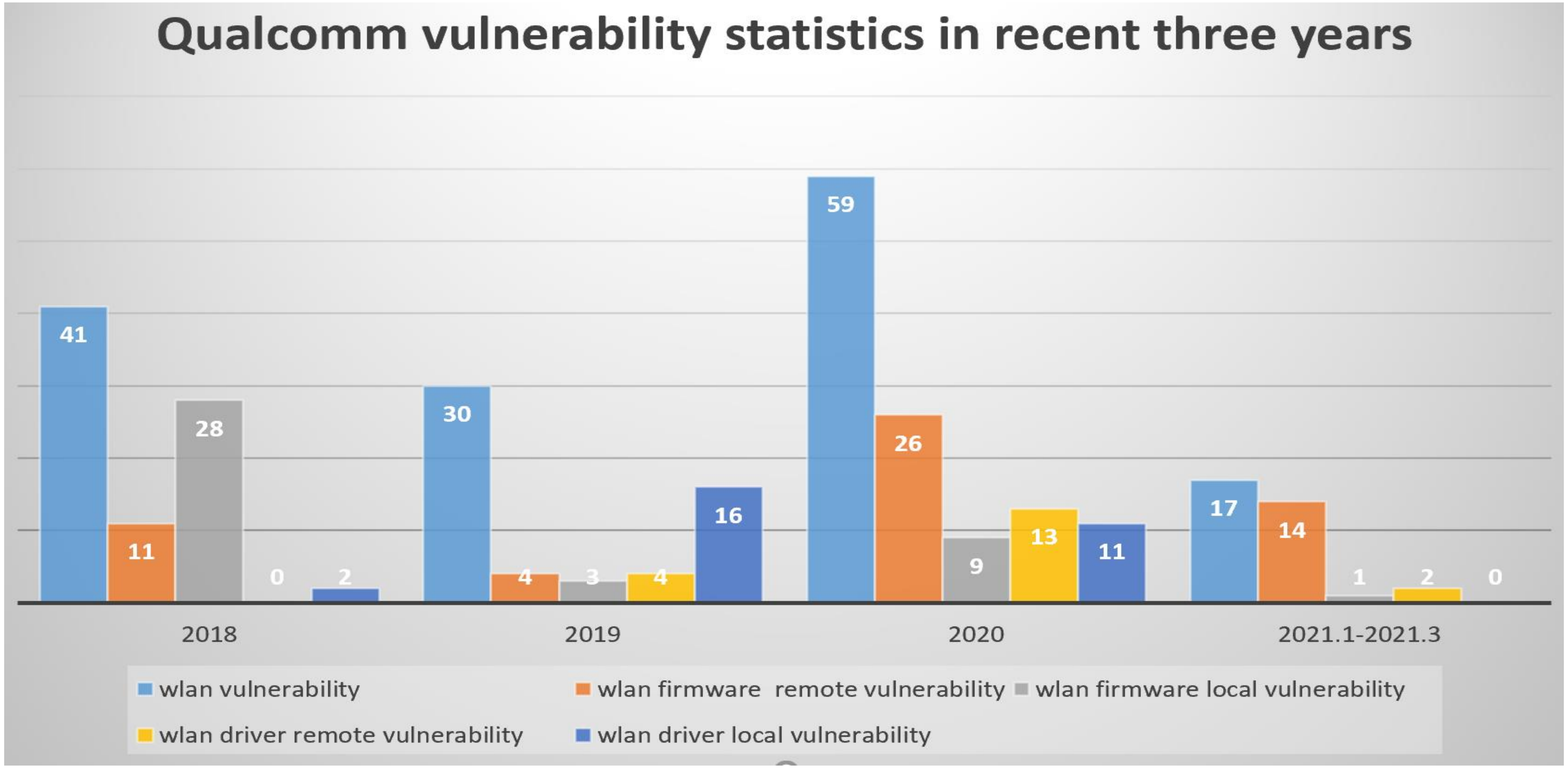
```
if (assoc_req->wpaPresent && (NULL == wpsie)) {
        rsn_len = assoc_ind->rsnIE.length;
        if ((rsn_len + assoc_req->wpa.length)
                >= SIR_MAC_MAX_IE_LENGTH) {
                pe_err("rsnIEdata index out of bounds: %d",
                        rsn_len);
                qdf_mem_free(assoc_ind);
                return;
        }
```

Wi-Fi Driver Security

Qualcomm mitigation

Qualcomm Wi-Fi driver used stack cookie、heap cookie、KASLR、W^X、CFI。

Qualcomm Wi-Fi Security research trends

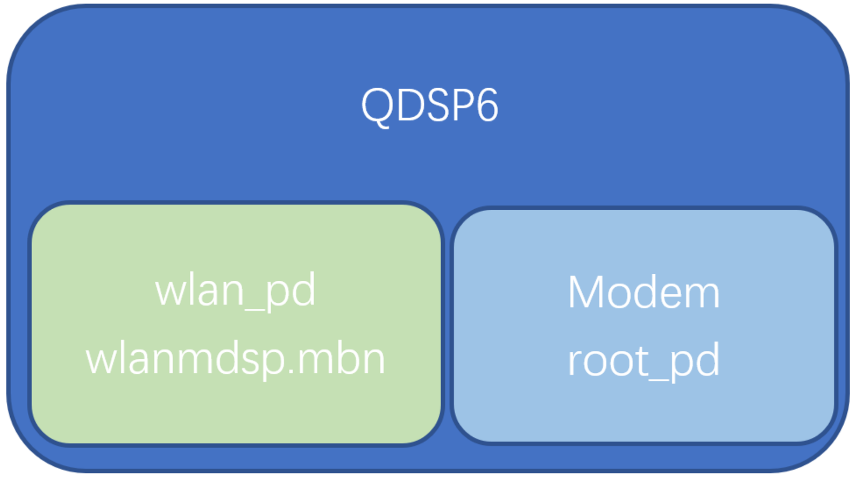


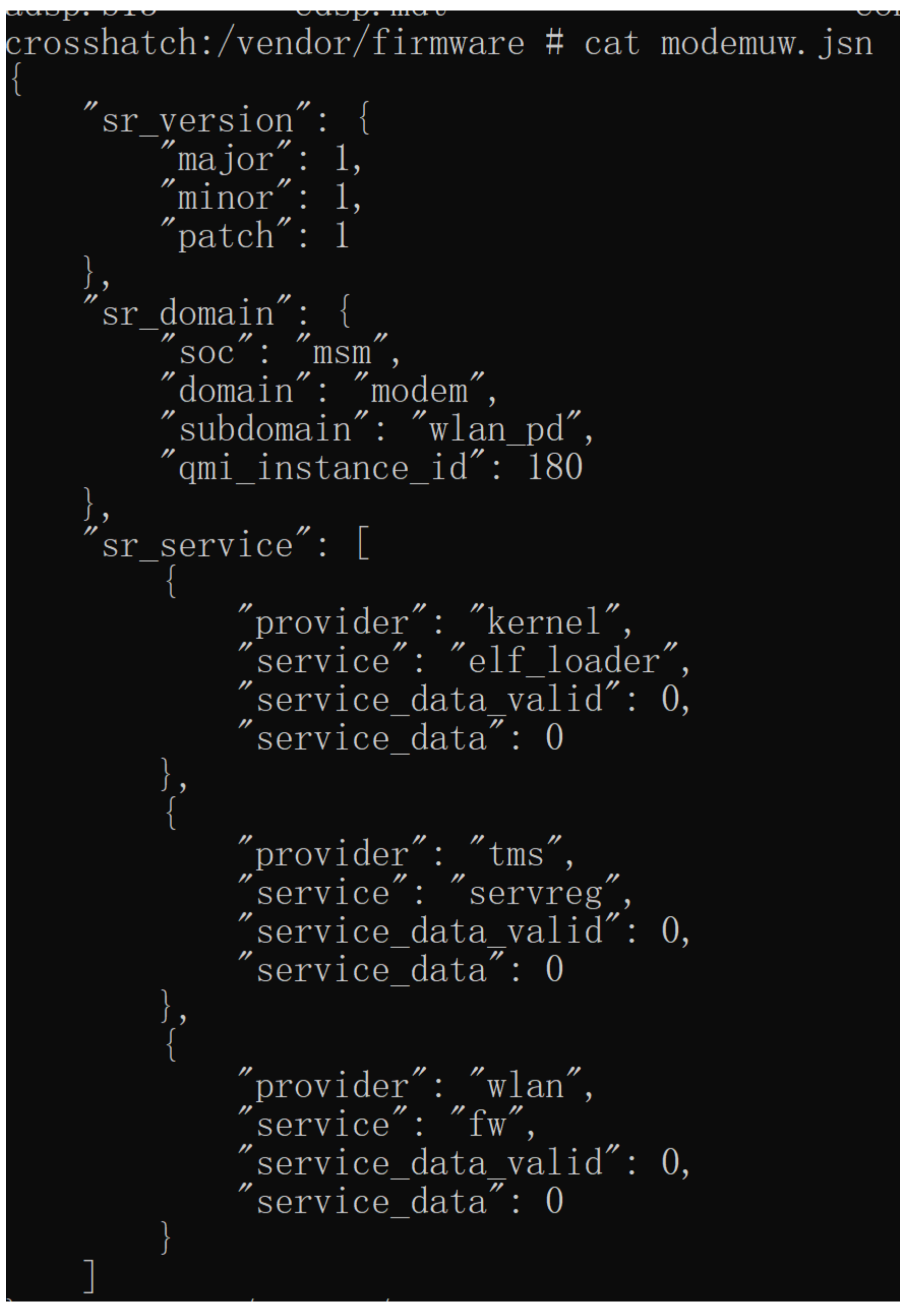
Wi-Fi Firmware Security

Qualcomm Wi-Fi firmware of SDM845

/vendor/firmware/wlanmdsp.mbn

modemuw.jsn :SDM845 modem configuration





```
crosshatch:/vendor/firmware # cat modemuw.jsn
{
    "sr_version": {
        "major": 1,
        "minor": 1,
        "patch": 1
    },
    "sr_domain": {
        "soc": "msm",
        "domain": "modem",
        "subdomain": "wlan_pd",
        "qmi_instance_id": 180
    },
    "sr_service": [
        {
            "provider": "kernel",
            "service": "elf_loader",
            "service_data_valid": 0,
            "service_data": 0
        },
        {
            "provider": "tms",
            "service": "servreg",
            "service_data_valid": 0,
            "service_data": 0
        },
        {
            "provider": "wlan",
            "service": "fw",
            "service_data_valid": 0,
            "service_data": 0
        }
    ]
```

Wi-Fi Firmware Security

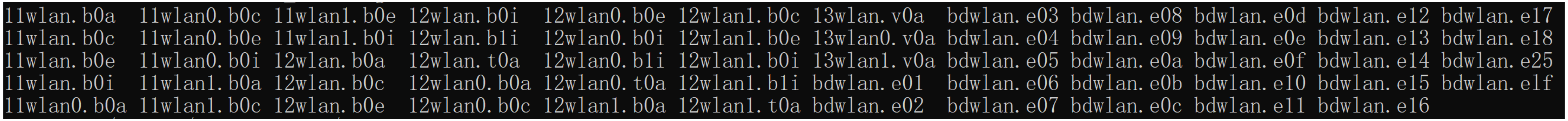
Snapdragon 865

firmware

/vendor/firmware_mnt/image/amss20.bin

bdf: board data file

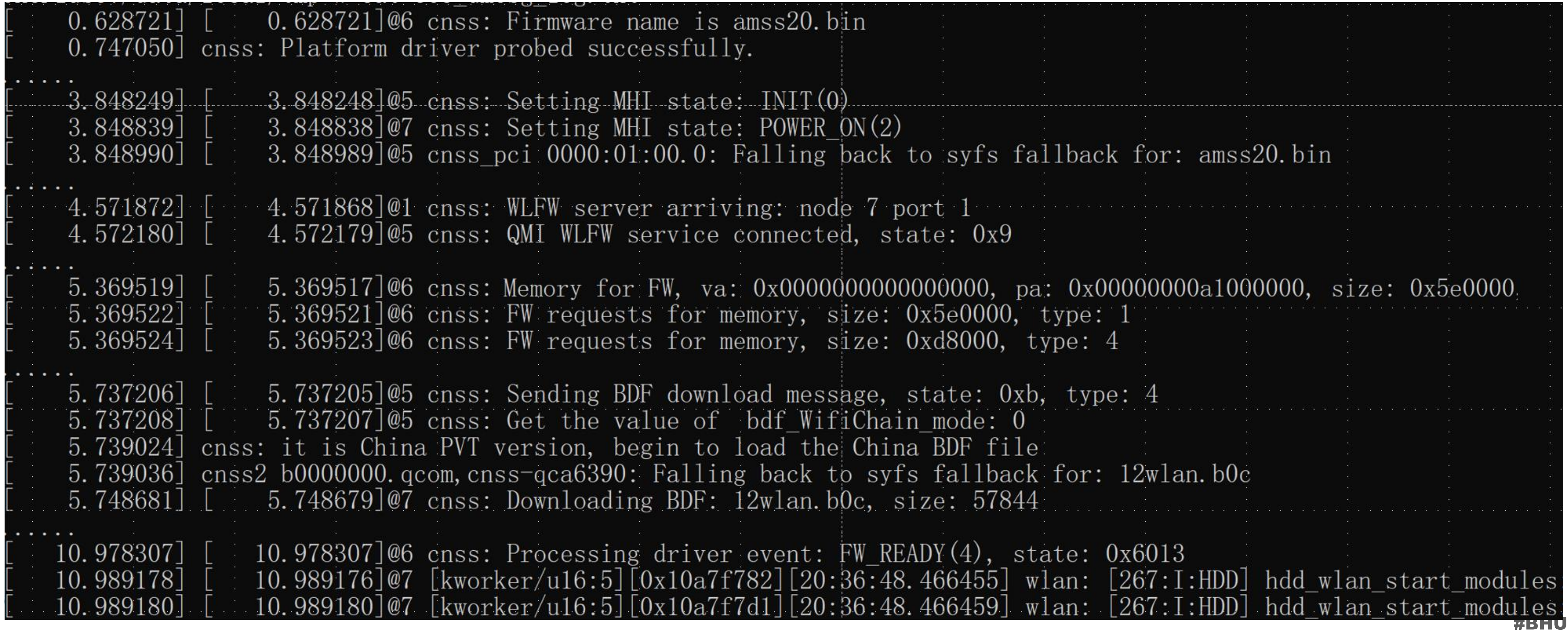
cnss: connectivity subsystem



```
11wlan.b0a  11wlan0.b0c 11wlan1.b0e 12wlan.b0i  12wlan0.b0e 12wlan1.b0c 13wlan.v0a  bdwlan.e03 bdwlan.e08 bdwlan.e0d bdwlan.e12 bdwlan.e17
11wlan.b0c  11wlan0.b0e 11wlan1.b0i 12wlan.b1i  12wlan0.b0i 12wlan1.b0e 13wlan0.v0a bdwlan.e04 bdwlan.e09 bdwlan.e0e bdwlan.e13 bdwlan.e18
11wlan.b0e  11wlan0.b0i 12wlan.b0a  12wlan.t0a  12wlan0.b1i 12wlan1.b0i 13wlan1.v0a bdwlan.e05 bdwlan.e0a bdwlan.e0f bdwlan.e14 bdwlan.e25
11wlan.b0i  11wlan1.b0a 12wlan.b0c  12wlan0.b0a 12wlan0.t0a 12wlan1.b1i bdwlan.e01  bdwlan.e06 bdwlan.e0b bdwlan.e10 bdwlan.e15 bdwlan.e1f
11wlan0.b0a 11wlan1.b0c 12wlan.b0e  12wlan0.b0c 12wlan1.b0a 12wlan1.t0a bdwlan.e02  bdwlan.e07 bdwlan.e0c bdwlan.e11 bdwlan.e16
```

Wi-Fi Firmware Security

Snapdragon 865 Wi-Fi firmware loading process



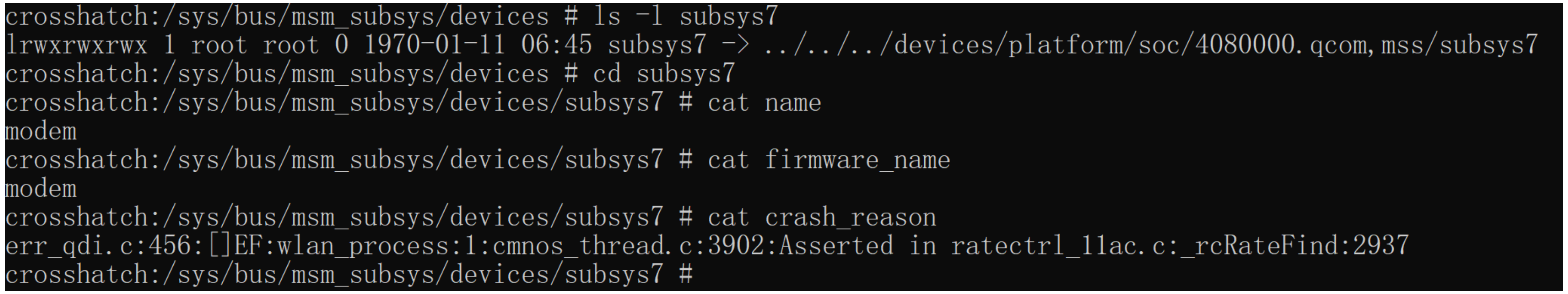
```
[    0.628721] [    0.628721]@6 cnss: Firmware name is amss20.bin
[    0.747050] cnss: Platform driver probed successfully.
......
[    3.848249] [    3.848248]@5 cnss: Setting MHI state: INIT(0)
[    3.848839] [    3.848838]@7 cnss: Setting MHI state: POWER_ON(2)
[    3.848990] [    3.848989]@5 cnss_pci 0000:01:00.0: Falling back to syfs fallback for: amss20.bin
......
[    4.571872] [    4.571868]@1 cnss: WLFW server arriving: node 7 port 1
[    4.572180] [    4.572179]@5 cnss: QMI WLFW service connected, state: 0x9
......
[    5.369519] [    5.369517]@6 cnss: Memory for FW, va: 0x0000000000000000, pa: 0x00000000a1000000, size: 0x5e0000
[    5.369522] [    5.369521]@6 cnss: FW requests for memory, size: 0x5e0000, type: 1
[    5.369524] [    5.369523]@6 cnss: FW requests for memory, size: 0xd8000, type: 4
......
[    5.737206] [    5.737205]@5 cnss: Sending BDF download message, state: 0xb, type: 4
[    5.737208] [    5.737207]@5 cnss: Get the value of  bdf_WifiChain_mode: 0
[    5.739024] cnss: it is China PVT version, begin to load the China BDF file
[    5.739036] cnss2 b0000000.qcom,cnss-qca6390: Falling back to syfs fallback for: 12wlan.b0c
[    5.748681] [    5.748679]@7 cnss: Downloading BDF: 12wlan.b0c, size: 57844
......
[   10.978307] [   10.978307]@6 cnss: Processing driver event: FW_READY(4), state: 0x6013
[   10.989178] [   10.989176]@7 [kworker/u16:5][0x10a7f782][20:36:48.466455] wlan: [267:I:HDD] hdd_wlan_start_modules
[   10.989180] [   10.989180]@7 [kworker/u16:5][0x10a7f7d1][20:36:48.466459] wlan: [267:I:HDD] hdd_wlan_start_modules
```

Wi-Fi Firmware Security

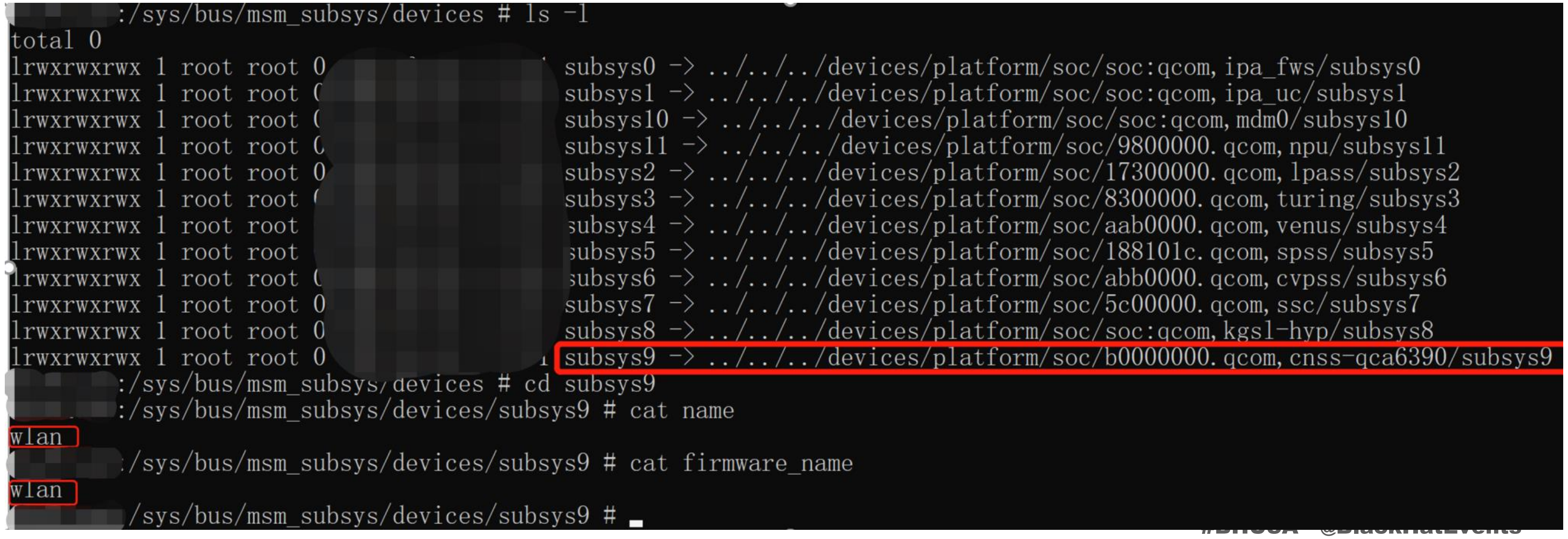
Qualcomm Subsystem

e.g.

- MODEM
- GPU
- LPASS
- VENUS
- WCNSS
- ……



```
crosshatch:/sys/bus/msm_subsys/devices # ls -l subsys7
lrwxrwxrwx 1 root root 0 1970-01-11 06:45 subsys7 -> ../../../devices/platform/soc/4080000.qcom,mss/subsys7
crosshatch:/sys/bus/msm_subsys/devices # cd subsys7
crosshatch:/sys/bus/msm_subsys/devices/subsys7 # cat name
modem
crosshatch:/sys/bus/msm_subsys/devices/subsys7 # cat firmware_name
modem
crosshatch:/sys/bus/msm_subsys/devices/subsys7 # cat crash_reason
err_qdi.c:456:[]EF:wlan_process:1:cmnos_thread.c:3902:Asserted in ratectrl_11ac.c:_rcRateFind:2937
crosshatch:/sys/bus/msm_subsys/devices/subsys7 #
```



```
[redacted]:/sys/bus/msm_subsys/devices # ls -l
total 0
lrwxrwxrwx 1 root root 0 [redacted] subsys0 -> ../../../devices/platform/soc/soc:qcom,ipa_fws/subsys0
lrwxrwxrwx 1 root root 0            subsys1 -> ../../../devices/platform/soc/soc:qcom,ipa_uc/subsys1
lrwxrwxrwx 1 root root 0            subsys10 -> ../../../devices/platform/soc/soc:qcom,mdm0/subsys10
lrwxrwxrwx 1 root root 0            subsys11 -> ../../../devices/platform/soc/9800000.qcom,npu/subsys11
lrwxrwxrwx 1 root root 0            subsys2 -> ../../../devices/platform/soc/17300000.qcom,lpass/subsys2
lrwxrwxrwx 1 root root 0            subsys3 -> ../../../devices/platform/soc/8300000.qcom,turing/subsys3
lrwxrwxrwx 1 root root              subsys4 -> ../../../devices/platform/soc/aab0000.qcom,venus/subsys4
lrwxrwxrwx 1 root root              subsys5 -> ../../../devices/platform/soc/188101c.qcom,spss/subsys5
lrwxrwxrwx 1 root root 0            subsys6 -> ../../../devices/platform/soc/abb0000.qcom,cvpss/subsys6
lrwxrwxrwx 1 root root 0            subsys7 -> ../../../devices/platform/soc/5c00000.qcom,ssc/subsys7
lrwxrwxrwx 1 root root 0            subsys8 -> ../../../devices/platform/soc/soc:qcom,kgsl-hyp/subsys8
lrwxrwxrwx 1 root root 0            subsys9 -> ../../../devices/platform/soc/b0000000.qcom,cnss-qca6390/subsys9
[redacted]:/sys/bus/msm_subsys/devices # cd subsys9
[redacted]:/sys/bus/msm_subsys/devices/subsys9 # cat name
wlan
[redacted]:/sys/bus/msm_subsys/devices/subsys9 # cat firmware_name
wlan
[redacted]/sys/bus/msm_subsys/devices/subsys9 #
```

Wi-Fi Firmware Security

SSR:Subsystem restart

/sys/bus/msm_subsys/devices/subsysx/restart_level

“related” : enable SSR, only the subsystem will be restarted when the subsystem is abnormal, and the main system and other subsystems will not be affected. ramdump of the subsystem will be collected

“system” : Disable SSR, the main system restarts, and ramdump will not be collected

Wi-Fi Firmware Security

Get ramdump

```
echo 1 > /sys/module/subsystem_restart/parameters/enable_ramdumps

echo 1 > /sys/module/subsystem_restart/parameters/enable_debug

echo 1 > /sys/module/subsystem_restart/parameters/enable_mini_ramdumps

subsystem_ramdump 1
```

When subsystem crash occurs，ramdump file(ramdump_wlan_*.elf) will be generated in /data/vendor/ramdump

Wi-Fi Firmware Security

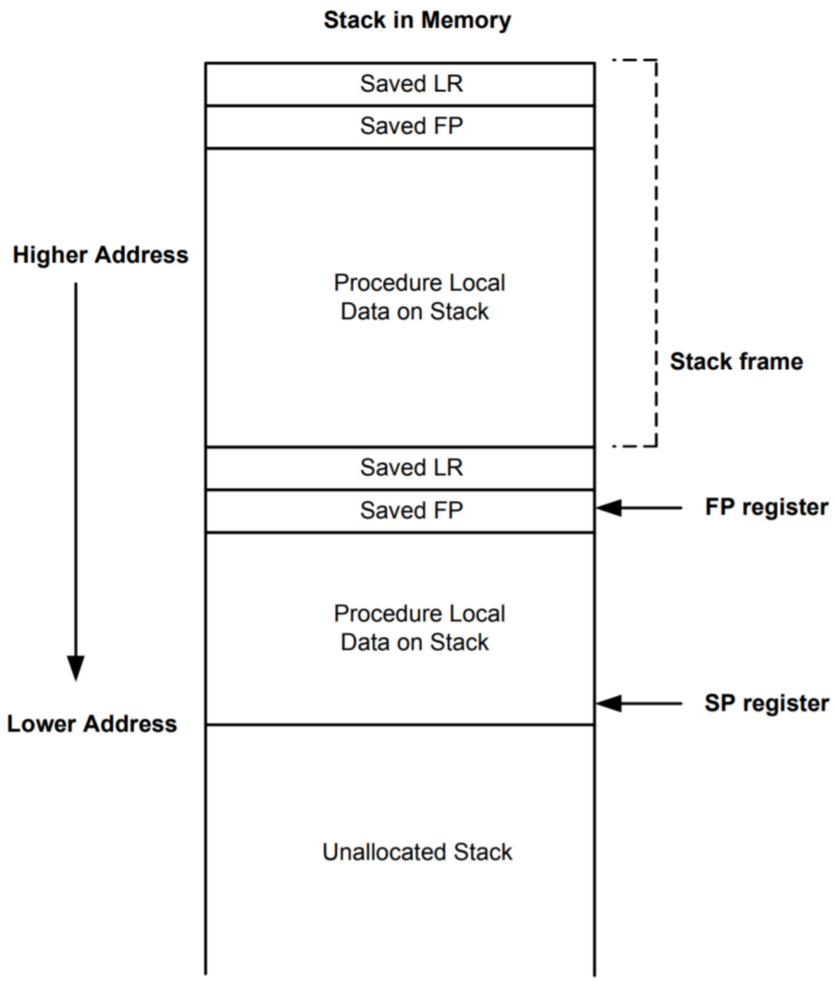
Qualcomm hexagon

arg1:R0 arg2:R1 arg3:R2

ret:R0

https://developer.qualcomm.com/qfile/67415/80-n2040-42_a_qualcomm_hexagon_v66_programmer_reference_manual.pdf

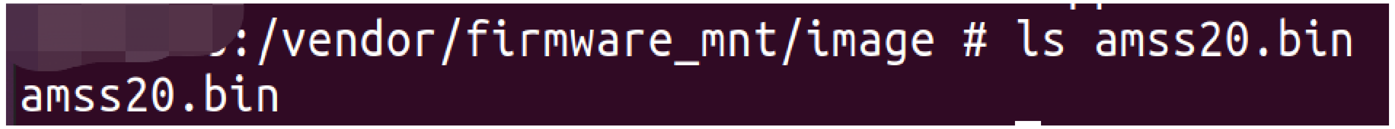
https://developer.qualcomm.com/qfile/67417/80-n2040-45_b_qualcomm_hexagon_v67_programmer_reference_manual.pdf



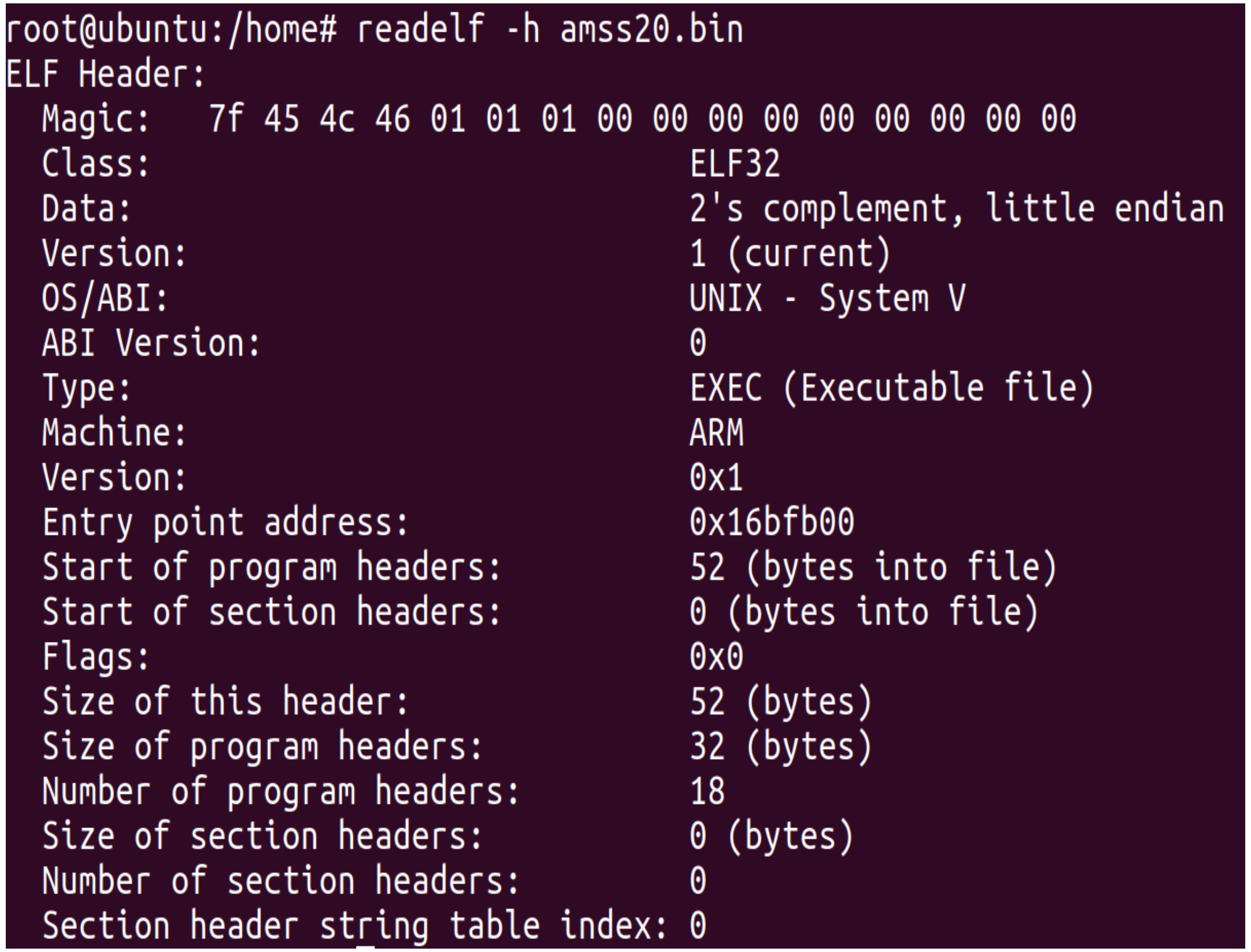
Wi-Fi Firmware Security

Firmware extraction

amss20.bin is the Wi-Fi firmware of qca639x



```
[redacted]:/vendor/firmware_mnt/image # ls amss20.bin
amss20.bin
```



```
root@ubuntu:/home# readelf -h amss20.bin
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x16bfb00
  Start of program headers:          52 (bytes into file)
  Start of section headers:          0 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         18
  Size of section headers:           0 (bytes)
  Number of section headers:         0
  Section header string table index: 0
```

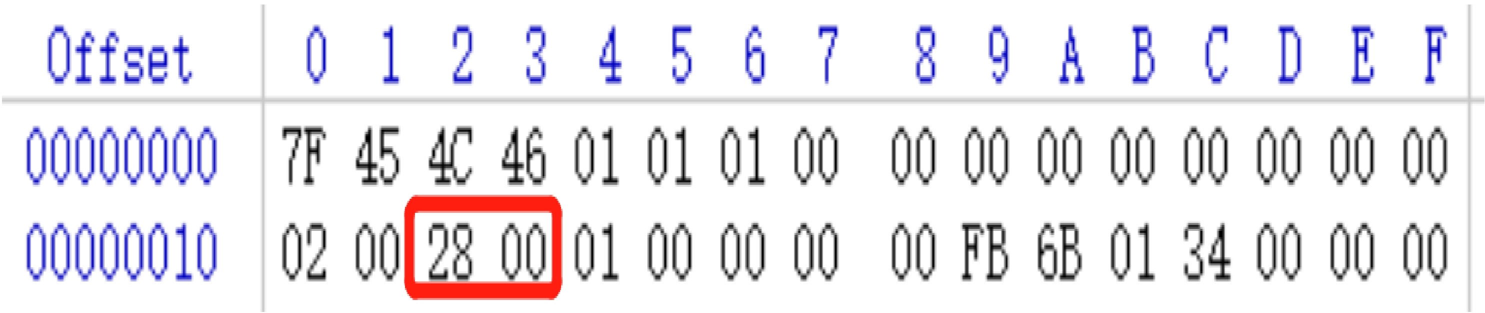
Wi-Fi Firmware Security

Firmware extraction

Architecture of amss20.bin :

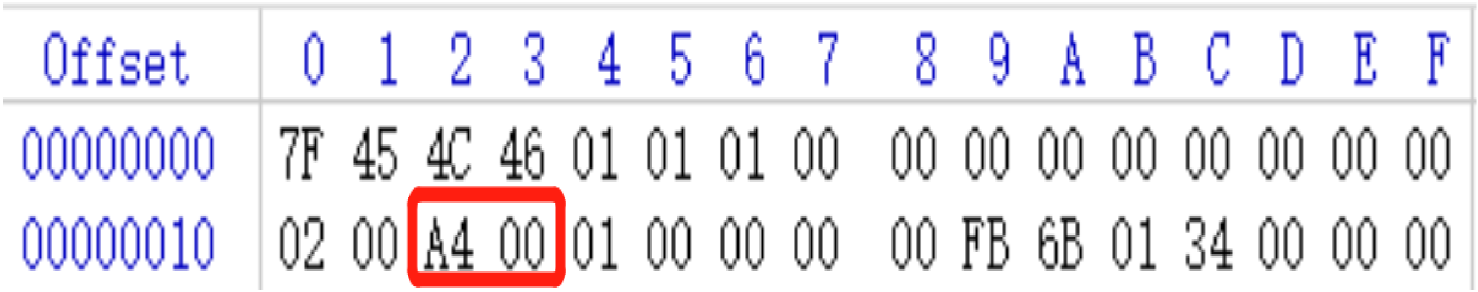
ARM ✗

QUALCOMM DSP6 Processor ✓



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  7F 45 4C 46 01 01 01 00  00 00 00 00 00 00 00 00
00000010  02 00 28 00 01 00 00 00  00 FB 6B 01 34 00 00 00
```

Old elf machine field of amss20.bin



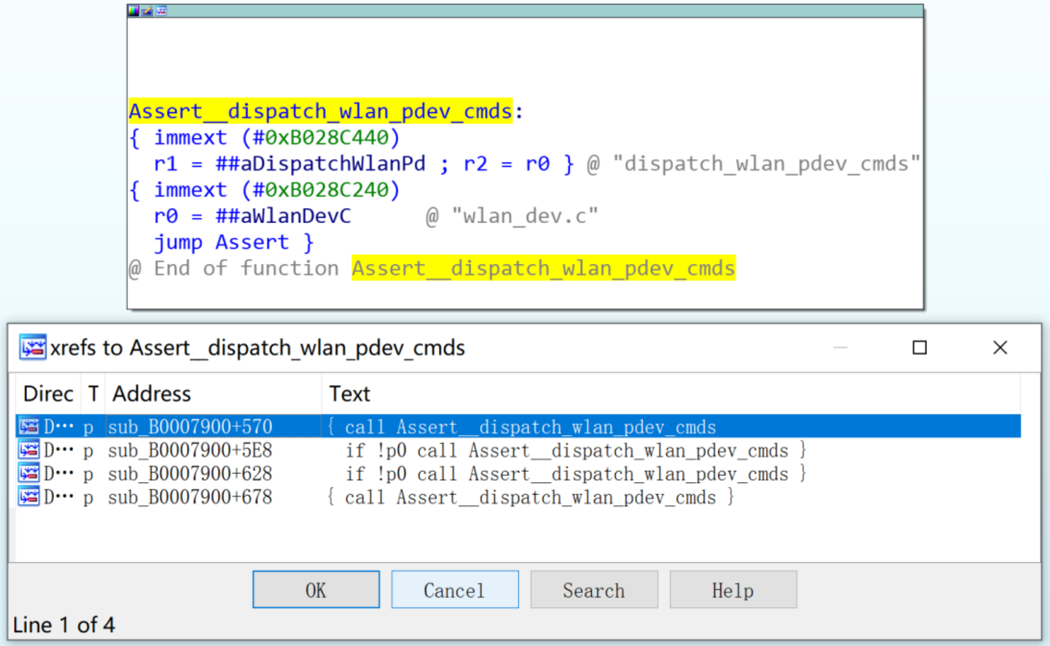
```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  7F 45 4C 46 01 01 01 00  00 00 00 00 00 00 00 00
00000010  02 00 A4 00 01 00 00 00  00 FB 6B 01 34 00 00 00
```

New elf machine field of amss20.bin

Wi-Fi Firmware Security

Firmware symbol

wlanmdsp.mbn of snapdragon 845

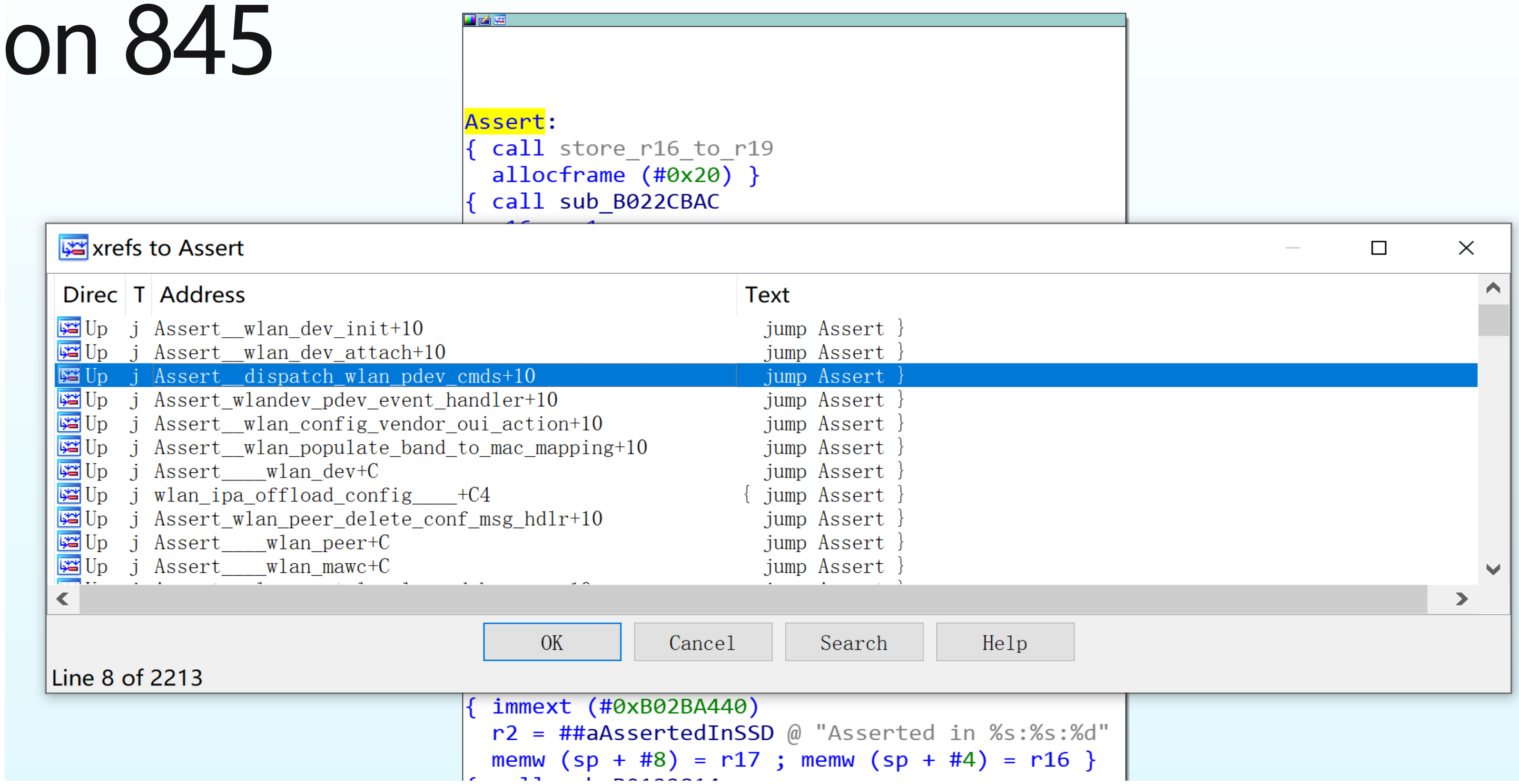


```
Assert__dispatch_wlan_pdev_cmds:
{ immext (#0xB028C440)
  r1 = ##aDispatchWlanPd ; r2 = r0 } @ "dispatch_wlan_pdev_cmds"
{ immext (#0xB028C240)
  r0 = ##aWlanDevC       @ "wlan_dev.c"
  jump Assert }
@ End of function Assert__dispatch_wlan_pdev_cmds
```

xrefs to Assert__dispatch_wlan_pdev_cmds

Direc	T	Address	Text
D···	p	sub_B0007900+570	{ call Assert__dispatch_wlan_pdev_cmds
D···	p	sub_B0007900+5E8	if !p0 call Assert__dispatch_wlan_pdev_cmds }
D···	p	sub_B0007900+628	if !p0 call Assert__dispatch_wlan_pdev_cmds }
D···	p	sub_B0007900+678	{ call Assert__dispatch_wlan_pdev_cmds }

Line 1 of 4



```
Assert:
{ call store_r16_to_r19
  allocframe (#0x20) }
{ call sub_B022CBAC
```

xrefs to Assert

Direc	T	Address	Text
Up	j	Assert__wlan_dev_init+10	jump Assert }
Up	j	Assert__wlan_dev_attach+10	jump Assert }
Up	j	Assert__dispatch_wlan_pdev_cmds+10	jump Assert }
Up	j	Assert_wlandev_pdev_event_handler+10	jump Assert }
Up	j	Assert__wlan_config_vendor_oui_action+10	jump Assert }
Up	j	Assert__wlan_populate_band_to_mac_mapping+10	jump Assert }
Up	j	Assert____wlan_dev+C	jump Assert }
Up	j	wlan_ipa_offload_config____+C4	{ jump Assert }
Up	j	Assert_wlan_peer_delete_conf_msg_hdlr+10	jump Assert }
Up	j	Assert____wlan_peer+C	jump Assert }
Up	j	Assert____wlan_mawc+C	jump Assert }

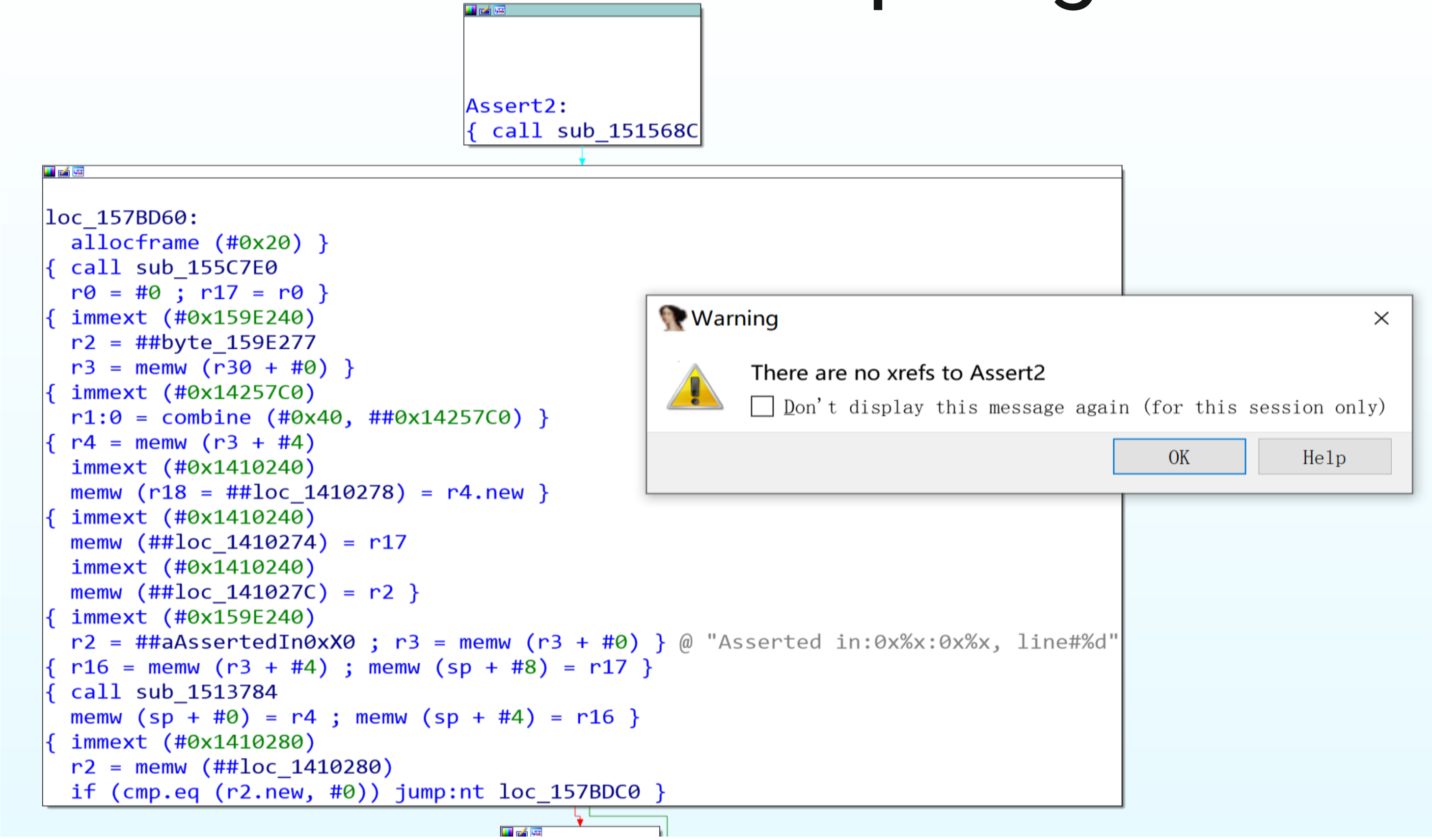
Line 8 of 2213

```
{ immext (#0xB02BA440)
  r2 = ##aAssertedInSSD @ "Asserted in %s:%s:%d"
  memw (sp + #8) = r17 ; memw (sp + #4) = r16 }
```

Wi-Fi Firmware Security

Firmware symbol

amss20.bin of Snapdragon 865

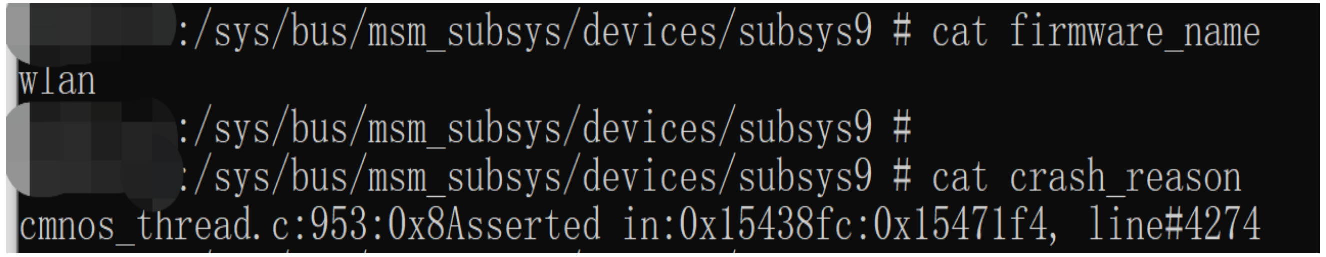


```
Assert2:
{ call  sub_151568C

loc_157BD60:
  allocframe (#0x20) }
{ call  sub_155C7E0
  r0 = #0 ; r17 = r0 }
{ immext (#0x159E240)
  r2 = ##byte_159E277
  r3 = memw (r30 + #0) }
{ immext (#0x14257C0)
  r1:0 = combine (#0x40, ##0x14257C0) }
{ r4 = memw (r3 + #4)
  immext (#0x1410240)
  memw (r18 = ##loc_1410278) = r4.new }
{ immext (#0x1410240)
  memw (##loc_1410274) = r17
  immext (#0x1410240)
  memw (##loc_141027C) = r2 }
{ immext (#0x159E240)
  r2 = ##aAssertedIn0xX0 ; r3 = memw (r3 + #0) } @ "Asserted in:0x%x:0x%x, line#%d"
{ r16 = memw (r3 + #4) ; memw (sp + #8) = r17 }
{ call  sub_1513784
  memw (sp + #0) = r4 ; memw (sp + #4) = r16 }
{ immext (#0x1410280)
  r2 = memw (##loc_1410280)
  if (cmp.eq (r2.new, #0)) jump:nt loc_157BDC0 }
```

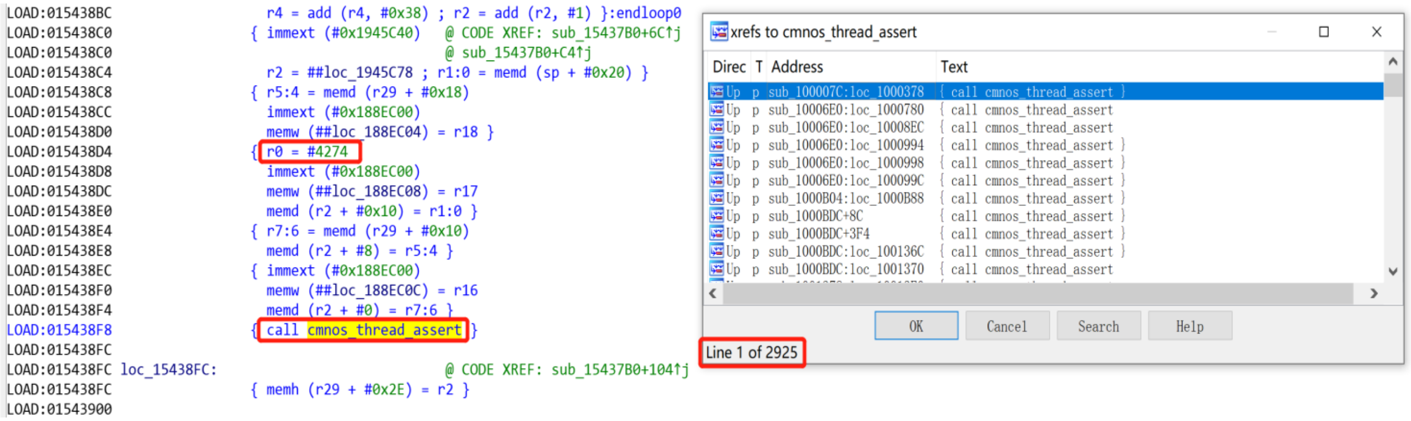
Warning

There are no xrefs to Assert2

□ Don't display this message again (for this session only)



```
[redacted]:/sys/bus/msm_subsys/devices/subsys9 # cat firmware_name
wlan
[redacted]:/sys/bus/msm_subsys/devices/subsys9 #
[redacted]:/sys/bus/msm_subsys/devices/subsys9 # cat crash_reason
cmnos_thread.c:953:0x8Asserted in:0x15438fc:0x15471f4, line#4274
```



```
LOAD:015438BC                 r4 = add (r4, #0x38) ; r2 = add (r2, #1) }:endloop0
LOAD:015438C0               { immext (#0x1945C40)   @ CODE XREF: sub_15437B0+6C↑j
LOAD:015438C0                                       @ sub_15437B0+C4↑j
LOAD:015438C4                 r2 = ##loc_1945C78 ; r1:0 = memd (sp + #0x20) }
LOAD:015438C8               { r5:4 = memd (r29 + #0x18)
LOAD:015438CC                 immext (#0x188EC00)
LOAD:015438D0                 memw (##loc_188EC04) = r18 }
LOAD:015438D4               { r0 = #4274
LOAD:015438D8                 immext (#0x188EC00)
LOAD:015438DC                 memw (##loc_188EC08) = r17
LOAD:015438E0                 memd (r2 + #0x10) = r1:0 }
LOAD:015438E4               { r7:6 = memd (r29 + #0x10)
LOAD:015438E8                 memd (r2 + #8) = r5:4 }
LOAD:015438EC               { immext (#0x188EC00)
LOAD:015438F0                 memw (##loc_188EC0C) = r16
LOAD:015438F4                 memd (r2 + #0) = r7:6 }
LOAD:015438F8               { call  cmnos_thread_assert }
LOAD:015438FC
LOAD:015438FC loc_15438FC:                          @ CODE XREF: sub_15437B0+104↑j
LOAD:015438FC               { memh (r29 + #0x2E) = r2 }
LOAD:01543900
```

xrefs to cmnos_thread_assert

Direc	T	Address	Text
Up	p	sub_100007C:loc_1000378	{ call cmnos_thread_assert }
Up	p	sub_10006E0:loc_1000780	{ call cmnos_thread_assert
Up	p	sub_10006E0:loc_10008EC	{ call cmnos_thread_assert
Up	p	sub_10006E0:loc_1000994	{ call cmnos_thread_assert }
Up	p	sub_10006E0:loc_1000998	{ call cmnos_thread_assert }
Up	p	sub_10006E0:loc_100099C	{ call cmnos_thread_assert }
Up	p	sub_1000B04:loc_1000B88	{ call cmnos_thread_assert }
Up	p	sub_1000BDC+8C	{ call cmnos_thread_assert }
Up	p	sub_1000BDC+3F4	{ call cmnos_thread_assert }
Up	p	sub_1000BDC:loc_100136C	{ call cmnos_thread_assert }
Up	p	sub_1000BDC:loc_1001370	{ call cmnos_thread_assert

Line 1 of 2925

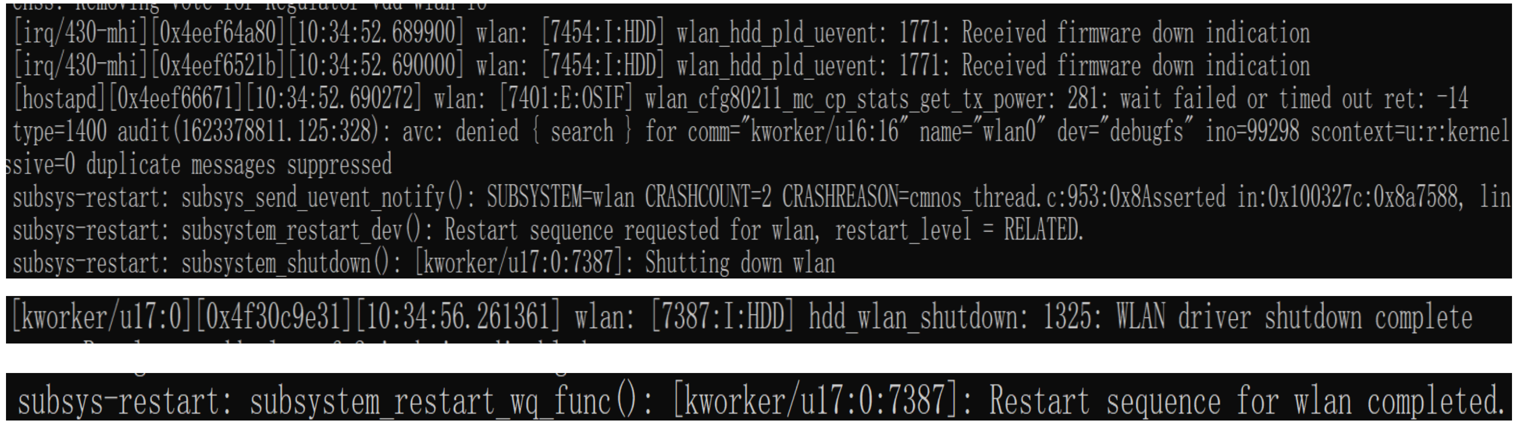
Wi-Fi Firmware Security

Firmware exception capture

crash_count: the number of subsystem crash

crash_reason: the last reason of subsystem crash

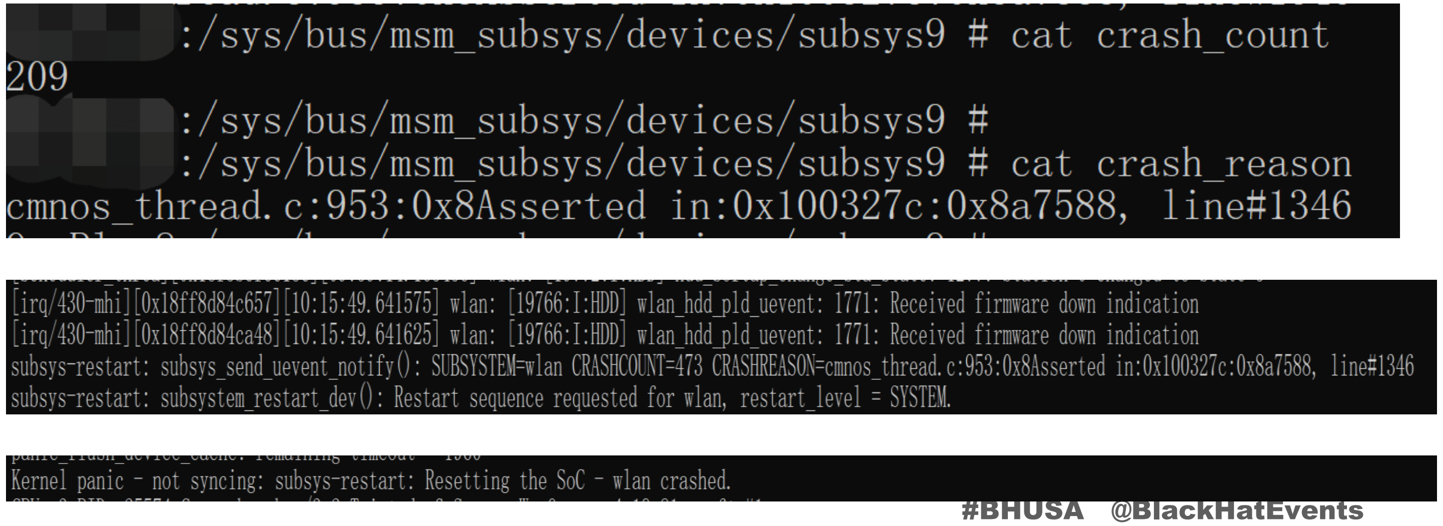
echo "system" > restart_level



```
[irq/430-mhi][0x4eef64a80][10:34:52.689900] wlan: [7454:I:HDD] wlan_hdd_pld_uevent: 1771: Received firmware down indication
[irq/430-mhi][0x4eef6521b][10:34:52.690000] wlan: [7454:I:HDD] wlan_hdd_pld_uevent: 1771: Received firmware down indication
[hostapd][0x4eef66671][10:34:52.690272] wlan: [7401:E:OSIF] wlan_cfg80211_mc_cp_stats_get_tx_power: 281: wait failed or timed out ret: -14
type=1400 audit(1623378811.125:328): avc: denied { search } for comm="kworker/u16:16" name="wlan0" dev="debugfs" ino=99298 scontext=u:r:kernel
ssive=0 duplicate messages suppressed
subsys-restart: subsys_send_uevent_notify(): SUBSYSTEM=wlan CRASHCOUNT=2 CRASHREASON=cmnos_thread.c:953:0x8Asserted in:0x100327c:0x8a7588, lin
subsys-restart: subsystem_restart_dev(): Restart sequence requested for wlan, restart_level = RELATED.
subsys-restart: subsystem_shutdown(): [kworker/u17:0:7387]: Shutting down wlan
```

```
[kworker/u17:0][0x4f30c9e31][10:34:56.261361] wlan: [7387:I:HDD] hdd_wlan_shutdown: 1325: WLAN driver shutdown complete
```

```
subsys-restart: subsystem_restart_wq_func(): [kworker/u17:0:7387]: Restart sequence for wlan completed.
```



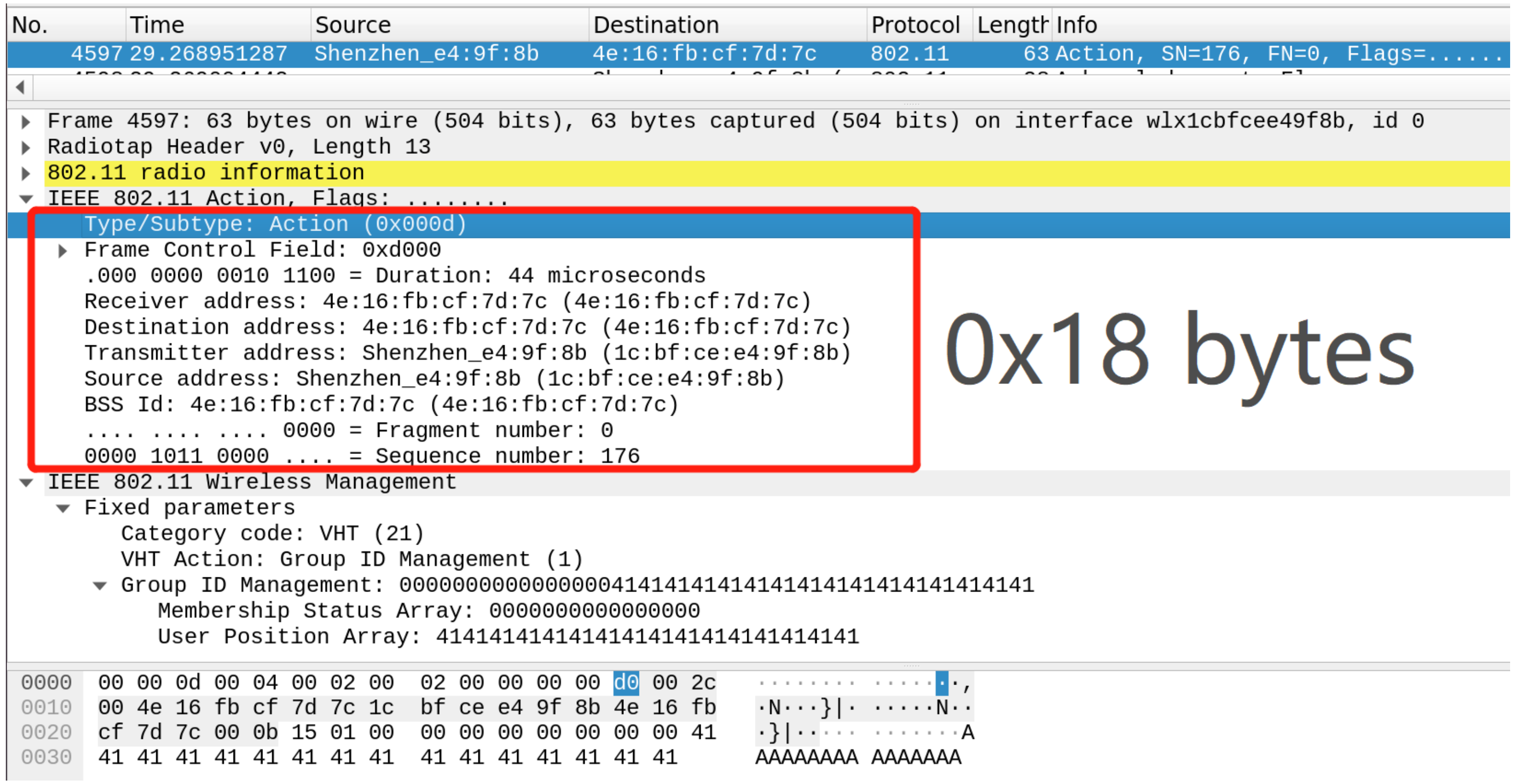
```
[redacted]:/sys/bus/msm_subsys/devices/subsys9 # cat crash_count
209
[redacted]:/sys/bus/msm_subsys/devices/subsys9 #
:/sys/bus/msm_subsys/devices/subsys9 # cat crash_reason
cmnos_thread.c:953:0x8Asserted in:0x100327c:0x8a7588, line#1346
```

```
[irq/430-mhi][0x18ff8d84c657][10:15:49.641575] wlan: [19766:I:HDD] wlan_hdd_pld_uevent: 1771: Received firmware down indication
[irq/430-mhi][0x18ff8d84ca48][10:15:49.641625] wlan: [19766:I:HDD] wlan_hdd_pld_uevent: 1771: Received firmware down indication
subsys-restart: subsys_send_uevent_notify(): SUBSYSTEM=wlan CRASHCOUNT=473 CRASHREASON=cmnos_thread.c:953:0x8Asserted in:0x100327c:0x8a7588, line#1346
subsys-restart: subsystem_restart_dev(): Restart sequence requested for wlan, restart_level = SYSTEM.
```

```
Kernel panic - not syncing: subsys-restart: Resetting the SoC - wlan crashed.
```

Wi-Fi Firmware Security

Type(1 byte)
Frame Control Field(1 byte)
Duration(2 bytes)
Destination address(6 bytes)
Source address(6 bytes)
BSSID(6 bytes)
Fragment/Sequence number(2 bytes)
Category code(1 byte)
Action id(1 byte)
……

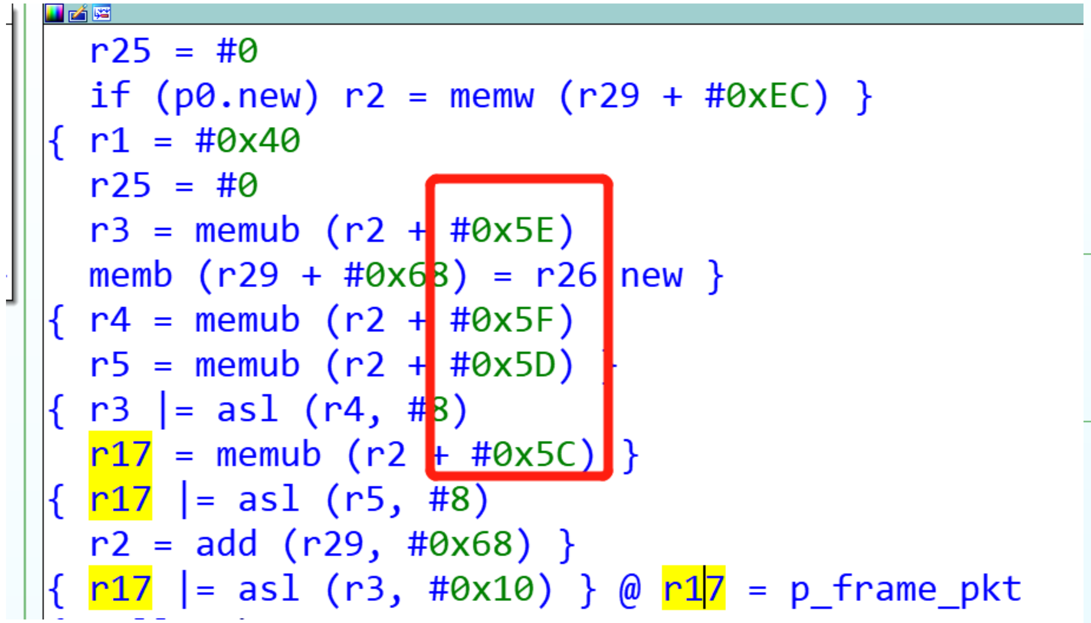


```
No.   Time           Source             Destination         Protocol  Length Info
 4597 29.268951287   Shenzhen_e4:9f:8b  4e:16:fb:cf:7d:7c   802.11        63 Action, SN=176, FN=0, Flags=......

▸ Frame 4597: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on interface wlx1cbfcee49f8b, id 0
▸ Radiotap Header v0, Length 13
▸ 802.11 radio information
▾ IEEE 802.11 Action, Flags: ........
    Type/Subtype: Action (0x000d)
  ▸ Frame Control Field: 0xd000
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: 4e:16:fb:cf:7d:7c (4e:16:fb:cf:7d:7c)
    Destination address: 4e:16:fb:cf:7d:7c (4e:16:fb:cf:7d:7c)
    Transmitter address: Shenzhen_e4:9f:8b (1c:bf:ce:e4:9f:8b)
    Source address: Shenzhen_e4:9f:8b (1c:bf:ce:e4:9f:8b)
    BSS Id: 4e:16:fb:cf:7d:7c (4e:16:fb:cf:7d:7c)
    .... .... .... 0000 = Fragment number: 0
    0000 1011 0000 .... = Sequence number: 176
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters
      Category code: VHT (21)
      VHT Action: Group ID Management (1)
    ▾ Group ID Management: 000000000000000041414141414141414141414141414141
        Membership Status Array: 0000000000000000
        User Position Array: 41414141414141414141414141414141

0000  00 00 0d 00 04 00 02 00  02 00 00 00 00 d0 00 2c   ········ ······,
0010  00 4e 16 fb cf 7d 7c 1c  bf ce e4 9f 8b 4e 16 fb   ·N···}|· ·····N··
0020  cf 7d 7c 00 0b 15 01 00  00 00 00 00 00 00 00 41   ·}|····· ·······A
0030  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41      AAAAAAAA AAAAAAA
```

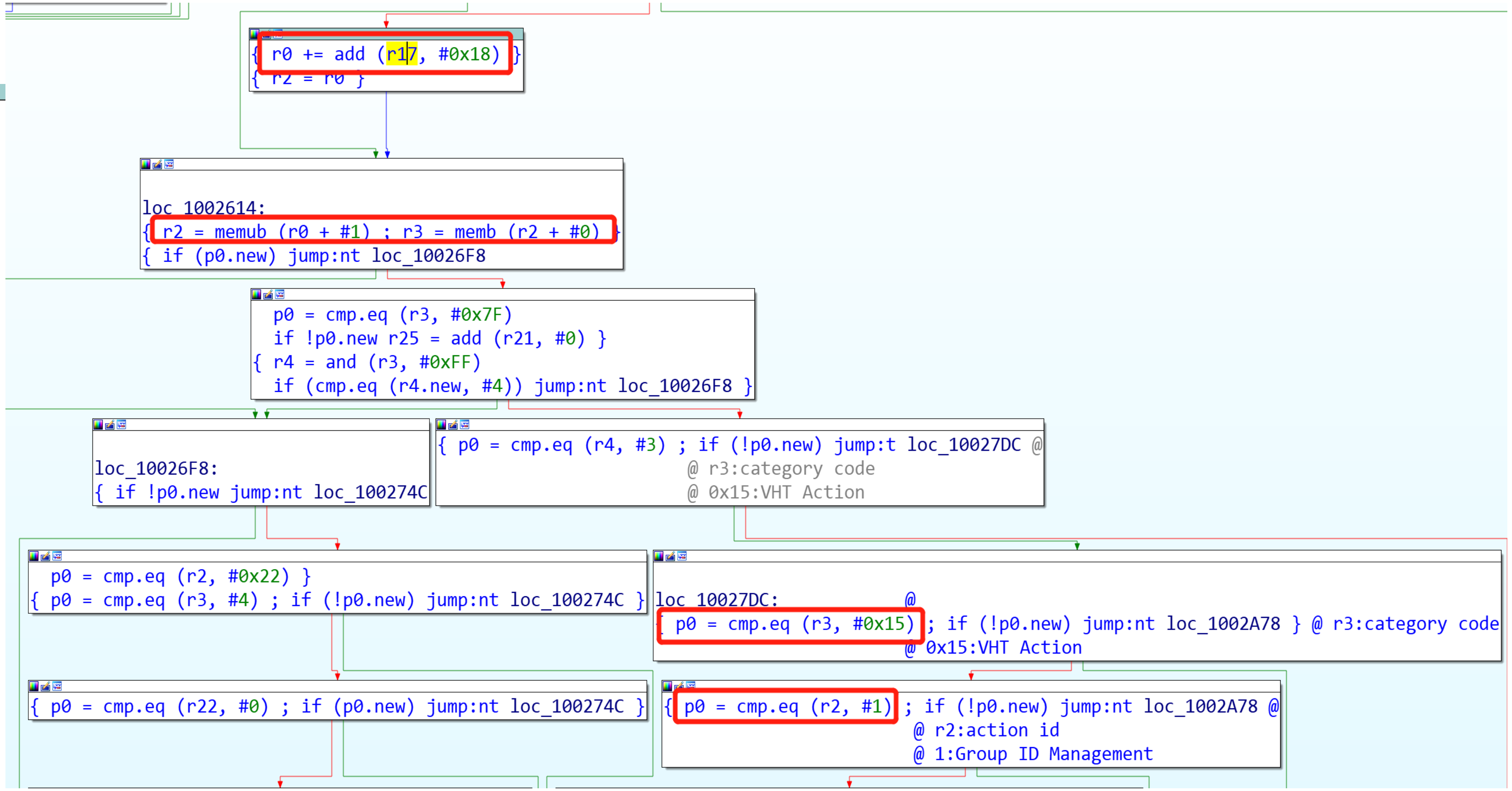
0x18 bytes

Wi-Fi Firmware Security

Finding vulnerabilities in Wi-Fi firmware



```
  r25 = #0
  if (p0.new) r2 = memw (r29 + #0xEC) }
{ r1 = #0x40
  r25 = #0
  r3 = memub (r2 + #0x5E)
  memb (r29 + #0x68) = r26 new }
{ r4 = memub (r2 + #0x5F)
  r5 = memub (r2 + #0x5D)
{ r3 |= asl (r4, #8)
  r17 = memub (r2 + #0x5C) }
{ r17 |= asl (r5, #8)
  r2 = add (r29, #0x68) }
{ r17 |= asl (r3, #0x10) } @ r17 = p_frame_pkt
```



```
{ r0 += add (r17, #0x18) }
{ r2 = r0 }
```

```
loc_1002614:
{ r2 = memub (r0 + #1) ; r3 = memb (r2 + #0) }
{ if (p0.new) jump:nt loc_10026F8
```

```
  p0 = cmp.eq (r3, #0x7F)
  if !p0.new r25 = add (r21, #0) }
{ r4 = and (r3, #0xFF)
  if (cmp.eq (r4.new, #4)) jump:nt loc_10026F8 }
```

```
loc_10026F8:
{ if !p0.new jump:nt loc_100274C
```

```
{ p0 = cmp.eq (r4, #3) ; if (!p0.new) jump:t loc_10027DC @
                                   @ r3:category code
                                   @ 0x15:VHT Action
```

```
  p0 = cmp.eq (r2, #0x22) }
{ p0 = cmp.eq (r3, #4) ; if (!p0.new) jump:nt loc_100274C }
```

```
loc_10027DC:                @
{ p0 = cmp.eq (r3, #0x15) ; if (!p0.new) jump:nt loc_1002A78 } @ r3:category code
                            @ 0x15:VHT Action
```

```
{ p0 = cmp.eq (r22, #0) ; if (p0.new) jump:nt loc_100274C }
```

```
{ p0 = cmp.eq (r2, #1) ; if (!p0.new) jump:nt loc_1002A78 @
                         @ r2:action id
                         @ 1:Group ID Management
```

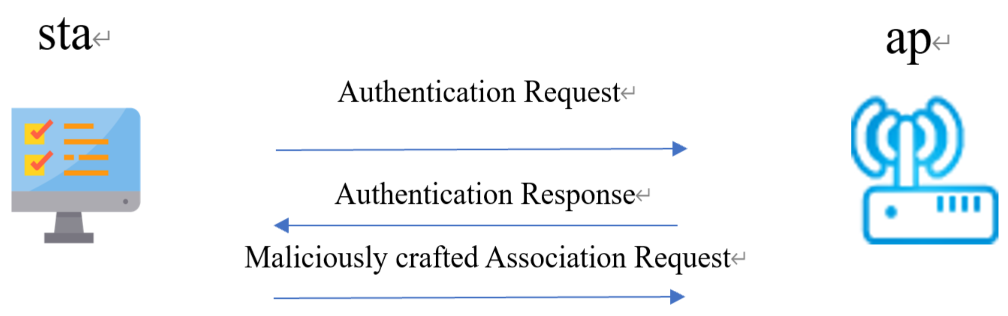
Wi-Fi Firmware Security

Finding vulnerabilities in Wi-Fi firmware

CVE-ID	Device	Disclosure time
CVE-2021-1925	SDM865	2021.5
CVE-2021-1937	SDM865	2021.6
CVE-2021-1938	SDM865	2021.7
CVE-2021-1907	SDM865	2021.7
CVE-2021-1953	SDM670	2021.7

Wi-Fi Firmware Security

CVE-2021-1937



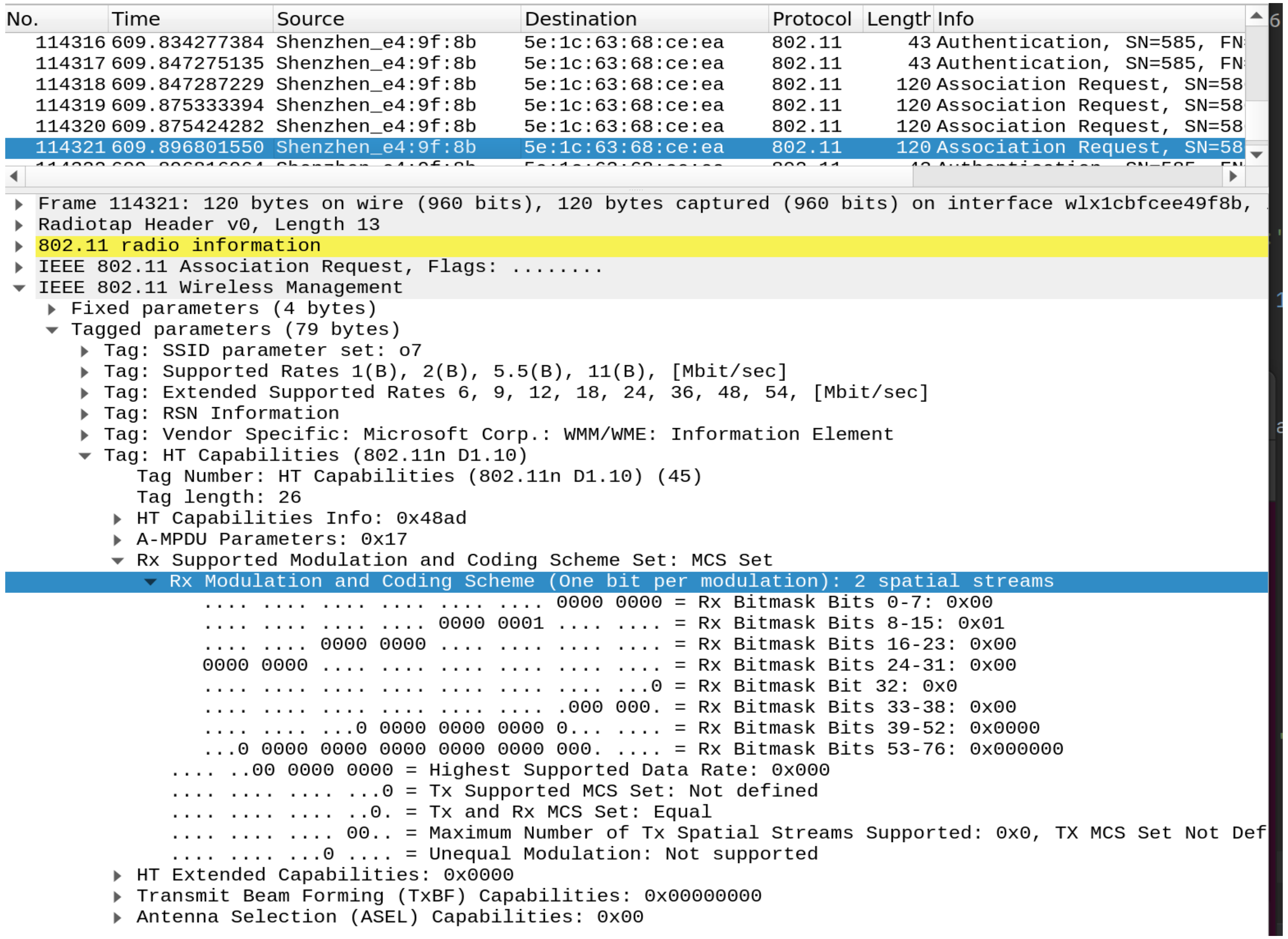
sta↵

ap↵

Authentication Request↵

Authentication Response↵

Maliciously crafted Association Request↵

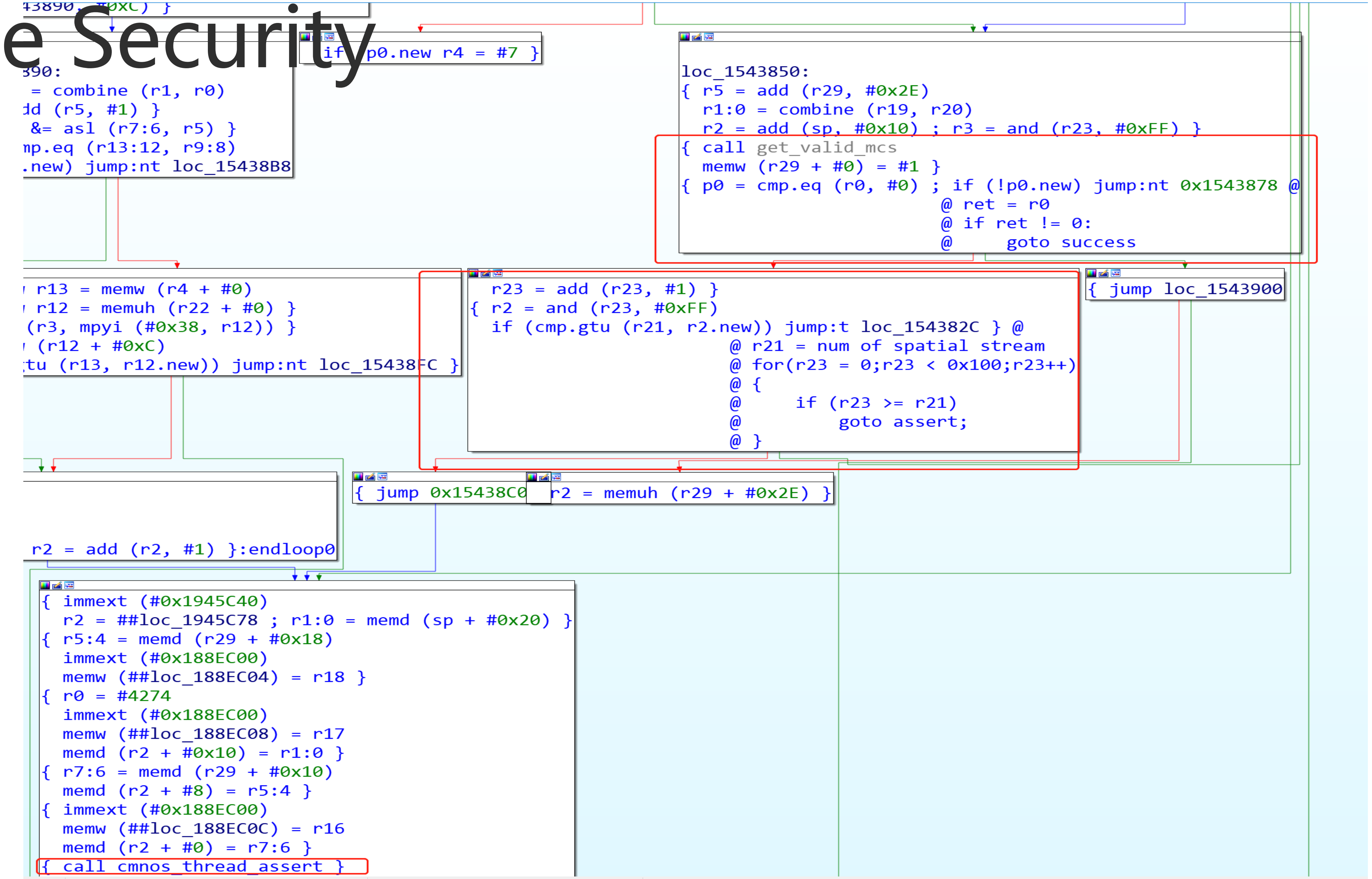


```
No.    Time           Source              Destination         Protocol  Lengtł Info
114316 609.834277384  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11      43 Authentication, SN=585, FN
114317 609.847275135  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11      43 Authentication, SN=585, FN
114318 609.847287229  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11     120 Association Request, SN=58
114319 609.875333394  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11     120 Association Request, SN=58
114320 609.875424282  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11     120 Association Request, SN=58
114321 609.896801550  Shenzhen_e4:9f:8b   5e:1c:63:68:ce:ea   802.11     120 Association Request, SN=58
```

```
▸ Frame 114321: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface wlx1cbfcee49f8b,
▸ Radiotap Header v0, Length 13
▸ 802.11 radio information
▸ IEEE 802.11 Association Request, Flags: ........
▾ IEEE 802.11 Wireless Management
  ▸ Fixed parameters (4 bytes)
  ▾ Tagged parameters (79 bytes)
    ▸ Tag: SSID parameter set: o7
    ▸ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), [Mbit/sec]
    ▸ Tag: Extended Supported Rates 6, 9, 12, 18, 24, 36, 48, 54, [Mbit/sec]
    ▸ Tag: RSN Information
    ▸ Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Information Element
    ▾ Tag: HT Capabilities (802.11n D1.10)
        Tag Number: HT Capabilities (802.11n D1.10) (45)
        Tag length: 26
      ▸ HT Capabilities Info: 0x48ad
      ▸ A-MPDU Parameters: 0x17
      ▾ Rx Supported Modulation and Coding Scheme Set: MCS Set
        ▾ Rx Modulation and Coding Scheme (One bit per modulation): 2 spatial streams
            .... .... .... .... .... .... 0000 0000 = Rx Bitmask Bits 0-7: 0x00
            .... .... .... .... 0000 0001 .... .... = Rx Bitmask Bits 8-15: 0x01
            .... .... 0000 0000 .... .... .... .... = Rx Bitmask Bits 16-23: 0x00
            0000 0000 .... .... .... .... .... .... = Rx Bitmask Bits 24-31: 0x00
            .... .... .... .... .... .... .... ...0 = Rx Bitmask Bit 32: 0x0
            .... .... .... .... .... .... .000 000. = Rx Bitmask Bits 33-38: 0x00
            .... .... ...0 0000 0000 0000 0... .... = Rx Bitmask Bits 39-52: 0x0000
            ...0 0000 0000 0000 0000 0000 000. .... = Rx Bitmask Bits 53-76: 0x000000
          .... ..00 0000 0000 = Highest Supported Data Rate: 0x000
          .... .... .... ...0 = Tx Supported MCS Set: Not defined
          .... .... .... ..0. = Tx and Rx MCS Set: Equal
          .... .... .... 00.. = Maximum Number of Tx Spatial Streams Supported: 0x0, TX MCS Set Not Def
          .... .... ...0 .... = Unequal Modulation: Not supported
      ▸ HT Extended Capabilities: 0x0000
      ▸ Transmit Beam Forming (TxBF) Capabilities: 0x00000000
      ▸ Antenna Selection (ASEL) Capabilities: 0x00
```

Wi-Fi Firmware Security

CVE-2021-1937



```
43890, #0xC) }
```

```
{ if p0.new r4 = #7 }
```

```
390:
 = combine (r1, r0)
dd (r5, #1) }
 &= asl (r7:6, r5) }
mp.eq (r13:12, r9:8)
.new) jump:nt loc_15438B8
```

```
loc_1543850:
{ r5 = add (r29, #0x2E)
  r1:0 = combine (r19, r20)
  r2 = add (sp, #0x10) ; r3 = and (r23, #0xFF) }
{ call get_valid_mcs
  memw (r29 + #0) = #1 }
{ p0 = cmp.eq (r0, #0) ; if (!p0.new) jump:nt 0x1543878 @
                                    @ ret = r0
                                    @ if ret != 0:
                                    @     goto success
```

```
 r13 = memw (r4 + #0)
 r12 = memuh (r22 + #0) }
 (r3, mpyi (#0x38, r12)) }
 (r12 + #0xC)
tu (r13, r12.new)) jump:nt loc_15438FC }
```

```
  r23 = add (r23, #1) }
{ r2 = and (r23, #0xFF)
  if (cmp.gtu (r21, r2.new)) jump:t loc_154382C } @
                              @ r21 = num of spatial stream
                              @ for(r23 = 0;r23 < 0x100;r23++)
                              @ {
                              @     if (r23 >= r21)
                              @         goto assert;
                              @ }
```

```
{ jump loc_1543900
```

```
{ jump 0x15438C0
```

```
r2 = memuh (r29 + #0x2E) }
```

```
 r2 = add (r2, #1) }:endloop0
```

```
{ immext (#0x1945C40)
  r2 = ##loc_1945C78 ; r1:0 = memd (sp + #0x20) }
{ r5:4 = memd (r29 + #0x18)
  immext (#0x188EC00)
  memw (##loc_188EC04) = r18 }
{ r0 = #4274
  immext (#0x188EC00)
  memw (##loc_188EC08) = r17
  memd (r2 + #0x10) = r1:0 }
{ r7:6 = memd (r29 + #0x10)
  memd (r2 + #8) = r5:4 }
{ immext (#0x188EC00)
  memw (##loc_188EC0C) = r16
  memd (r2 + #0) = r7:6 }
{ call cmnos_thread_assert }
```

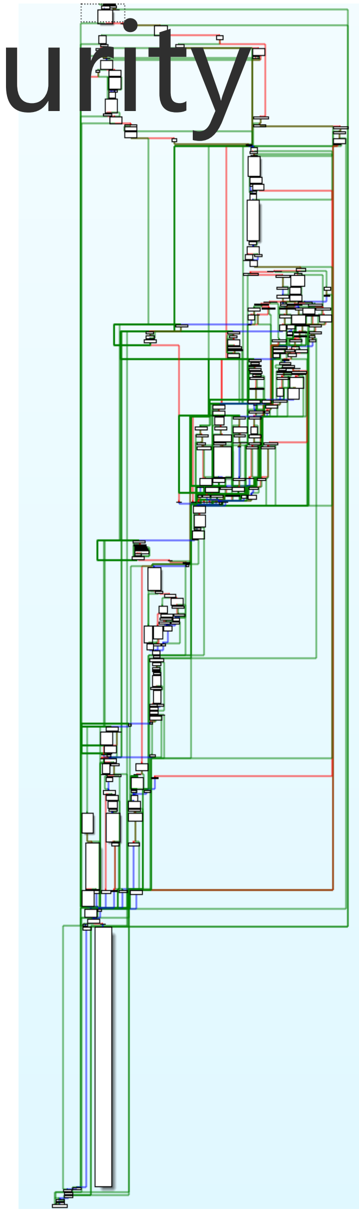
rcGetLowestValidTxMcsForBW

Wi-Fi Firmware Security

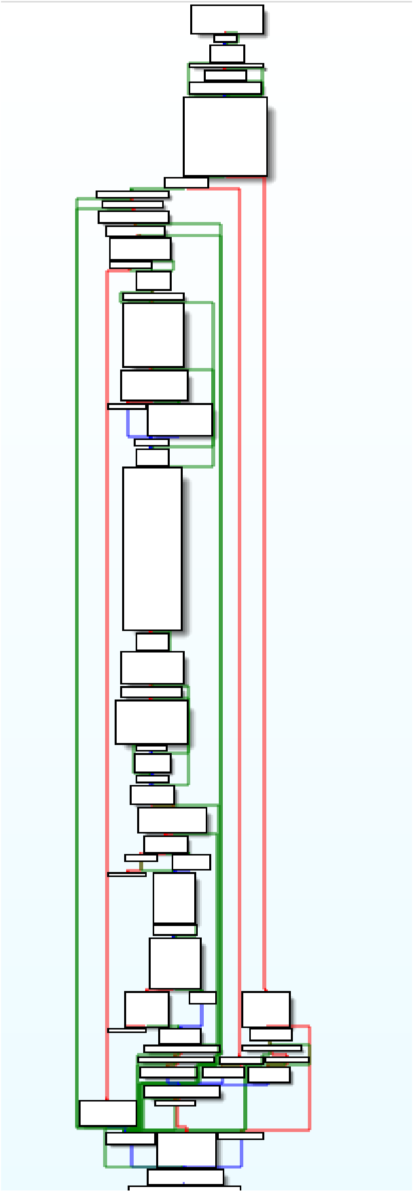
New vulnerability
in Wi-Fi firmware

CVE-2021-1925

wlan_txbfee_parse_gid

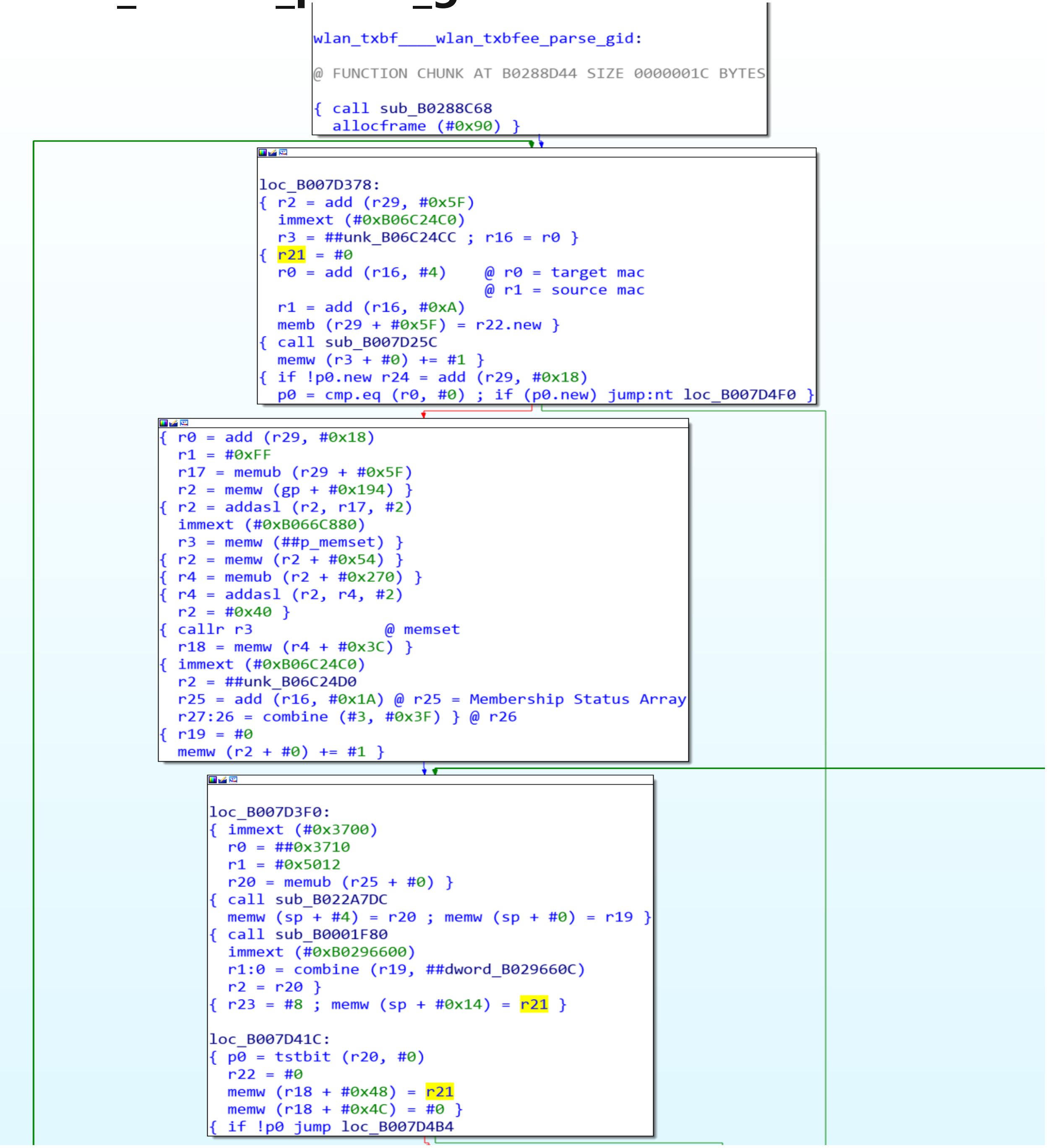


sdm865 wlan_mgmt_rx_frame_handler



sdm845 wlan_mgmt_rx_frame_handler

wlan_txbfee_parse_gid



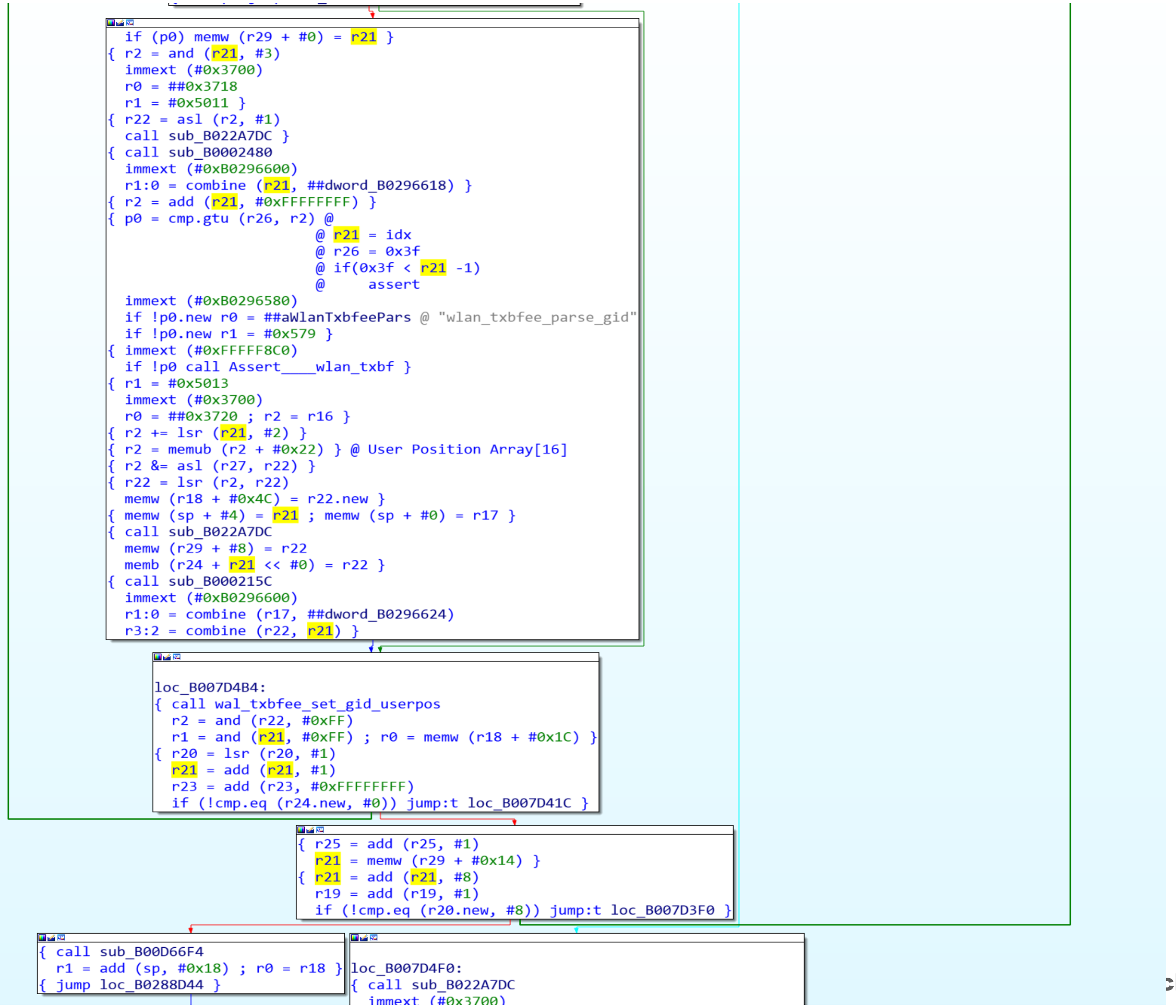
```
wlan_txbf____wlan_txbfee_parse_gid:

@ FUNCTION CHUNK AT B0288D44 SIZE 0000001C BYTES

{ call sub_B0288C68
  allocframe (#0x90) }
```

```
loc_B007D378:
{ r2 = add (r29, #0x5F)
  immext (#0xB06C24C0)
  r3 = ##unk_B06C24CC ; r16 = r0 }
{ r21 = #0
  r0 = add (r16, #4)     @ r0 = target mac
                         @ r1 = source mac
  r1 = add (r16, #0xA)
  memb (r29 + #0x5F) = r22.new }
{ call sub_B007D25C
  memw (r3 + #0) += #1 }
{ if !p0.new r24 = add (r29, #0x18)
  p0 = cmp.eq (r0, #0) ; if (p0.new) jump:nt loc_B007D4F0 }
```

```
{ r0 = add (r29, #0x18)
  r1 = #0xFF
  r17 = memub (r29 + #0x5F)
  r2 = memw (gp + #0x194) }
{ r2 = addasl (r2, r17, #2)
  immext (#0xB066C880)
  r3 = memw (##p_memset) }
{ r2 = memw (r2 + #0x54) }
{ r4 = memub (r2 + #0x270) }
{ r4 = addasl (r2, r4, #2)
  r2 = #0x40 }
{ callr r3               @ memset
  r18 = memw (r4 + #0x3C) }
{ immext (#0xB06C24C0)
  r2 = ##unk_B06C24D0
  r25 = add (r16, #0x1A) @ r25 = Membership Status Array
  r27:26 = combine (#3, #0x3F) } @ r26
{ r19 = #0
  memw (r2 + #0) += #1 }
```

```
loc_B007D3F0:
{ immext (#0x3700)
  r0 = ##0x3710
  r1 = #0x5012
  r20 = memub (r25 + #0) }
{ call sub_B022A7DC
  memw (sp + #4) = r20 ; memw (sp + #0) = r19 }
{ call sub_B0001F80
  immext (#0xB0296600)
  r1:0 = combine (r19, ##dword_B029660C)
  r2 = r20 }
{ r23 = #8 ; memw (sp + #0x14) = r21 }

loc_B007D41C:
{ p0 = tstbit (r20, #0)
  r22 = #0
  memw (r18 + #0x48) = r21
  memw (r18 + #0x4C) = #0 }
{ if !p0 jump loc_B007D4B4
```

CVE-2021-1925



```
  if (p0) memw (r29 + #0) = r21 }
{ r2 = and (r21, #3)
  immext (#0x3700)
  r0 = ##0x3718
  r1 = #0x5011 }
{ r22 = asl (r2, #1)
  call sub_B022A7DC }
{ call sub_B0002480
  immext (#0xB0296600)
  r1:0 = combine (r21, ##dword_B0296618) }
{ r2 = add (r21, #0xFFFFFFFF) }
{ p0 = cmp.gtu (r26, r2) @
                        @ r21 = idx
                        @ r26 = 0x3f
                        @ if(0x3f < r21 -1)
                        @     assert
  immext (#0xB0296580)
  if !p0.new r0 = ##aWlanTxbfeePars @ "wlan_txbfee_parse_gid"
  if !p0.new r1 = #0x579 }
{ immext (#0xFFFFF8C0)
  if !p0 call Assert____wlan_txbf }
{ r1 = #0x5013
  immext (#0x3700)
  r0 = ##0x3720 ; r2 = r16 }
{ r2 += lsr (r21, #2) }
{ r2 = memub (r2 + #0x22) } @ User Position Array[16]
{ r2 &= asl (r27, r22) }
{ r22 = lsr (r2, r22)
  memw (r18 + #0x4C) = r22.new }
{ memw (sp + #4) = r21 ; memw (sp + #0) = r17 }
{ call sub_B022A7DC
  memw (r29 + #8) = r22
  memb (r24 + r21 << #0) = r22 }
{ call sub_B000215C
  immext (#0xB0296600)
  r1:0 = combine (r17, ##dword_B0296624)
  r3:2 = combine (r22, r21) }
```

```
loc_B007D4B4:
{ call wal_txbfee_set_gid_userpos
  r2 = and (r22, #0xFF)
  r1 = and (r21, #0xFF) ; r0 = memw (r18 + #0x1C) }
{ r20 = lsr (r20, #1)
  r21 = add (r21, #1)
  r23 = add (r23, #0xFFFFFFFF)
  if (!cmp.eq (r24.new, #0)) jump:t loc_B007D41C }
```

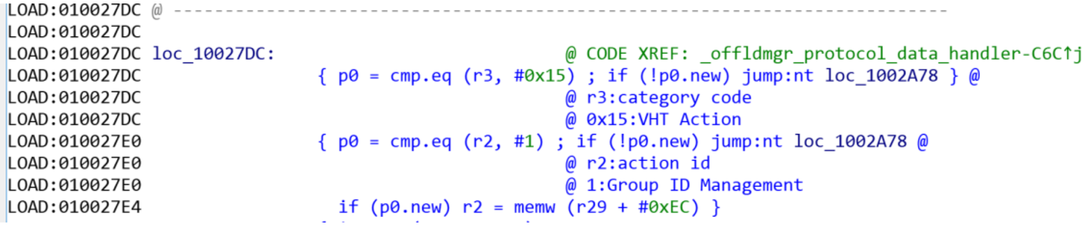
```
{ r25 = add (r25, #1)
  r21 = memw (r29 + #0x14) }
{ r21 = add (r21, #8)
  r19 = add (r19, #1)
  if (!cmp.eq (r20.new, #8)) jump:t loc_B007D3F0 }
```

```
{ call sub_B00D66F4
  r1 = add (sp, #0x18) ; r0 = r18 }
{ jump loc_B0288D44 }
```

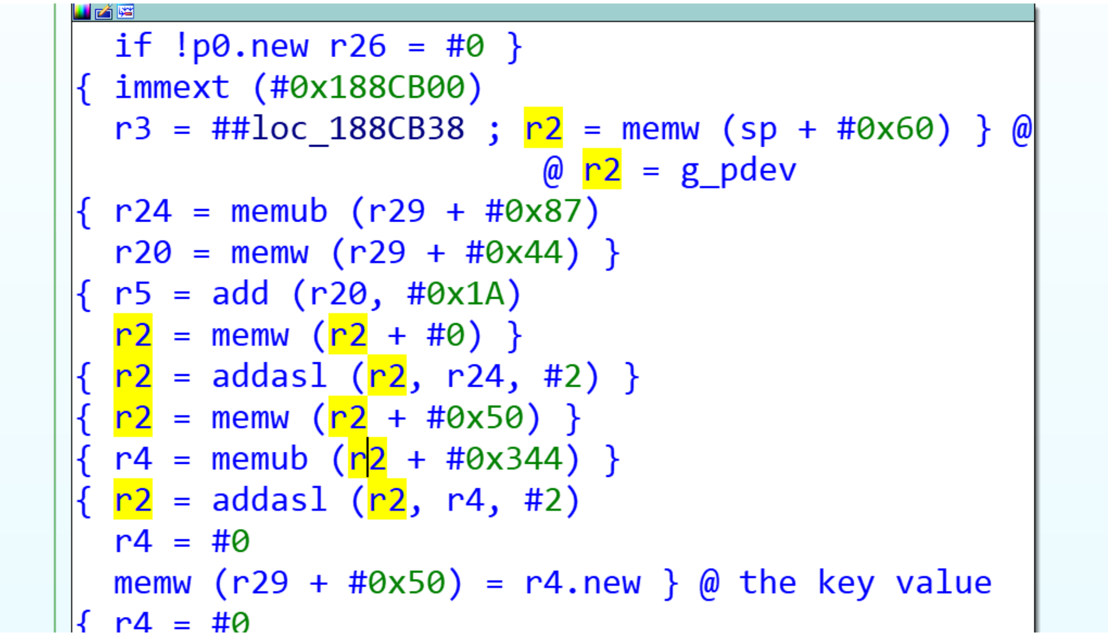
```
loc_B007D4F0:
{ call sub_B022A7DC
  immext (#0x3700)
```

Wi-Fi Firmware Security

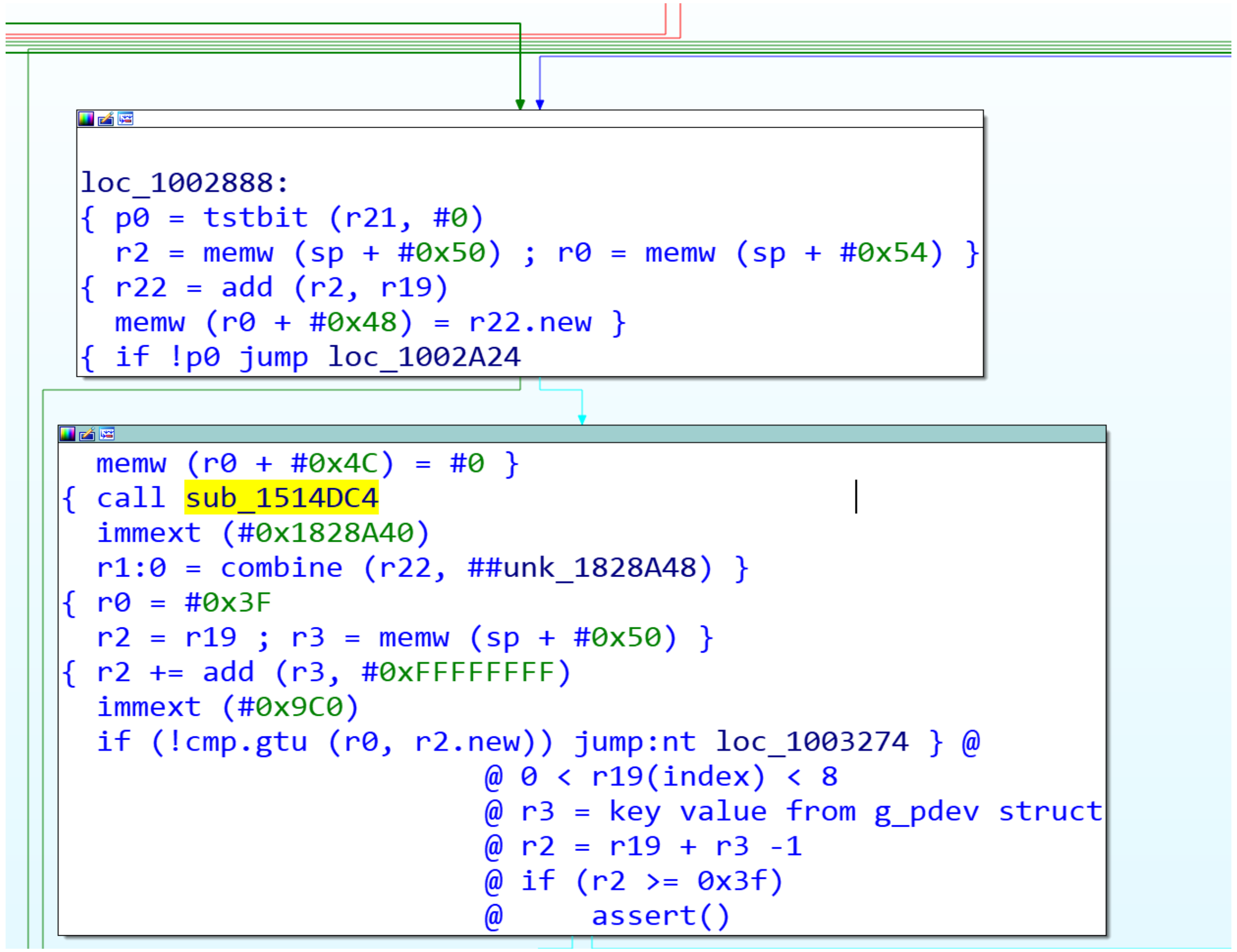
CVE-2021-1925



```
LOAD:010027DC @ ---------------------------------------------------------------
LOAD:010027DC
LOAD:010027DC loc_10027DC:                             @ CODE XREF: _offldmgr_protocol_data_handler-C6C↑j
LOAD:010027DC                { p0 = cmp.eq (r3, #0x15) ; if (!p0.new) jump:nt loc_1002A78 } @
LOAD:010027DC                                          @ r3:category code
LOAD:010027DC                                          @ 0x15:VHT Action
LOAD:010027E0                { p0 = cmp.eq (r2, #1) ; if (!p0.new) jump:nt loc_1002A78 @
LOAD:010027E0                                          @ r2:action id
LOAD:010027E0                                          @ 1:Group ID Management
LOAD:010027E4                  if (p0.new) r2 = memw (r29 + #0xEC) }
```



```
  if !p0.new r26 = #0 }
{ immext (#0x188CB00)
  r3 = ##loc_188CB38 ; r2 = memw (sp + #0x60) } @
                        @ r2 = g_pdev
{ r24 = memub (r29 + #0x87)
  r20 = memw (r29 + #0x44) }
{ r5 = add (r20, #0x1A)
  r2 = memw (r2 + #0) }
{ r2 = addasl (r2, r24, #2) }
{ r2 = memw (r2 + #0x50) }
{ r4 = memub (r2 + #0x344) }
{ r2 = addasl (r2, r4, #2)
  r4 = #0
  memw (r29 + #0x50) = r4.new } @ the key value
{ r4 = #0
```



```
loc_1002888:
{ p0 = tstbit (r21, #0)
  r2 = memw (sp + #0x50) ; r0 = memw (sp + #0x54) }
{ r22 = add (r2, r19)
  memw (r0 + #0x48) = r22.new }
{ if !p0 jump loc_1002A24
```

```
  memw (r0 + #0x4C) = #0 }
{ call sub_1514DC4
  immext (#0x1828A40)
  r1:0 = combine (r22, ##unk_1828A48) }
{ r0 = #0x3F
  r2 = r19 ; r3 = memw (sp + #0x50) }
{ r2 += add (r3, #0xFFFFFFFF)
  immext (#0x9C0)
  if (!cmp.gtu (r0, r2.new)) jump:nt loc_1003274 } @
                                     @ 0 < r19(index) < 8
                                     @ r3 = key value from g_pdev struct
                                     @ r2 = r19 + r3 -1
                                     @ if (r2 >= 0x3f)
                                     @     assert()
```

Wi-Fi Firmware Security

Qualcomm Wi-Fi firmware mitigation

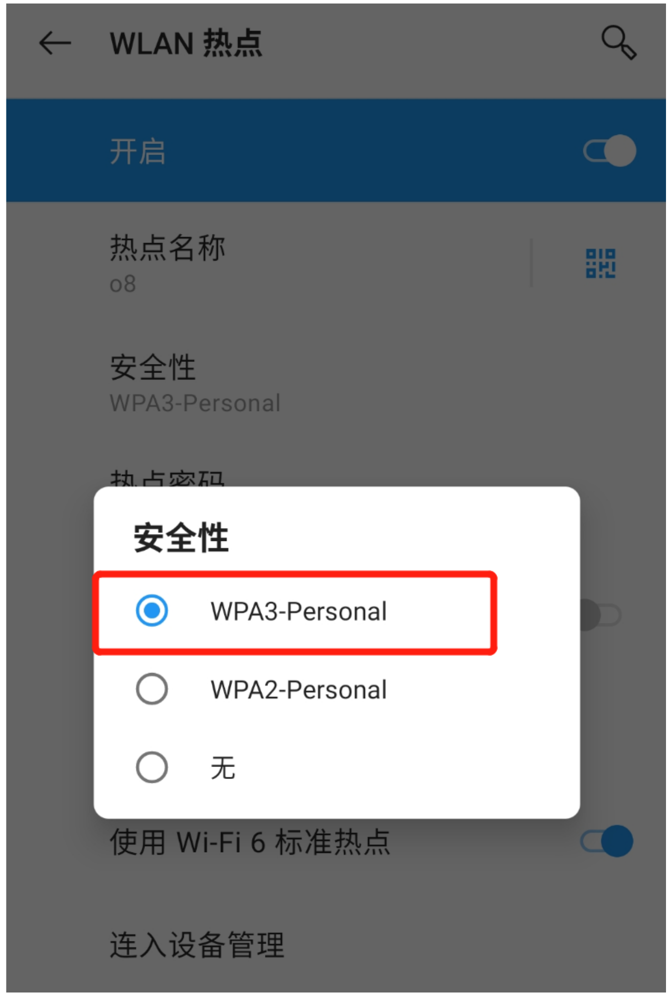
Stack cookie、Heap cookie, W^X

No ASLR,CFI

Wi-Fi Firmware Security

New features and functions make WLAN drivers and firmware constantly changing.

Code refactoring



WLAN 热点

开启

热点名称
o8

安全性
WPA3-Personal

热点密码

安全性

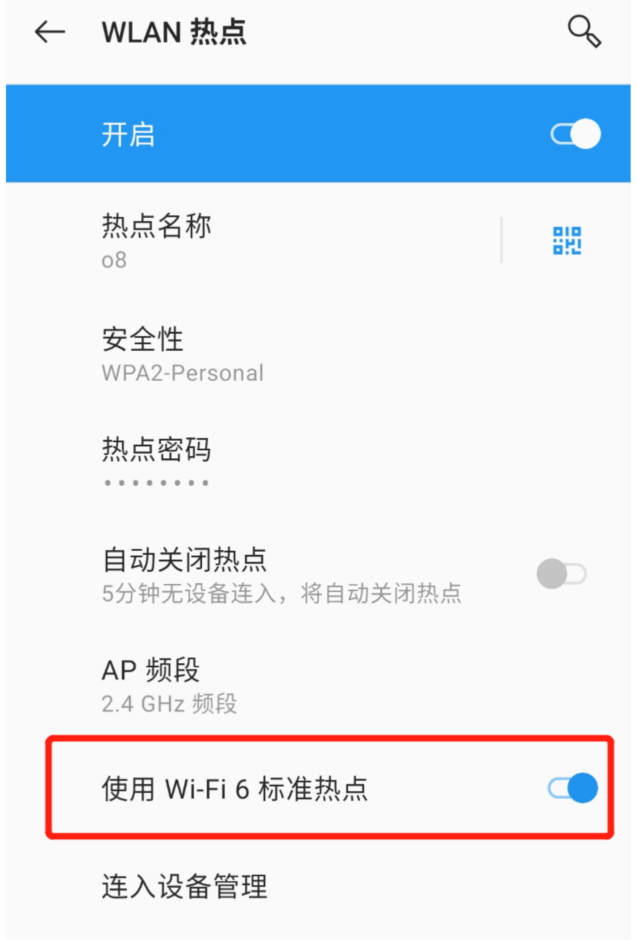
● WPA3-Personal

○ WPA2-Personal

○ 无

使用 Wi-Fi 6 标准热点

连入设备管理



WLAN 热点

开启

热点名称
o8

安全性
WPA2-Personal

热点密码
••••••••

自动关闭热点
5分钟无设备连入，将自动关闭热点

AP 频段
2.4 GHz 频段

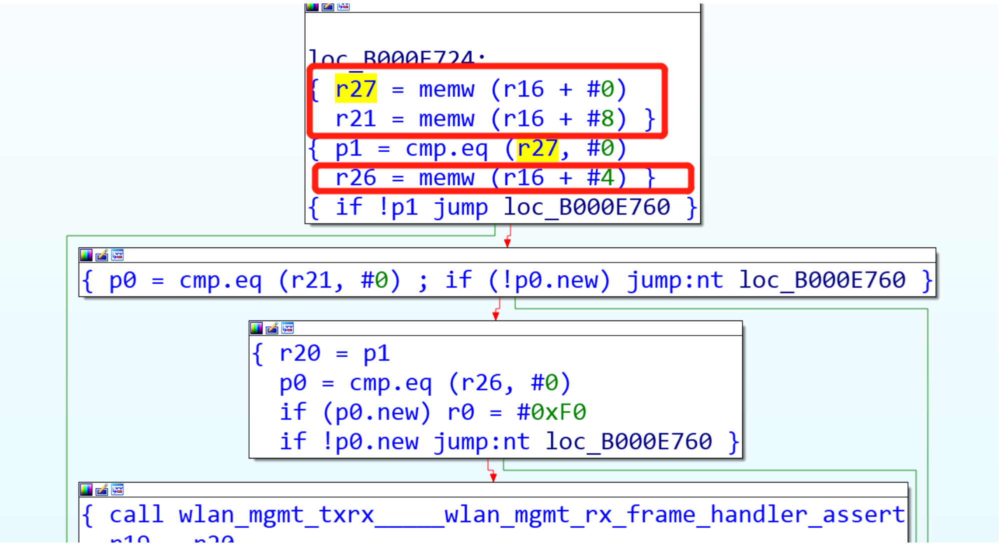
使用 Wi-Fi 6 标准热点

连入设备管理

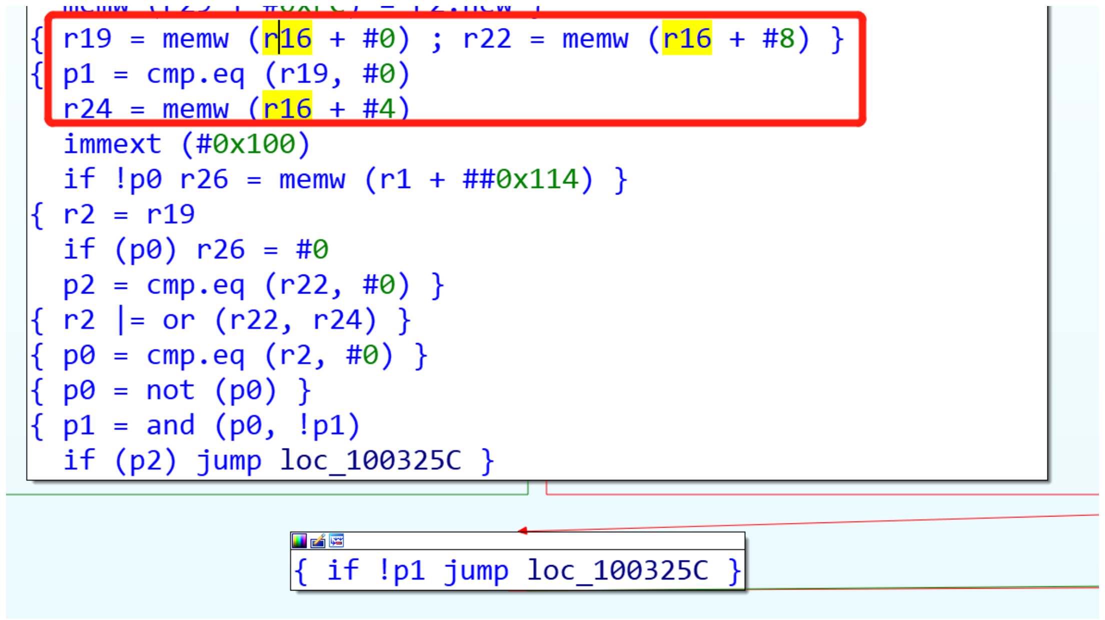
Wi-Fi Firmware Security

New features and functions make WLAN drivers and firmware constantly changing.

Code refactoring



```
loc_B000E724:
{ r27 = memw (r16 + #0)
  r21 = memw (r16 + #8) }
{ p1 = cmp.eq (r27, #0)
  r26 = memw (r16 + #4) }
{ if !p1 jump loc_B000E760 }

{ p0 = cmp.eq (r21, #0) ; if (!p0.new) jump:nt loc_B000E760 }

{ r20 = p1
  p0 = cmp.eq (r26, #0)
  if (p0.new) r0 = #0xF0
  if !p0.new jump:nt loc_B000E760 }

{ call wlan_mgmt_txrx_____wlan_mgmt_rx_frame_handler_assert
```



```
{ r19 = memw (r16 + #0) ; r22 = memw (r16 + #8) }
{ p1 = cmp.eq (r19, #0)
  r24 = memw (r16 + #4)
  immext (#0x100)
  if !p0 r26 = memw (r1 + ##0x114) }
{ r2 = r19
  if (p0) r26 = #0
  p2 = cmp.eq (r22, #0) }
{ r2 |= or (r22, r24) }
{ p0 = cmp.eq (r2, #0) }
{ p0 = not (p0) }
{ p1 = and (p0, !p1)
  if (p2) jump loc_100325C }

{ if !p1 jump loc_100325C }
```

Wi-Fi Firmware Security

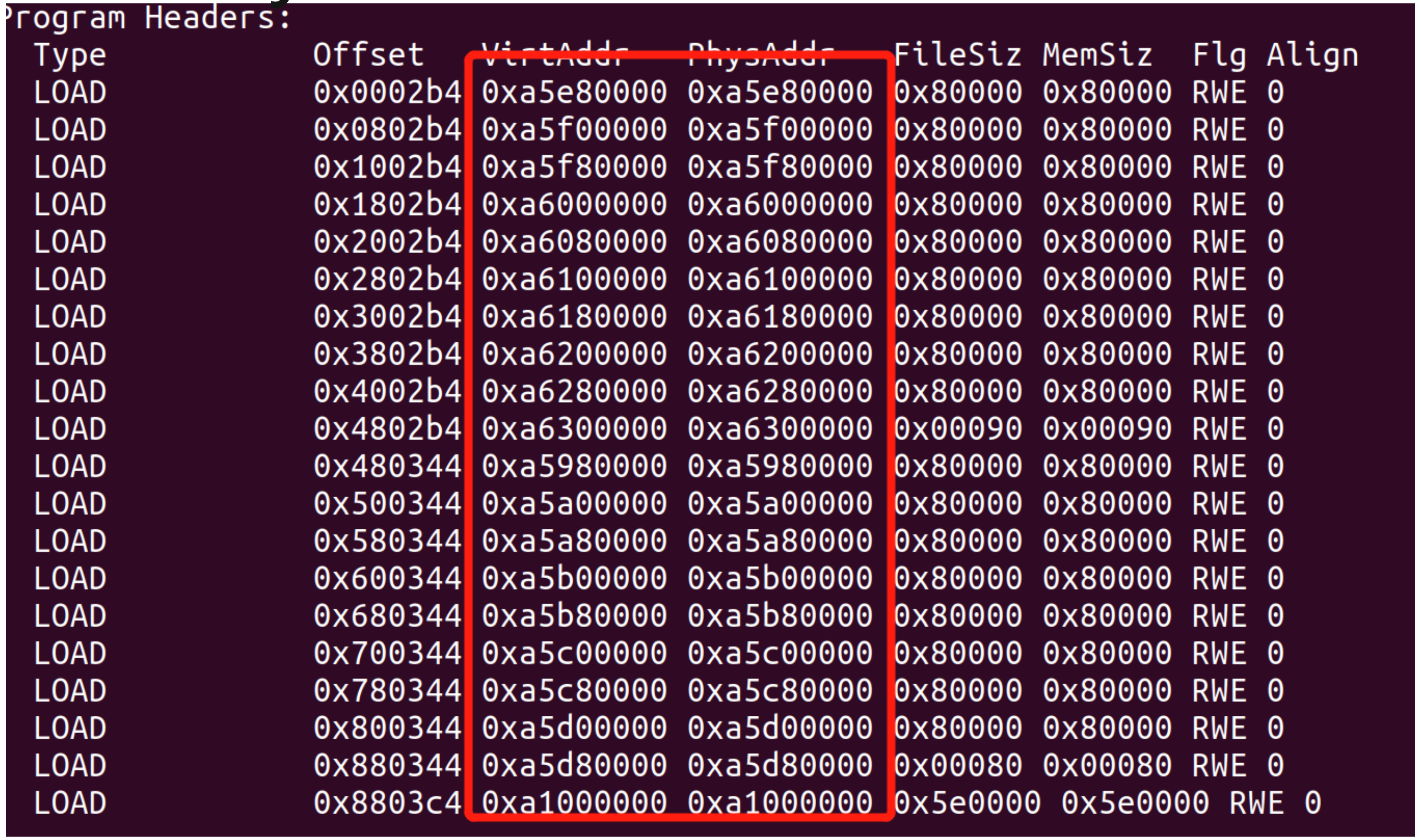
Analyze firmware memory

ramdump_wlan.elf, It contains the memory of Wi-Fi

heap block and code in RAM
are mixed together

0xa1000000—0xa15e0000

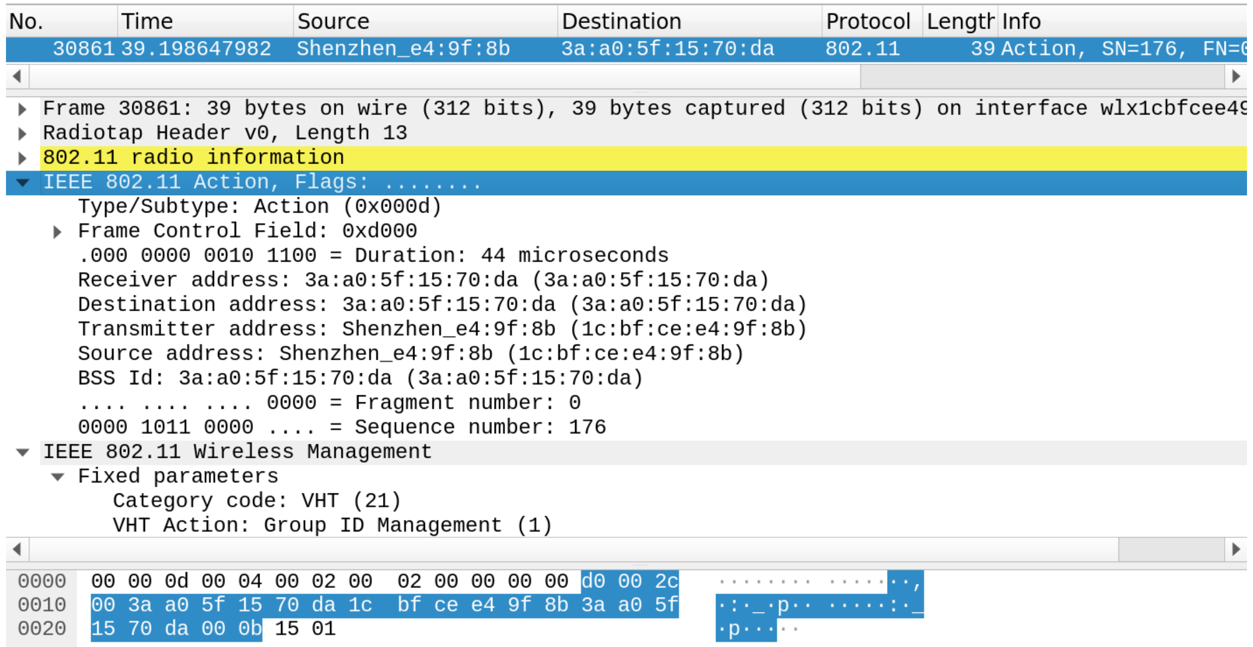
0xa5980000—0xa6300000



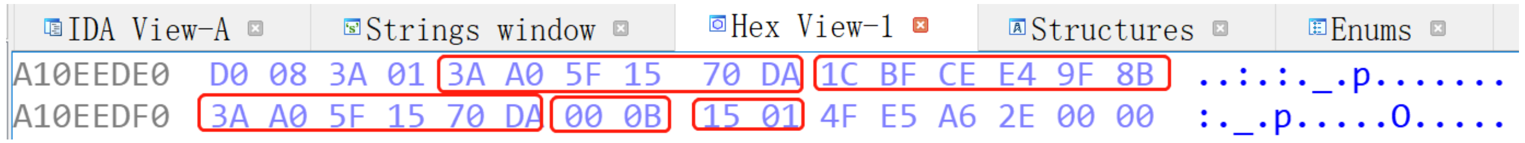
```
Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  LOAD           0x0002b4 0xa5e80000 0xa5e80000 0x80000 0x80000 RWE 0
  LOAD           0x0802b4 0xa5f00000 0xa5f00000 0x80000 0x80000 RWE 0
  LOAD           0x1002b4 0xa5f80000 0xa5f80000 0x80000 0x80000 RWE 0
  LOAD           0x1802b4 0xa6000000 0xa6000000 0x80000 0x80000 RWE 0
  LOAD           0x2002b4 0xa6080000 0xa6080000 0x80000 0x80000 RWE 0
  LOAD           0x2802b4 0xa6100000 0xa6100000 0x80000 0x80000 RWE 0
  LOAD           0x3002b4 0xa6180000 0xa6180000 0x80000 0x80000 RWE 0
  LOAD           0x3802b4 0xa6200000 0xa6200000 0x80000 0x80000 RWE 0
  LOAD           0x4002b4 0xa6280000 0xa6280000 0x80000 0x80000 RWE 0
  LOAD           0x4802b4 0xa6300000 0xa6300000 0x00090 0x00090 RWE 0
  LOAD           0x480344 0xa5980000 0xa5980000 0x80000 0x80000 RWE 0
  LOAD           0x500344 0xa5a00000 0xa5a00000 0x80000 0x80000 RWE 0
  LOAD           0x580344 0xa5a80000 0xa5a80000 0x80000 0x80000 RWE 0
  LOAD           0x600344 0xa5b00000 0xa5b00000 0x80000 0x80000 RWE 0
  LOAD           0x680344 0xa5b80000 0xa5b80000 0x80000 0x80000 RWE 0
  LOAD           0x700344 0xa5c00000 0xa5c00000 0x80000 0x80000 RWE 0
  LOAD           0x780344 0xa5c80000 0xa5c80000 0x80000 0x80000 RWE 0
  LOAD           0x800344 0xa5d00000 0xa5d00000 0x80000 0x80000 RWE 0
  LOAD           0x880344 0xa5d80000 0xa5d80000 0x00080 0x00080 RWE 0
  LOAD           0x8803c4 0xa1000000 0xa1000000 0x5e0000 0x5e0000 RWE 0
```

Wi-Fi Firmware Security

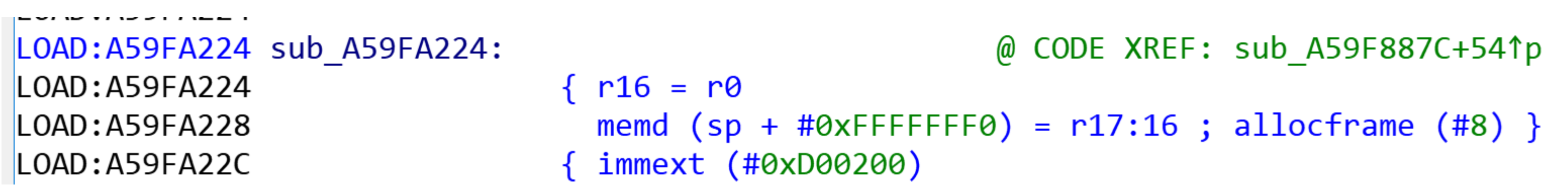
Analyze firmware memory



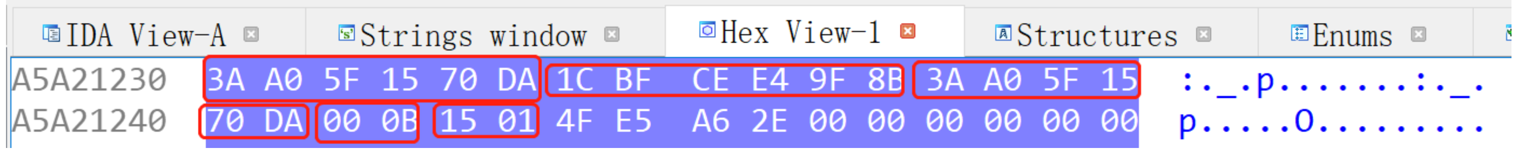
```
No.   Time          Source              Destination          Protocol  Length Info
30861 39.198647982  Shenzhen_e4:9f:8b   3a:a0:5f:15:70:da    802.11        39 Action, SN=176, FN=(

▸ Frame 30861: 39 bytes on wire (312 bits), 39 bytes captured (312 bits) on interface wlx1cbfcee49
▸ Radiotap Header v0, Length 13
▸ 802.11 radio information
▾ IEEE 802.11 Action, Flags: ........
    Type/Subtype: Action (0x000d)
  ▸ Frame Control Field: 0xd000
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: 3a:a0:5f:15:70:da (3a:a0:5f:15:70:da)
    Destination address: 3a:a0:5f:15:70:da (3a:a0:5f:15:70:da)
    Transmitter address: Shenzhen_e4:9f:8b (1c:bf:ce:e4:9f:8b)
    Source address: Shenzhen_e4:9f:8b (1c:bf:ce:e4:9f:8b)
    BSS Id: 3a:a0:5f:15:70:da (3a:a0:5f:15:70:da)
    .... .... .... 0000 = Fragment number: 0
    0000 1011 0000 .... = Sequence number: 176
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters
      Category code: VHT (21)
      VHT Action: Group ID Management (1)

0000  00 00 0d 00 04 00 02 00  02 00 00 00 00 d0 00 2c   ········ ·····.·,
0010  00 3a a0 5f 15 70 da 1c  bf ce e4 9f 8b 3a a0 5f   ·:·_·p·· ·····:·_
0020  15 70 da 00 0b 15 01                               ·p·····
```



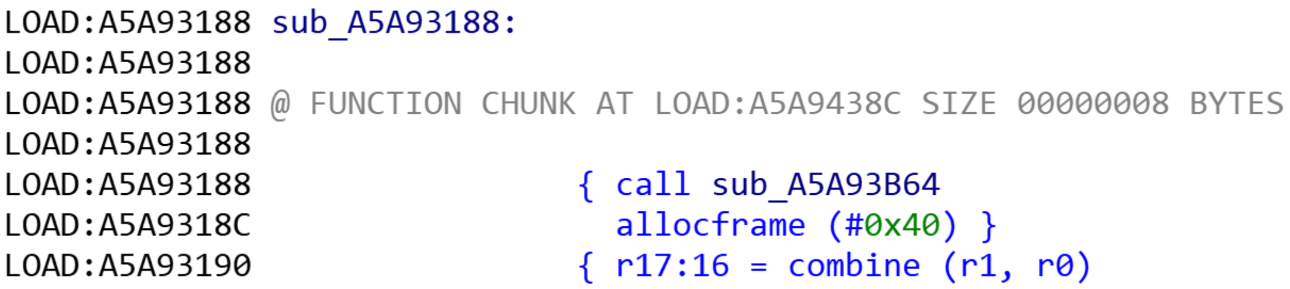
```
A10EEDE0  D0 08 3A 01 3A A0 5F 15  70 DA 1C BF CE E4 9F 8B  ..:.:._.p.......
A10EEDF0  3A A0 5F 15 70 DA 00 0B  15 01 4F E5 A6 2E 00 00  :._.p.....O.....
```



```
LOAD:A59FA224 sub_A59FA224:                           @ CODE XREF: sub_A59F887C+54↑p
LOAD:A59FA224                 { r16 = r0
LOAD:A59FA228                   memd (sp + #0xFFFFFFF0) = r17:16 ; allocframe (#8) }
LOAD:A59FA22C                 { immext (#0xD00200)
```



```
A5A21230  3A A0 5F 15 70 DA 1C BF  CE E4 9F 8B 3A A0 5F 15  :._.p.......:._.
A5A21240  70 DA 00 0B 15 01 4F E5  A6 2E 00 00 00 00 00 00  p.....O.........
```



```
LOAD:A5A93188 sub_A5A93188:
LOAD:A5A93188
LOAD:A5A93188 @ FUNCTION CHUNK AT LOAD:A5A9438C SIZE 00000008 BYTES
LOAD:A5A93188
LOAD:A5A93188                 { call sub_A5A93B64
LOAD:A5A9318C                   allocframe (#0x40) }
LOAD:A5A93190                 { r17:16 = combine (r1, r0)
```

Future work

Find memory write vulnerabilities in the firmware

Research new features

Optimize debugging method

Conclusion

There are still great risks in the security of Wi-Fi driver, but the mitigation make the attack more difficult

The security of Wi-Fi firmware is weaker and more and more attention has been paid to it

The vulnerability cannot be completely eliminated, and the research on Wi-Fi security is a continuous process

Thanks