DEVCORE

什麼！
原來連 Wi-Fi 不需要密碼！？

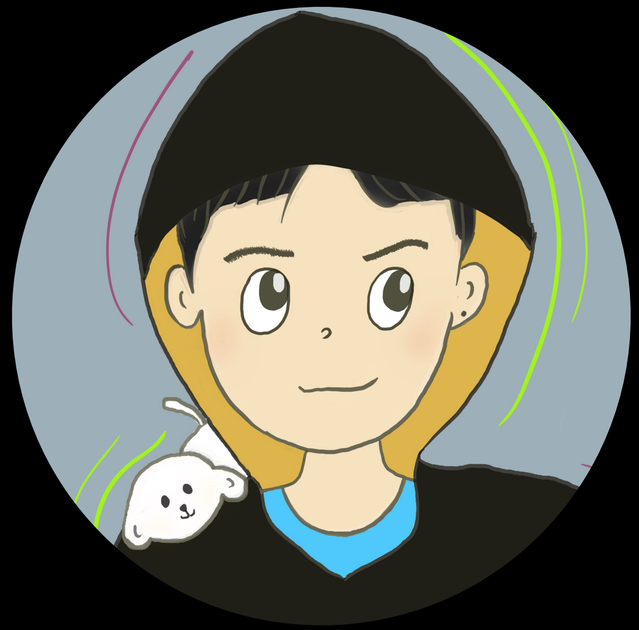
Wei Che Kao (@xiaobye_tw)

weichekao@devco.re

DEVCORE Conference 2026 | 2026.03.14

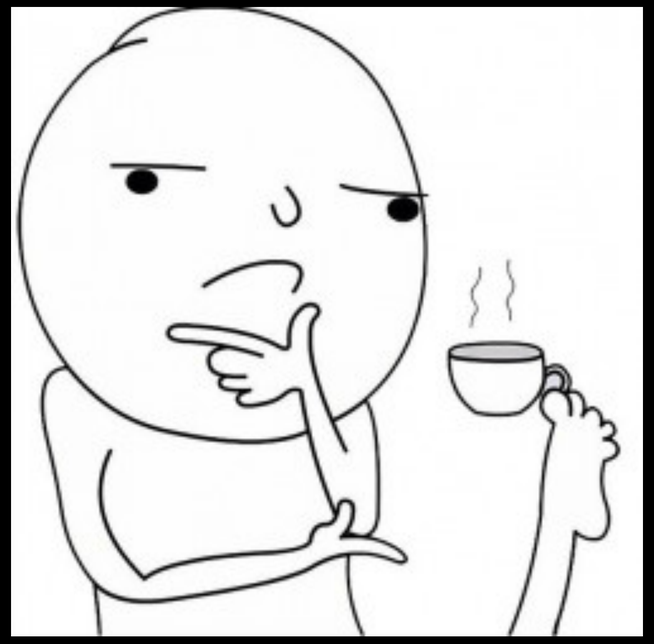
我是誰？

- Xiaobye (@xiaobye_tw)
- DEVCORE 資安研究員
- 略懂
  - Wireless Security
  - Virtual Machine
- 講過
  - USENIX Security, Hexacon, Hardwear.io



DEVCORE

變成研究員了，該研究啥呢？



DEVCORE

不是第1次！這款監視器13000使用者房間全被「看光光」 陷資安危機

全球超過四萬台監視攝影機暴露於網路可遭遠端入侵

上百萬台光纖路由器爆漏洞，變更網址即可避開認證機制

FBI緊急警告！舊路由器被駭有致命漏洞，速查危險WiFi機型和設定

網路銀行存款人間蒸發? 原來是家用路由器出現安全漏洞

你用它上網，我用它進你內網

知名電信商設備遠端代碼執行漏洞

🍊 Orange Tsai (@orange_8361)

DEVCORE

DEVCORE

第一次打 Pwn2Own 就 SOHO Smashup 是不是搞錯了什麼？

張書銘 (LJP Chang) / 彭建霖 (YingMuo Peng)

戴夫寇爾股份有限公司
research@devco.re

DEVCORE
SECURITY CONSULTING

From Zero to Hero:
從零開始的 Pwn2Own 奪冠之路

蔡政達 (Orange Tsai) / 楊安傑 (Angelboy Yang)

戴夫寇爾股份有限公司

research@devco.re

2023.03.11

DEVCORE Conference

只需一次 API 呼叫的致命一擊
從硬體逆向到突破保護機制的精準攻擊

DEVCORE
SECURITY CONSULTING

DEVCORE

寫作 Sync，唸作 Shell ~#

Pumpkin & Orange

戴夫寇爾股份有限公司
pumpkin@devco.re
orange@devco.re

DEVCORE CONFERENCE 2025

那我也來挑一台打？

DEVCORE

DEVCORE

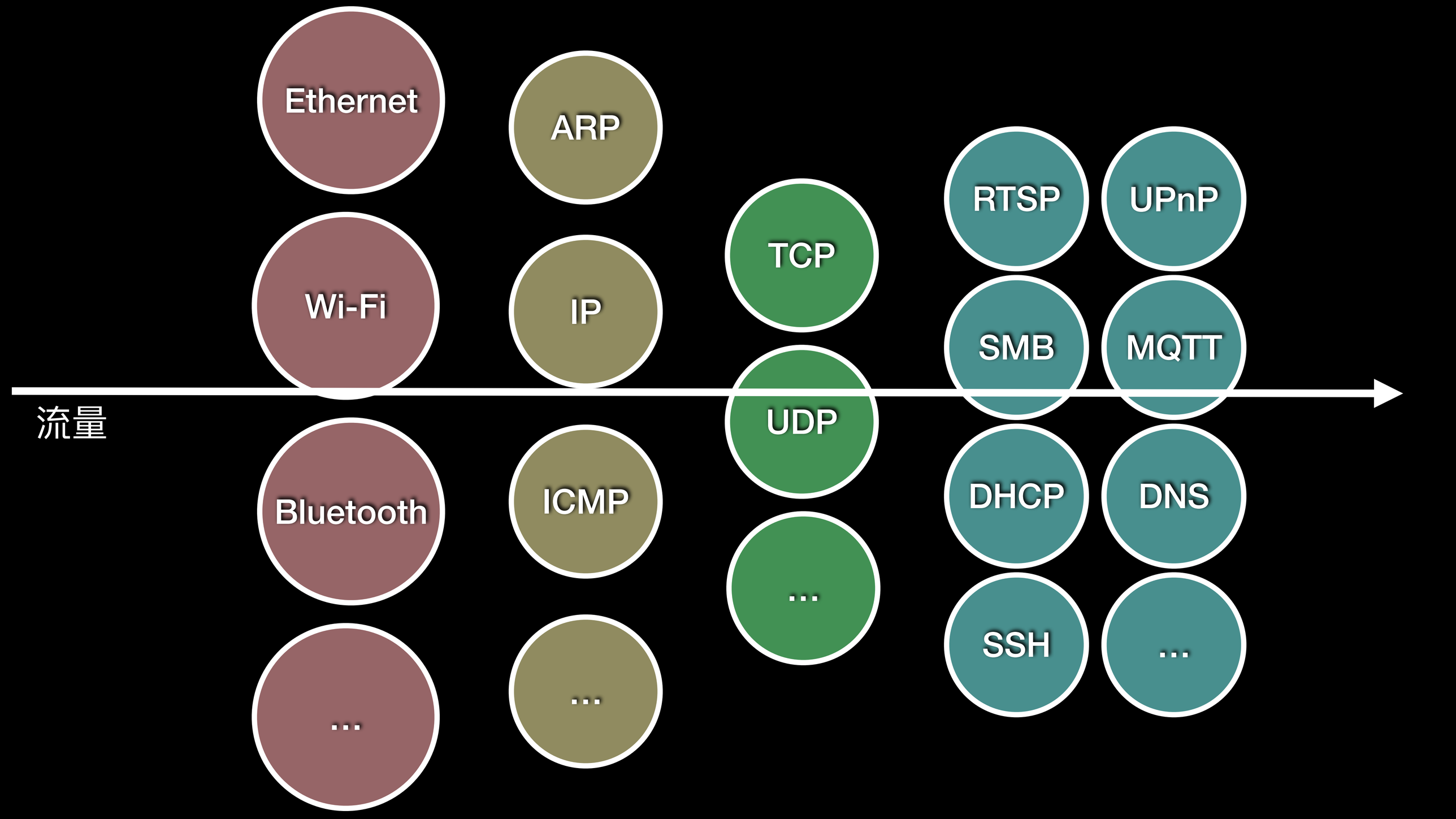
DEVCORE

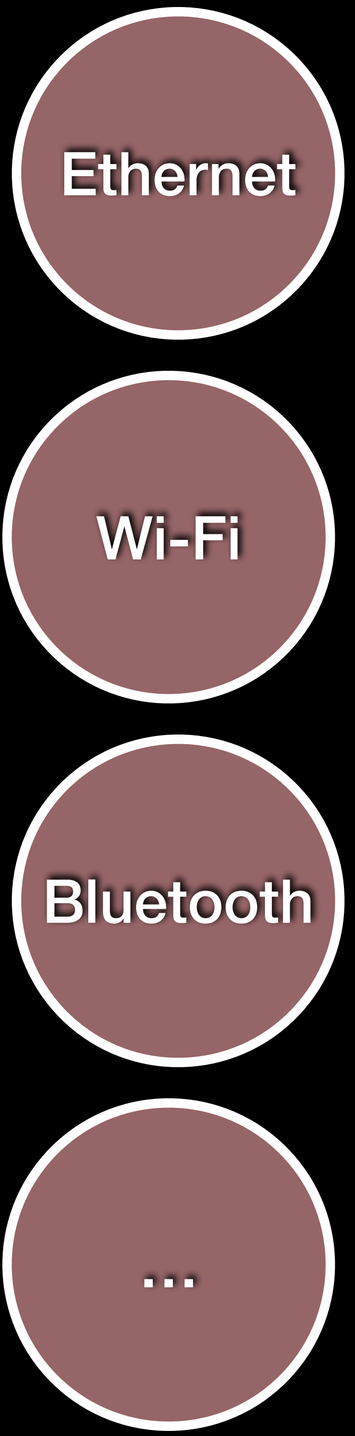
追求卓越

DEVCORE

DEVCORE

DEVCORE


Ethernet
ARP
Wi-Fi
IP
TCP
RTSP
UPnP
SMB
MQTT
流量
UDP
Bluetooth
ICMP
DHCP
DNS
...
SSH
...
...
...


Ethernet
Wi-Fi
Bluetooth
...


Wi-Fi

Wi-Fi 網路受到密碼保護

真的嗎？

過去研究

- 2017 - Broadpwn: Remotely Compromising Android and iOS via a Bug in Broadcom's Wi-Fi Chipsets
- 2018 - Researching Marvell Avastar Wi-Fi: From Zero Knowledge to Over-the-Air Zero-Touch RCE
- 2019 - Exploiting Qualcomm WLAN and Modem Over the Air
- 2021 - Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation
- 2021 - Qualcomm WiFi: Infinity War
- 2022 - BadMesher: New Attack Surfaces of Wi-Fi Mesh Network
- 2022 - Ghost in the Wireless, iwlwifi edition
- 2022 - WIFI Security - From 0 To 1
- 2024 - Listen-Up: Sonos Over-The-Air Remote Kernel Exploitation and Covert Wiretap
- 2025 - Unlock hidden Superpowers in MediaTek Wi-Fi Chips

DEVCORE

為發哥安全獻上祝福

DEVCORE

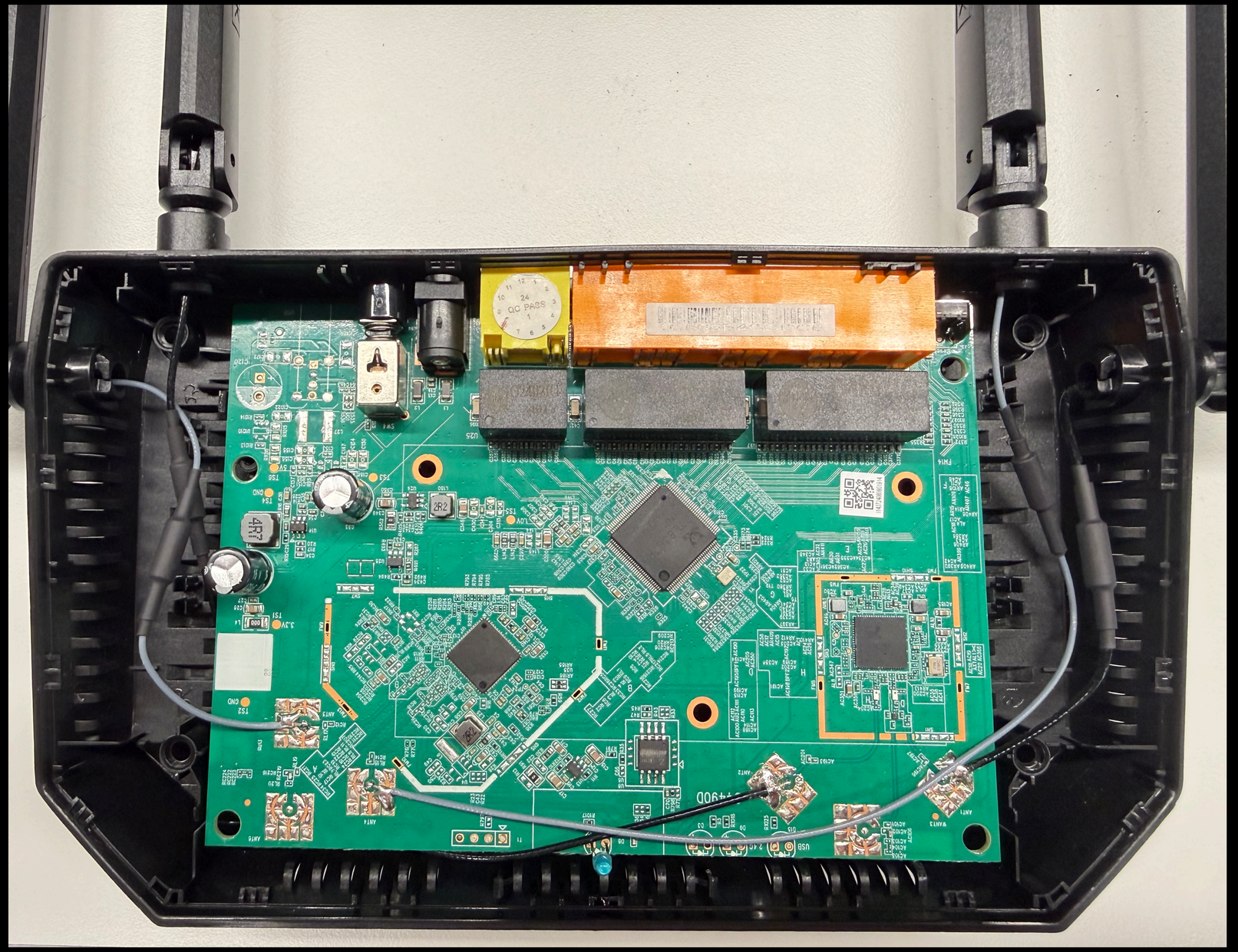
重視承諾

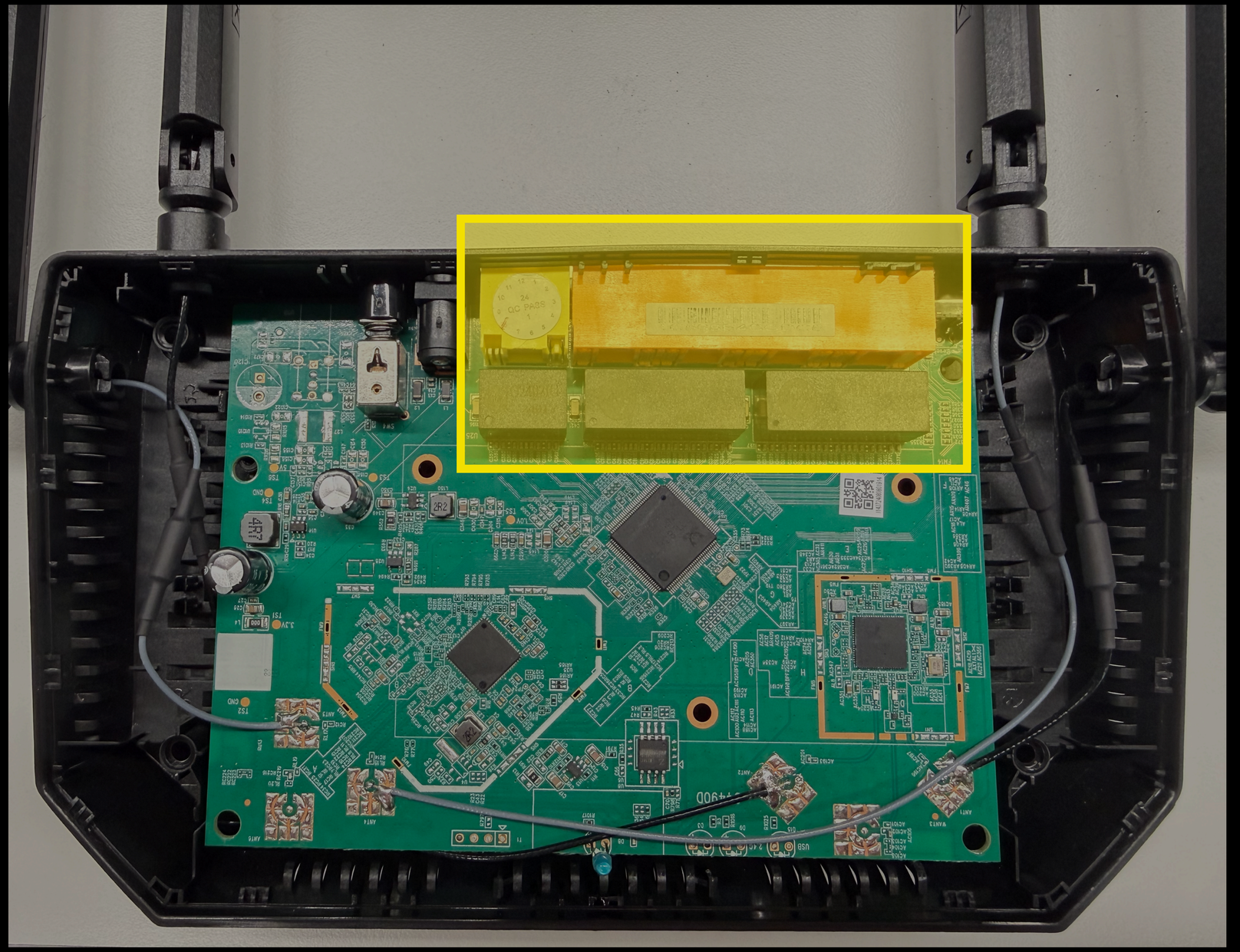
DEVCORE

盒子裡面裝什麼？
（非當事機）

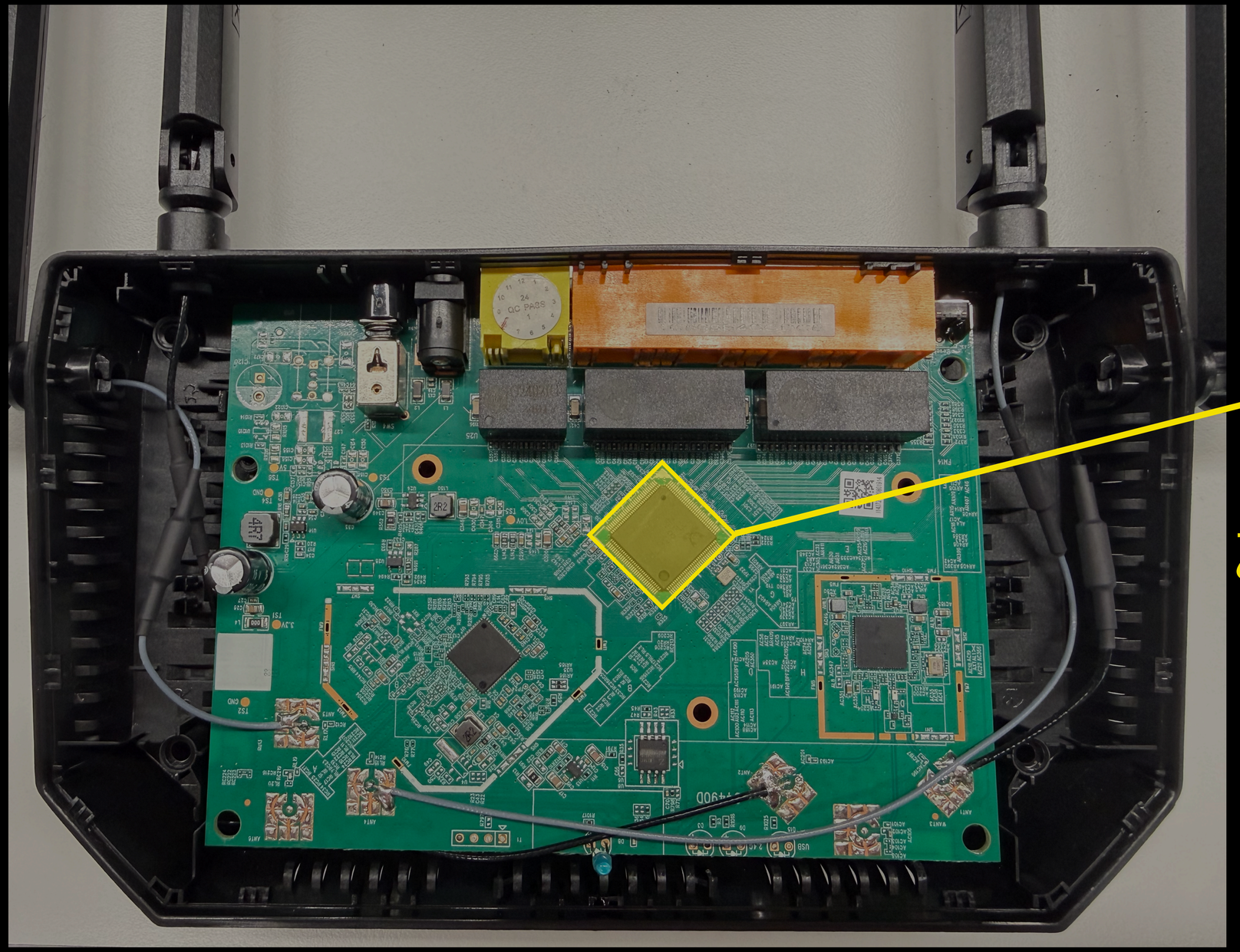
DEVCORE

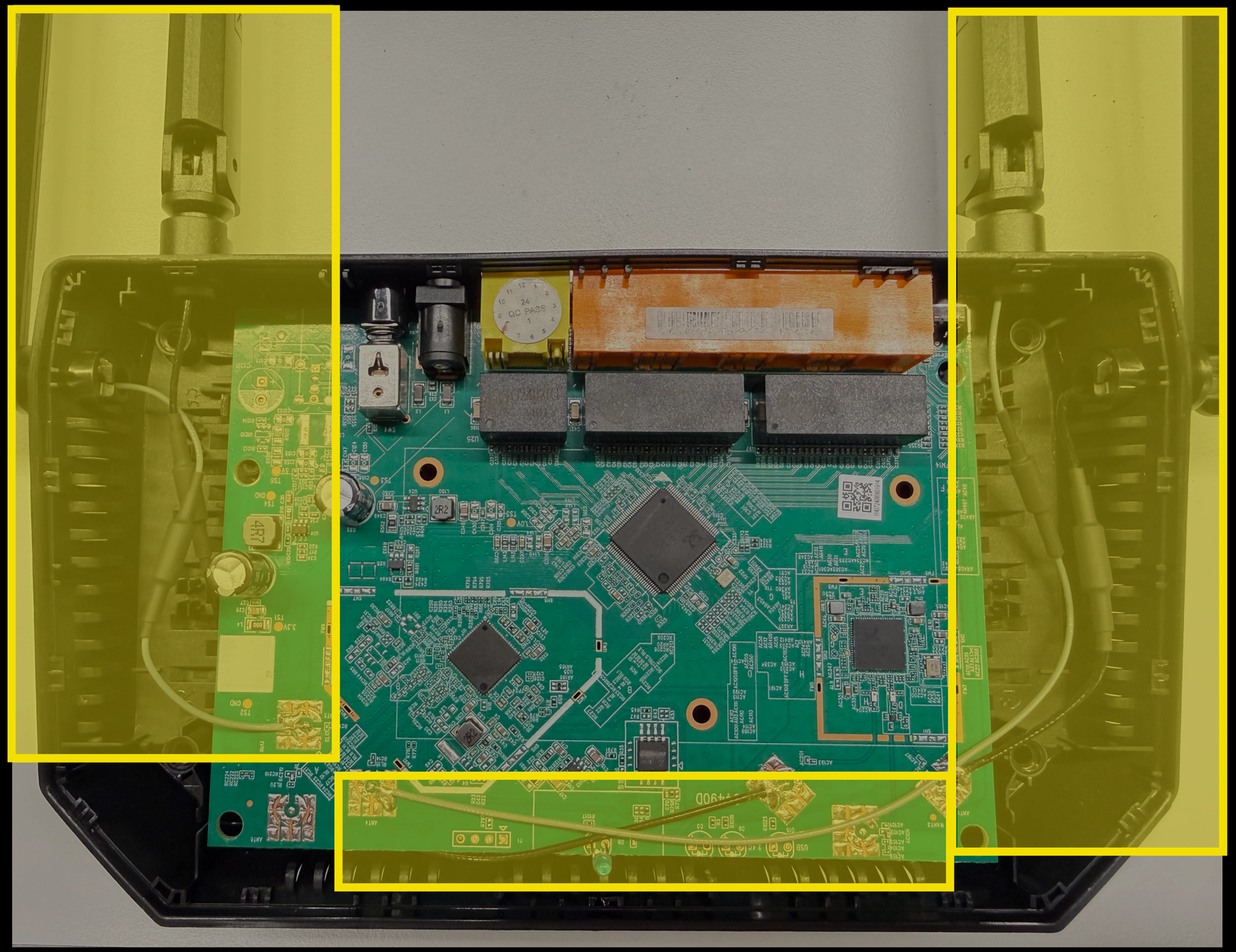
某台
Wi-Fi 路由器



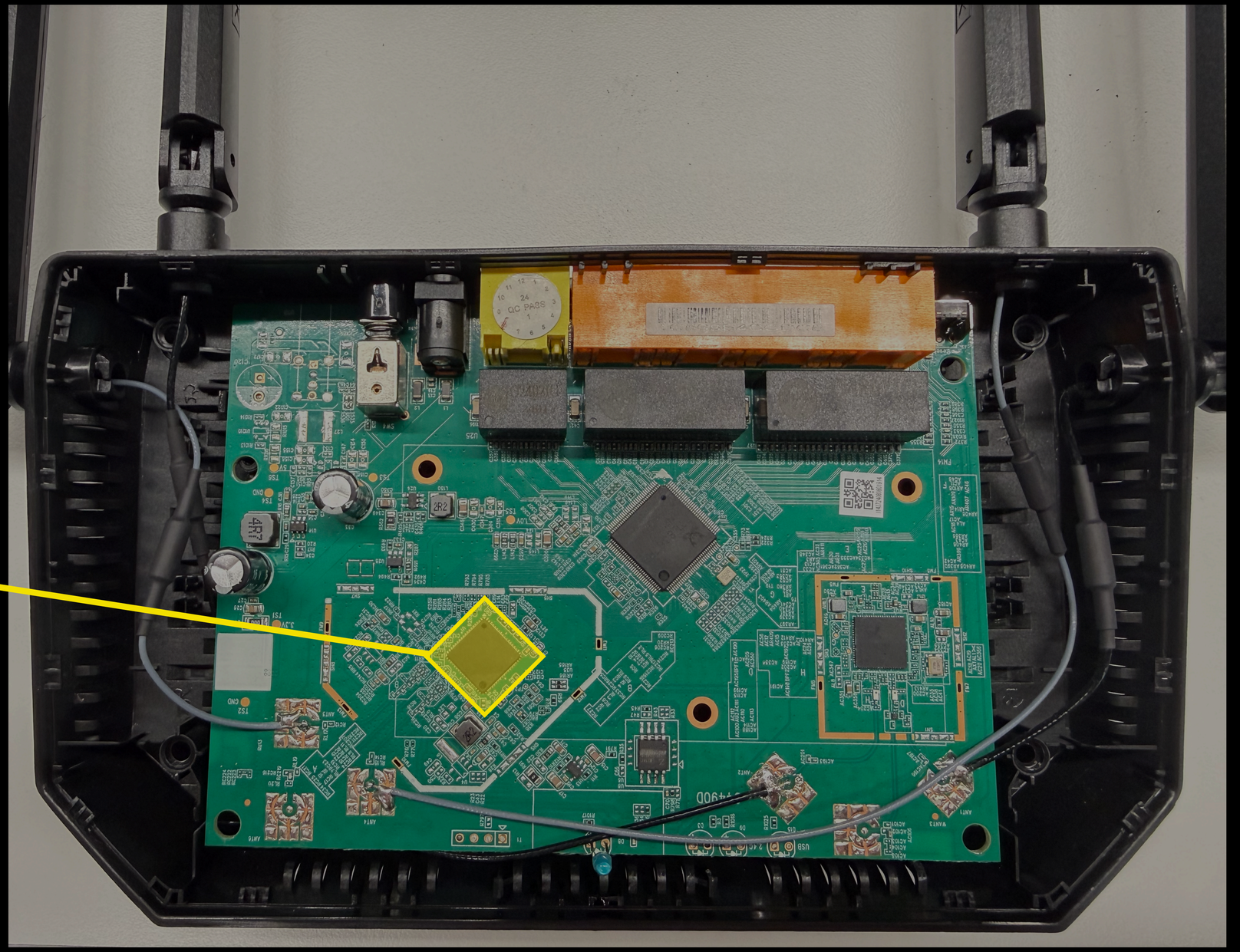


RJ45 網路孔


乙太網交換晶片

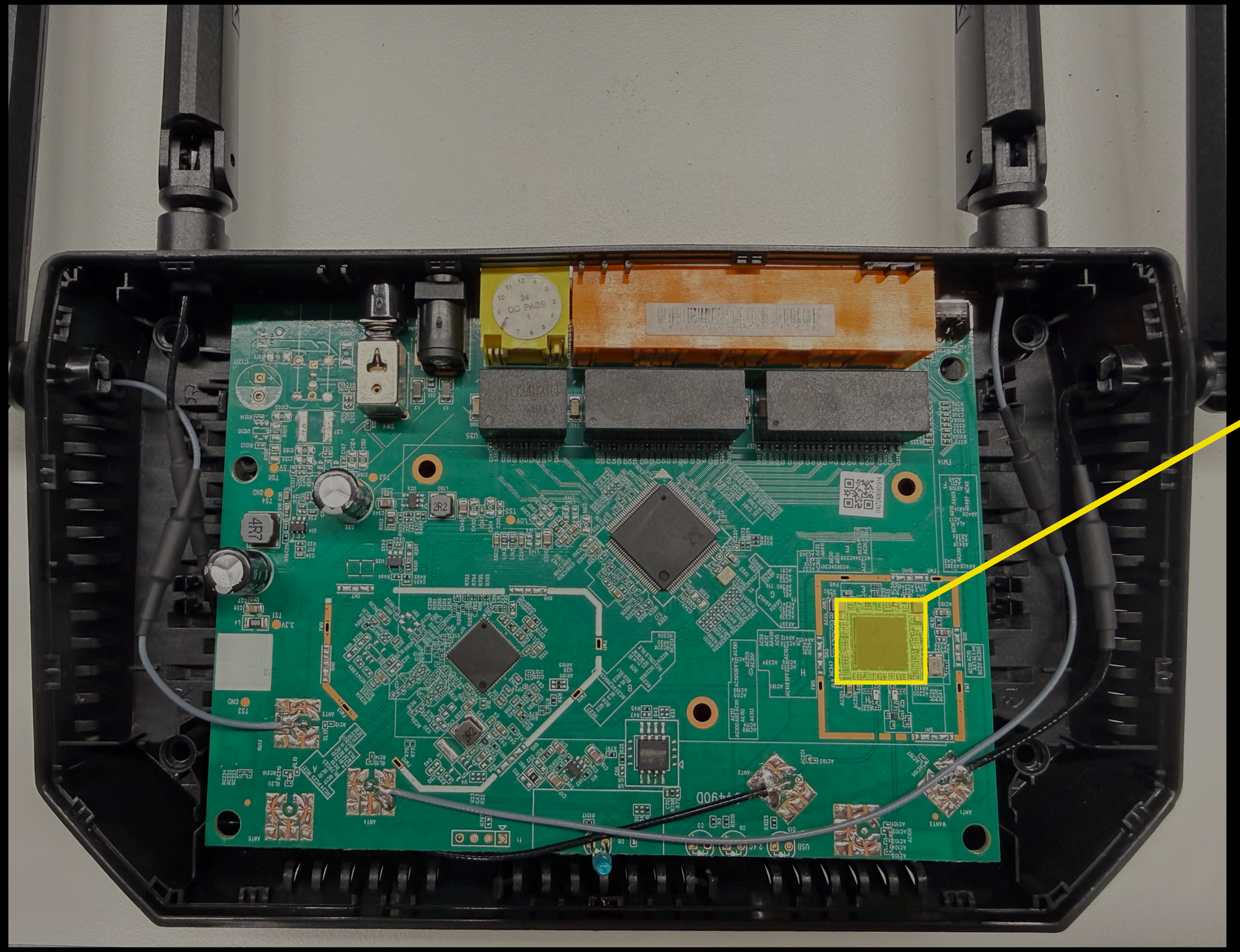


Wi-Fi 天線
2.4 GHz x 2
5 GHz x 2



Wi-Fi SoC

- 主處理器
  - 跑 Linux
- 2.4 GHz Wi-Fi

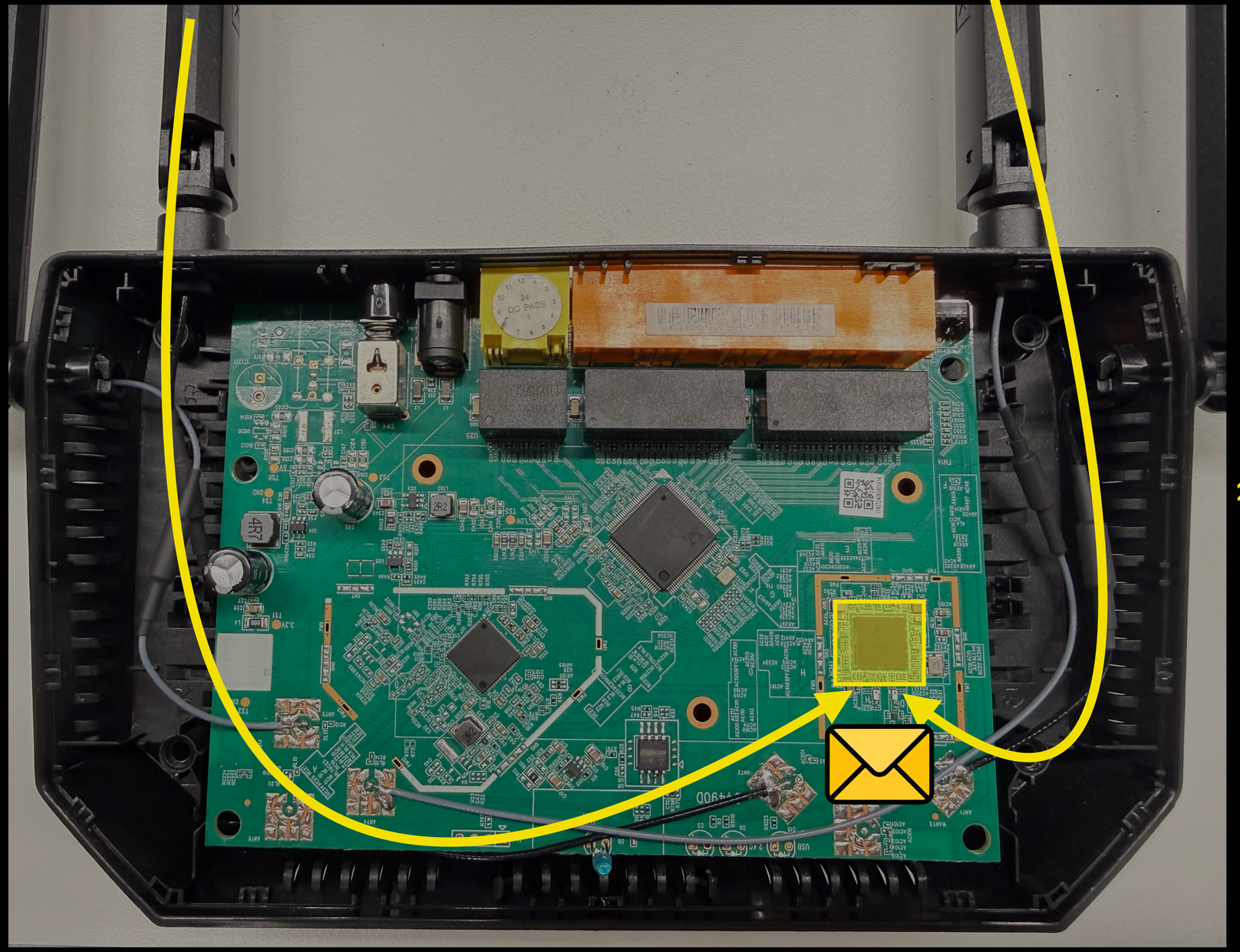


Wi-Fi MCU

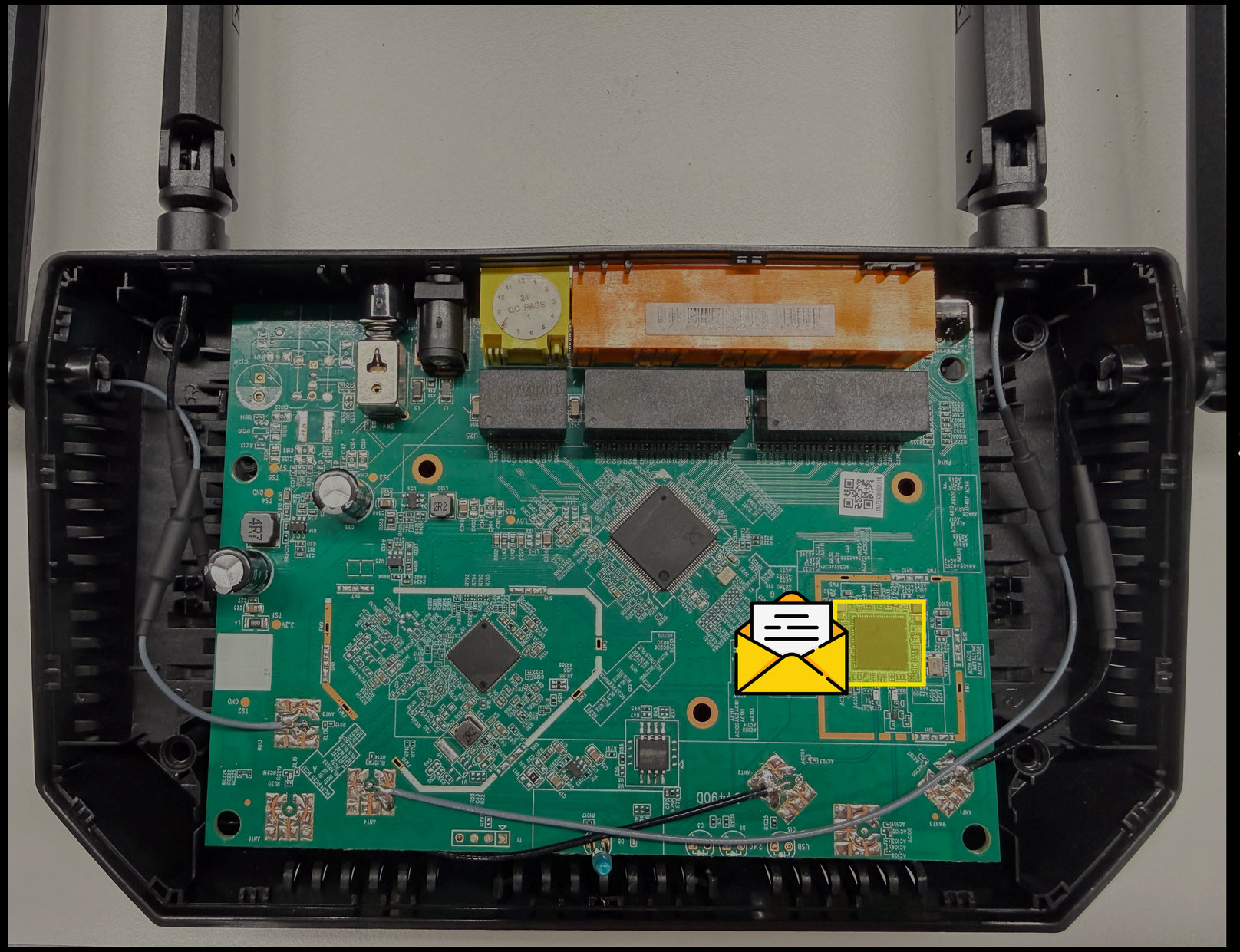
- 5 GHz Wi-Fi
- 跑 RTOS

Wi-Fi 內網使用者收封包

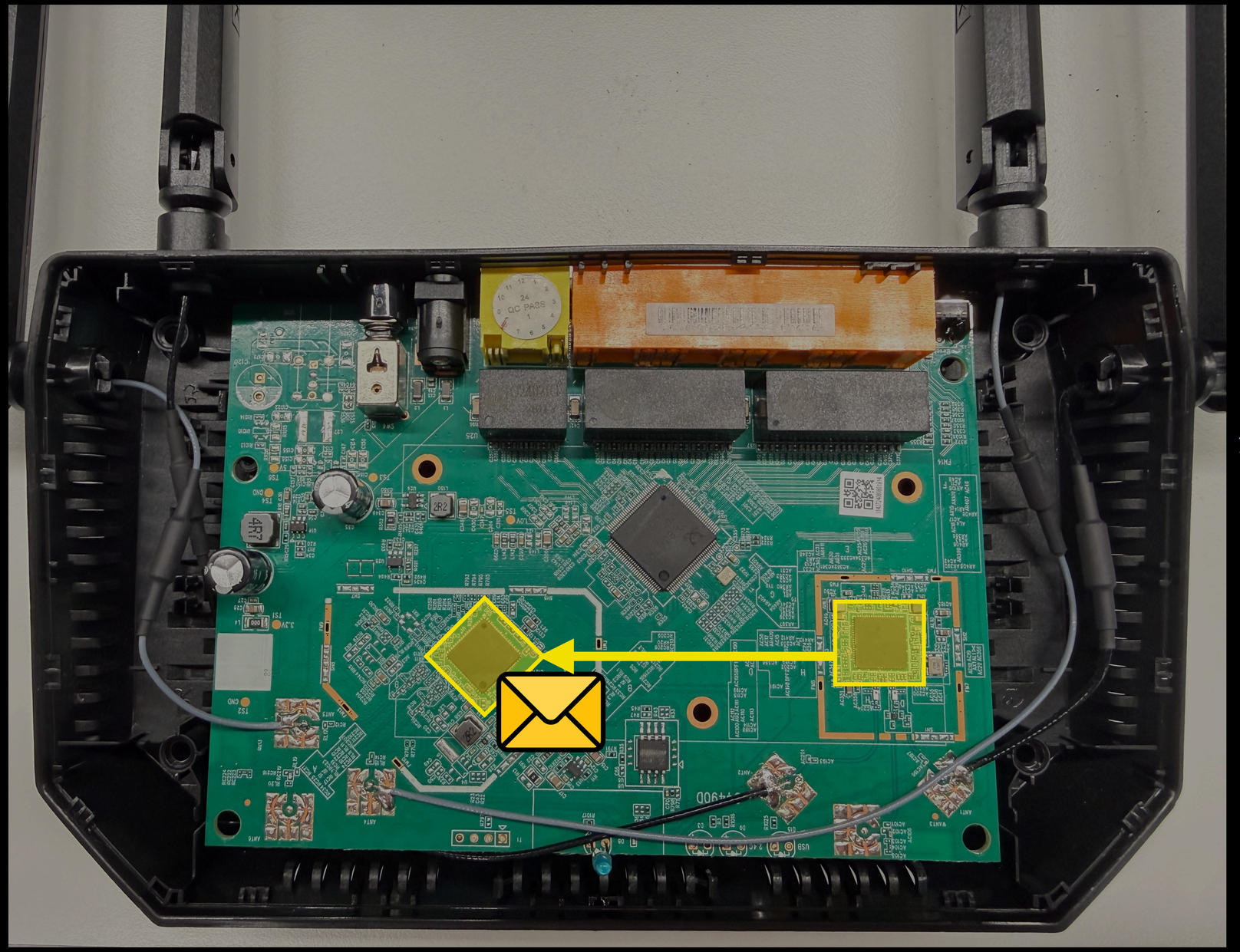
DEVCORE



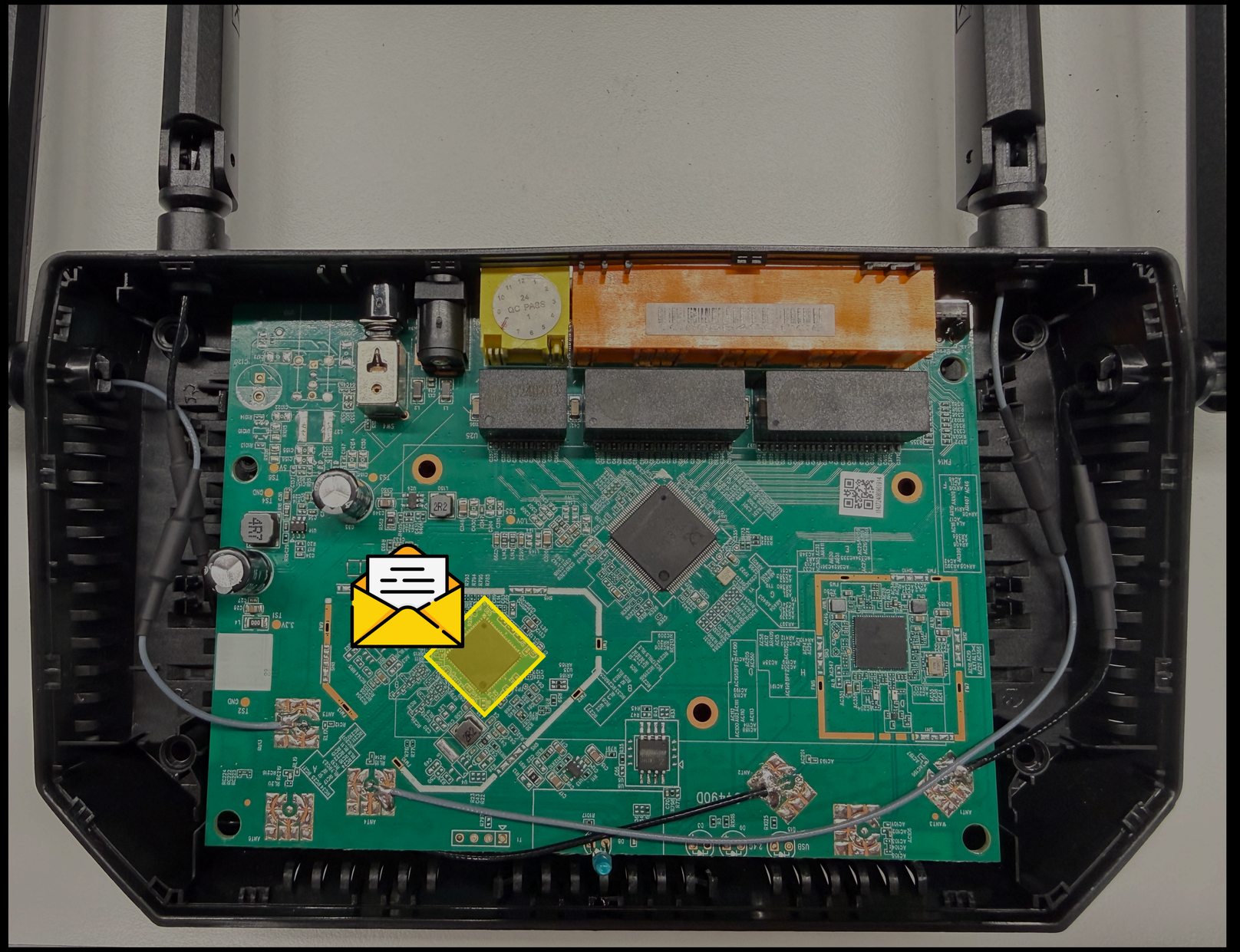
Wi-Fi Frame
從 5GHz 頻段傳入



Wi-Fi MCU 的硬體和韌體會解析 Wi-Fi Frame (Layer 1, 2)

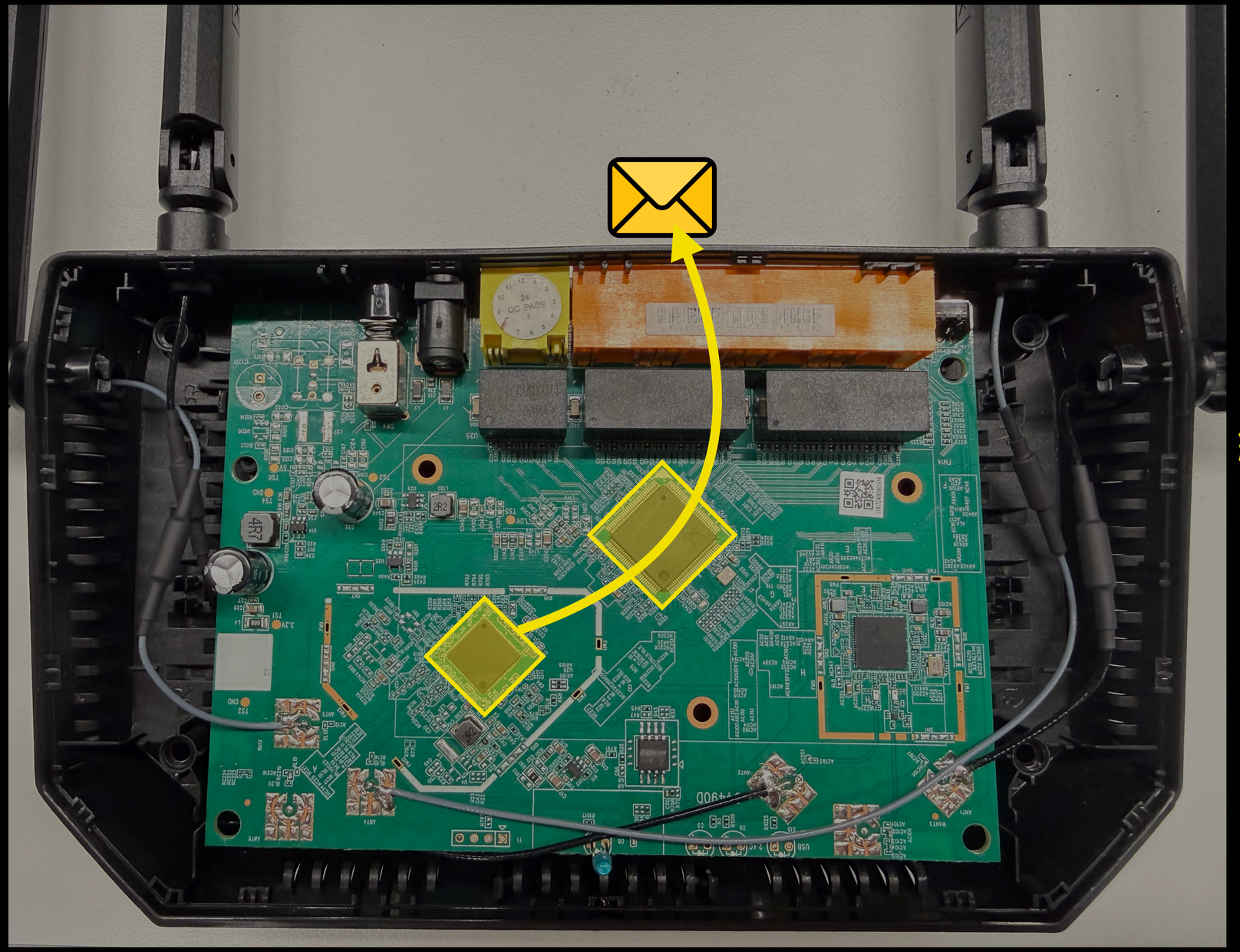


然後轉給主處理器
(Wi-Fi SoC)



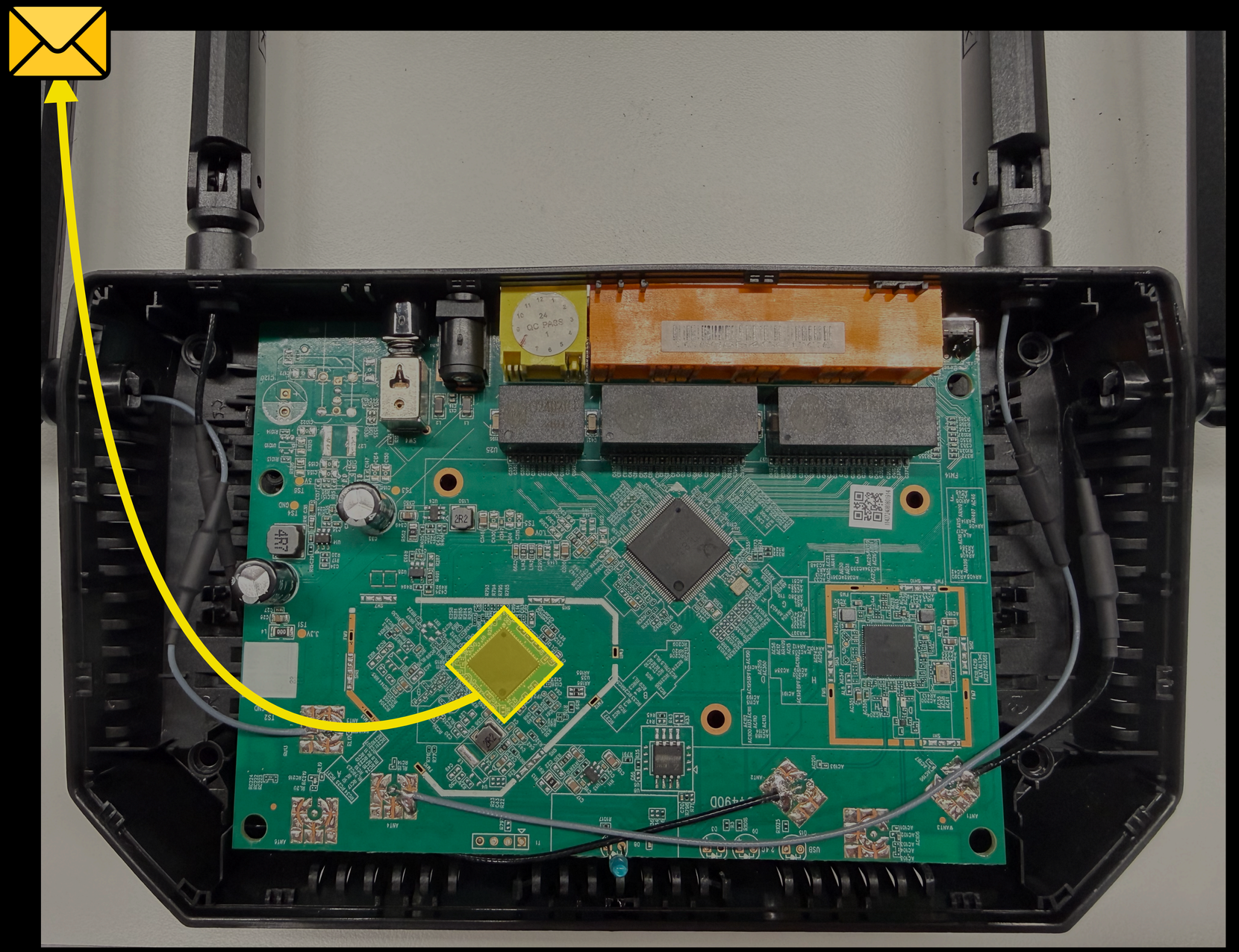
主處理器
處理 Wi-Fi Frame
(Layer 2 以上)

情況一：乙太內網

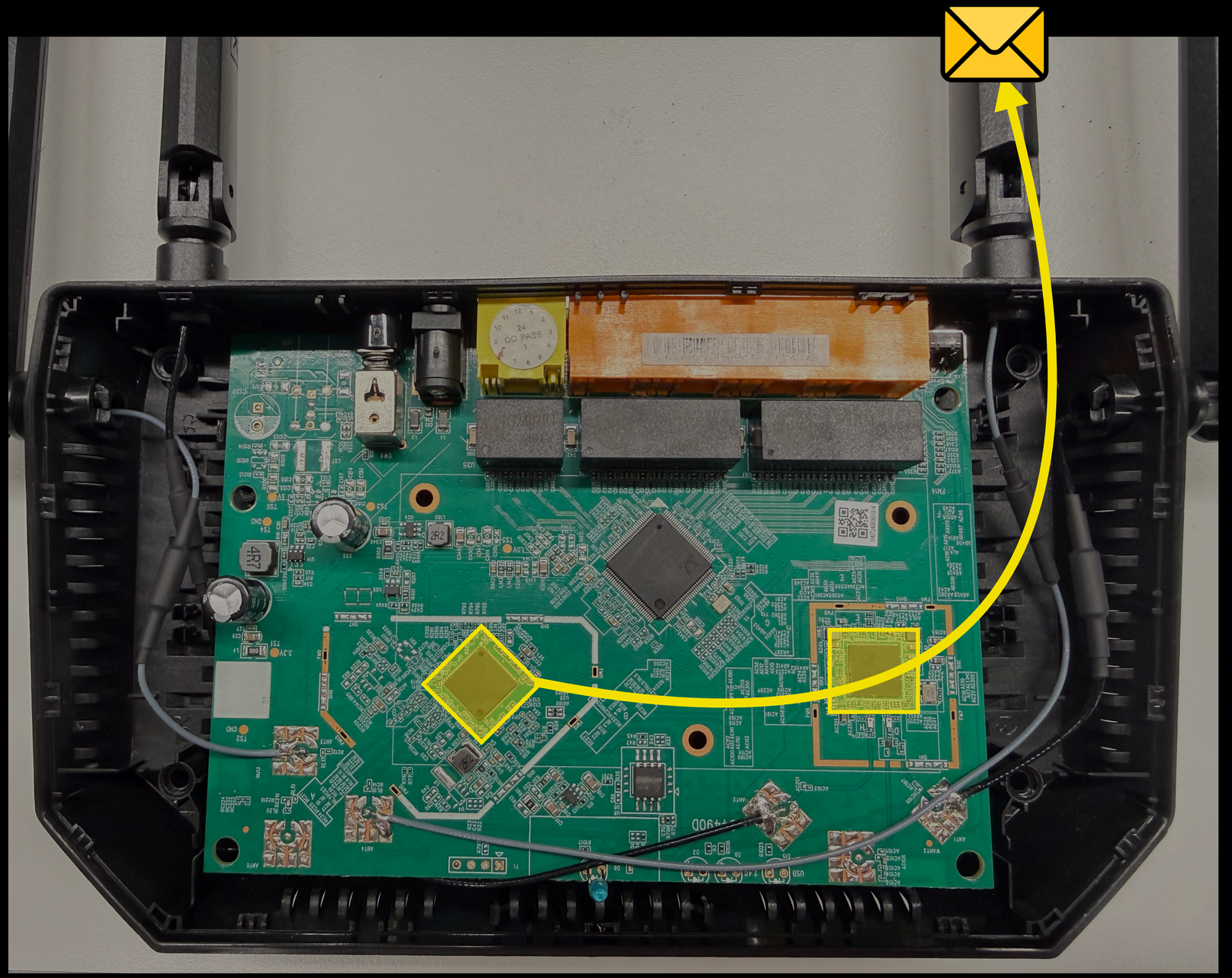


經乙太網交換晶片
轉發到 RJ45

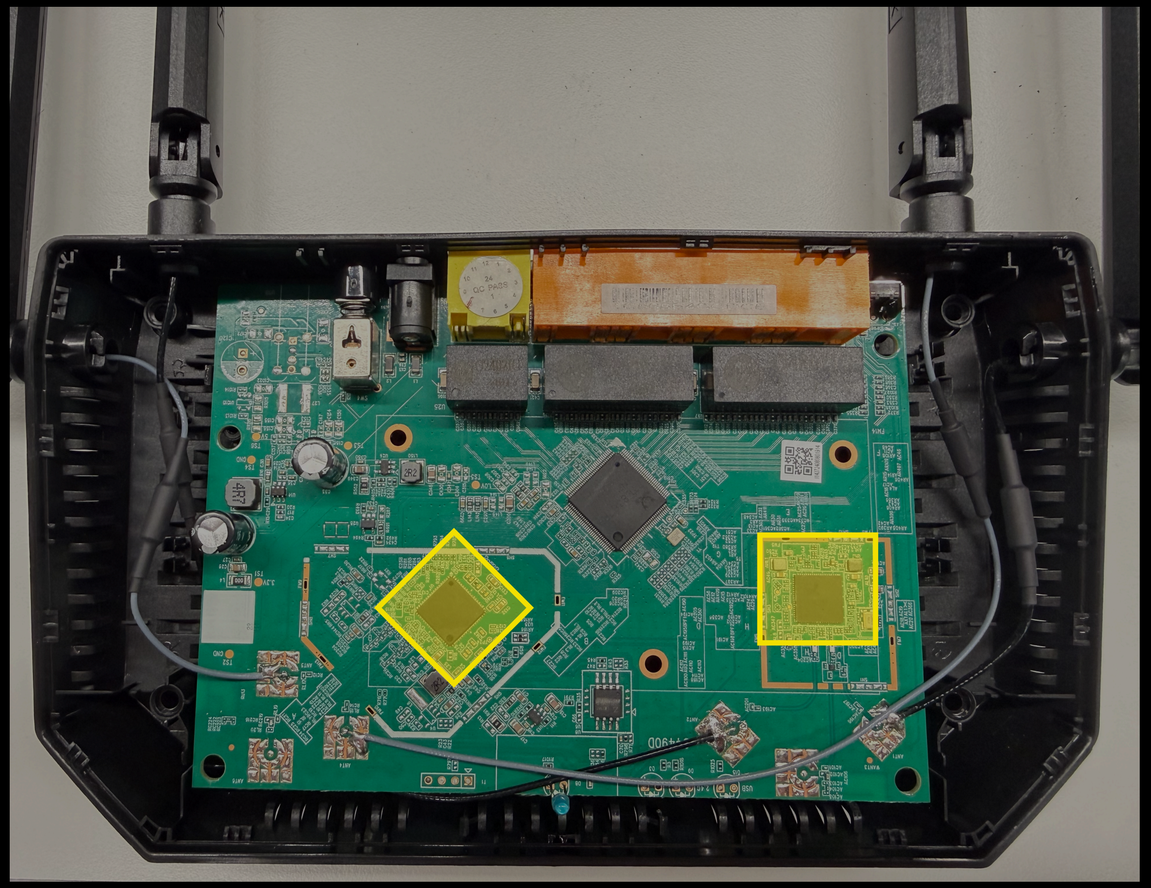
情況二：2.4 GHz Wi-Fi 內網


轉發到乙太內網

情況三：5 GHz Wi-Fi 內網


轉發到乙太內網

本日主角:
MediaTek MT7981

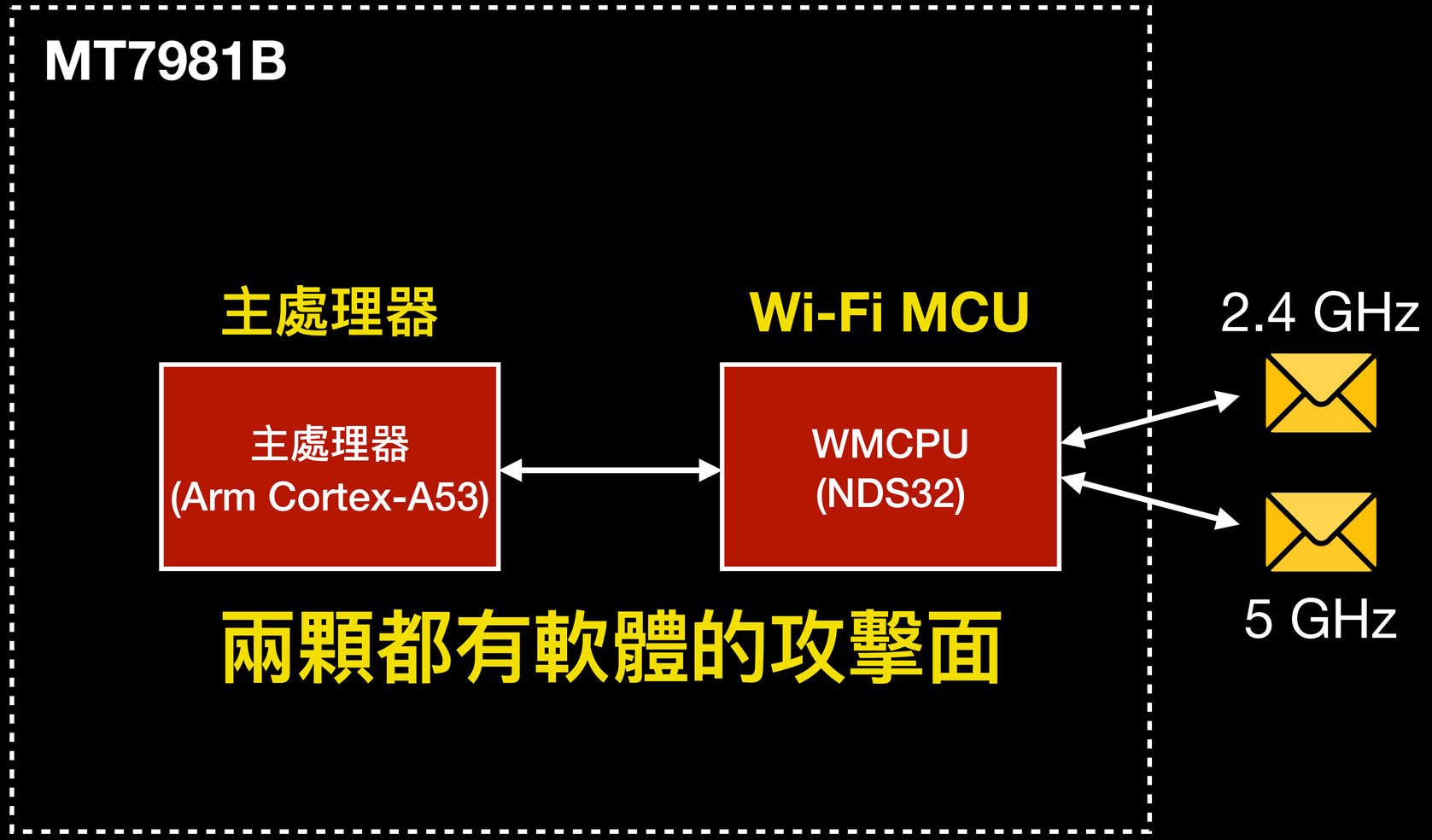


主處理器和 Wi-Fi MCU
分開來整合在兩個不同晶片

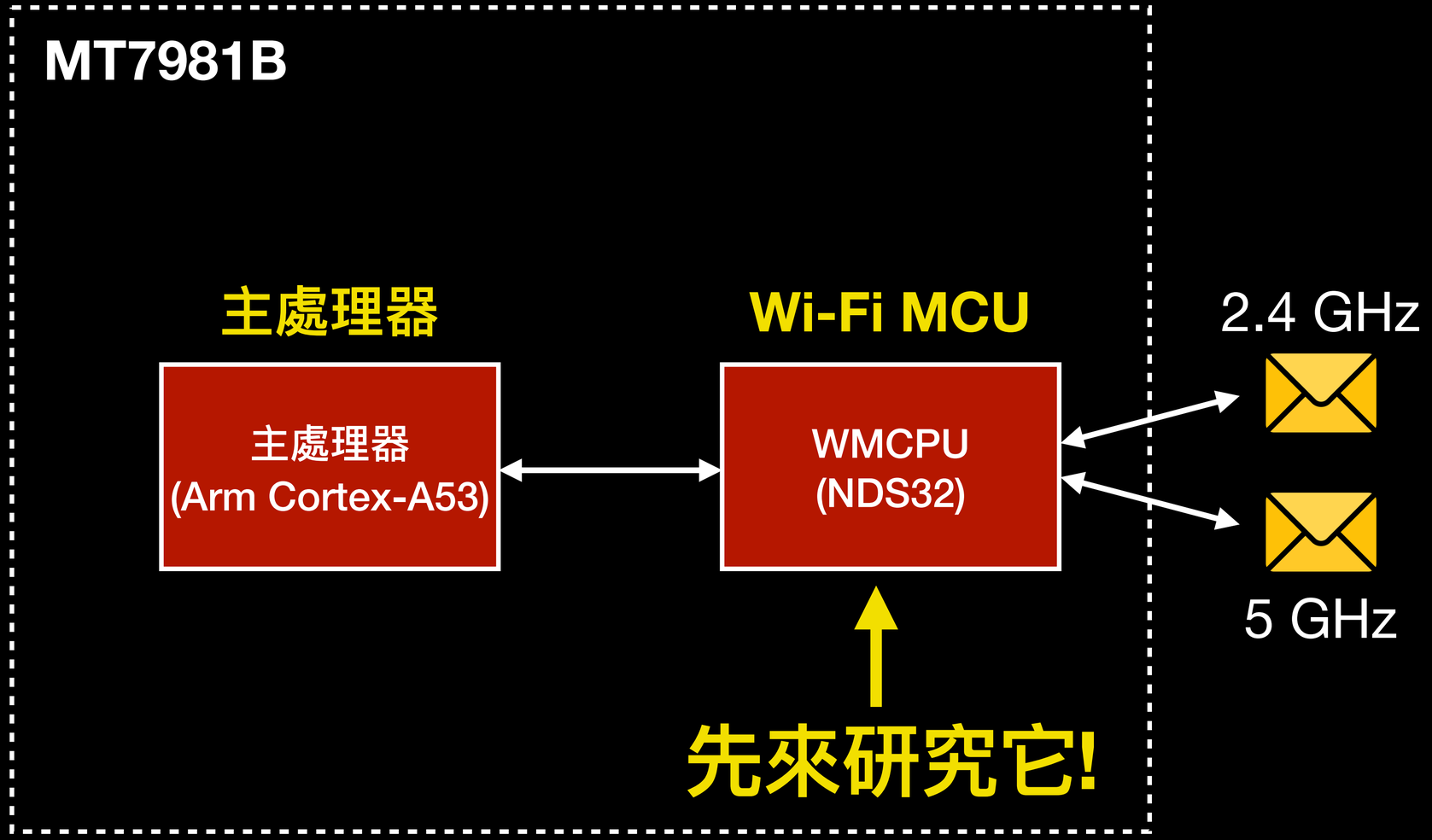
兩者類比

MT7981 則是
整個整合在同一個晶片

Wi-Fi SoC
- 整合 2.4 和 5 GHz
- NDS32 Wi-Fi MCU
  - 運行 RTOS
- Arm 主處理器
  - 運行 Linux

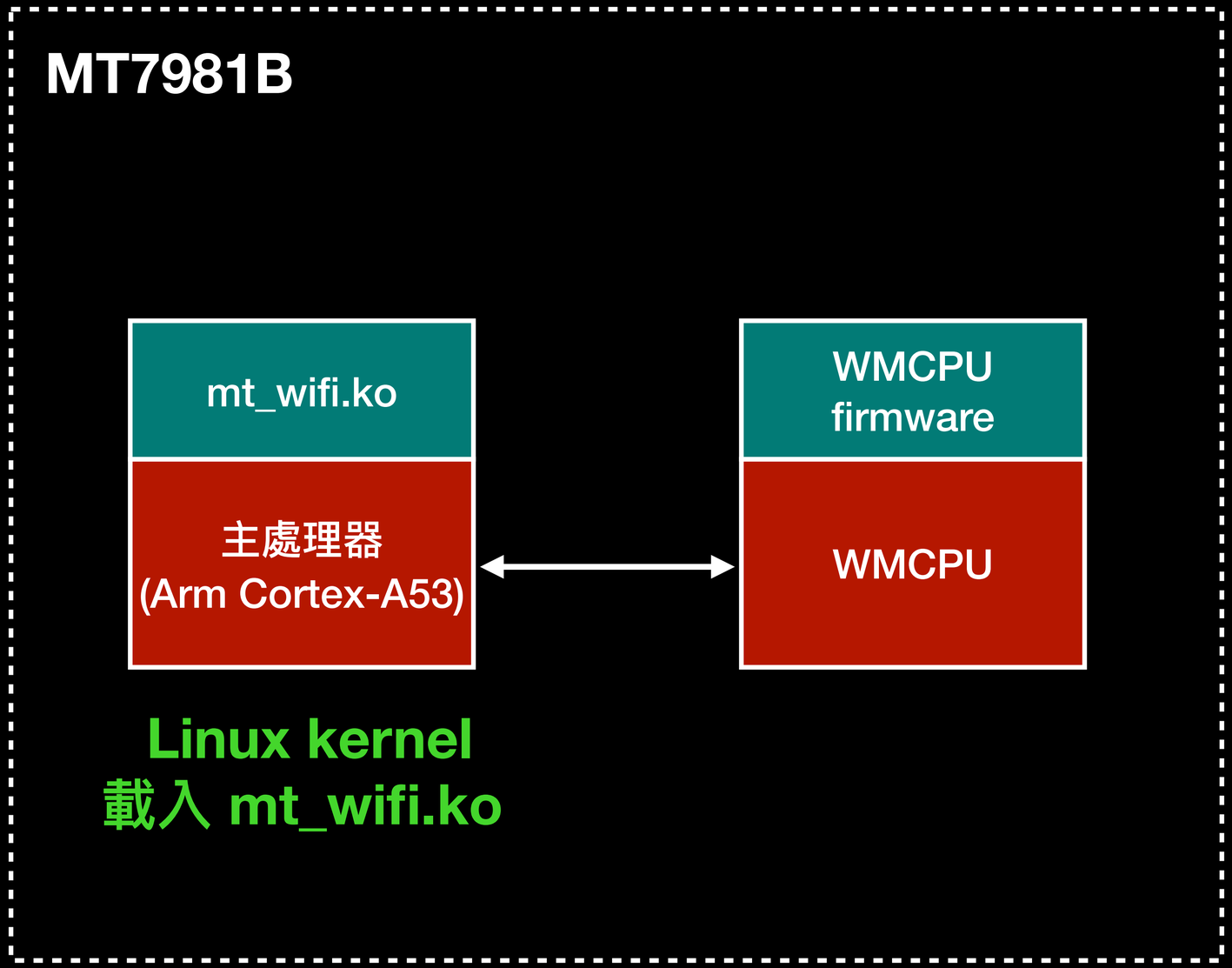


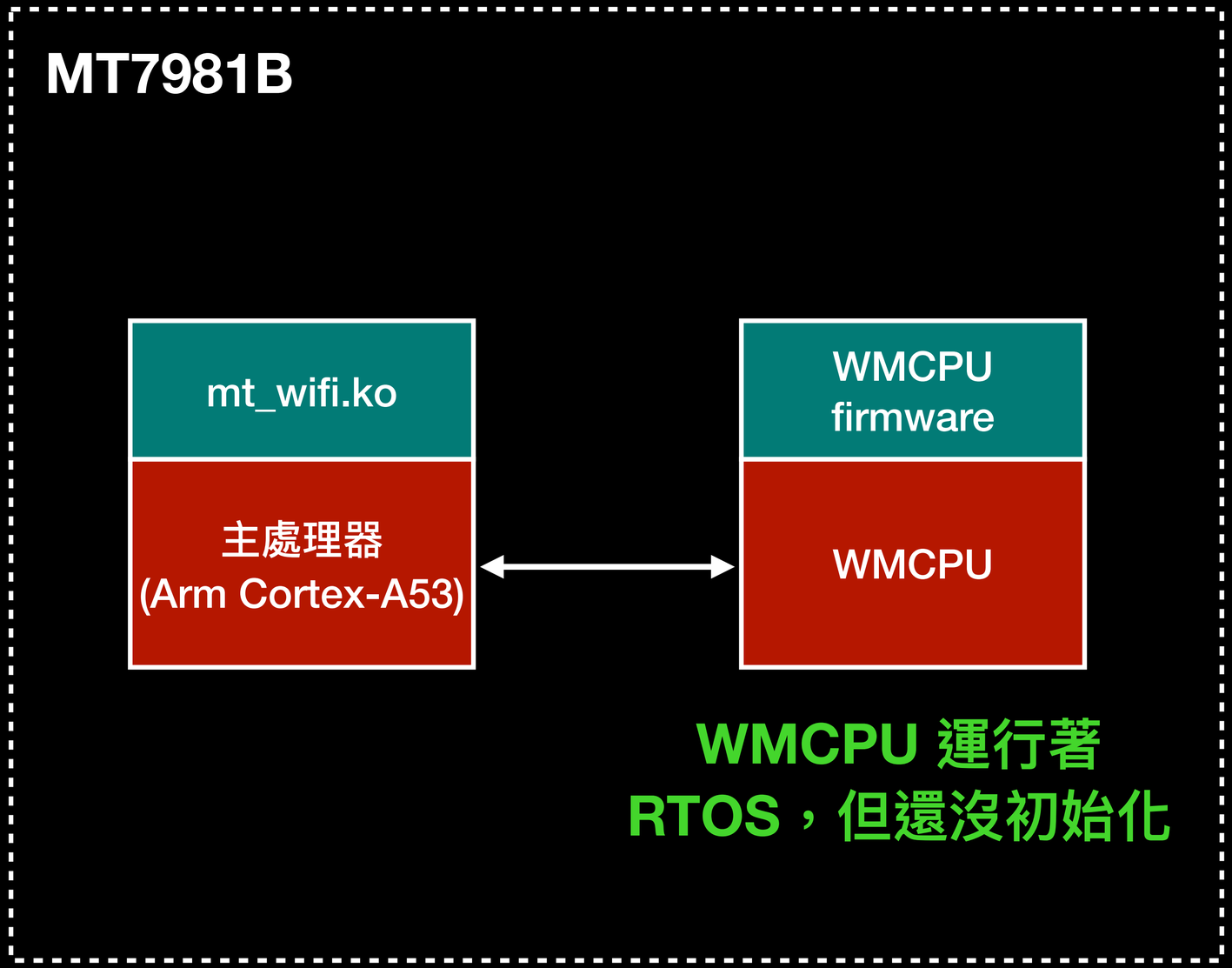
Wi-Fi SoC
- 整合 2.4 和 5 GHz
- NDS32 Wi-Fi MCU
  - 運行 RTOS
- Arm 主處理器
  - 運行 Linux



Wi-Fi MCU 初始化

DEVCORE


MT7981B
mt_wifi.ko
主處理器
(Arm Cortex-A53)
WMCPU
firmware
WMCPU
Linux kernel
載入 mt_wifi.ko


MT7981B
mt_wifi.ko
主處理器
(Arm Cortex-A53)
WMCPU
firmware
WMCPU
WMCPU 運行著
RTOS，但還沒初始化

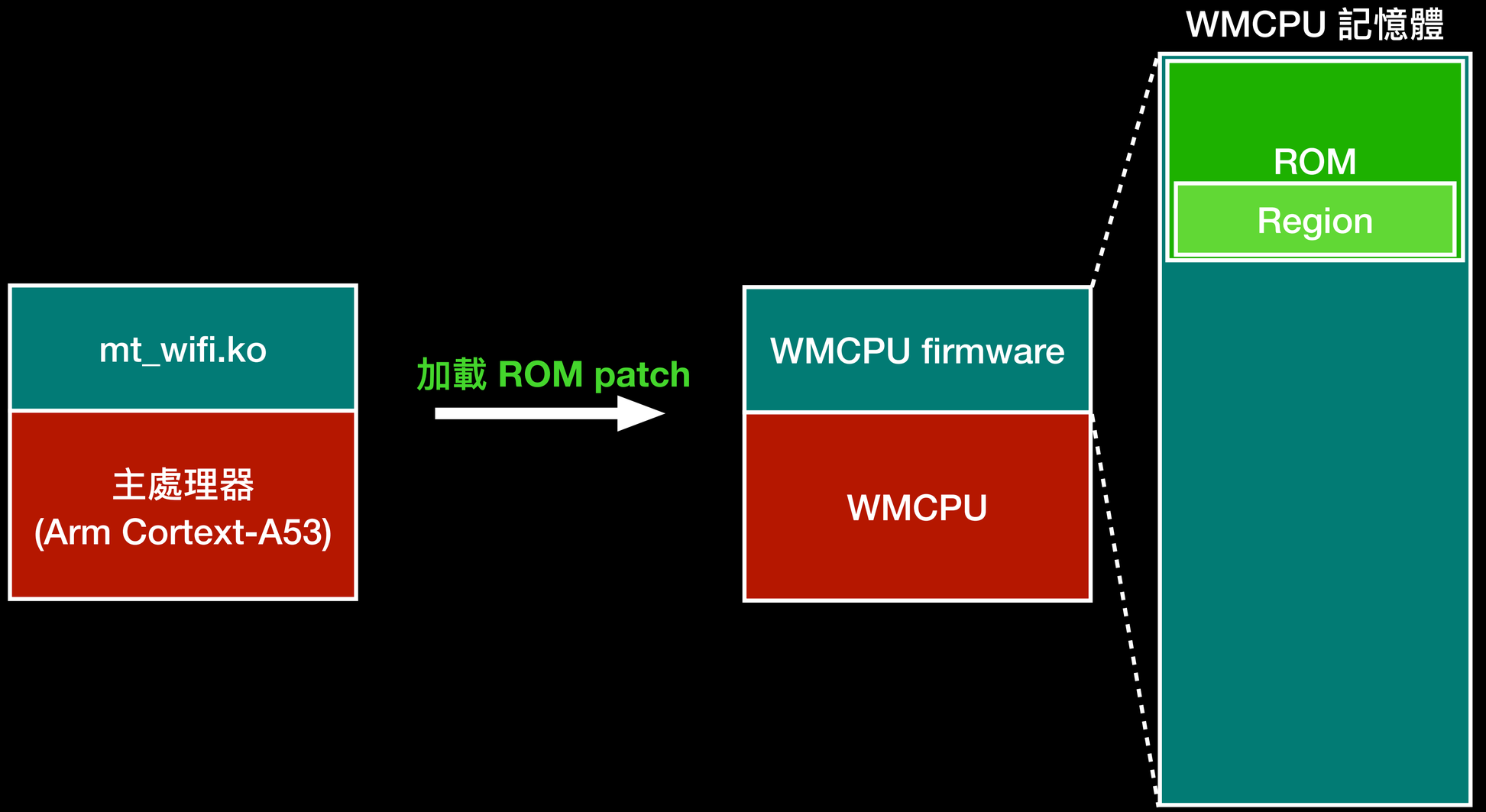
ROM Patch 加載

DEVCORE

載入 ROM Patch


mt7981_patch_e1_hdr.bin
Header
Region
mt_wifi.ko
主處理器
(Arm Cortext-A53)
WMCPU firmware
WMCPU
WMCPU 記憶體
ROM

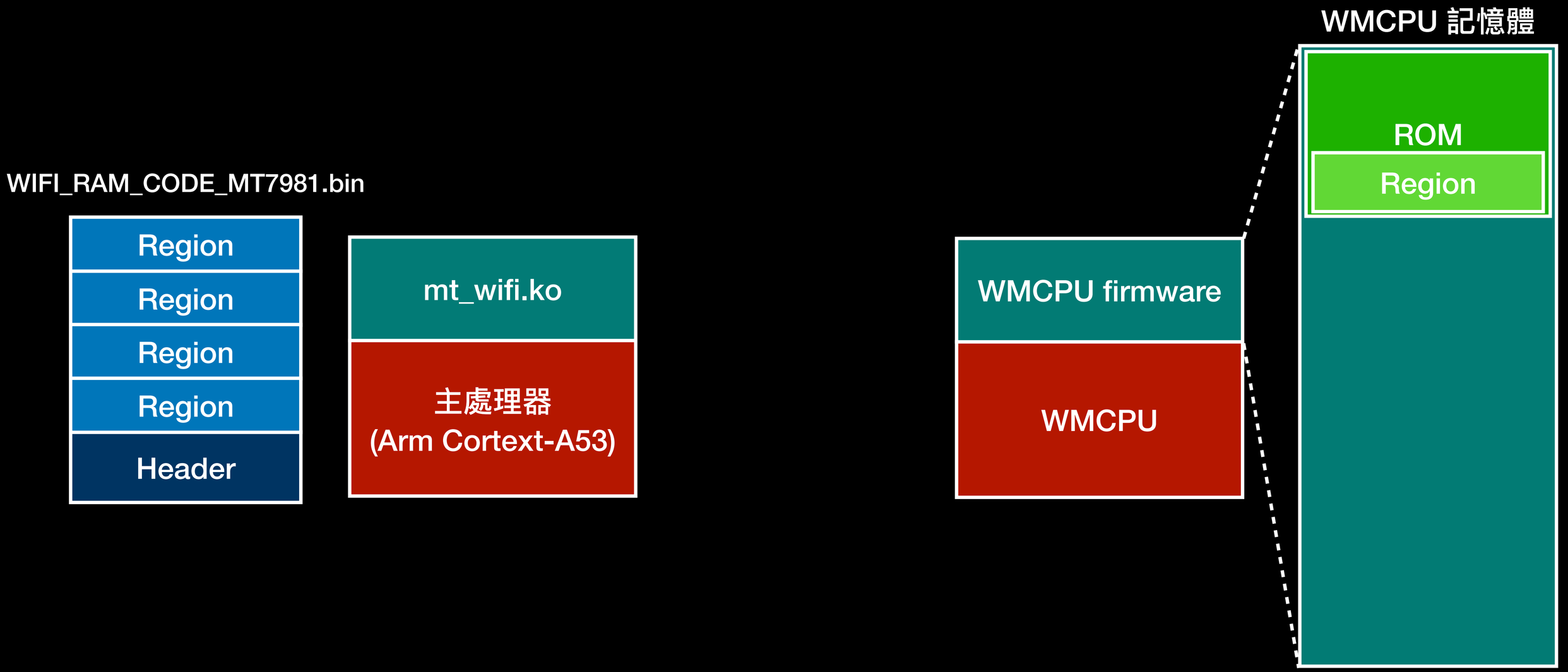
載入 ROM Patch


WMCPU 記憶體
ROM
Region
mt_wifi.ko
主處理器
(Arm Cortext-A53)
加載 ROM patch
WMCPU firmware
WMCPU

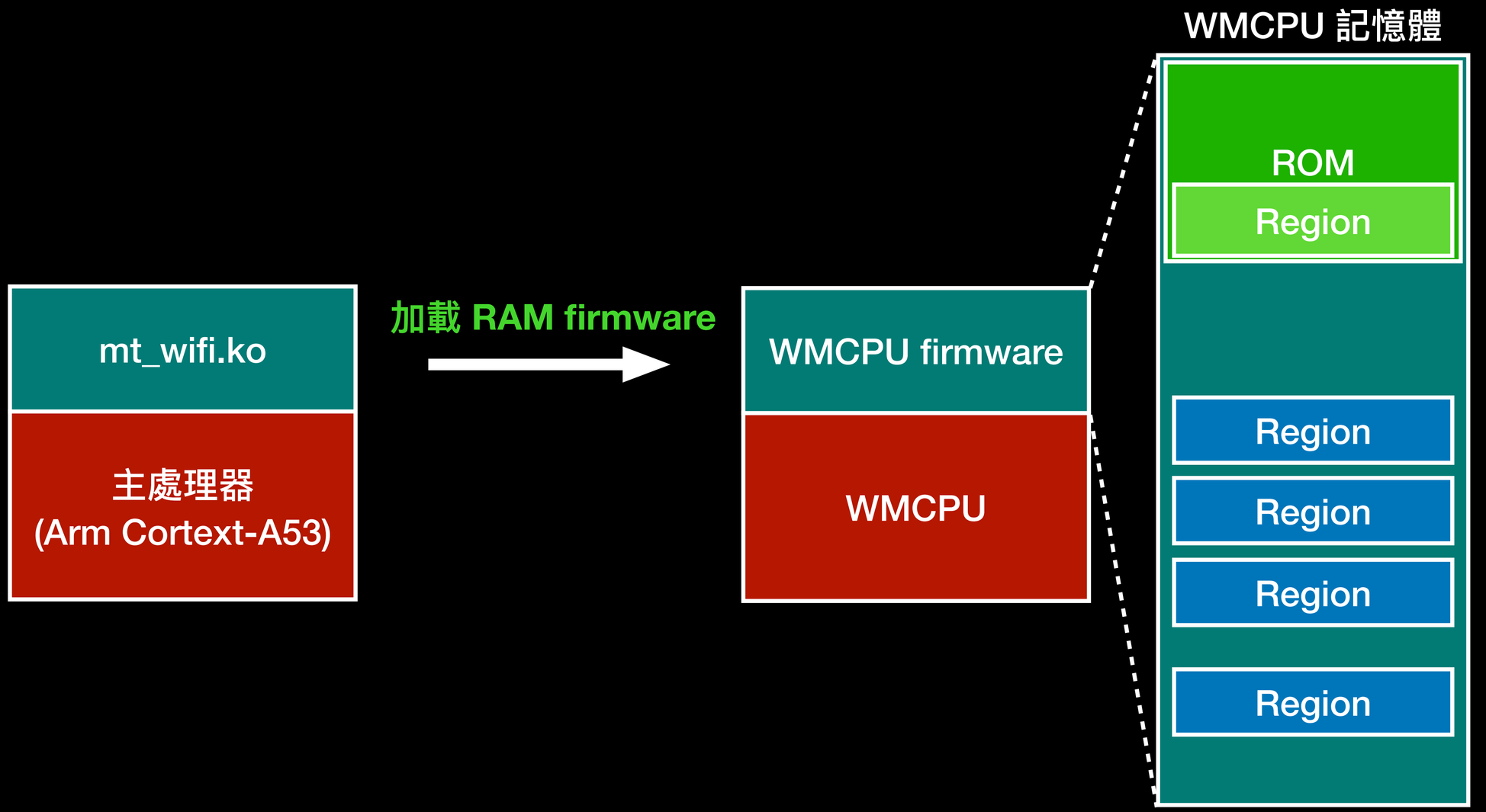
RAM Firmware 加載

DEVCORE

載入 RAM Firmware

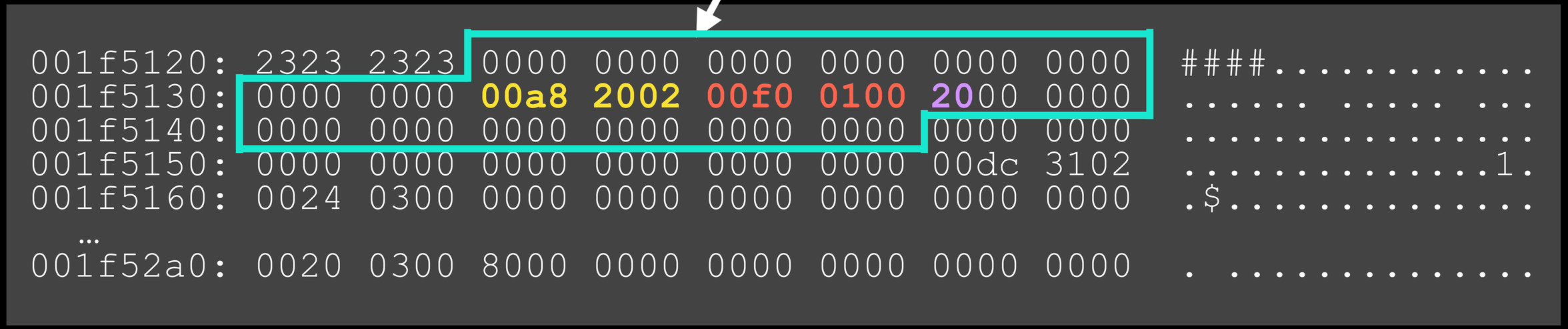


載入 RAM Firmware



RAM Region Header 格式

0x20 是 FW_FEATURE_OVERRIDE_RAM_ADDR,
所以入口點會被設置成 0x0220a800

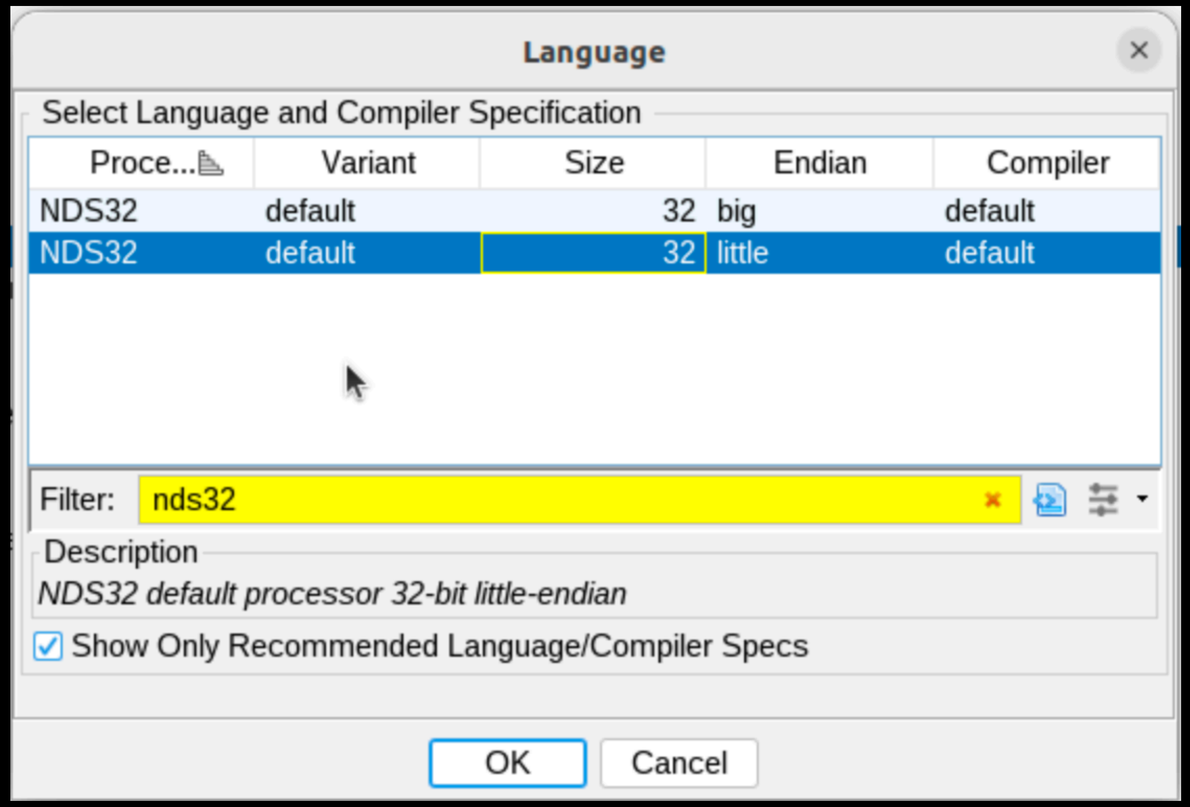


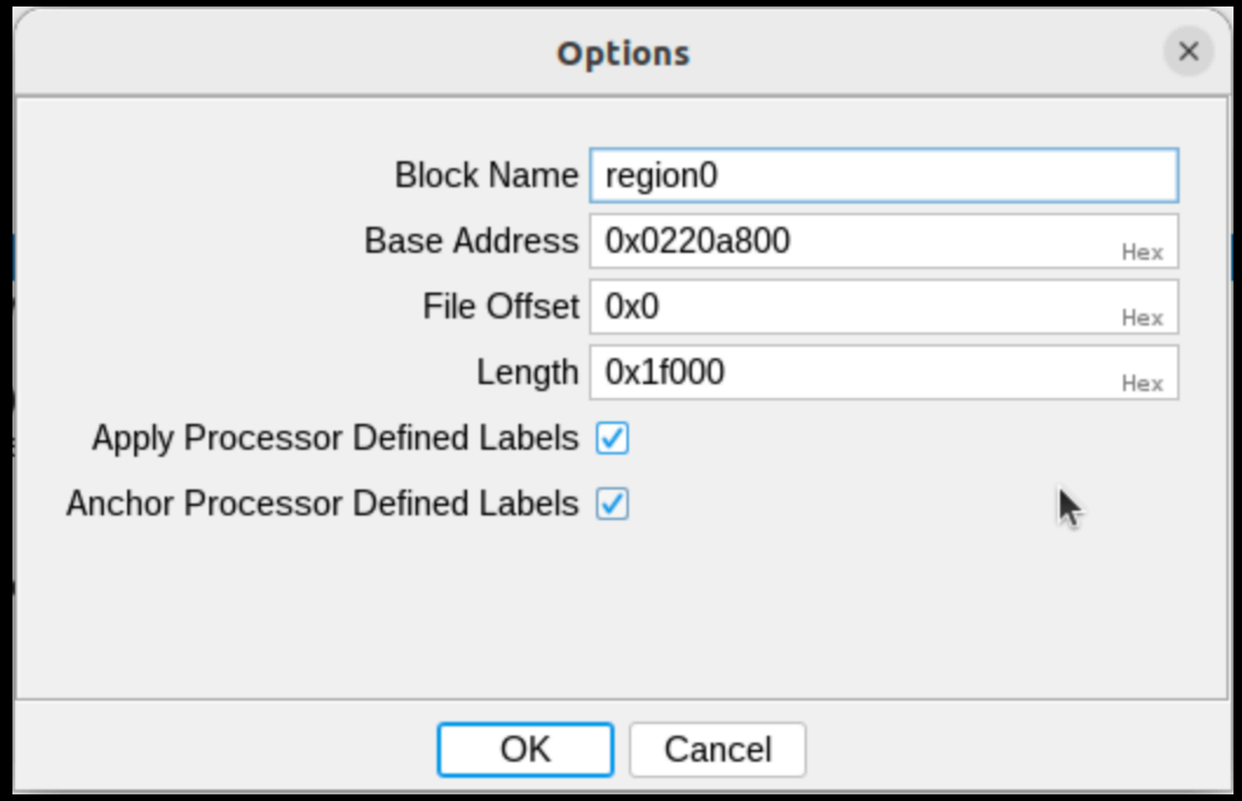
Wi-Fi 韌體加載完畢
開逆！

DEVCORE

NDS32 Ghidra 配置

- 我們使用 ghidra-staging 的 NDS32 patch 編譯 Ghidra
- 接著加載 Region 0 到 Ghidra






全都是 unaff_gp
LIGHTYEAR
字串呢？函數呢？跑哪了？

未知的 GP 暫存器

```
57        else {
58          iVar1 = FUN_e004c970();
59          if (iVar1 != 1) {
60            FUN_e003b50c(5,0x45);
61            iVar1 = FUN_e004c9f0();
62            if (iVar1 == 1) {
63              FUN_e00405d6();
64            }
65            goto LAB_0220be8c;
66          }
67          FUN_e0042482(0x15,4);
68          FUN_e004232c(s_Skip_send_notify_due_to_hif_susp_e00a6410);
69  LAB_0220be8e:
70          FUN_e003b50c(5,0x4a);
71          iVar1 = FUN_e003da16(&DAT_e00f9294,unaff_gp + 0x1b58c,0x803)
72          if (iVar1 != 0) {
73            FUN_e003dbfc(&DAT_e00f9294,unaff_gp + 0x1b58c,0x804);
74          }
75          FUN_e003b50c(5,0x4b);
76          FUN_e003dfd0(&DAT_e00f9294,5,unaff_gp + 0x1b58c,0x807);
77        }
78        uVar3 = 0x4c;
79        goto LAB_0220bf06;
80      }
81    }
82    FUN_e003b50c(5,0x26);
83    FUN_e00597c0(0);
```

unaff_gp => 未知的 gp register 值

gp 指向 data 段，有字串、函數指標，等資料

GP 肯定得在某個地方吧？

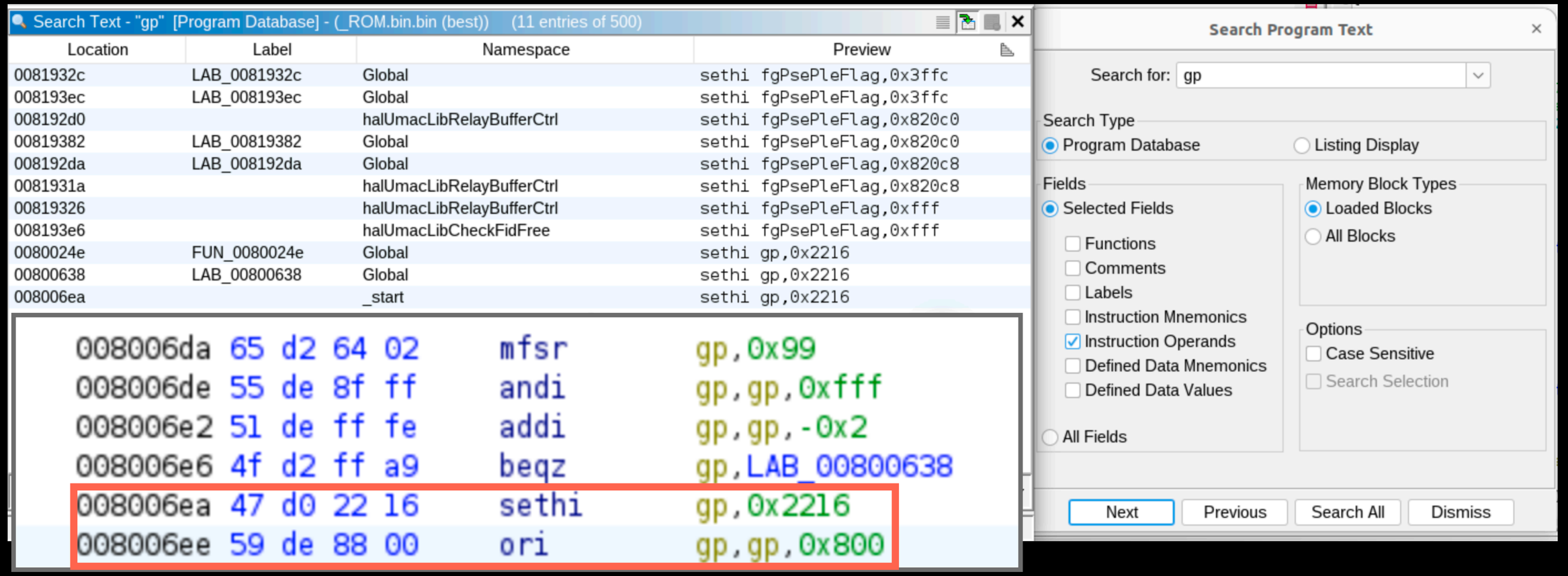
DEVCORE

Debug 指令

- WMCPU 的 coredump 指令
  - iwpriv wl0 show core_dump

```
/etc/_ROM.bin.bin       addr:0x800000
/etc/ULM1.bin           addr:0x900000
/etc/ULM2.bin           addr:0x2200000
/etc/ULM3.bin           addr:0x2300000
/etc/SRAM.bin           addr:0x400000
/etc/CRAM.bin           addr:0xe0000000
```

GP Register 設置



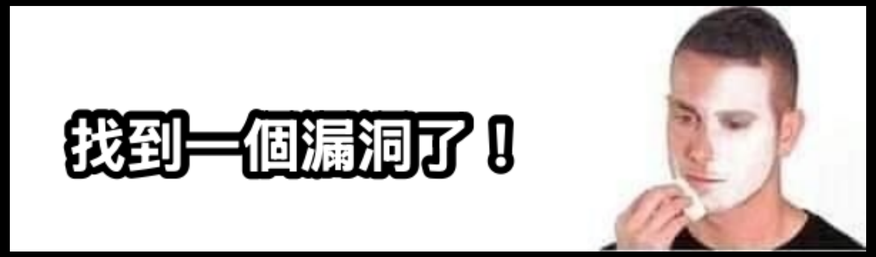
gp 暫存器會被初始化成 0x2216800

反編譯大成功

```
110   else {
111     iVar1 = FUN_e004c970();
112     if (iVar1 != 1) {
113       FUN_e003b50c(5,0x45);
114       iVar1 = FUN_e004c9f0();
115       if (iVar1 == 1) {
116         FUN_e00405d6();
117       }
118       goto LAB_0220be8c;
119     }
120     FUN_e0042482(0x15,4);
121     FUN_e004232c(s_Skip_send_notify_due_to_hif_susp_e00a6410);
122 LAB_0220be8e:
123     FUN_e003b50c(5,0x4a);
124     iVar1 = FUN_e003da16(&DAT_e00f9294,s_pwrCtrlInactivateAllHw_02231d8c,0x803);
125     if (iVar1 != 0) {
126       FUN_e003dbfc(&DAT_e00f9294,s_pwrCtrlInactivateAllHw_02231d8c,0x804);
127     }
128     FUN_e003b50c(5,0x4b);
129     FUN_e003dfd0(&DAT_e00f9294,5,s_pwrCtrlInactivateAllHw_02231d8c,0x807);
130   }
131   uVar2 = 0x4c;
132 LAB_0220bf06:
133   FUN_e003b50c(5,uVar2);
```

可讀字串

找到漏洞了!



找到漏洞了!!

MCU Exploit Mitigation:
- NX
- ASLR
- Stack Canary

找到漏洞了!!

MCU E[illegible]oit M[illegible]gation:
- NX
- ASLR
- Sta[illegible]ana

找到漏洞了!!!

找到漏洞了嗎???

找到漏洞了嗎????

設備韌體版本

- 我非常確定我的設備韌體已經更新到最新了
  - 最新的設備韌體在 2023/12 釋出

設備韌體版本

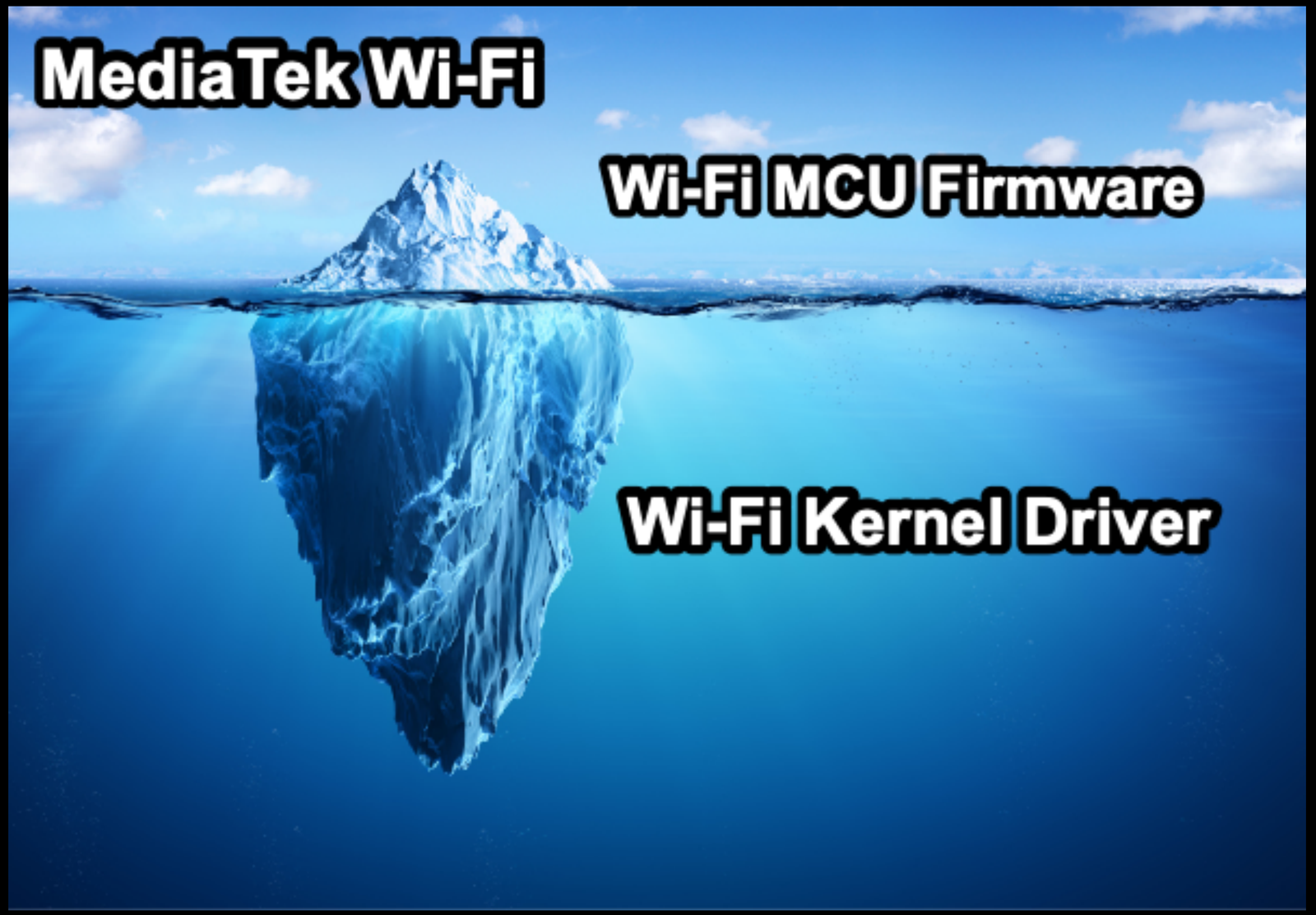
- 我非常確定我的設備韌體已經更新到最新了
  - 最新的設備韌體在 2023/12 釋出
- 但設備韌體裡面的 WMCPU 韌體編譯時間是 2023/07
  - 然而這個漏洞已經在 2023/10 修補了

設備韌體版本

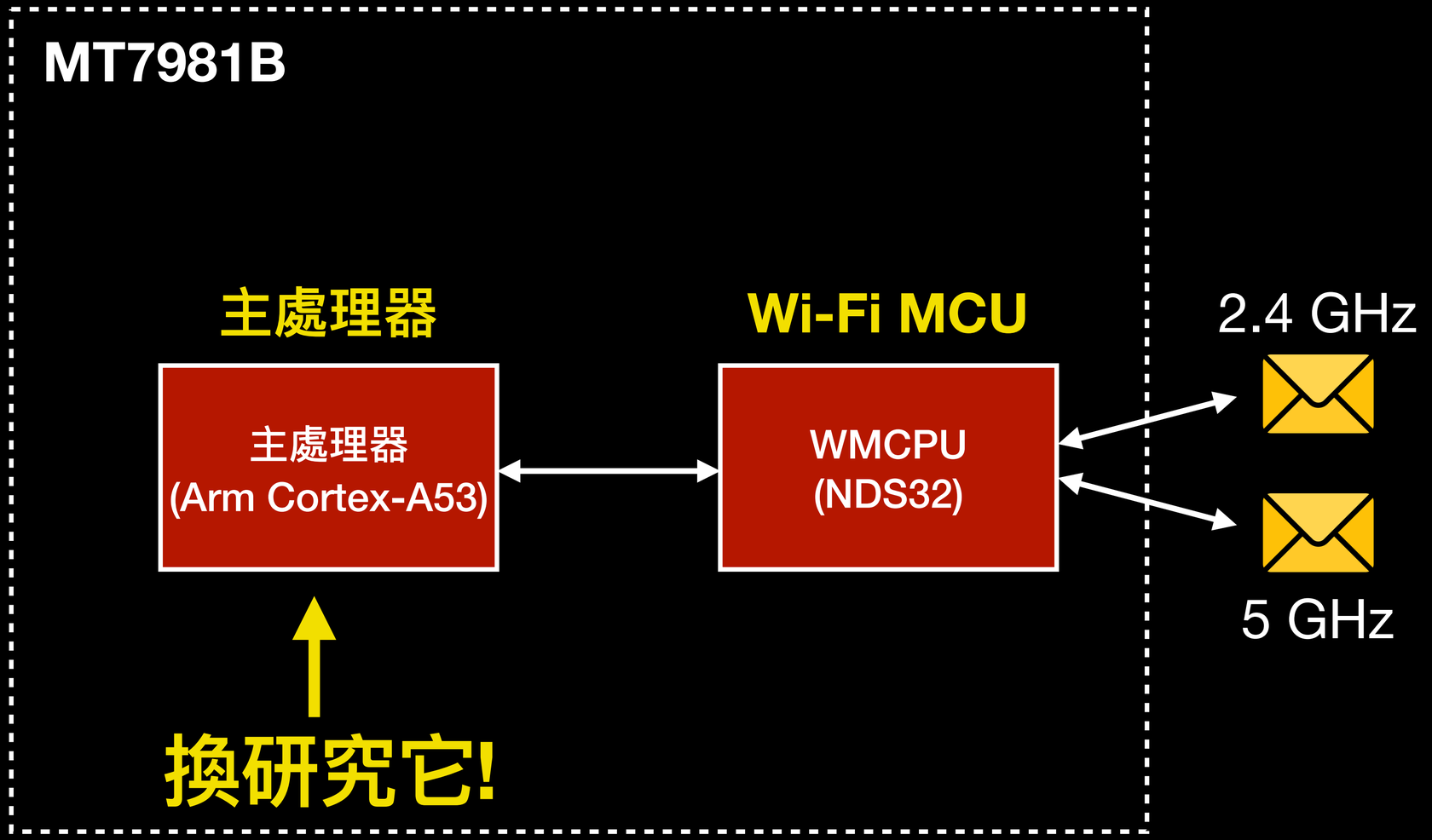
- 我非常確定我的
  - 最新的設備韌
- 但設備韌體裡面
  - 然而這個漏洞

小小的心得

- 設備廠商有得時候會忽略晶片廠商的修補
- 在我的研究過程中，一直遇到這樣的問題


MediaTek Wi-Fi
Wi-Fi MCU Firmware
Wi-Fi Kernel Driver

挖！繼續挖！


MT7981B
主處理器
Wi-Fi MCU
主處理器
(Arm Cortex-A53)
WMCPU
(NDS32)
2.4 GHz
5 GHz
換研究它!

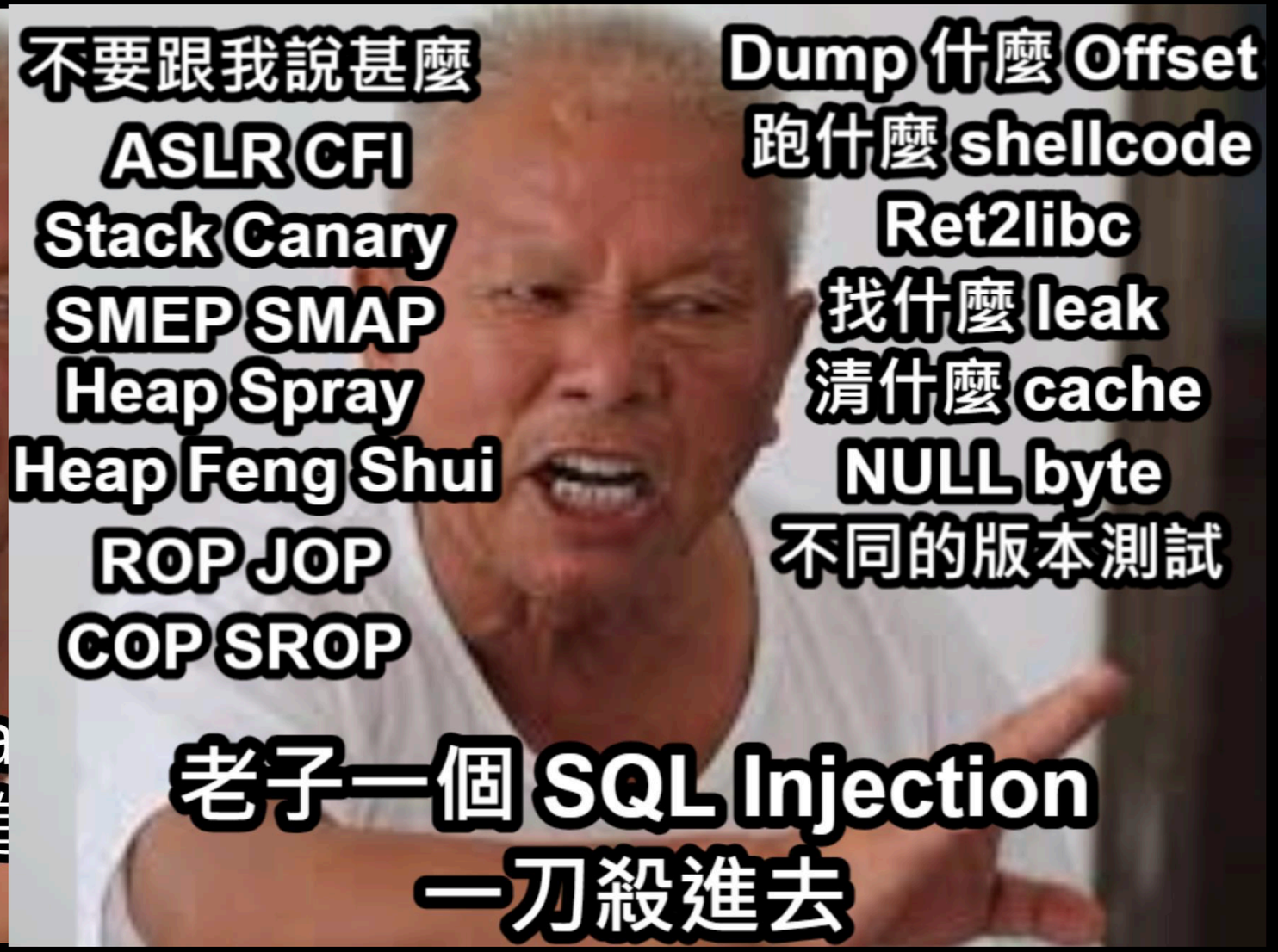
為什麼 Wi-Fi 連線要密碼驗證？

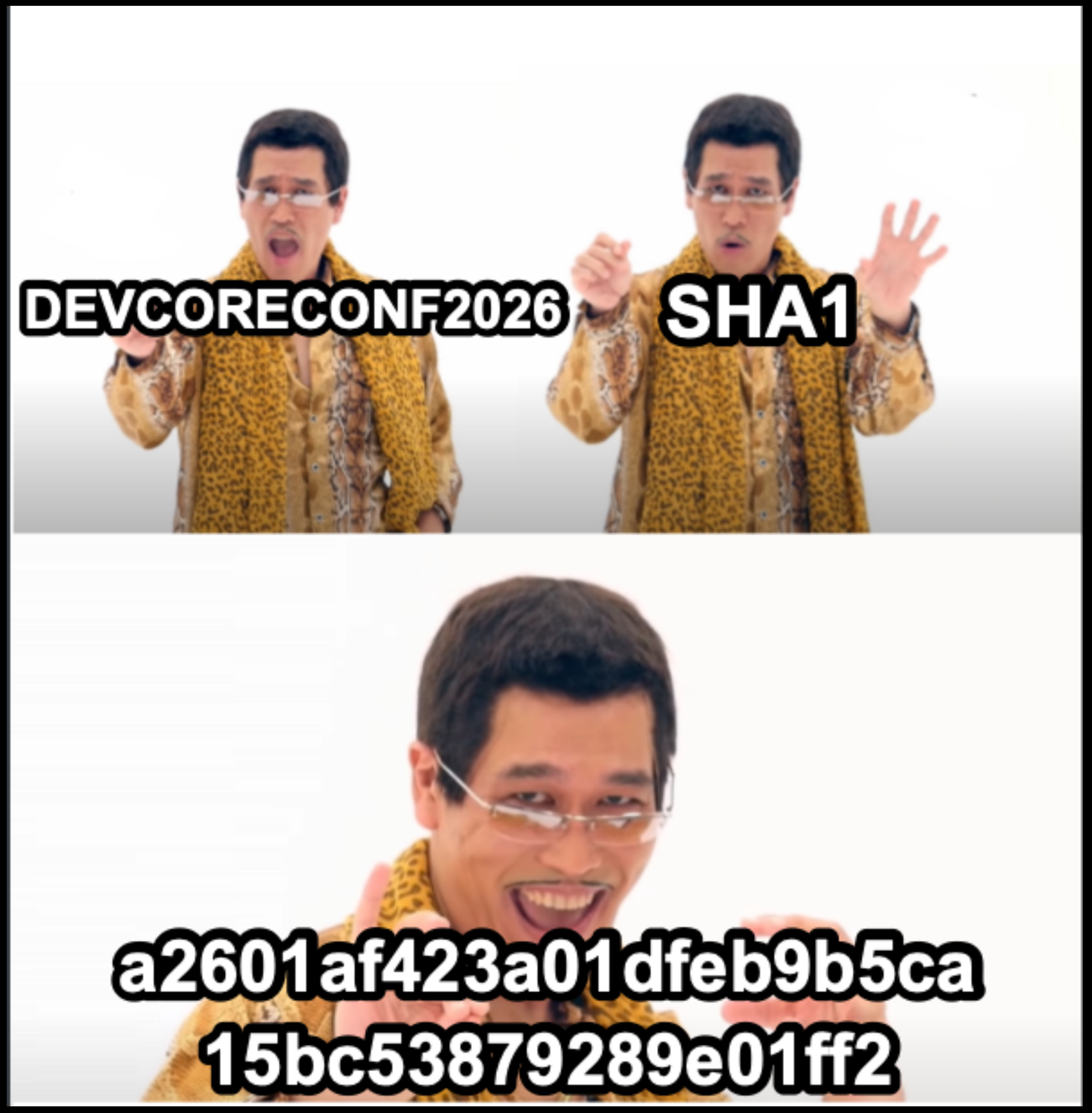
DEVCORE


對於大部分用戶而言
進入內網後就會像家人一樣，
打招呼之後進房間，大家相親相愛
DEVCORE SECURITY CONSULTING
By Orange Tsai

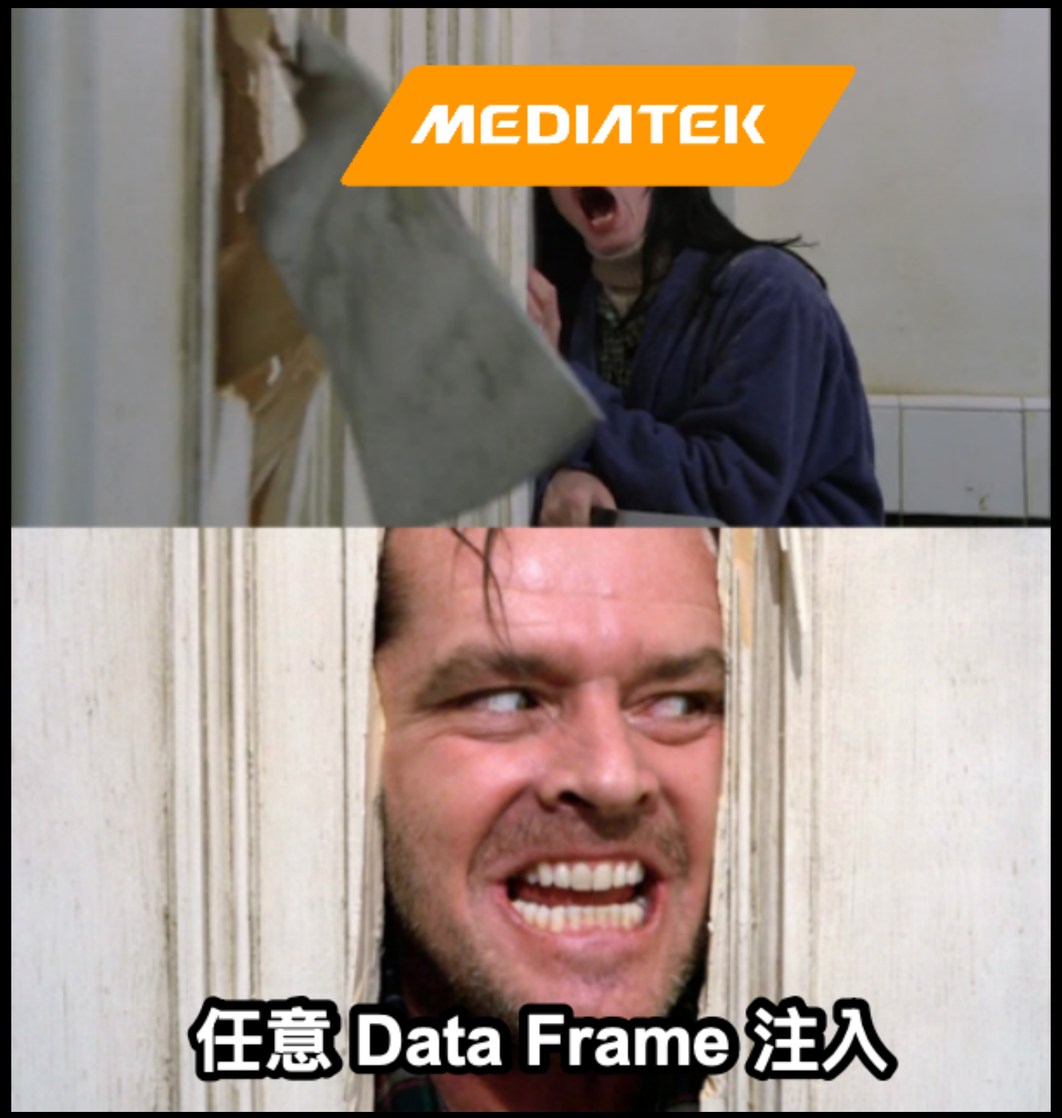

不要跟我說甚麼
ASLR CFI
Stack Canary
SMEP SMAP
Heap Spray
Heap Feng Shui
ROP JOP
COP SROP
Dump 什麼 Offset
跑什麼 shellcode
Ret2libc
找什麼 leak
NULL byte
清什麼 cache
不同的版本測試
老子一個 Command Injection
一刀殺進去


不要跟我說甚麼
ASLR CFI
Stack Canary
SMEP SMAP
Heap Spray
Heap Feng Shui
ROP JOP
COP SROP
老子一個 Comma
一刀殺進

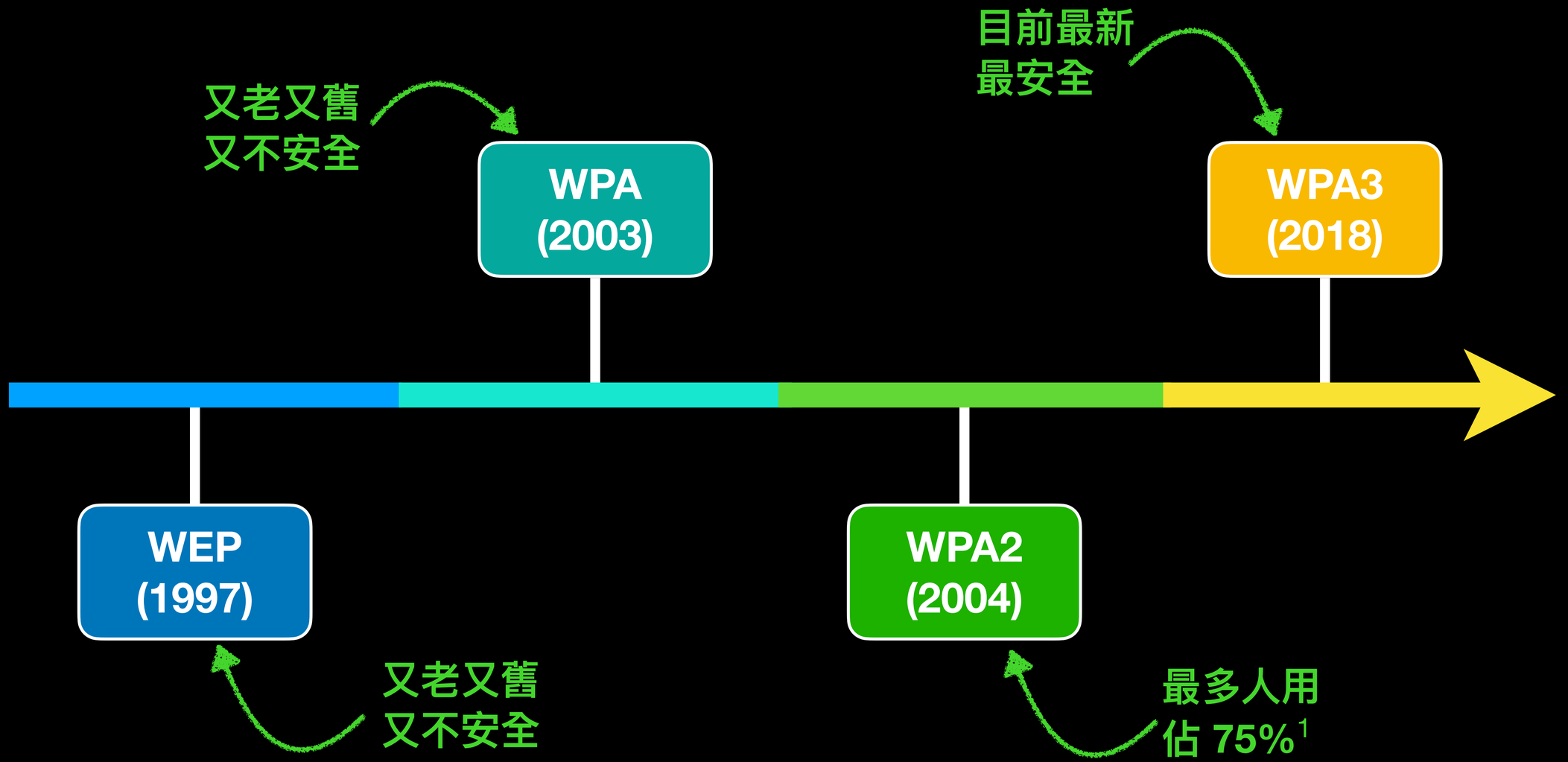

不要跟我說甚麼
ASLR CFI
Stack Canary
SMEP SMAP
Heap Spray
Heap Feng Shui
ROP JOP
COP SROP
Dump 什麼 Offset
跑什麼 shellcode
Ret2libc
找什麼 leak
清什麼 cache
NULL byte
不同的版本測試
老子一個 SQL Injection
一刀殺進去


DEVCORECONF2026
SHA1
a2601af423a01dfeb9b5ca
15bc53879289e01ff2

首殺：CVE-2025-20674

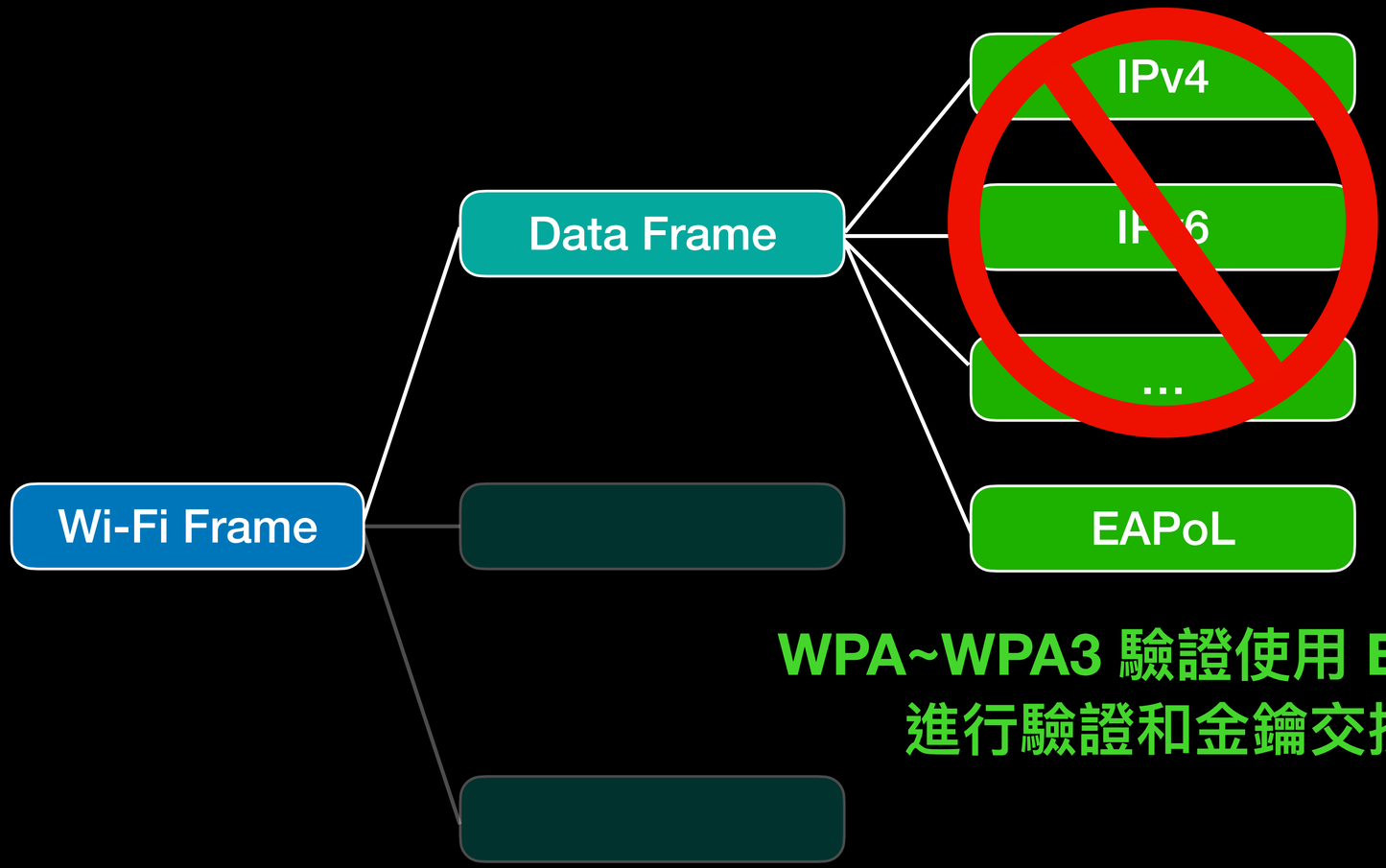


Wi-Fi 驗證協議



[1] https://wigle.net/stats

未驗證使用者

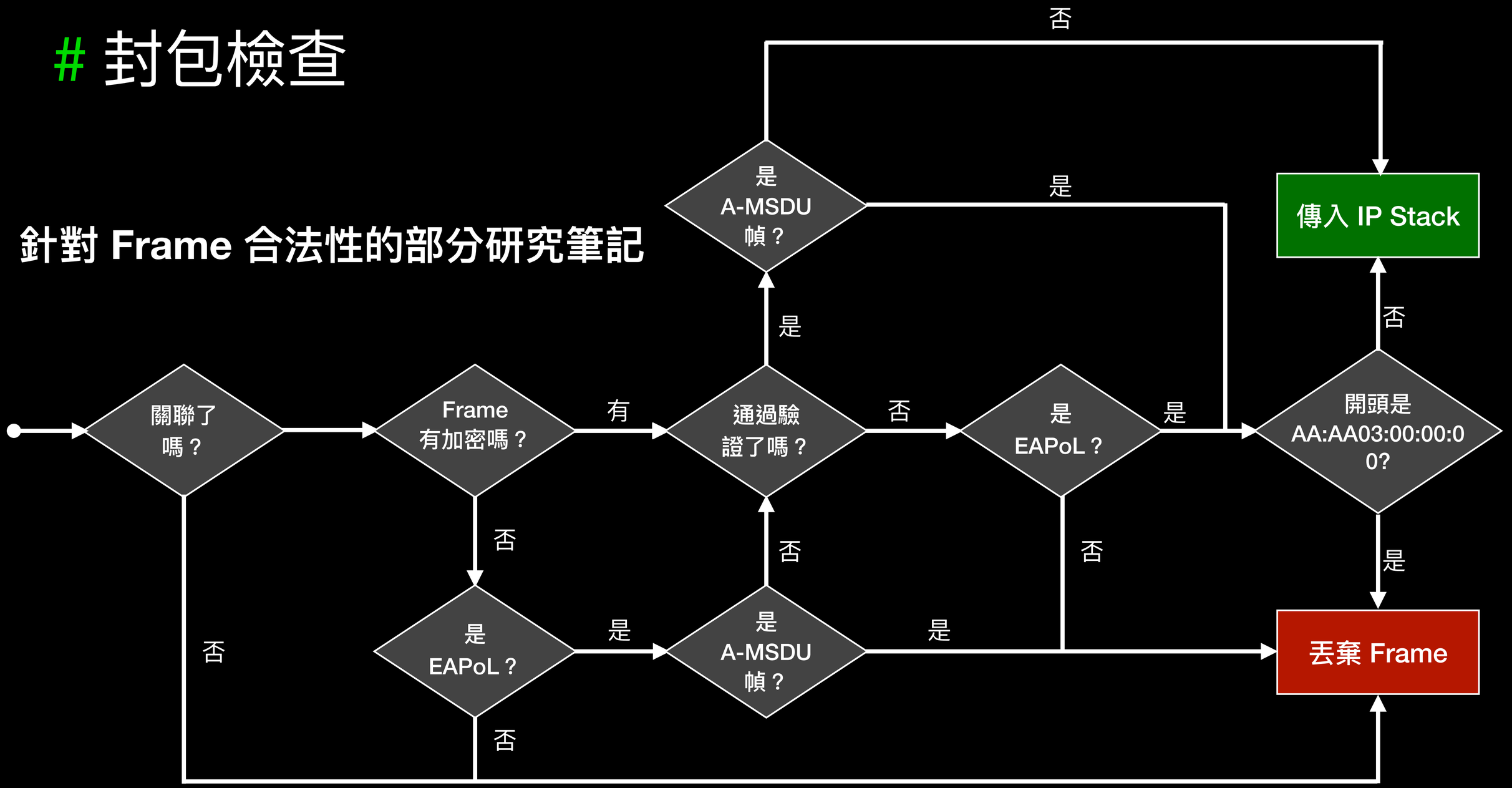


未驗證的使用者
只能傳送 EAPoL

WPA~WPA3 驗證使用 EAPoL
進行驗證和金鑰交換

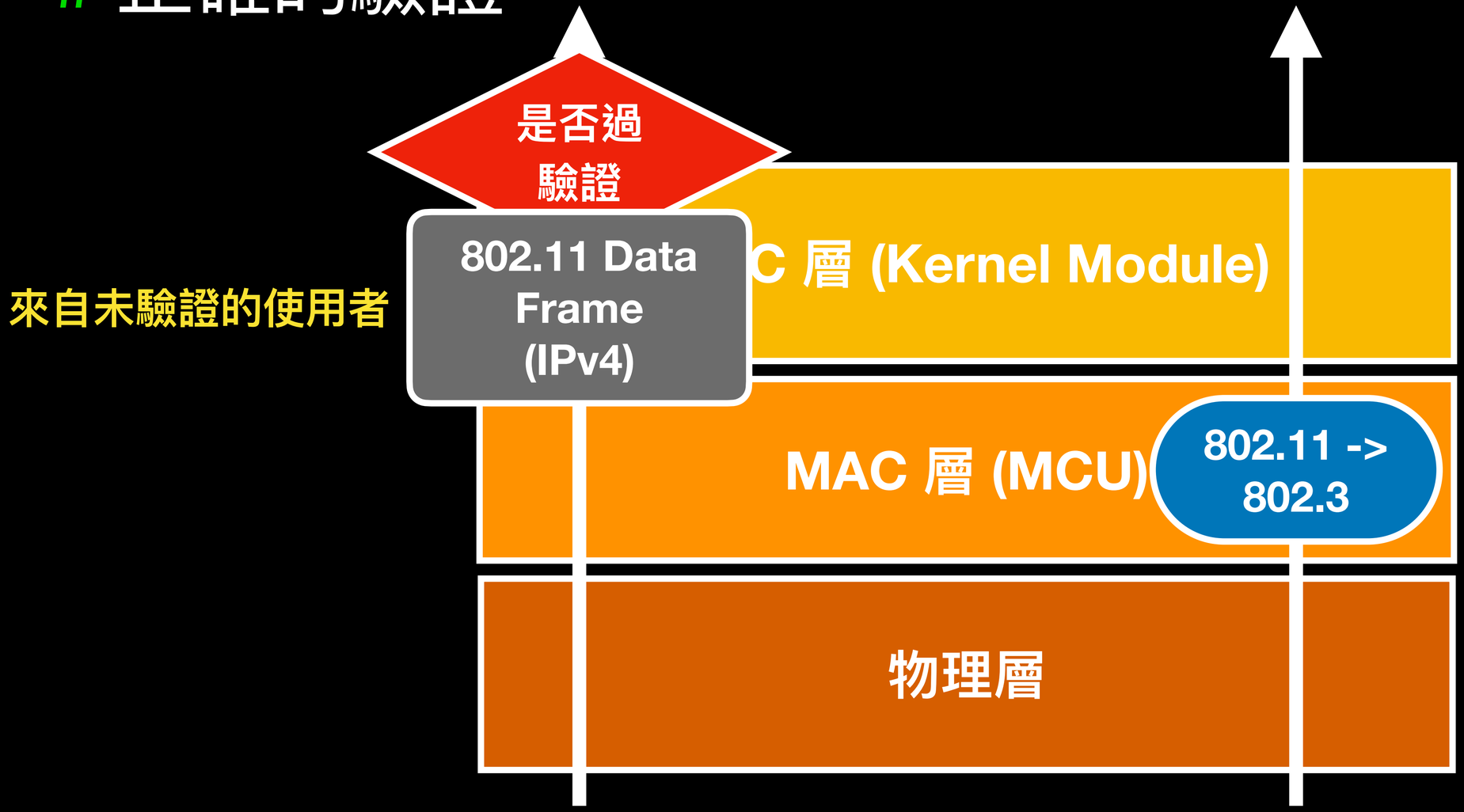
封包檢查

針對 Frame 合法性的部分研究筆記



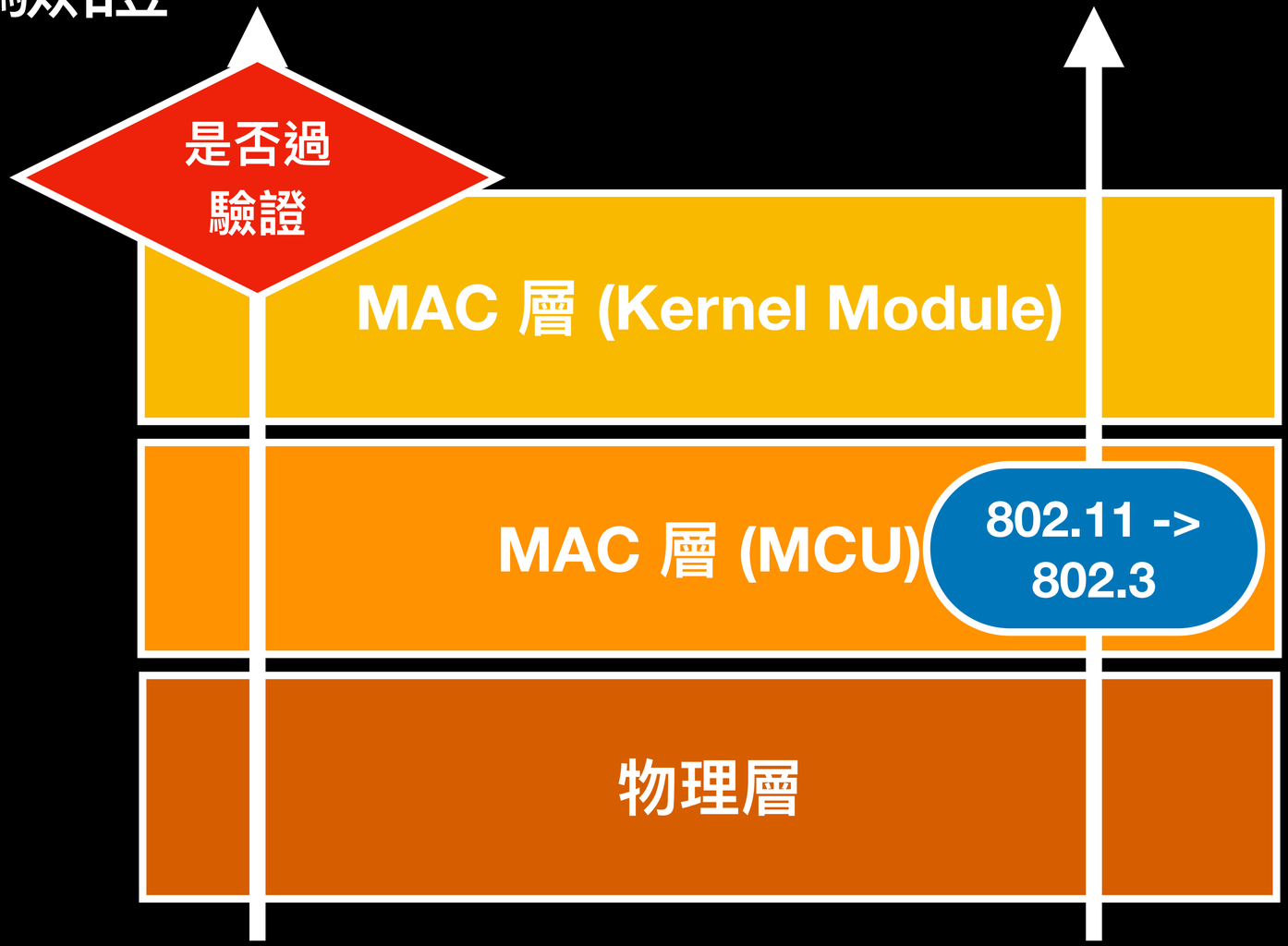

這麼多東西要檢查
肯定要有漏洞吧

正確的驗證

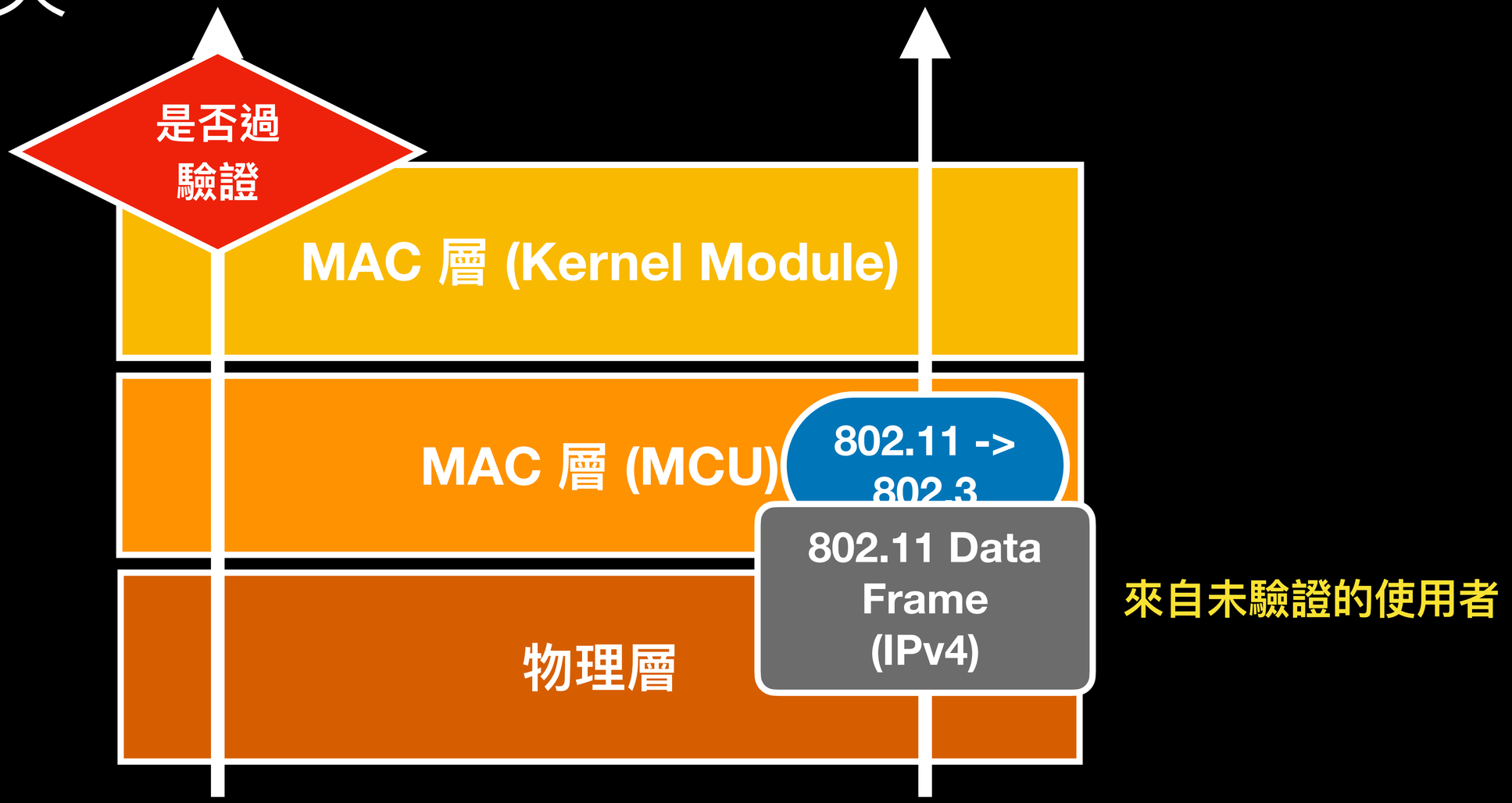

是否過
驗證
來自未驗證的使用者
802.11 Data
Frame
(IPv4)
C 層 (Kernel Module)
MAC 層 (MCU)
802.11 ->
802.3
物理層

正確的驗證

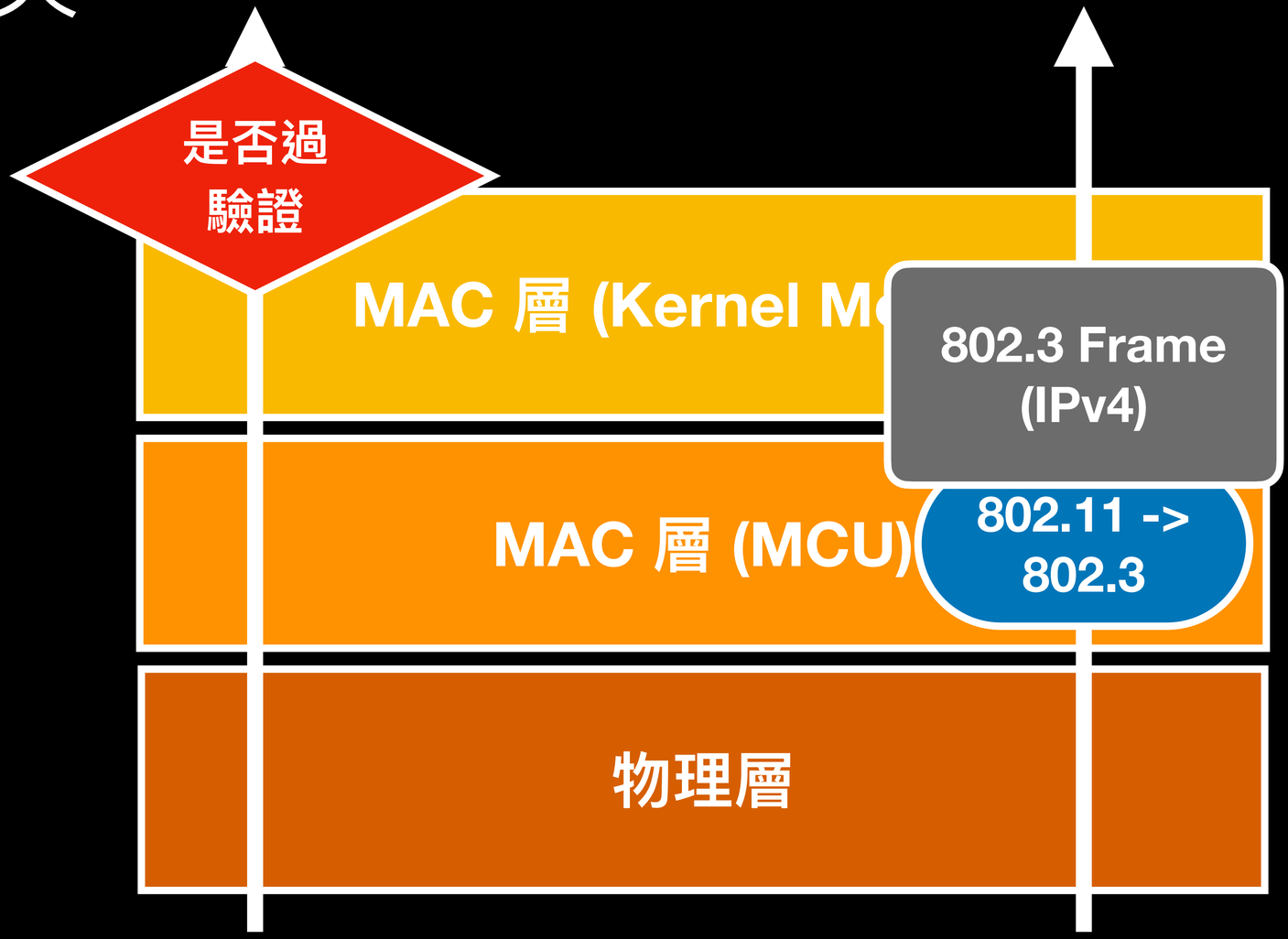
丟棄
來源未驗證的
Data Frame



驗證缺失

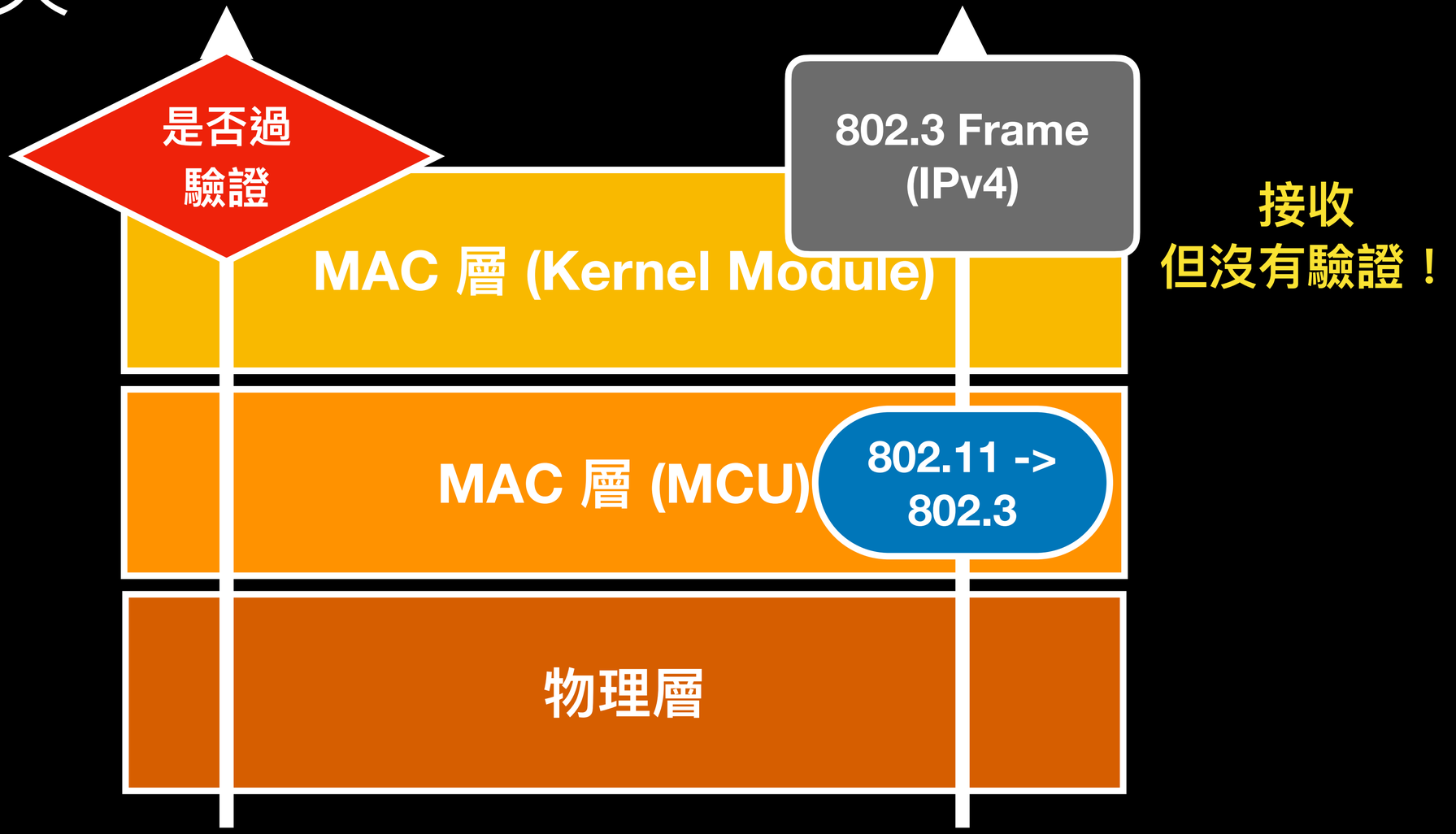

是否過
驗證
MAC 層 (Kernel Module)
MAC 層 (MCU)
802.11 ->
802.3
802.11 Data
Frame
(IPv4)
物理層
來自未驗證的使用者

驗證缺失



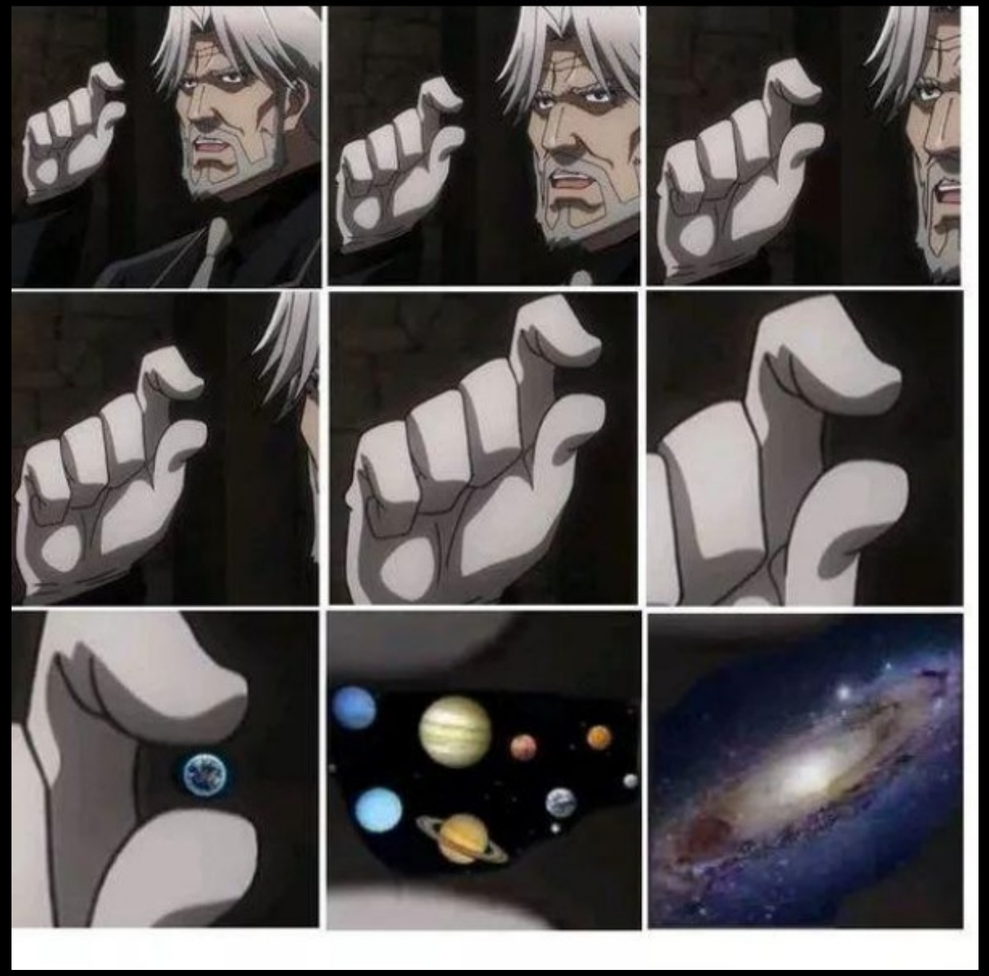
將 Wi-Fi Data Frame 轉換成 Ethernet Frame

驗證缺失

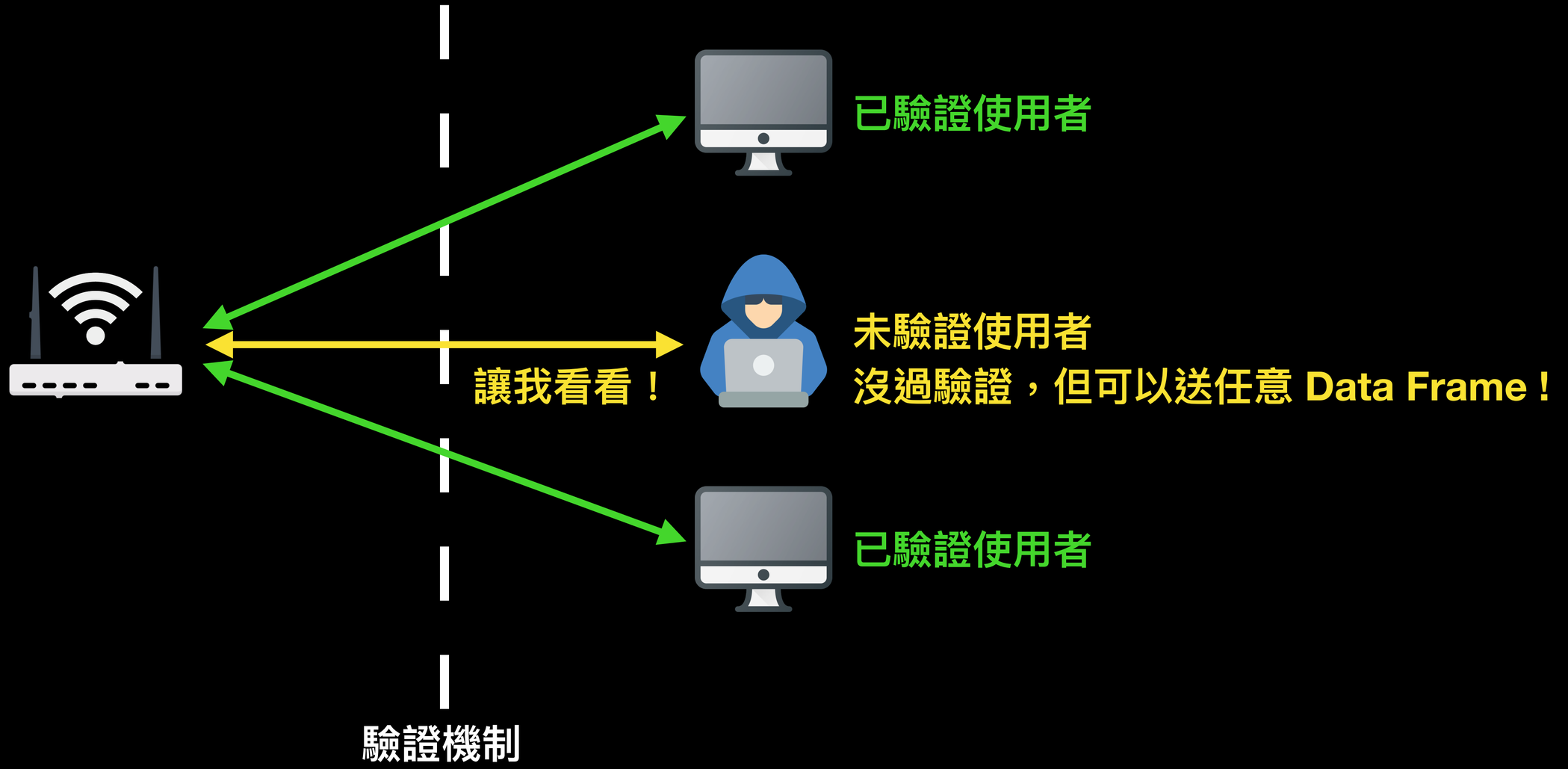


接收
但沒有驗證！

看到沒檢查的時候
我只有億點點驚訝



Wi-Fi (沒有) 驗證



利用情境

DEVCORE

攻擊情境

AWS EC2
(EC2)

NAT

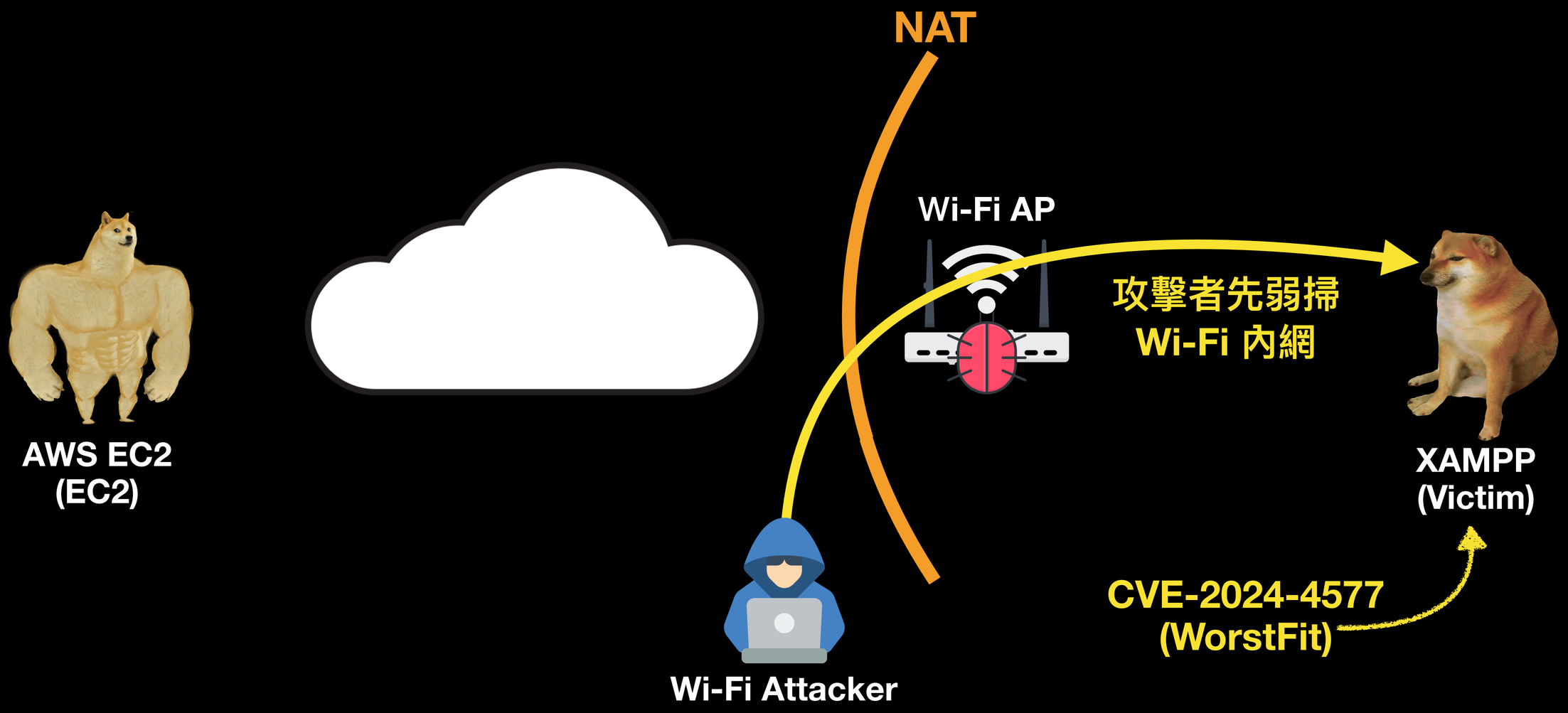
CVE-2025-20674
(Wi-Fi 任意 Data Frame
注入)

Wi-Fi AP

XAMPP
(Victim)

Wi-Fi Attacker

攻擊情境


NAT
Wi-Fi AP
攻擊者先弱掃
Wi-Fi 內網
AWS EC2
(EC2)
XAMPP
(Victim)
CVE-2024-4577
(WorstFit)
Wi-Fi Attacker

NAT 打洞

AWS EC2
(EC2)

NAT

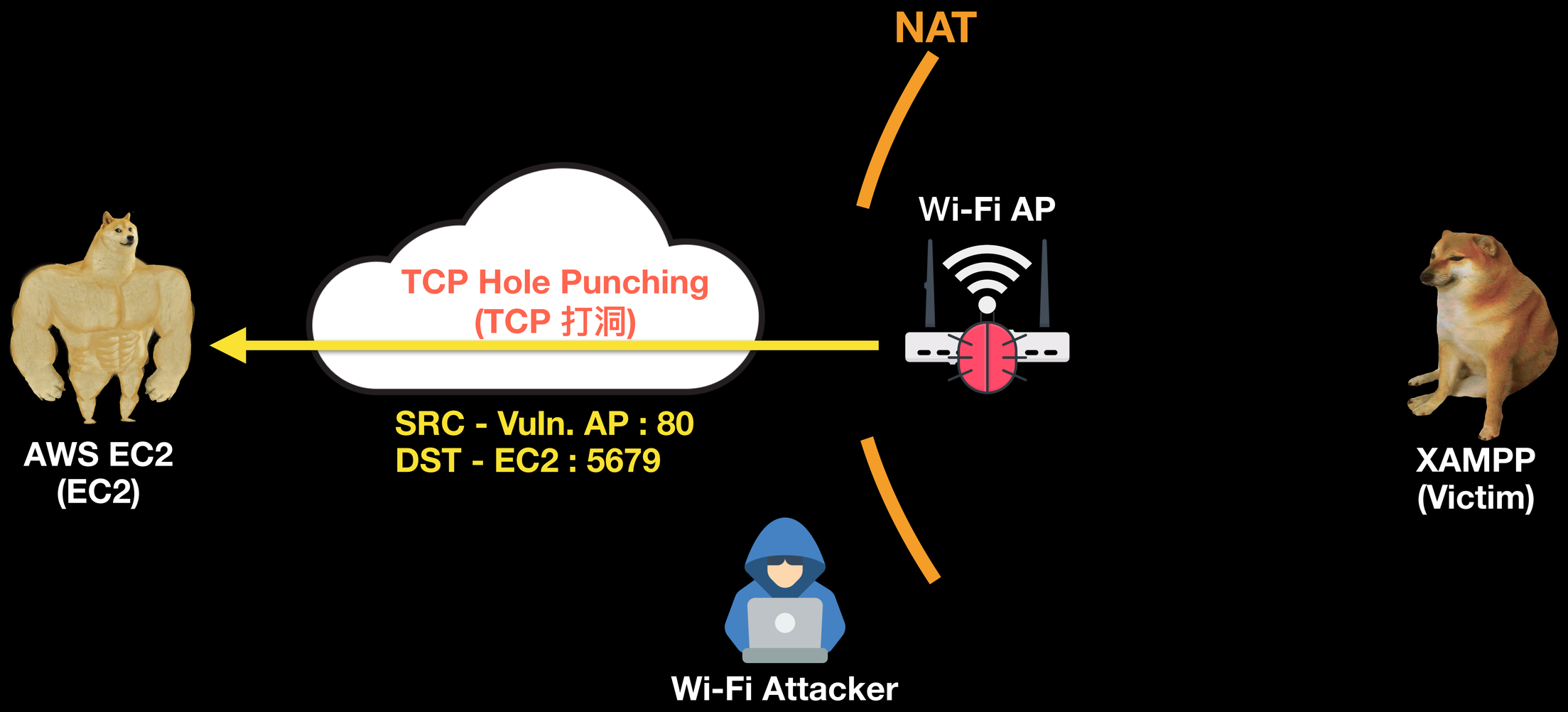
Wi-Fi AP

SRC - Victim : 80
DST - EC2 : 5679
TCP SYN

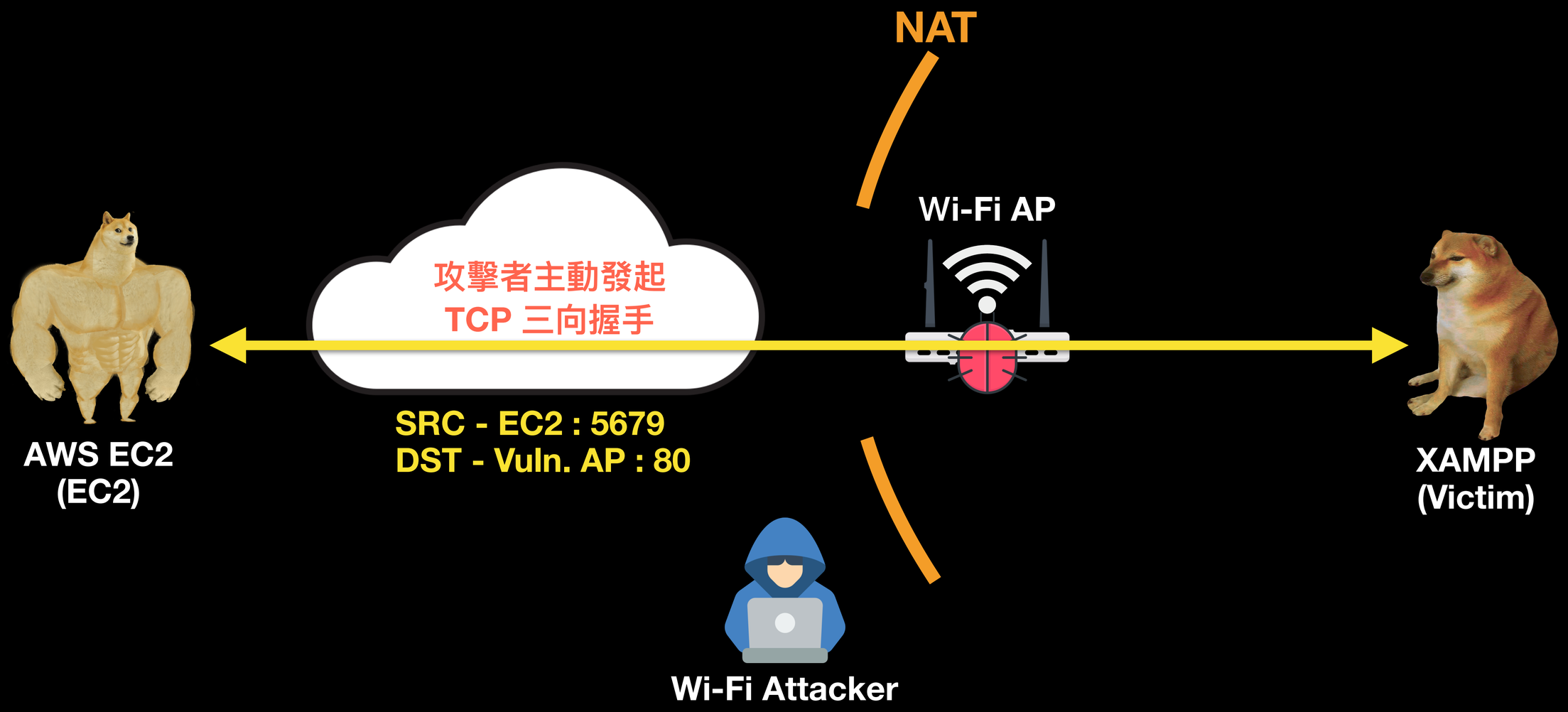
Wi-Fi Attacker

XAMPP
(Victim)

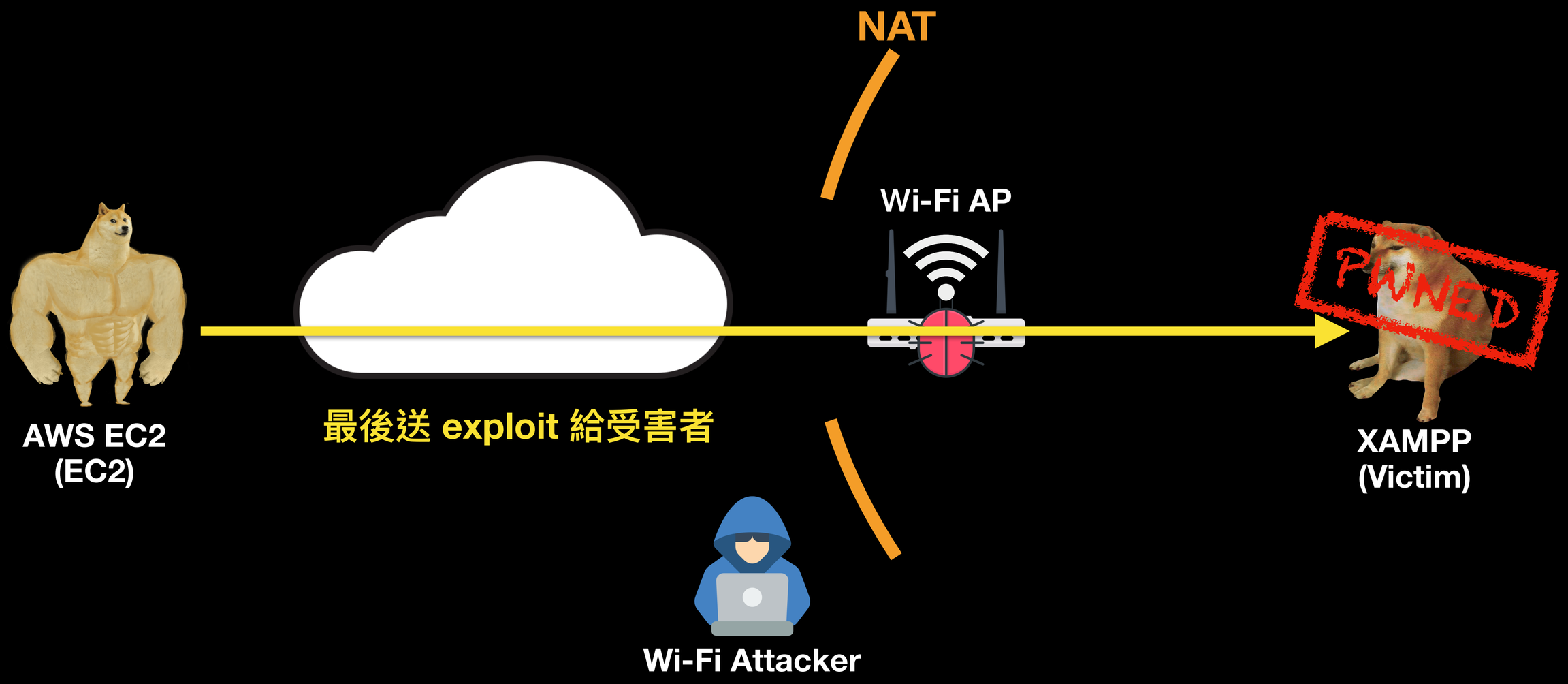
WorstFit 發射！


NAT
Wi-Fi AP
TCP Hole Punching
(TCP 打洞)
SRC - Vuln. AP : 80
DST - EC2 : 5679
AWS EC2
(EC2)
XAMPP
(Victim)
Wi-Fi Attacker

WorstFit 發射！


NAT
Wi-Fi AP
攻擊者主動發起
TCP 三向握手
SRC - EC2 : 5679
DST - Vuln. AP : 80
AWS EC2
(EC2)
XAMPP
(Victim)
Wi-Fi Attacker

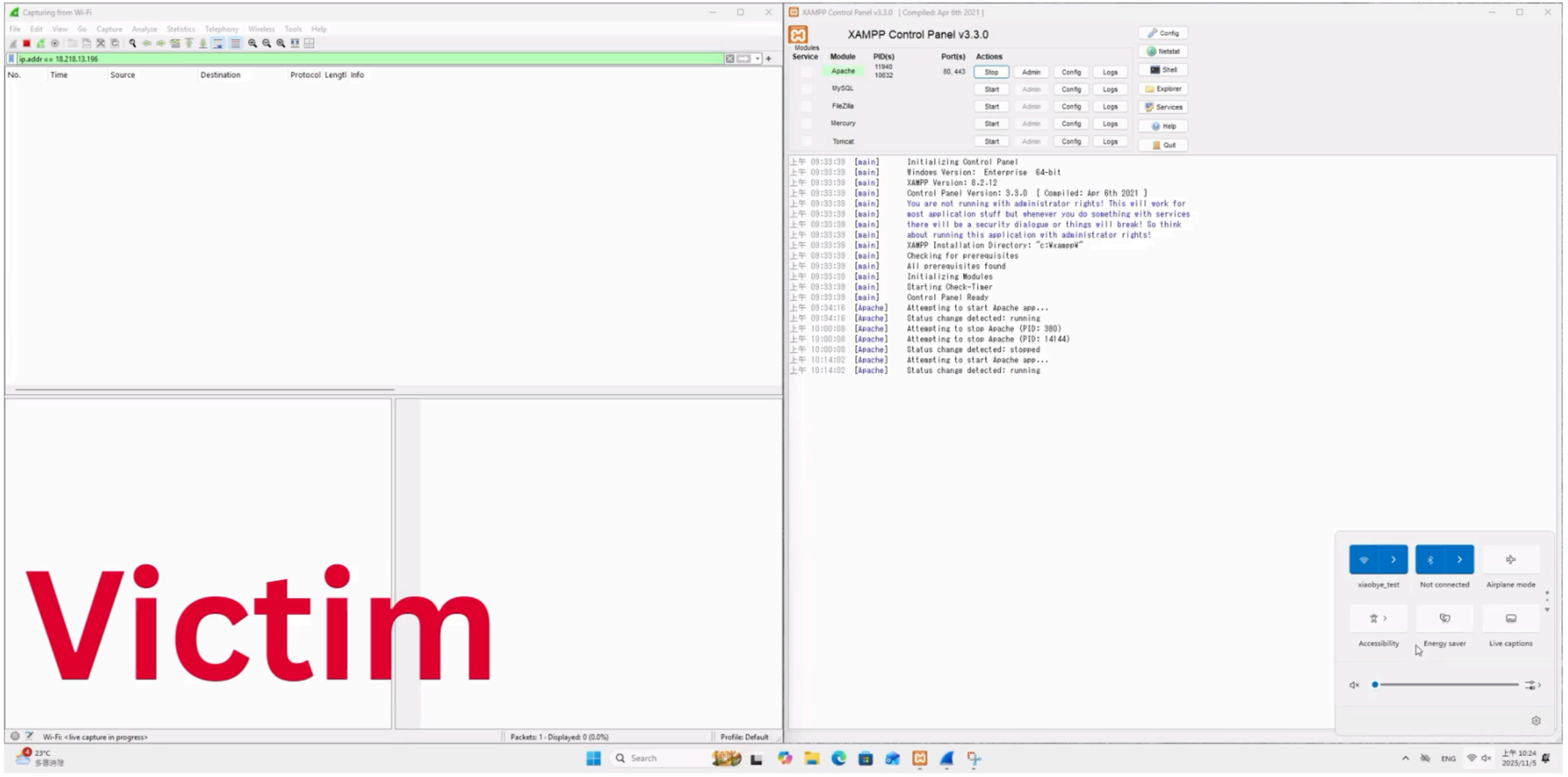
WorstFit 發射！


NAT
Wi-Fi AP
PWNED
AWS EC2
(EC2)
最後送 exploit 給受害者
XAMPP
(Victim)
Wi-Fi Attacker

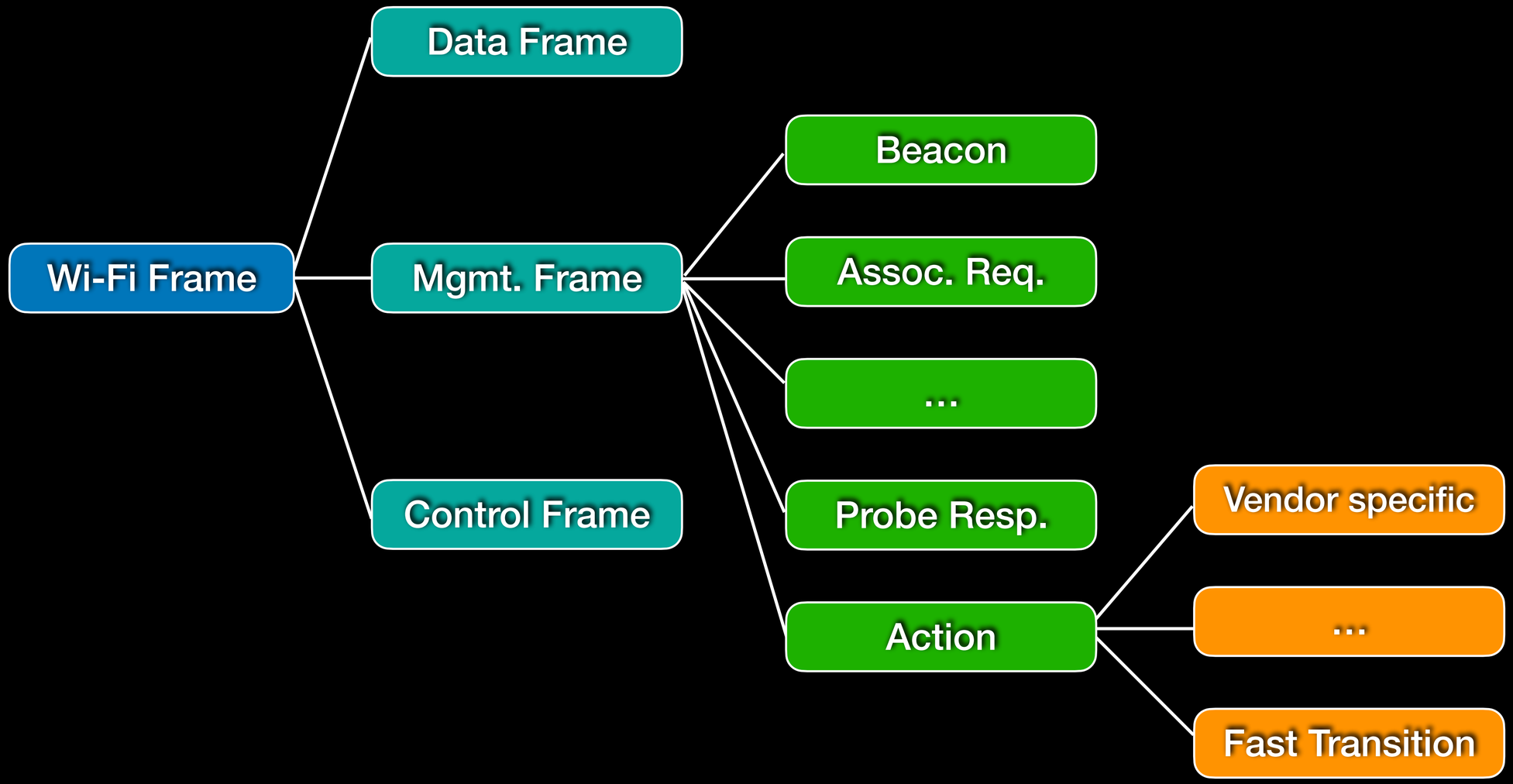
我們的內網

我:

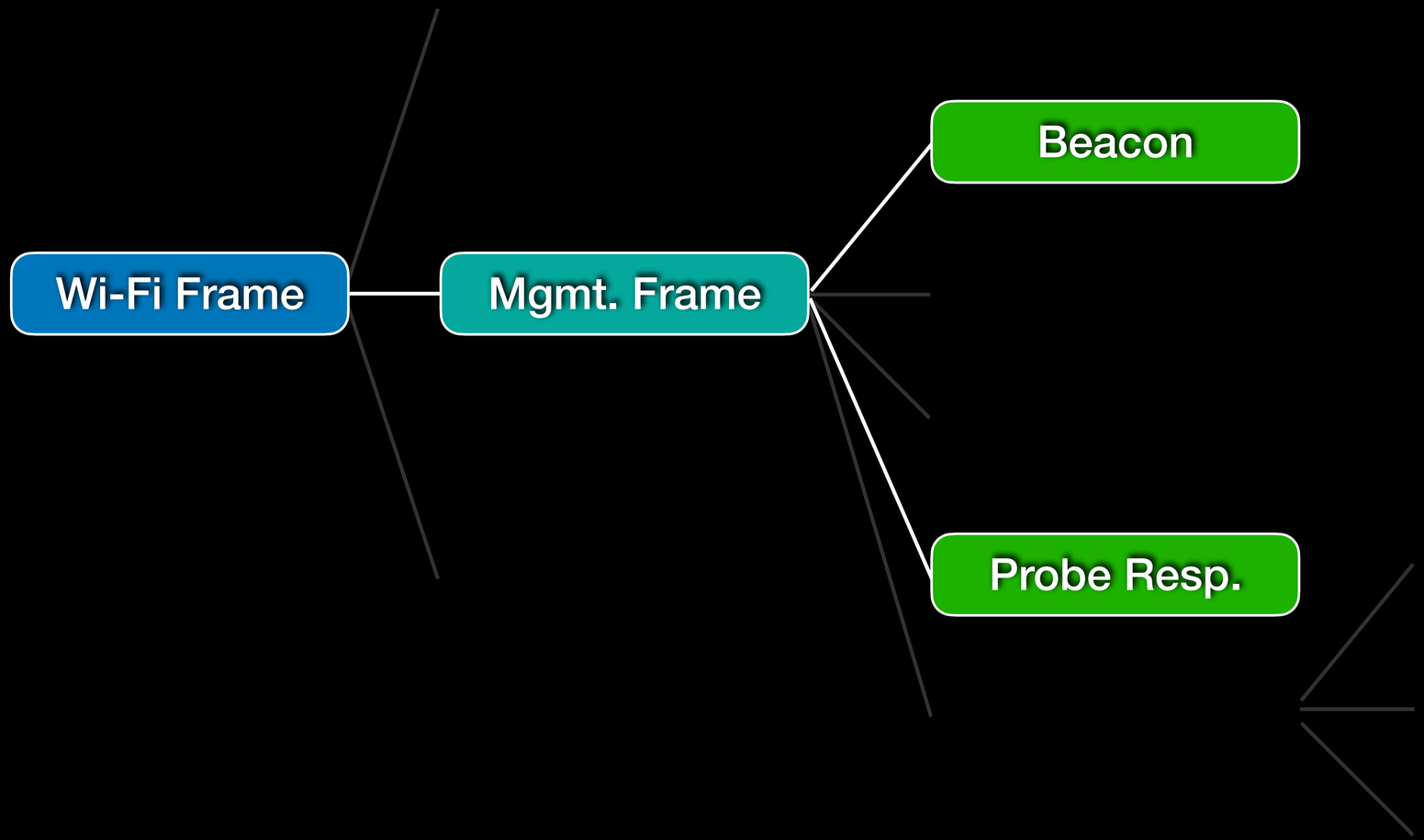
大內網 homelab
弱密碼設備
一堆洞的設備
自己寫的服務


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 18.218.13.196
No. Time Source Destination Protocol Lengtl Info
Wi-Fi: <live capture in progress>
Packets: 1 · Displayed: 0 (0.0%)
Profile: Default
XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service Module PID(s) Port(s) Actions
Apache 11940 10632 80, 443 Stop Admin Config Logs
MySQL Start Admin Config Logs
FileZilla Start Admin Config Logs
Mercury Start Admin Config Logs
Tomcat Start Admin Config Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
上午 09:33:39 [main] Initializing Control Panel
上午 09:33:39 [main] Windows Version: Enterprise 64-bit
上午 09:33:39 [main] XAMPP Version: 8.2.12
上午 09:33:39 [main] Control Panel Version: 3.3.0 [Compiled: Apr 6th 2021]
上午 09:33:39 [main] You are not running with administrator rights! This will work for
上午 09:33:39 [main] most application stuff but whenever you do something with services
上午 09:33:39 [main] there will be a security dialogue or things will break! So think
上午 09:33:39 [main] about running this application with administrator rights!
上午 09:33:39 [main] XAMPP Installation Directory: "c:¥xampp¥"
上午 09:33:39 [main] Checking for prerequisites
上午 09:33:39 [main] All prerequisites found
上午 09:33:39 [main] Initializing Modules
上午 09:33:39 [main] Starting Check-Timer
上午 09:33:39 [main] Control Panel Ready
上午 09:34:16 [Apache] Attempting to start Apache app...
上午 09:34:16 [Apache] Status change detected: running
上午 10:00:08 [Apache] Attempting to stop Apache (PID: 380)
上午 10:00:08 [Apache] Attempting to stop Apache (PID: 14144)
上午 10:00:08 [Apache] Status change detected: stopped
上午 10:14:02 [Apache] Attempting to start Apache app...
上午 10:14:02 [Apache] Status change detected: running
xiaobye_test
Not connected
Airplane mode
Accessibility
Energy saver
Live captions
Victim
23°C
Search
ENG
上午 10:24
2025/11/5


這樣你就滿足了嗎？

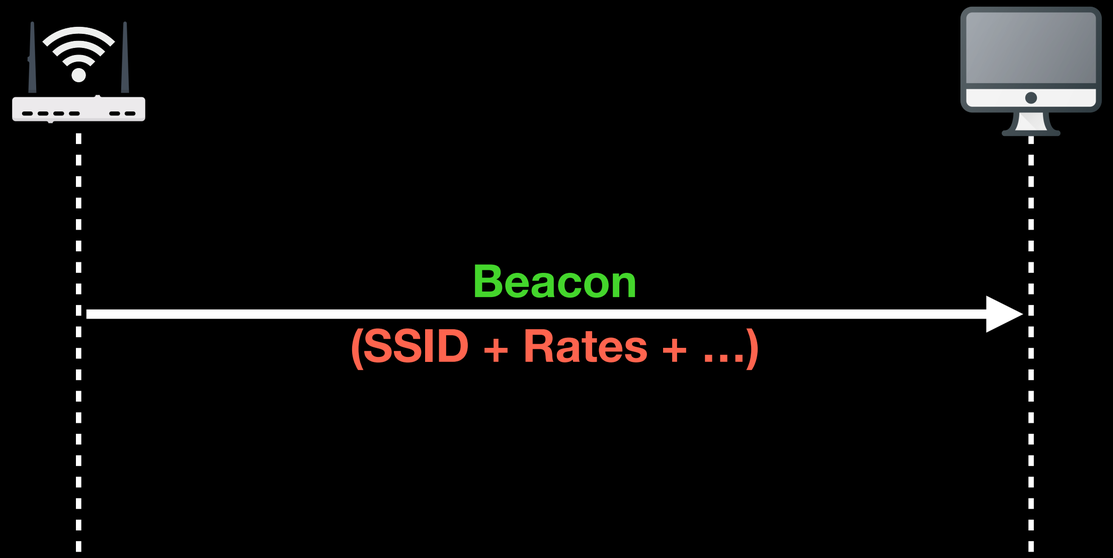

Wi-Fi Frame
Data Frame
Mgmt. Frame
Control Frame
Beacon
Assoc. Req.
...
Probe Resp.
Action
Vendor specific
...
Fast Transition


他一定在想其他女人
研討會要講哪個洞


Wi-Fi Frame
Mgmt. Frame
Beacon
Probe Resp.

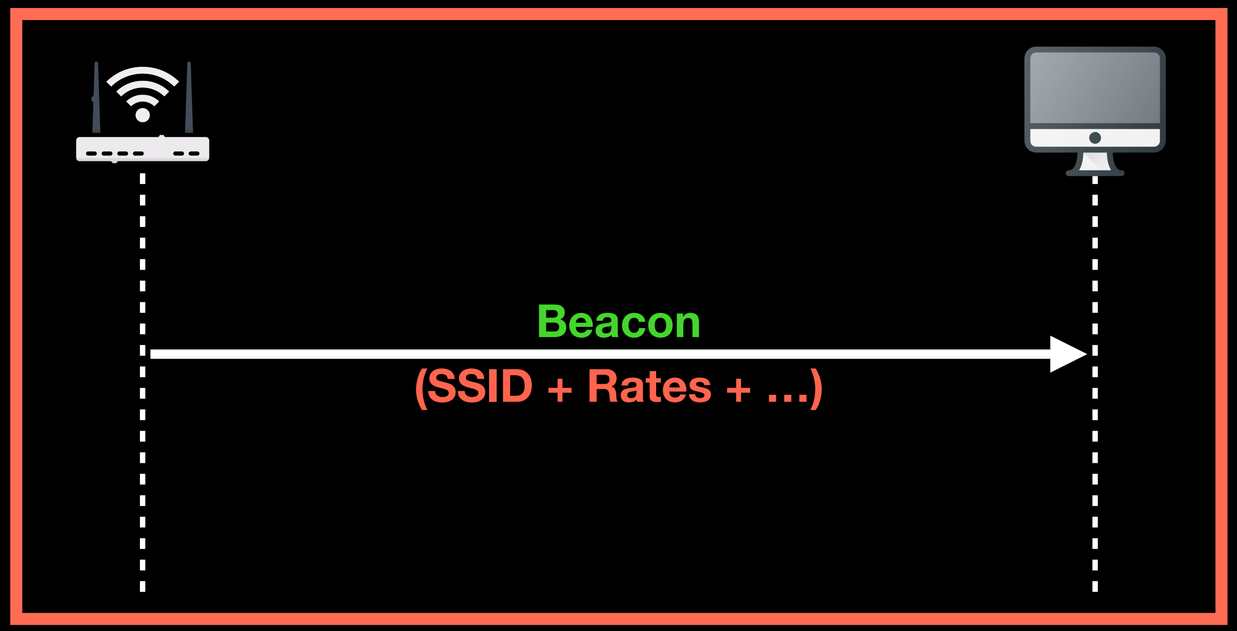
Wi-Fi 掃描

- 主動掃描
  - Wi-Fi 設備傳送 Probe Frame 探測指定AP是否存在
- 被動掃描
  - Wi-Fi AP 定期廣播 Beacon Frame，告知週邊設備自身的存在



Wi-Fi 掃描

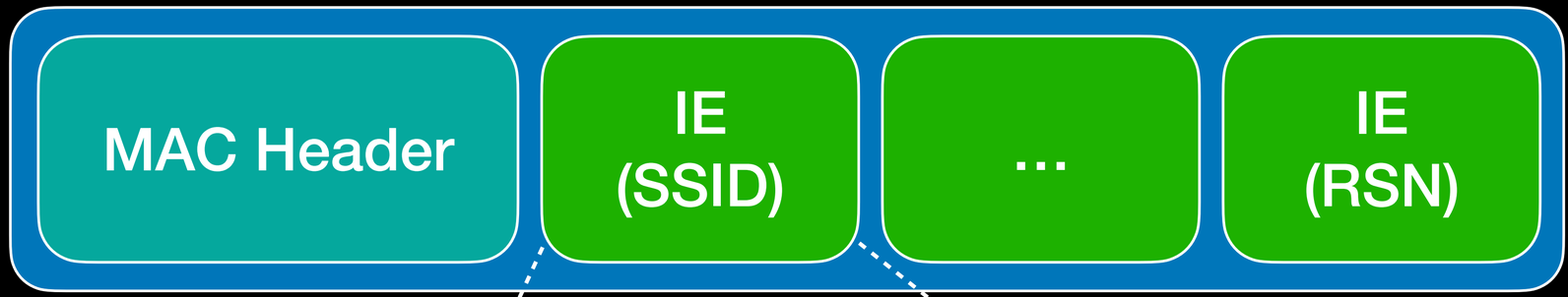
- 主動掃描
  - Wi-Fi 設備傳送 Probe Frame 探測指定AP是否存在
- 被動掃描
  - Wi-Fi AP 定期廣播 Beacon Frame，告知週邊設備自身的存在



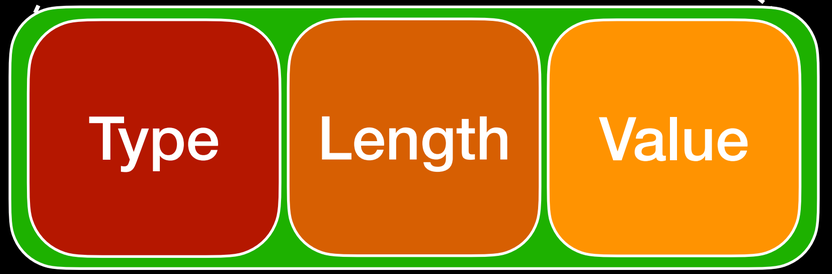
CVE-2025-20685

如何交換設定和資訊?

Beacon



Information Element (IE)



成因分析

```
int PeerBeaconAndProbeRspSanity(...){
    while(...){
        // Check if ie_buf can store ie
        switch(ie->type){
            case ...:
                ...
            case 244:
                // Copy ie into ie_buf
            case 221:
                // Check if wsc can store ie
                // Copy ie into wsc
        }
    }
    // Copy wsc into ie_buf
}
```

成因分析

```
int PeerBeaconAndProbeRspSanity(...){
    while(...){
        // Check if ie_buf can store ie
        switch(ie->type){
            case ...:
                ...
            case 244:
                // Copy ie into ie_buf
            case 221:
                // Check if wsc can store ie
                // Copy ie into wsc
        }
    }
    // Copy wsc into ie_buf
}
```

成因分析

ie_buf: 1024 bytes

```
int PeerBeaconAndProbeRspSanity(...){
    while(...){
        // Check if ie_buf can store ie
        switch(...){
            case ...:
                ...
            case 244:
                // Copy ie into ie_buf
            case 221:
                // Check if wsc can store ie
                // Copy ie into wsc
        }
    }
    // Copy wsc into ie_buf
}
```

成因分析

ie_buf: 1024 bytes
wsc: 512 bytes

```
int PeerBeaconAndProbeRspSanity(...){
    while(...){
        // Check if ie_buf can store ie
        switch(ie->type){
case 244:
    // Copy ie into ie_buf
case 221:
    // Check if wsc can store ie
    // Copy ie into wsc
        }
        // Copy wsc into ie_buf
}
```

但是

DEVCORE

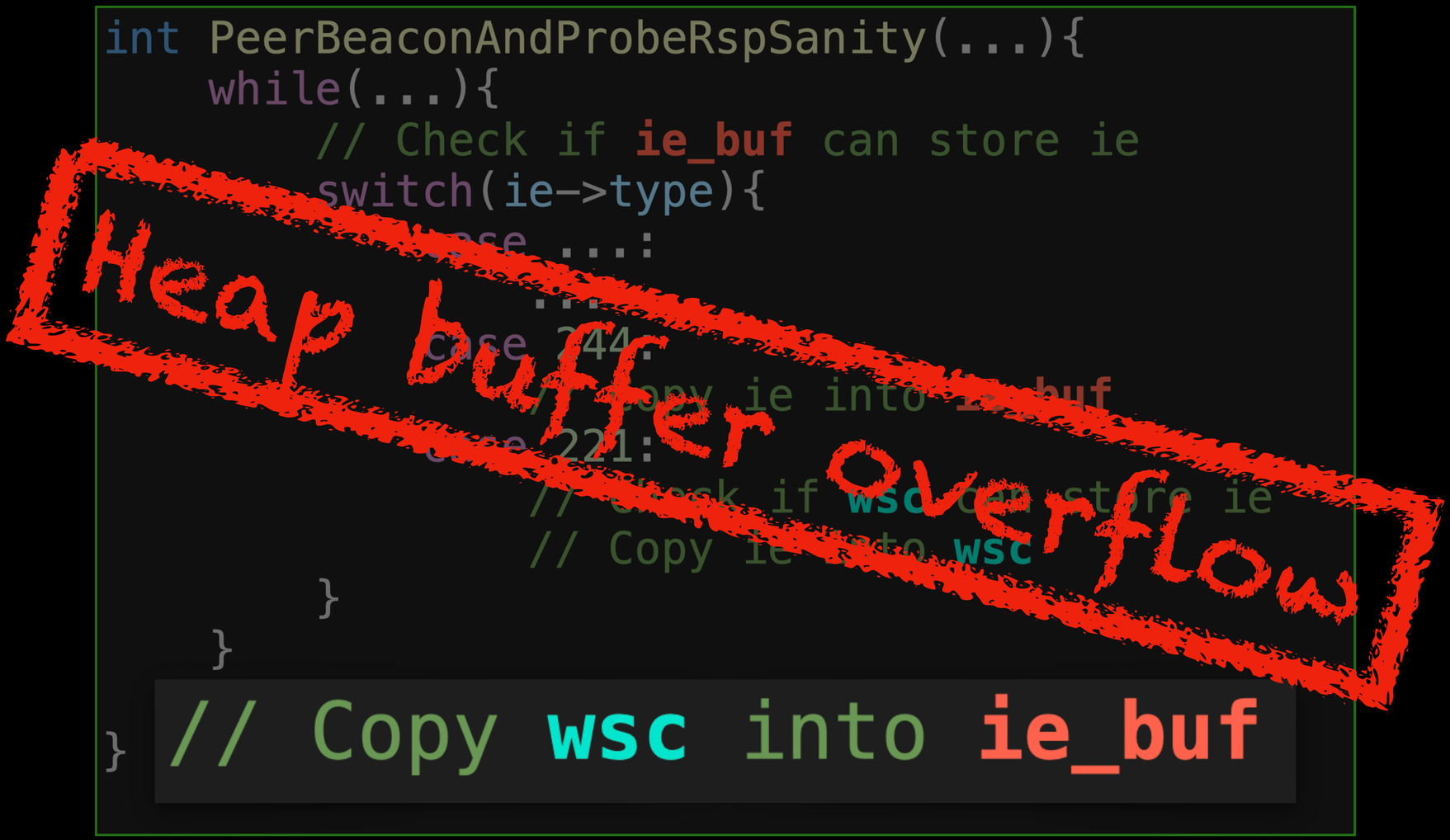
成因分析

ie_buf: 1024 bytes
wsc: 512 bytes

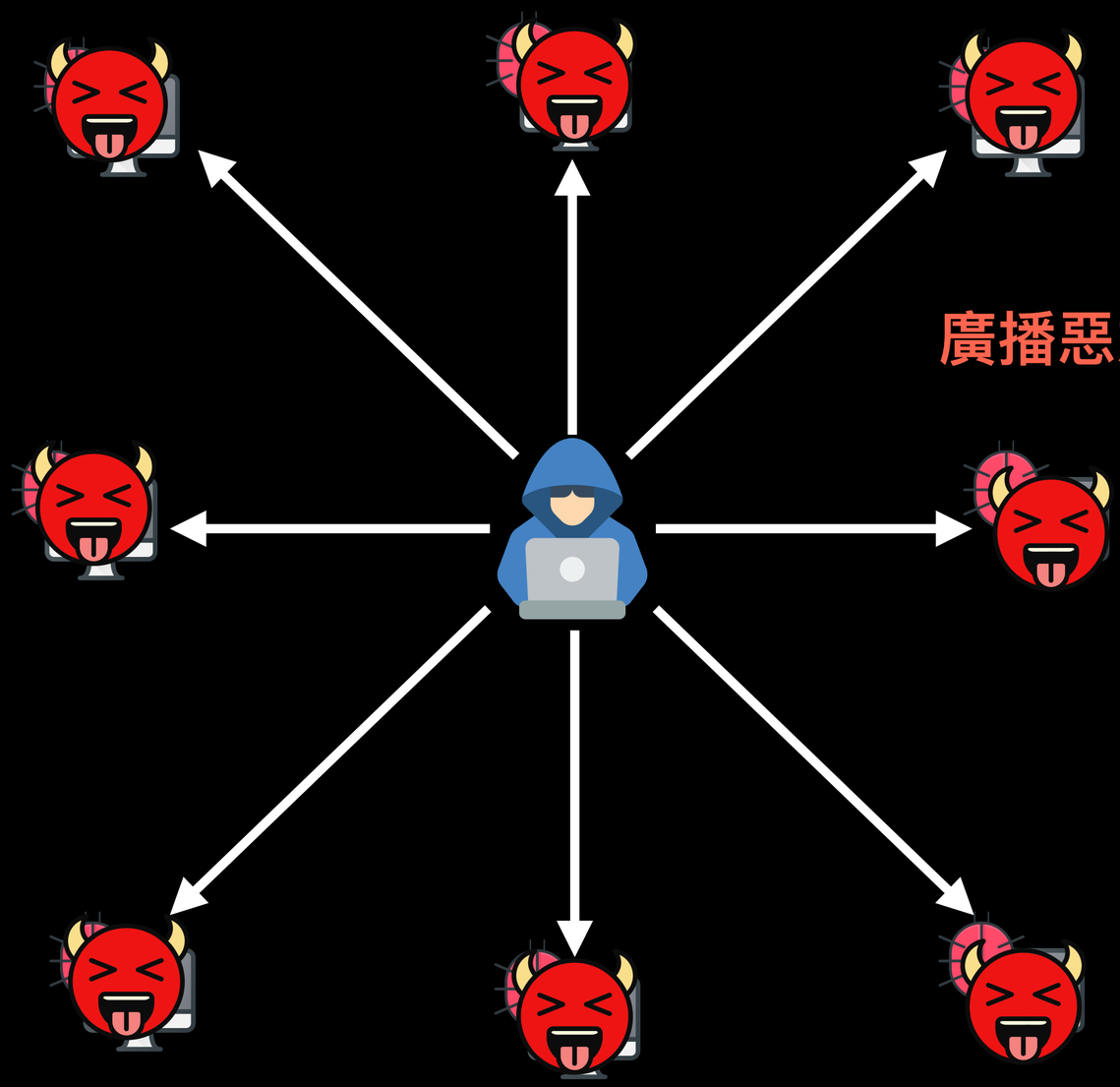
```
int PeerBeaconAndProbeRspSanity(...){
    while(...){
        // Check if ie_buf can store ie
        switch(ie->type){
            case ...:
                ...
            case 244:
                // Copy ie into ie_buf
            case 221:
                // Check if wsc can store ie
                // Copy ie into wsc
        }
    }
} // Copy wsc into ie_buf
```

成因分析

ie_buf: 1024 bytes
wsc: 512 bytes



無差別攻擊



廣播惡意 Frame 給附近的設備

無差別攻擊

利用 CVE-2025-20685

- 設備有什麼漏洞？
  - Heap-based Buffer Overflow (kmalloc-1k)
  - 溢出 508 bytes
- 設備 Kernel 有哪些保護？
  - 🟢 Stack Canary
  - 🟢 NX
  - ❌ KASLR
  - ❌ Freelist Hardening
  - ❌ KCFI

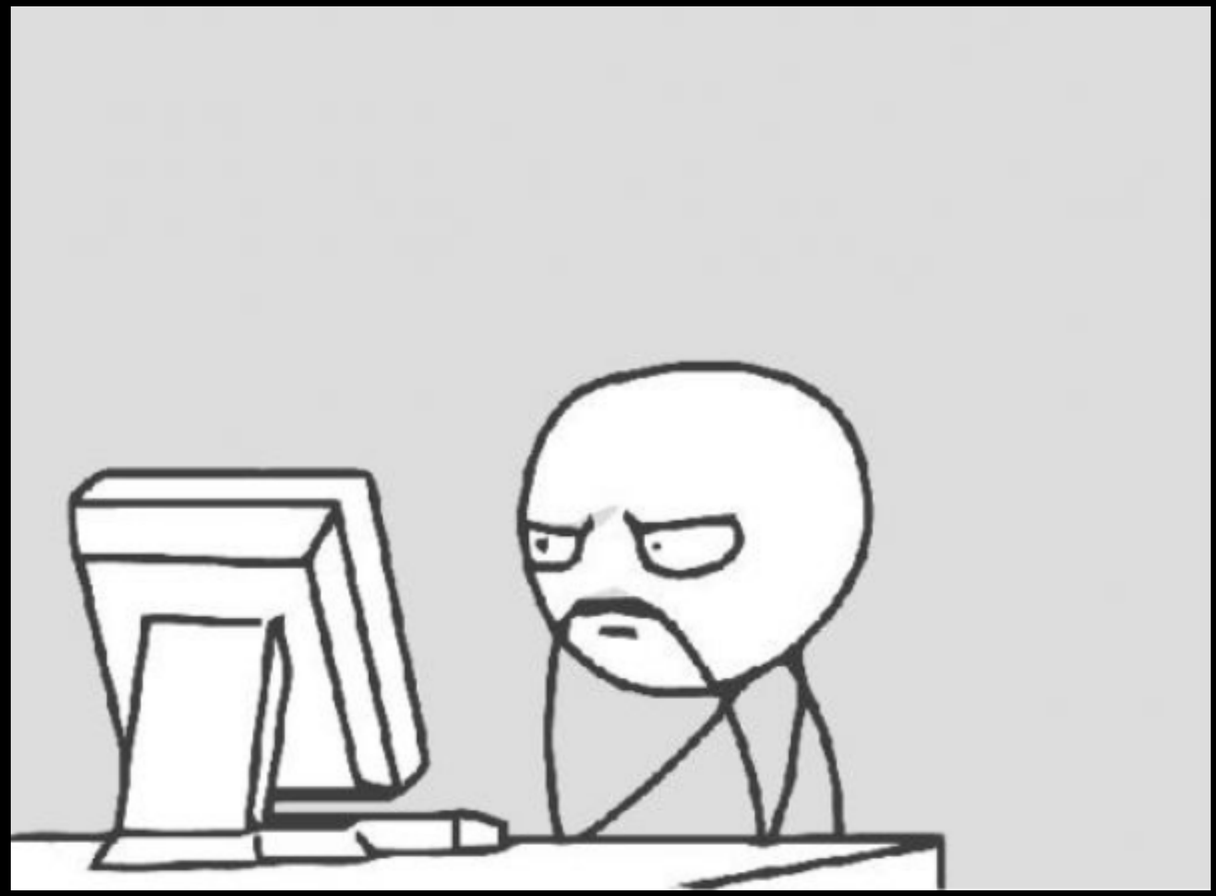
怎麼 Debug?

- 沒有 GDB, 但 crash dump 已經足夠了

```
[   76.248093] CPU: 1 PID: 2875 Comm: RtmpMlmeTask Tainted: P                  5.4.171 #0
[   76.255991] Hardware name: MediaTek MT7981 RFB (DT)
[   76.260855] pstate: 20000005 (nzCv daif -PAN -UAO)
[   76.265637] pc : __kmalloc+0x84/0x208
[   76.269287] lr : __kmalloc+0x40/0x208
[   76.272935] sp : ffffffc0125c3c30
[   76.276236] x29: ffffffc0125c3c30 x28: 0000000000000000
[   76.281533] x27: ffffff80074d7a38 x26: ffffffc012c815f8
[   76.286831] x25: ffffffc008b8ff00 x24: ffffffc012daabc8
[   76.292127] x23: ffffff8007b49cc0 x22: ffffffc012690000
[   76.297425] x21: 0000000000000a20 x20: 00006f666e697265
[   76.302722] x19: ffffff800f003800 x18: 0000000000000000
[   76.308019] x17: 0000000000000000 x16: 0000000000000000
[   76.313315] x15: 0000000000000000 x14: 0000000000000000
[   76.318612] x13: 0000000000000000 x12: 0000000000000000
[   76.323909] x11: 0000000000000000 x10: 00000000000007e0
[   76.329205] x9 : 0000000000000000 x8 : ffffffc008eec2b8
[   76.334502] x7 : 0000000000000000 x6 : 000000000000003f
[   76.339799] x5 : 0000000000000040 x4 : ffffff800fcb7780
[   76.345096] x3 : ffffffbfff325000 x2 : ffffff800fb5fb80
[   76.350393] x1 : 0000000000002645 x0 : 0000000000000000
[   76.355691]
[   76.355691] PC: 0xffffffc0101c63ac:
[   76.360640] 63ac  f140081f 910003fd a90153f3 54000c88 a9025bf5 2a0103f5 97ff39e7 aa0003f3
[   76.368806] 63cc  f100401f 54000ce9 f0003401 b9499834 0a1402b4 2a1403e1 97ff39fd 35000a40
[   76.376970] 63ec  d503201f b4000a13 f9400260 910003e2 d538d081 91002000 f8616801 d538d083
[   76.385135] 640c  f9400260 8b030002 f9400844 f8636814 f100009f fa401a84 54000aa0 b9402260
[   76.393299] 642c  f8606a83 d538d082 91000424 f9400260 8b020000 f9800011 c87f1406 ca1400c6
[   76.401464] 644c  ca0100a5 aa0500c5 b5000065 c8261003 35ffff46 b5fffca5 b9402260 f8a06860
[   76.409628] 646c  d503201f d503201f d34822b5 35000515 d503201f a9425bf5 aa1403e0 a94153f3
[   76.417793] 648c  a8c37bfd d65f03c0 37b00074 b9400a60 36d7faa0 aa1303e0 94002a9f aa0003f3
[   76.425958]
[   76.425958] LR: 0xffffffc0101c6368:
[   76.430906] 6368  d4210000 b0002ba0 aa1903e2 913a2000 d2800001 9411067d f900b660 b5fffae0
[   76.439071] 6388  aa1903e0 12800174 9411045a 35fff597 17ffff8f aa1903e0 94110456 17ffff8c
[   76.447235] 63a8  a9bd7bfd f140081f 910003fd a90153f3 54000c88 a9025bf5 2a0103f5 97ff39e7
[   76.455400] 63c8  aa0003f3 f100401f 54000ce9 f0003401 b9499834 0a1402b4 2a1403e1 97ff39fd
[   76.463564] 63e8  35000a40 d503201f b4000a13 f9400260 910003e2 d538d081 91002000 f8616801
[   76.471729] 6408  d538d083 f9400260 8b030002 f9400844 f8636814 f100009f fa401a84 54000aa0
[   76.479894] 6428  b9402260 f8606a83 d538d082 91000424 f9400260 8b020000 f9800011 c87f1406
[   76.488059] 6448  ca1400c6 ca0100a5 aa0500c5 b5000065 c8261003 35ffff46 b5fffca5 b9402260
[   76.496225]
[   76.496225] SP: 0xffffffc0125c3bb0:
[   76.501173] 3bb0  12daabc8 ffffffc0 08b8ff00 ffffffc0 12c815f8 ffffffc0 074d7a38 ffffff80
[   76.509337] 3bd0  00000000 00000000 125c3c30 ffffffc0 101c63e8 ffffffc0 125c3c30 ffffffc0
[   76.517502] 3bf0  101c642c ffffffc0 20000005 00000000 12690000 ffffffc0 00000004 ff000000
[   76.525667] 3c10  ffffffff ffffffff 100d4e2c ffffffc0 125c3c30 ffffffc0 101c642c ffffffc0
[   76.533831] 3c30  125c3c60 ffffffc0 08d07764 ffffffc0 125c3cf0 ffffffc0 128b7340 ffffffc0
[   76.541996] 3c50  12d113e0 ffffffc0 12690000 ffffffc0 125c3ca0 ffffffc0 08b5a1e8 ffffffc0
[   76.550161] 3c70  12690000 ffffffc0 100d112c ffffffc0 0f0b2cc0 ffffff80 0fb5c980 ffffff80
[   76.558326] 3c90  0f0b2cd8 ffffff80 0fb5c9b8 ffffff80 125c3d30 ffffffc0 08b8f6cc ffffffc0
```

可利用結構

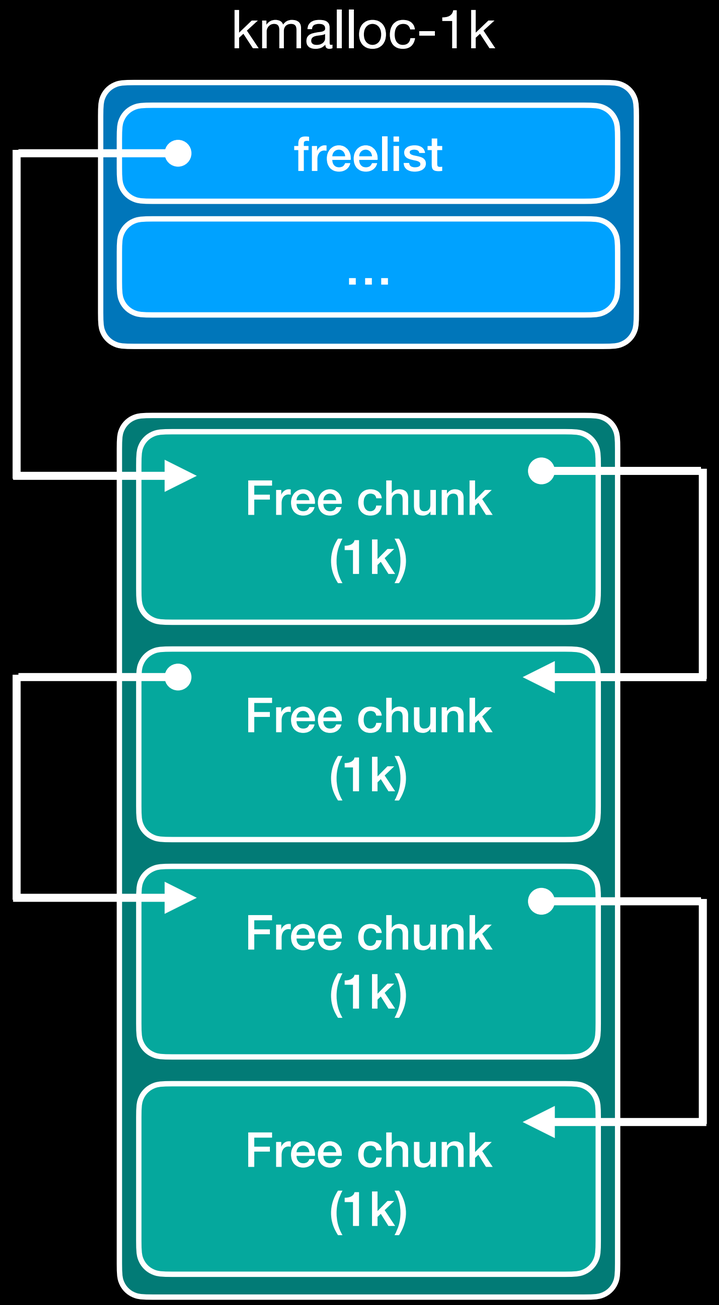
- 不知道可以控制什麼結構.....



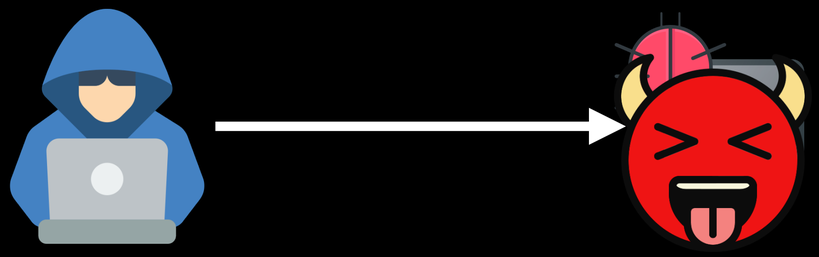
可利用結構

- 不知道可以控制什麼結構.....
- 持續 crash 設備並且:
  - 蒐集 stack traces
  - 辨識出有趣的 stack traces

Freelist 劫持


kmalloc-1k
freelist
...
Free chunk
(1k)
Free chunk
(1k)
Free chunk
(1k)
Free chunk
(1k)

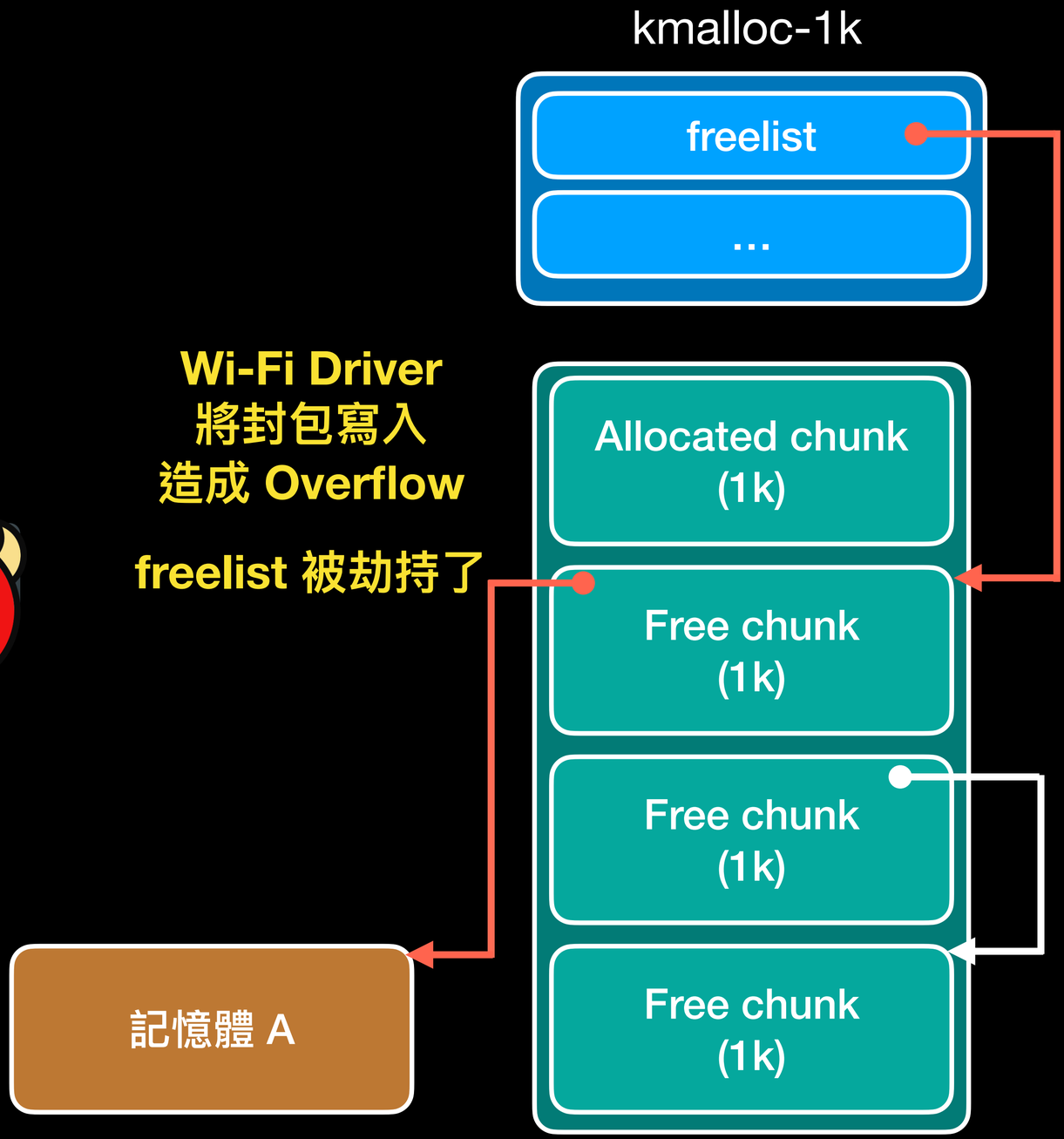
Freelist 劫持



傳送第一個 Beacon
劫持 freelist

Wi-Fi Driver
將封包寫入
造成 Overflow

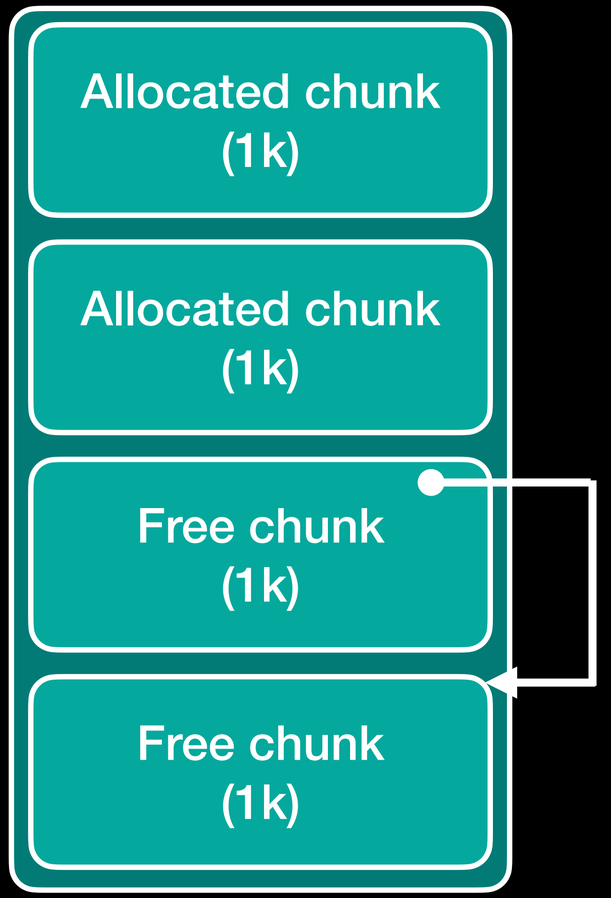
freelist 被劫持了



Freelist 劫持

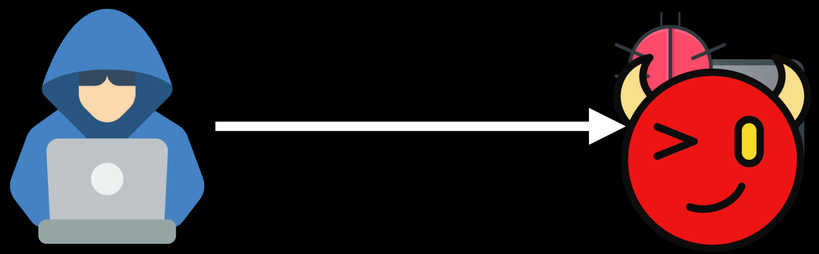
Kernel 某段程式碼
拿走一個 free chunk

kmalloc-1k



Chunk A

任意地址寫



送第二個 Beacon
任意地址寫入惡意 payload

Wi-Fi Driver
將 payload
寫入控制的地址

Allocated chunk
A

kmalloc-1k

freelist

...

Allocated chunk
(1k)

Allocated chunk
(1k)

Free chunk
(1k)

Free chunk
(1k)

利用計劃

- 可以達成一次性的任意地址寫入
  - 在 payload 中放置 shellcode
  - 控制 Wi-Fi Driver 中的一個函數指標，替換成 cpu_switch_to

```
[    77.443850] Hardware name: MediaTek MT7981 RFB (DT)
[    77.448714] pstate: 00000005 (nzcv daif -PAN -UAO)
[    77.453494] pc : 0xaaaaaaaaaaaaaaaa
…
[    77.519602] x9 : 8a8a8a8a8a8a8a8a x8 : 9191919191919191
[    77.524898] x7 : 9090909090909090 x6 : 0000000000000000
[    77.530195] x5 : 0000000000000000 x4 : aaaaaaaaaaaaaaaa
[    77.535492] x3 : ffffffc012cad028 x2 : ffffff80038c6000
[    77.540789] x1 : ffffffc0128c5bd0 x0 : ffffffc012688000
```

```
LDP     X19, X20, [X8],#0x10
LDP     X21, X22, [X8],#0x10
LDP     X23, X24, [X8],#0x10
LDP     X25, X26, [X8],#0x10
LDP     X27, X28, [X8],#0x10
LDP     X29, X9,  [X8],#0x10
LDR     X30, [X8]
MOV     SP, X9
MSR     SP_EL0, X1
RET
```

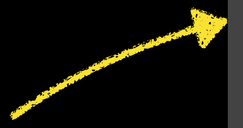
x8 可控

cpu_switch_to

利用計劃

- 可以達成一次性的任意地址寫入
  - 在 payload 中放置 shellcode
  - 控制 Wi-Fi kernel module 中的一個函數指標，替換成 cpu_switch_to
- 執行三個 ROP & JOP gadgets
  - 使得 shellcode 可執行
  - Return 回 shellcode

我用到的 3 個 gadgets



```
mov x19, x1
mov x1, x22
blr x23
…

mov x0, x20
blr x25
…

blr x21
ldp x21, x22, [sp, #0x20]
ldp x29, x30, [sp], #0x30
ret
```

利用限制

- 機率問題
  - Kernel Driver 載入時，地址是部分隨機
  - Freelist 的順序也有一定的隨機性

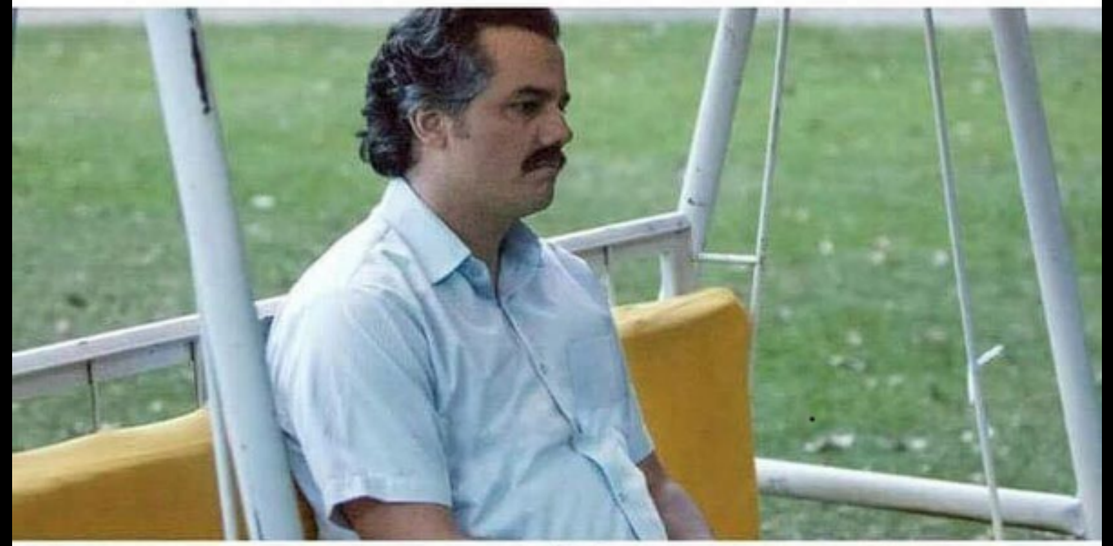
利用限制

- 機率問題
  - Kernel Driver 載入時，地址是部分隨機
  - Freelist 的順序也有一定的隨機性
- 某些設備沒有開 KASLR 且 Wi-Fi Driver 靜態連結到 Kernel
  - 不需要撞地址，利用時間大幅縮短

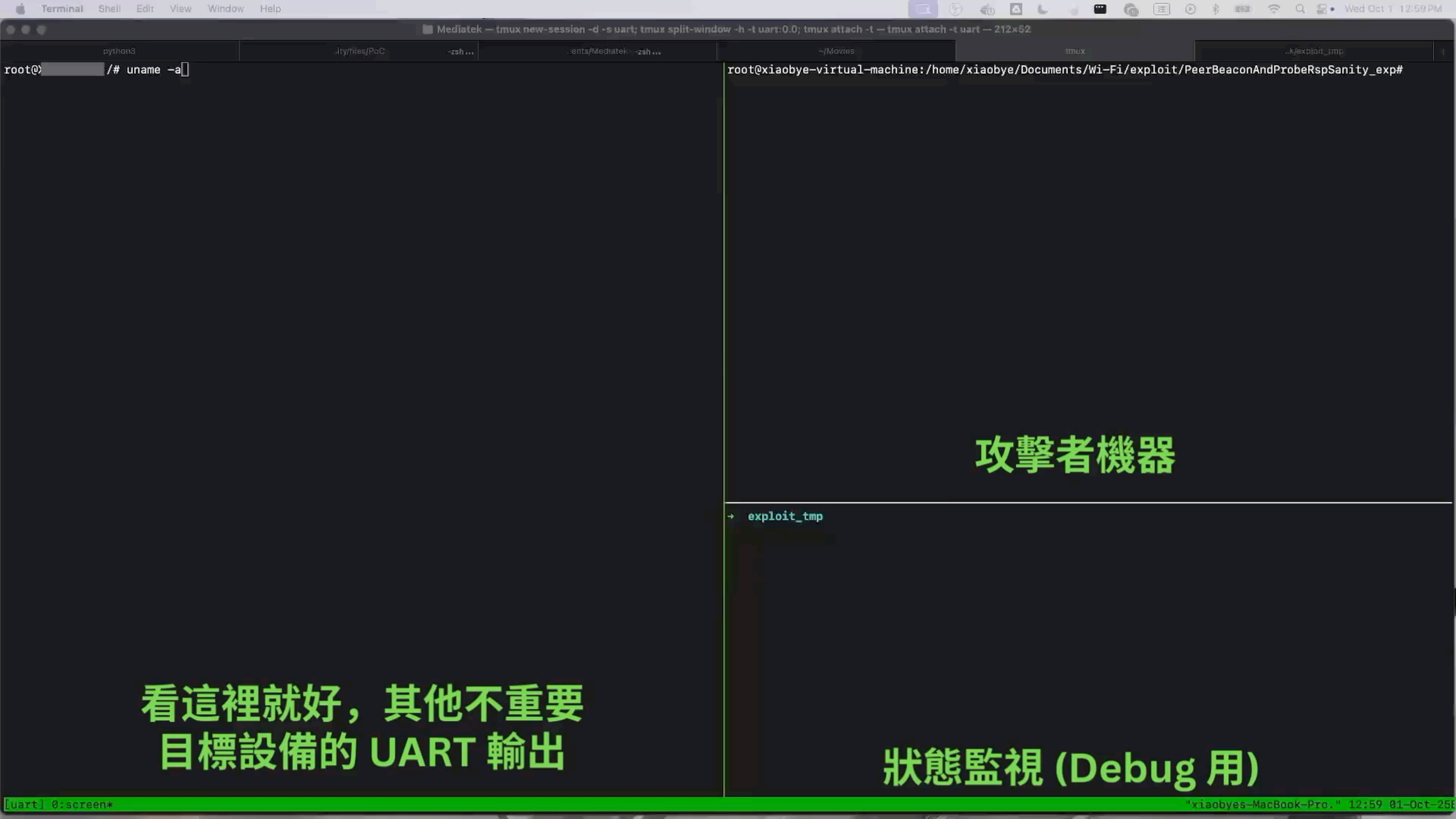
利用限制

- 機率問題
  - Kernel Driver 載入時，地址是部分隨機
  - Freelist 的順序也有一定的隨機性
- 某些設備沒有開 KASLR 且 Wi-Fi Driver 靜態連結到 Kernel
  - 不需要撞地址，利用時間大幅縮短
- 某些設備是 32 位元處理器
  - 很好撞，利用時間大幅縮短

Crash 後，開一次機要 30 秒
寫 Exploit 就像坐牢一樣



Crash! Crash! Crash!


Terminal Shell Edit View Window Help
Wed Oct 1 12:59 PM
Mediatek — tmux new-session -d -s uart; tmux split-window -h -t uart:0.0; tmux attach -t — tmux attach -t uart — 212×52
python3
-zsh ...
-zsh ...
~/Movies
tmux
root@ /# uname -a
root@xiaobye-virtual-machine:/home/xiaobye/Documents/Wi-Fi/exploit/PeerBeaconAndProbeRspSanity_exp#
攻擊者機器
→ exploit_tmp
看這裡就好，其他不重要
目標設備的 UART 輸出
狀態監視 (Debug 用)
[uart] 0:screen*
"xiaobyes-MacBook-Pro." 12:59 01-Oct-25

結語

DEVCORE


Wi-Fi 密碼設這麼強，連我
自己都背不起來
啊怎麼還是被打穿

建議

- 研究員
  - Wi-Fi Driver 和廠商基於 Wi-Fi protocol 的客製協議都是相當大的攻擊面
  - MediaTek Wi-Fi SDK 中的 wf_rom.axd，大幅減少逆向 Wi-Fi 6 MCU 的成本
- Wi-Fi 使用者
  - 高安全性的密碼不一定能夠保護到你的 AP
    - 定期追蹤設備韌體的更新
    - 隔離網段，減少任意 Data Frame 注入帶來的影響

DEVCORE

建議

- 設備廠商
  - 啟用 KASLR 和 Freelist Hardening，可以大幅提升遠端 Heap 利用的難度
  - 持續追蹤晶片廠商釋出的修補

DEVCORE

聯發科 PSIRT

- MediaTek 有完整的漏洞回報、處理和公開流程
- 我們簡報中提到的 CVEs 和 vulnerabilities 已經被修補了
  - CVE-2025-20674 已經被修補 June 2025 MediaTek Security Bulletin
  - CVE-2025-20685, CVE-2025-20686 已經被修補 July 2025 MediaTek Security Bulletin
  - CVE-2025-20710, CVE-2025-20711 和 CVE-2025-20712 已經被修補 October 2025 MediaTek Security Bulletin
- MediaTek 會先對客戶公開修補內容，並給予客戶兩個月的時間修補

DEVCORE

Q&A

DEVCORE