

2016 / TDOH Conf

CTF From Zero To One

Inndy / inndy.tw@gmail.com

\$ whoami

- Inndy (a.k.a. 木棍)
- 台科資工大三
- 業餘 Security Researcher
- 業餘工程師 at Sharelike
- 業餘學生
- CTF 隊伍 forx 隊長
- 金盾 2016 冠軍

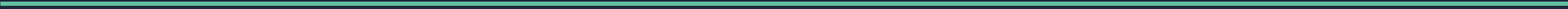
WTF is CTF?



「搶棋遊戲由兩隊人馬互相前往對方的基地奪旗，把敵方的旗從敵方的基地帶回自己隊伍的基地。」

- 維基百科

駭客的搶棋遊戲



為什麼打 CTF？

- 在外面亂打太危險，今天拿 root 明天查水錶
- 和其他人切磋、交流
- 藉由 CTF 學資安

今天我要說的故事是：
從什麼都不會到帶團拿下第一名的過程

今天我要說的故事是：
從什麼都不會到一個人打CTF的過程



From Zero!

- 想要寫自己的遊戲外掛
- 然後就學會了寫程式和逆向工程
- 至少先瞭解一個領域，醬油味才不會太重 (Binary, Web, Crypto)

Form Zero!

- 跟在 HackStuff 的大大們旁邊打醬油，試著解 HITCON Wargame 逆向題
- 終於知道什麼是 IDA Pro —— 已知用火！
 - 在這之前都是用 Cheat Engine + Notepad++ 做逆向
- HITCON Wargame 轉型成 HITCON CTF，開始變成國際賽事

To One!

- 這段時間開始學逆向之外的東西，Web, Crypto, Binary Exploitation
- 身邊沒有人一起打 CTF，只好一個人玩
- 一人戰隊是痛苦的，沒有隊友只好自己學各個領域的知識
- 但是難以集中精神看難題，有些題目不難但是花時間

Then, To One!

- 去年九月時組了 forx，帶幾個同學一起玩，但是戰力實在不夠 XD
- 年底的時候 🍊 大大帶我們一起打 XCTF 聯賽
- RCTF 2016 Final 2_{nd} place
- SSCTF 2016 Final 1_{st} place
- XCTF 2016 Final 5_{th} place
- 金盾獎 2016 1_{st} place

不同的 CTF 賽制

- Jeopardy - 答題挑戰
- Attack & Defense - 攻防競賽
 - DARPA CGC - 機器人攻防
- King of the Hill - 佔領領地

Jeopardy

計分板

\$ 100,000
\$ 60,000
\$ 48,000
\$ 36,000
\$ 24,000
\$ 12,000
\$ 8,000
\$ 5,000
\$ 3,000
\$ 1,000

10萬挑戰題 台視

題目

6年級 藝術與人文	6年級 英文
5年級 國語	5年級 健康與體育
4年級 數學	4年級 社會
2年級 鄉土	3年級 自然與生活科技
1年級 生活	

Jeopardy

- 解題拿 flag 換分數
- first-blood 有時候有 bonus
- 不同類型的題目有各種難度
- 參與門檻低，有電腦有網路即可

Roulette	Web Security	Binary Processing	Forensics	Pwn3d
100	100	100	100	100
200	200	200	200	200
300	300	300	300	300
400	400	400	400	400
500	500	500	500	500

Forensics

200 points

Desc
Find Key ^_^

Link
Tw.pptx

Hint 1
password: hit2013

Hint 2
ILSpy will help you ^_^

Enter key Send

Jeopardy - Web

- 會上網就可以開始打，夠簡單了吧？
- 資訊洩露
 - robots.txt / .git / .svn / error message / shared host
- 注入攻擊
 - SQL Injection
 - ' or " = '--
 - UNION SELECT
 - Blind Injection

Jeopardy - Web

- 注入攻擊（續）
 - Command Injection
 - 你跟 Linux command 有多熟？
 - 通常會要繞過某些黑名單（空白、引號、分號...）
 - ; cat flag;
 - %0a cat<flag%00
 - {grep,-r,.}
 - sh /proc/self/env + HTTP header (apache)

Jeopardy - Web

- 注入攻擊（續）
 - Cross Site Scripting (XSS)
 - 最近不少 Web 都要先 XSS 再往後打
 - XML Entity Injection (XXE)
 - 以 Jeopardy 來說不多見，因為可以 DoS (Billion laughs)
 - 玩滲透的說，入侵的時候很實用

Jeopardy - Web

- Race Condition
 - 資料庫（或其他操作）沒有用 Transaction （Lock）造成的狀態不一致，產生間隙可以利用
 - Ex：轉賬邏輯
 - 讀取餘額（SELECT * 2），扣款（in PHP），改變餘額（UPDATE * 2）
 - 如果伺服器 loading 較重，SELECT 與 UPDATE 操作會有時間差，可以重複轉出
 - 在 PTT 發生過很多，近期的 Dirty COW 漏洞也是 Race Condition

Race Condition on PTT

```
* 13f5e343 - (1 year, 9 months ago) boardd: fix another race when bufferevent is freed. - robertabcd
* cd77b482 - (1 year, 9 months ago) Fix race of freeing bufferevent when client error. - robertabcd
* 3f365ebd - (2 years, 9 months ago) Create URL tags earlier to avoid race condition. - Hung-Te Lin (piaip)
* bb4343ff - (5 years ago) fix race condition for making vote results -- contributed from robertabcd - Hung-Te Lin (pia
* 94b94383 - (7 years ago) * badpost can be only assigned if the file deletion was success, to prevent race condition.
* 1b737a75 - (9 years ago) * fix crash bug: race condition between idle and talk interrupt. - Kuang-che Wu
* 79972df1 - (10 years ago) not necessary to dereference mark race - Victor Hsieh
* b53a36ec - (11 years ago) random sleep to make race condition harder. - Kuang-che Wu
* e1ffa495 - (11 years ago) reduce race condition period which lead to leave utmp record after user has been kick out.
* 40f41982 - (11 years ago) deny user login in less than 3 second, to prevent flooding and race condition of multilogin
* 2c7e0c96 - (11 years ago) prevent board master use gamble race condition to earn illegal money. - Kuang-che Wu
* c18b4471 - (11 years ago) reduce crash possibility due to race condition. - Kuang-che Wu
* 9cd8f279 - (11 years ago) fix unpaired brace in comment - Kuang-che Wu
* 91361879 - (11 years ago) fix bug, board class disappear because incorrect bsorted[] made by race condition. - Kuang-
* 831949f1 - (11 years ago) make it harder to multi-login by race condition. prevent logout function reenter. - Kuang-c
* edb33b9d - (11 years ago) dirty workaround for race condition in multi-login checking. - Kuang-che Wu
* 6d6d428a - (11 years ago) fix race condition of user registration (as well as while restoring a backed-up account):
* 7063ef2e - (12 years ago) fix race of assessment note that SHM_t was modified - Victor Hsieh
* 80b0c558 - (12 years ago) fix bvote problem use filename0 instead of filename (use hardlink to avoid race, it'll be
* 1e7cb256 - (12 years ago) race - Victor Hsieh
* 0ce30fd0 - (12 years ago) solved a frequent race - Victor Hsieh
* 2daadf06 - (13 years ago) add stampfilefd() to avoid race condition - in2
* 517171da - (13 years ago) avoid race condition at tag board->paste board - Ethan Tu
* 99a9a261 - (13 years ago) Remove altering wbtime in nkwd, avoiding race. - Shu-Chun Weng
* c670e566 - (13 years ago) fix race in write_request() - in2
```

Jeopardy - Web

- 語言特性、Bug : PHP
 - 地表最多人用的語言 (Web) , 最慘的 codebase
 - unserialize - memory corruption, use after free
 - parse_url - parser 邏輯錯誤
 - `strcmp($_GET['secret'], 'p9dMhRV5VQc4MwjYRyK3U2sS')`
 - `a.php?secret[]=&secret[]=%00`
 - `md5('240610708') == md5('QNKCZDZO')`

Jeopardy - Web

- 語言特性、Bug：Ruby
 - Ruby 用 C 寫 interpreter，而且幾乎沒有 library dependency (zlib, openssl, libunwind)
 - 這代表很多東西都自幹，不依賴外部程式庫，以程式安全的角度來看...
 - 浮點數處理時有 Overflow
 - `BigDecimal("9E69999999").to_s("F"); ("0."+"1"**300000).to_f`
 - YAML parser 導致 RCE - CVE-2013-0156
 - 相對來說，Ruby 社群注重安全得多

Jeopardy - Web

- Crypto Fail
- 你會需要檢查資料有沒有被改過，你會...
 - 用 md5 ?
 - $\text{md5}(\text{secret} + \text{data})$?
 - Length Extension Attack !
- 你會需要加密資料，也許是 Session
 - AES-256-CBC ?
 - Padding Oracle Attack !

Jeopardy - Web

- Web 打起來像是在猜謎？
- 請關注資安新聞，最近的公開漏洞
 - structs2
- Web出題會有流行趨勢
 - 前陣子有：Server Side Template Injection, Padding Oracle, 混合 XSS
 - 關注出題團隊的 Web 專家最近在研究什麼
 - 🍊 出題常常中（？）

Jeopardy - Binary / Reverse

- 紿你一個 Binary，想辦法搞懂他在做什麼！
- 典型的 Reversing 題目會是 native code，通常用 C 語言寫
- 需要讀組合語言的能力
- 題目大概有幾種長相
 - 輸入 flag，題目會告訴你是不是對的
 - 輸入密碼，解密 flag 後印出來
 - 紿你加密的 flag 和加密程式，有時候會有一組已知明文的密文
 - 註冊機，幫指定的 User 算出他的序號

Jeopardy - Binary / Reverse

- 典型的題目像是這樣

```
[ 12/16 22:10:53 ] inndy /tmp (N/A)
$ ./rev
./rev FLAG
[ 12/16 22:10:56 ] inndy /tmp (N/A)
$ ./rev 123123
No
[ 12/16 22:10:58 ] inndy /tmp (N/A)
$ ./rev 'FLAG{This is a flag for you}'
Yes, It's flag!
[ 12/16 22:10:59 ] inndy /tmp (N/A)
$ █
```

Jeopardy - Binary / Reverse

- 進階一點的題目...
- 加殼 / 混淆過的程式，分析前要先脫殼 / Deobfuscate
- 手工組合的執行檔（手寫 assembly，不是高階語言編譯出來的）
- C++ 執行檔加編譯最佳化
 - string 的處理就夠難看了，optimize 開下去更想死
- 奇怪的語言編譯出來的 binary，如：golang, swift, haskell
- Virtual Machine （像是 GBA Emulator 那類的東西）
- static compiled and striped binary... WTF?!

Jeopardy - Binary / Reverse

- 偏門一點的題目
- 某些編譯成 binary code 但不是 native code 的 binary，通常沒有 decompiler 可以用，就算有，也會想辦法搞壞 decompiler
 - VB6 P-Code (SECCON 2016 retrospective)
 - Squirrel lang (BCTF 2016 sif)
 - 光是找出他是什麼語言就花了我好久的時間 QQ
 - Python byte code (pyc), Java class, obfuscated .NET
 - 手寫一些奇怪的 byte code pattern 會搞壞 decompiler

Jeopardy - Binary / Reverse

- 偏門一點的題目 (續)
 - 冷門平台的 native binary
 - ARM, MIPS 都算小事了，用 QEMU 就可以跑
 - 沒有 decompiler 可以用 QQ
 - 寫一大堆 code 然後開 compiler optimize 又是 C++ 寫的...
 - 洗洗睡比較快...都怪現代 CTF 發展太可怕 QQ
 - 非常底層的 binary
 - BIOS ROM, Driver, DOS binary

Jeopardy - Binary / Reverse : 工具篇

- 學習工具是很重要的，才能過當一個稱職的工具人
- 好的工具帶你上天堂，壞的工具讓你崩潰 QQ
- 動態分析工具 - Debugger
 - OllyDbg, x64dbg, gdb, evan's debugger (GUI debugger for Linux)
 - 有些題目設定好 breakpoint，密碼馬上就出現在你面前了
 - 有些題目可以 byte-by-byte 的暴力破解
 - 學習怎麼寫 debugger script !
- strace, ltrace : 觀察程式的 system call, library call 行為

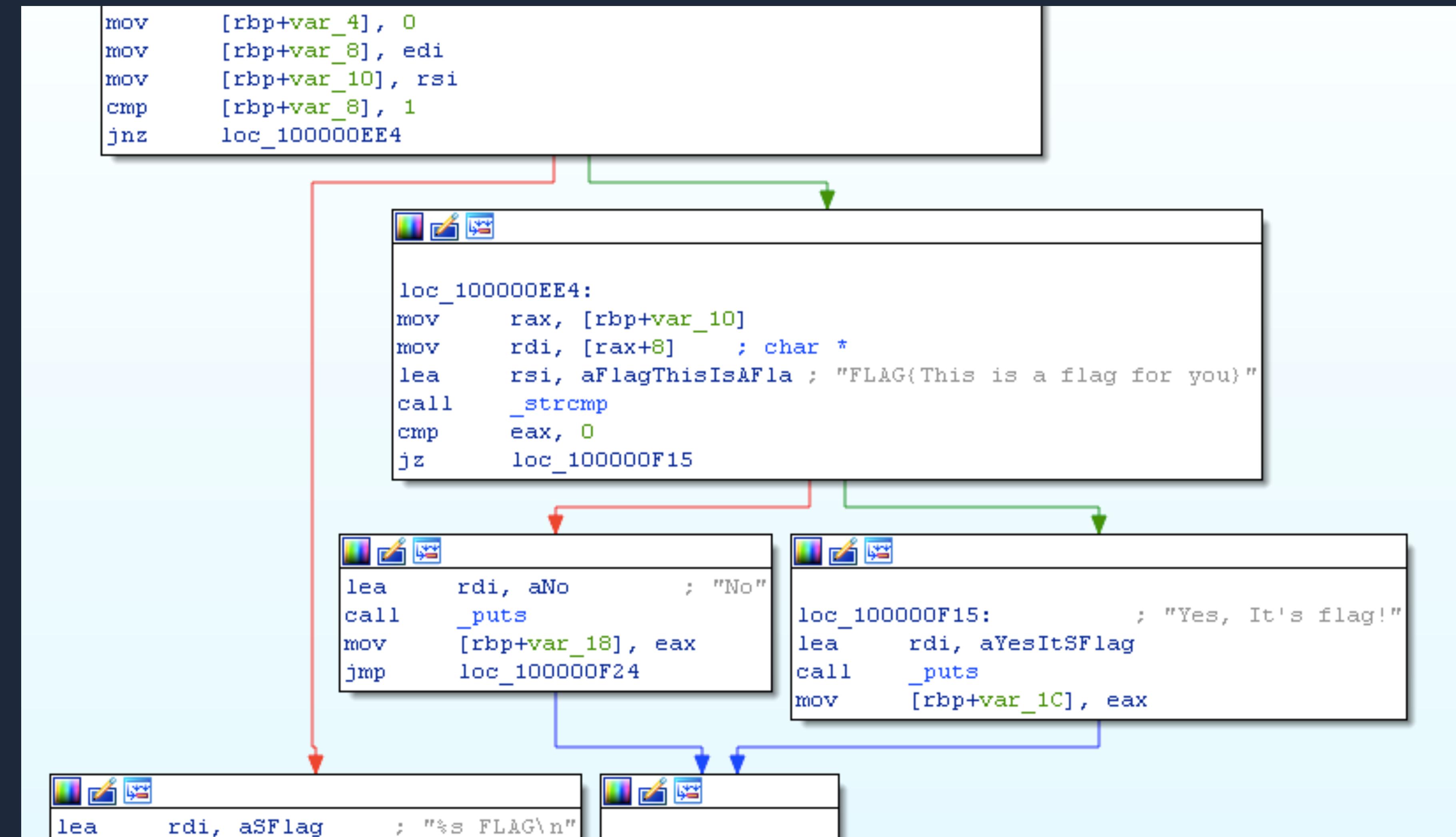
Jeopardy - Binary / Reverse : 工具篇

- 靜態分析工具
 - objdump 可以看組合語言！
 - IDA Pro with Hex-ray decompiler
 - 超超超超級貴貴貴死人，包含 decompiler 的授權要接近十萬臺幣
 - 超級強大，幾乎是標準配備
 - radare2
 - 很強大的工具，IDA Pro 之於 radare2 就像是 Visual Studio 之於 vim
 - 學習曲線很可怕，有些奇怪的 bug QQ

Jeopardy - Binary / Reverse : 工具篇

- 靜態分析工具
 - binary ninja
 - Hop Disassembler
 - 特長是 Objective-C
 - Retargetable decompiler
 - 線上反編譯服務，不過免費的有限制

IDA Pro



radare2

[0x100000ea0]> VV @ entry0 (nodes 7 edges 8 zoom 100%) BB-NORM mouse:canvas-y movements-speed:5

```
2. radare2
; var int local_4h @ rbp-0x4
push rbp
mov rbp, rsp
sub rsp, 0x20
mov dword [rbp - local_4h], 0
mov dword [rbp - local_8h], edi
mov qword [rbp - local_10h], rsi
cmp dword [rbp - local_8h], 1
jne 0x100000ee4 ;[a]

t f
' '-----'
'-----'-----'

0x100000ee4
mov rax, qword [rbp - local_10h]
mov rdi, qword [rax + 8]
lea rsi, [rip + 0x88] ;[b]
call sym.imp.strcmp ;[c]
cmp eax, 0
je 0x100000f15 ;[d]

t f
|-----'|-----'
|-----'|-----'

0x100000ec0
lea rdi, [rip + 0xab] ;[j]
mov rax, qword [rbp - local_10h]
mov rsi, qword [rax]
mov al, 0
call sym.imp.printf ;[k]
mov dword [rbp - local_4h], 0
mov dword [rbp - local_14h], eax
jmp 0x100000f29 ;[i]

v
|-----'|-----'
|-----'|-----'

0x100000f15
lea rdi, [rip + 0x7f] ;[e]
call sym.imp.puts ;[f]
mov dword [rbp - local_1ch], eax

v
|-----'|-----'

0x100000f01
lea rdi, [rip + 0x90] ;[g]
call sym.imp.puts ;[f]
mov dword [rbp - local_18h], eax
jmp 0x100000f24 ;[h]

v
|-----'|-----'
```

Jeopardy - Binary / Reverse : 工具篇

- 動態分析工具 (進階)
- Binary instrumentation - Intel pintools
 - 寫程式動態的觀察、Hook 另一個程式
 - 可以用來做 side channel attack (timing attack)
 - 線性比對字串的時候，計算指令數量就可以知道猜對了幾個字
- SMT solver - Z3 by Microsoft
 - $X > 87 \&& (X^{**} 2 + X * 77) \% 999 == 894; X = ?$
 - 神奇的自動暴力破解

Jeopardy - Binary / Reverse : 巫術篇

- Symbolic Execution
- angr
 - 對 binary 做 symbolic execution，自動帶入 SMT solver (angr 用 Z3) ，找出所有條件跳轉的條件是什麼
 - 從那裡開始，走到那裡，避開那裡
 - 好，你告訴我要滿足的條件是什麼？
 - 慢，吃記憶體，而且有些狀況不見得能解

Jeopardy - Binary / Reverse : 巫術篇

- Symbolic Execution
- klee
 - 基於 llvm 架構下，原始碼層級的 symbolic execution
 - C / C++ / other-language → compiled to llvm IR → klee
 - 寫一個 C 程式，叫 klee 告訴你要滿足什麼條件才能觸發 assert
 - 可以拿 hex-ray decompiler 出來的 code 進去跑 XD
 - LazyKLEE by L4ys - <https://github.com/L4ys/LazyKLEE>

Pwnable

- 你寫過 C 語言嗎？

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char name[100];
6     printf("What's your name?");
7     scanf("%s", name);
8     printf("Hello, %s\n", name);
9     return 0;
10 }
```

Pwnable

- 你寫過 C 語言嗎？

```
inndy@ntust-sec demo$ ./buffer-overflow
What's your name?Inndy
Hello, Inndy
inndy@ntust-sec demo$ ./buffer-overflow
What's your name?AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hello, AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*** stack smashing detected ***: ./buffer-overflow terminated
Aborted (core dumped)
inndy@ntust-sec demo$ █
```

Pwnable

```
int main() {  
    ...  
    printf("your name:");  
    scanf("%s", name);  
    ...  
    return 0;  
}
```

(low address)
AAAAAAA <--- esp
0000000
0000000 <--- buffer
0000000
0000000
0000000
0000000
0000000
0083af3c <--- stack canary
ffeeddcc <--- ebp, saved ebp
00401234 <--- return address
abcdabcd
(high address)

Pwnable

```
int main() {  
    ...  
    printf("your name:");  
    scanf("%s", name);  
    ...  
    return 0;  
}
```

(low address)
AAAAAAA <--- esp
0000000
41414141 <--- buffer
41414141
41414141
41414141
41414141
41414141 <--- stack canary
41414141 <--- ebp, saved ebp
41414141 <--- return address
41414141
(high address)

Pwnable

```
int main() {  
    ...  
    printf("your name:");  
    scanf("%s", name);  
    ...  
    return 0;  
}
```

(low address)
AAAAAAA <--- esp
0000000
41414141 <--- buffer ***
41414141
41414141
41414141
41414141
41414141
41414141 <--- stack canary
41414141 <--- ebp, saved ebp
41414141 <--- return address
41414141
(high address)

stack smashing detected

Pwnable

- stack canary 是現代編譯器的保護機制，防止 stack overflow 漏洞被利用
- 進入 function 時，寫入 OS 隨機產生的 stack canary
- 離開 function 時，檢查 stack canary 是否被篡改
 - (連續寫入) 覆蓋 ebp, retn address 之前一定會先蓋到 stack canary
 - Boooooooooooooom!
- 我們來看一個沒有保護的例子

Pwnable

- name 在堆疊上
- data 在靜態的位址
- scanf("%s", name);
 - 造成 buffer overflow
- 知道 stack 是怎麼運作的很重要

```
1 #include <stdio.h>
2 #include <string.h>
3
4 char data[1024];
5
6 int main()
7 {
8     char name[16];
9
10    printf("What's your name? ");
11    scanf("%s", name);
12
13    strcpy(data, " Hello, ");
14    strcat(data, name);
15    puts(data);
16
17    return 0;
18 }
```

Pwnable

- scanf("%s", name);
- AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA

```
ababcdcd <- esp  
00000000  
41414141 <- &name  
41414141  
41414141  
41414141  
41414141  
41414141  
41414141 <- ebp  
41414141 <- ret addr  
41414141  
41414141
```

Pwnable

- ret
- EIP = 0x41414141
- Crash!
- EIP 可以控制
 - 那我們可不可以叫 bash 出來？
 - Demo !

```
ababcdcd <- esp  
00000000  
41414141 <- &name  
41414141  
41414141  
41414141  
41414141  
41414141  
41414141 <- ebp  
41414141 <- ret addr  
41414141  
41414141
```

Pwnable：記憶體戰爭的攻防歷史

- Buffer overflow, return to stack
 - Data Execution Prevention，讓資料不可執行
 - Return Oriented Programming (ROP輕鬆談 by Lays)
 - Stack canary 檢查有沒有 overflow
 - ASLR 讓你猜不中
 - Shadow Stack，用另外一個 stack 存 return address
 - ROP is dead !
- 延伸閱讀：Jump Oriented Programming

Pwnable：CTF常用攻擊手法

- Return to Stack
 - 把 shellcode 寫在 stack 上，想辦法 return 到 stack 上執行
 - 找到 jmp esp
- Return to libc
 - 用 ROP 執行 libc 裡面的 function
- Return to dlresolve
 - 利用 Dynamic Linking ，讓他直接帶我們去 system()

Pwnable : CTF常用攻擊手法

- format string 漏洞 00401234 <- esp, ret addr
- printf("%.4d-%.2d-%.2d", 2016, 1, 1); 2016 <- arg1
- 你知道 %n, %p 嗎 ? 1 <- arg2
- printf("%s", data); 1 <- arg3
- printf(data); aaaaaaaaaa <- int a
- bbbbbbbb <- int b
- 41414141 <- char data[8]
- 41414141
- 00ae5d25 <- stack canary
- cccccccc <- saved ebp
- 00405566 <- ret addr

Pwnable : CTF常用攻擊手法

- format string 漏洞 00401234 <- esp, ret addr
- printf("%.4d-%.2d-%.2d", 2016, 1, 1); 2016 <- arg1
- 你知道 %n, %p 嗎 ? 1 <- arg2
- printf("%s", data); 1 <- arg3
- printf(data); aaaaaaaaaa <- arg4
- %p%p%p%p%p%p%p%p%p bbbbbbbb <- arg5
- Demo! 41414141 <- arg6
- 00ae5d25 <- arg7
- cccccccc <- arg8
- 00405566 <- arg9
- 00405566 <- arg10

Pwnable : CTF常用攻擊手法

- Binary Exploitation 的本質是：讀取或寫入了不該讓你碰到的地方
- 覆蓋的地方是：
 - return address
 - data pointer
 - function pointer
 - GOT entry
 - ... more?

Crypto

- 編碼？隱藏？加密？
 - 加密一定有一把鑰匙！
- 編碼：
 - 你一定聽過：Hex, base64, urlencode
 - 比較少見：uuencode, base85, base32
- 隱藏：
 - 跟編碼一樣，但是目的不同
 - rot13
 - #1NueKY_M

Crypto

- 古典密碼學
 - DBFTBS DJQIFS FODSZQUJPO , Caesar Cipher, key = A (1)
 - Vigenere Cipher : 凱撒加密進階版本
 - 代換加密 (Substitution Cipher)
 - 把每個字母一對一的轉換到另一組字母
 - 頻率分析攻擊，針對明文的特性做分析

Crypto

- 現代密碼學：
 - Block Cipher：把資料切成一塊一塊做加解密
 - DES, AES, Blowfish
 - 如果資料很長？
 - Block Cipher Operation Mode
 - PKCS7 + CBC Mode 有 Padding Oracle Attack
 - Stream Cipher
 - 用 PRNG（亂數產生器）根據 Key 產生連續的 Stream Key 做 Xor
 - OFB Mode 很像是 Stream Cipher

Crypto

- Diffie-Hellman Key Exchange
 - 如果連線被竊聽，要怎麼安全的進行加密連線？
- Hash (雜湊)
 - 純粹用來給你一些資料，產出固定長度的數值
 - Hash Collision
 - Length Extension Attack
 - 密碼用的 Hash 演算法：
 - scrypt, bcrypt, pbkdf2, argon2

Crypto

- 非對稱密碼學：
- 課本上一定有的：RSA， $N = pq$
 - RSA 需要安全的 padding：OAEP
- 楕圓曲線密碼學（ECC）
- ElGamal、Rabin
- 基於離散數學的加密系統
 - RSA 系統基於費馬小定理和 Finite Field Arithmetic
 - 瞭解 RSA 的運作，不會證明至少可以背出產生 Key 和加解密的方式

Crypto : 工具

- rsatool.py
- xor tool
- <http://factordb.com/>

Forensic

- 數位鑑識，從不會動的 Memory Dump / Disk Image 挖資料
- Memory Forensic
 - 知道 kernel 裡面有些什麼結構體，要如何定位、找出這些結構體
 - 什麼是 Virtual Memory、TLB？
 - Windows: EPROCESS / Linux: task_struct, mm_struct
 - 從 Process List 開始
 - Dump 可疑的 process，還原出 executable file 進行分析
 - File Cache、Network Connections

Forensic

- Disk Forensic
 - Registry HIVE file (Windows)
 - Dump LM/NTLM hash (Windows 登入密碼 hash)
 - 抓 rootkit
 - 加密隱藏磁區？ (TrueCrypt)
 - EFI rootkit

Forensic : Tool

- volatility
- 一個超強的 Memory forensic framework
- 除了商業軟體之外，最好的 Free software 選擇
- 可以加掛 plugin 來擴充功能
- pslist, dump process memory, dump module file, dump driver file, dump cached file, dump registry, dump password hash (mimikatz), dump AES key...

Misc & Stego

- 紿你一些奇怪的挑戰，隱藏資訊在資料中
 - 圖片 LSB
 - 在圖片後面附加一個壓縮檔
 - base64 padding bit 可以藏資料
 - 壞掉的 binary data
 - 摩斯電碼
- 工具：
 - openstego, stegosolver, 010 Editor (binary template)

Misc & Stego : Tool

- 010 Editor
- Binary template 超好用
- openstego
- stegosolver
 - 圖片題可以用這個看 LSB
- snow (<http://www.darkside.com.au/snow/>)

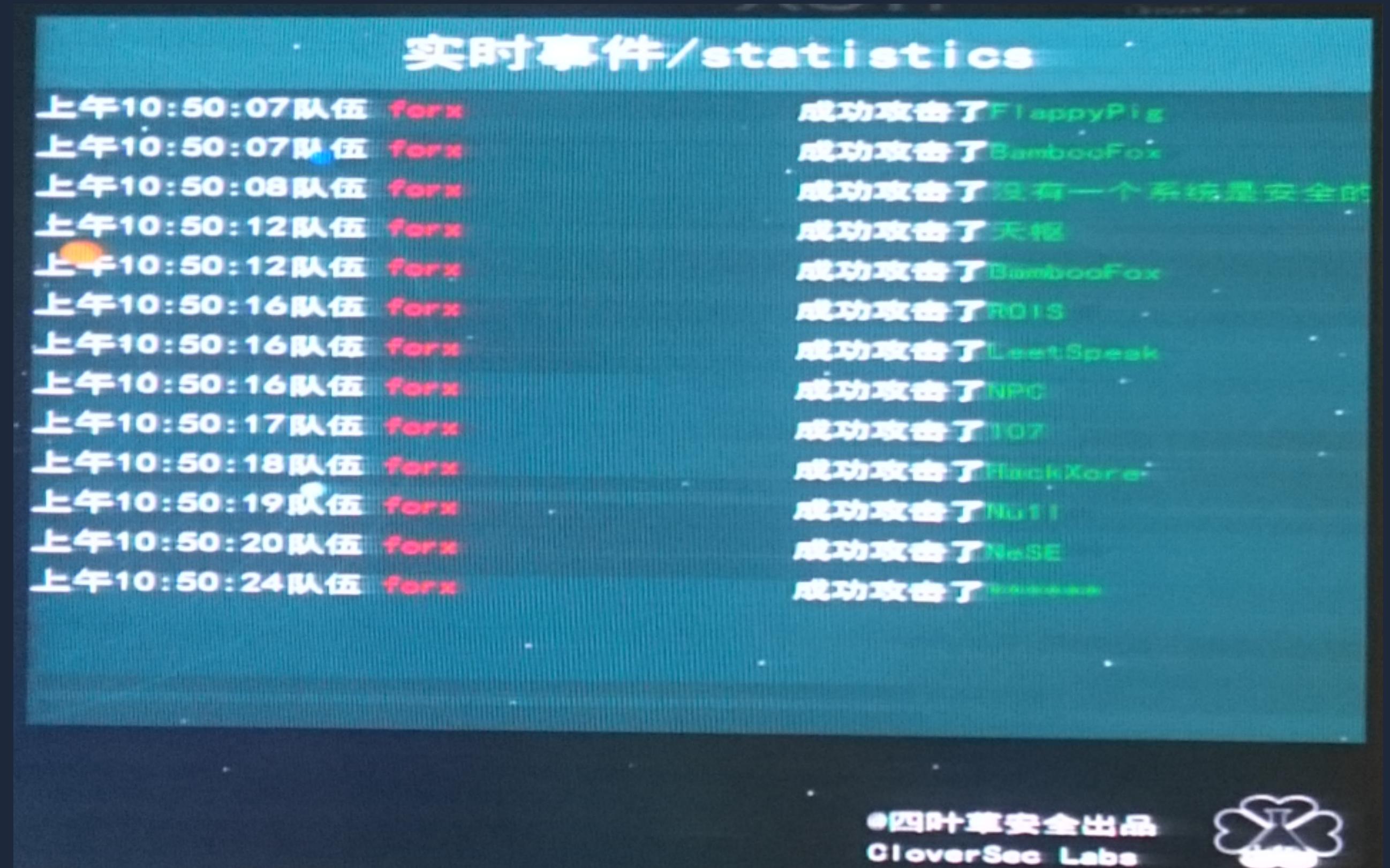
Attack & Defense

- 計分板上會有：
 - 各個服務的狀態
 - Alive, Pwned, Down
 - 該回合的分數變動量
 - 總分數

2015 HITCON CTF FINAL								
Round: 217								
Rank	Team	Avatar	hitree	webful	otherworld	ghostparty	Score	Δ
1	Cykorkinesis		ALIVE	ALIVE	ALIVE	ALIVE	43869.91	-
2	Blue-Lotus		ALIVE	ALIVE	ALIVE	ALIVE	36763.42	-
3	LC↯BC		ALIVE	ALIVE	ALIVE	ALIVE	28742.91	-
4	Oops		ALIVE	ALIVE	ALIVE	ALIVE	27388.03	-
5	PPP		ALIVE	ALIVE	ALIVE	ALIVE	24940.60	-
6	Shellphish		ALIVE	ALIVE	ALIVE	ALIVE	19502.03	-
7	!SpamAndHex		ALIVE	ALIVE	ALIVE	ALIVE	17149.05	-
8	Dragon Sector		ALIVE	ALIVE	ALIVE	ALIVE	15954.68	-
9	fuzzi3		ALIVE	ALIVE	ALIVE	ALIVE	13617.66	-
10	BambooFox-DSNS		ALIVE	ALIVE	ALIVE	ALIVE	13014.71	-
11	Samurai		ALIVE	ALIVE	ALIVE	ALIVE	12753.95	-
12	TokyoWesterns		ALIVE	ALIVE	ALIVE	ALIVE	10093.76	-
13	<(_ _)> shik		ALIVE	ALIVE	ALIVE	ALIVE	8891.58	-

Attack & Defense

- 每個隊伍有自己的 Gamebox
- 有一樣的題目，一樣的漏洞
- 攻擊其他隊伍，防禦自己
- 以 Pwn 為主，亞洲偶爾有 Web
- 主辦方有 Service Check

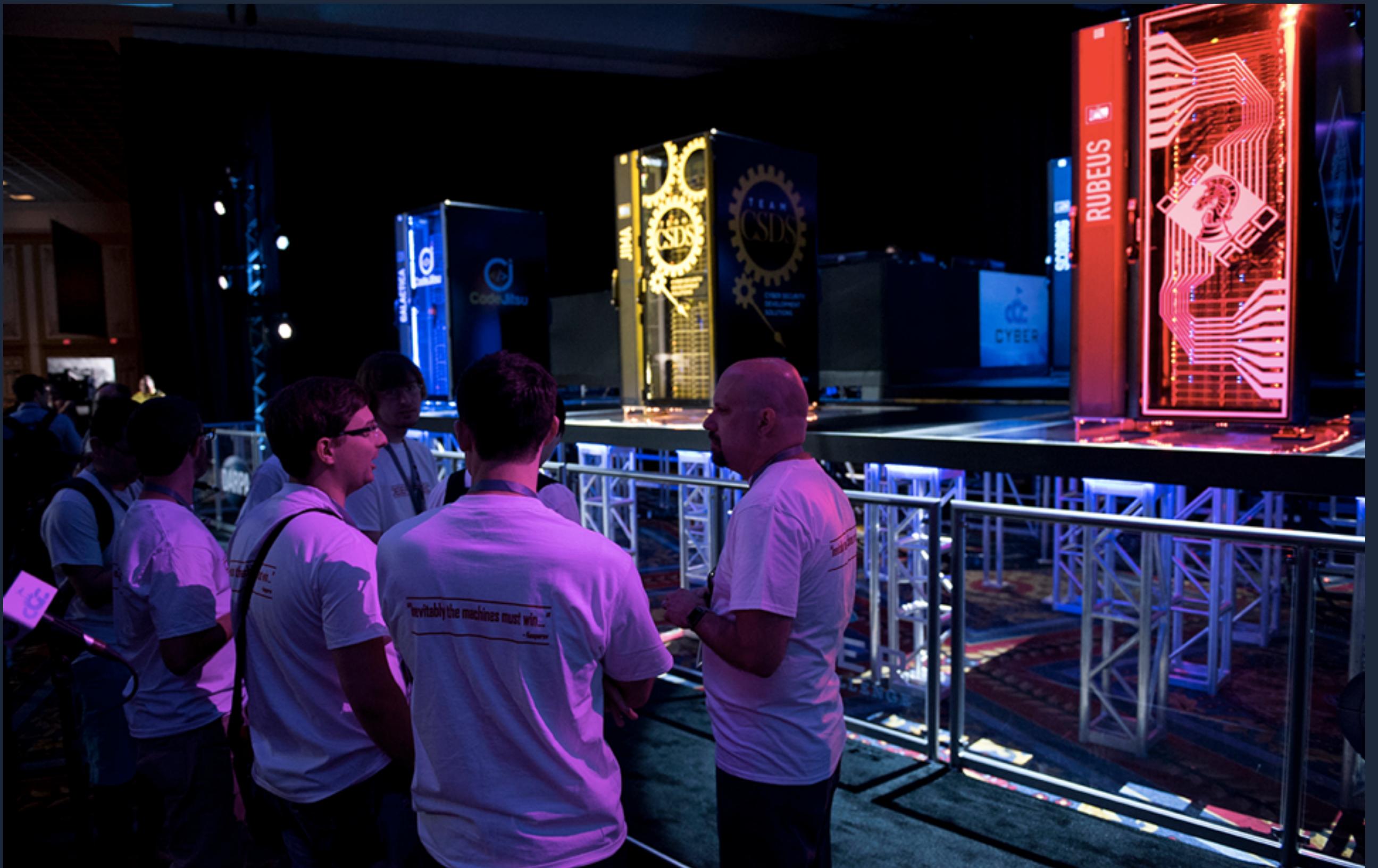


Attack & Defense

- 策略很重要
- 流量分析於重放攻擊
- 放後門
- 如何 binary patch
 - IDA Pro
 - heditor
 - cheat engine
 - LD_PRELOAD

DARPA CGC

- 機器人打 Attack & Defense
- 預知詳情請看上午的場次 XD



Q & A
