決戰 CTF

組員： B05901011 許秉倫 B06901145 李子筠 B06901090 沈哲瑋

競賽成績

- RITSEC CTF 2019 - 60th out of 900 teams (2790 points)
- TUCTF 2019 - 510th out of 1005 teams (1394 points)
- X-MAS CTF 2019 - 1375th out of 1744 teams (25 points)

PWN題解

安全機制：

```
Arch:     amd64      - 64 - little
RELRO:    Full RELRO
Stack:    Canary found
NX:       NX enabled
PIE:      PIE enabled
FORTIFY:  Enabled
```

ROP：

Return-oriented programming 是用於繞過 NX 安全機制的方法。

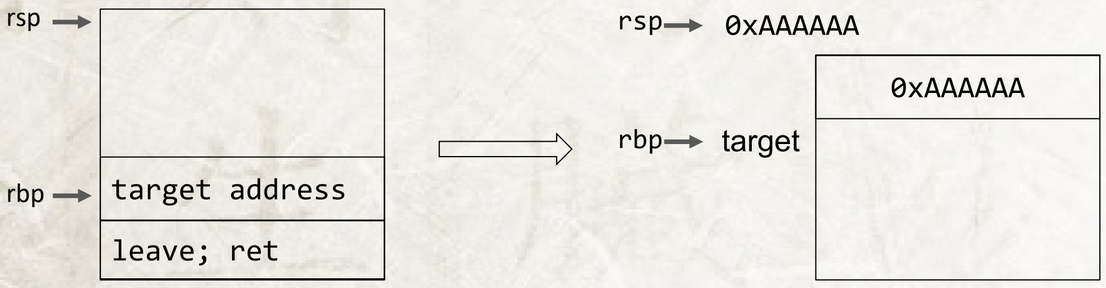
ROP 主要由以 ret/jmp 結尾的程式片段組成，稱之為 gadget。

在 stack 上寫入 gadget，並控制程式流程，return 到 gadget 串上面，就可以觸發 gadget 執行，再利用 ret/jmp 回到 stack 上。
如此往復執行，就能從程式片段組合出目標功能。

0x400686	→	pop rdi; ret
0x6b6000		
0x4100f3	→	pop rsi; ret
0x68732f6e69622f	→	“/bin/sh\0”
0x44709b	→	mov rdi; mov rsi; ret
0x44beb9	→	pop rdx; pop rsi; ret
0		
0		
0x415714	→	pop rdx; pop rsi; ret
0x3b		
0x47b68f	→	syscall

Stack Pivoting：

ROP 常需要很大的記憶體來存放 gadget，而 stack 上可寫入的地方有限，因此需要改變 stack pointer register 的值，把可寫入區域當成 stack。
透過將 rbp 修改成目標 address - 8，return 到 leave; ret 的 gadget 上，可讓 rsp 指向 目標 address，轉移 stack。



解題過程：

Stack canary → 利用 buffer 長度不一致，利用 memcmp 取得 stack canary 以及 image base address

Stack Pivoting → 利用 buffer overflow 將 stack pointer 移動至 buffer 上

ROP → 利用 buffer 中的 gadget 達到 ROP，取得 libc address

Shell → 利用 read 修改 buffer 中的值，寫入開 shell 的 gadget

Reverse題解

解題過程；

用 IDA 打開 KeyChecker.exe，再看其Decompile 結果。
首先會看到

```
if ( dword_40C040 != 1985 )
    puts("[!] WARNING: \n\tit might be some trouble if
you're not in 1985 year.");
```

提示我們 dword_40C040 就是現在年分，且年分不在1985可能會錯。

接著再往後看，可以看到三個迴圈：

1. 輸入的內容全部跟 0x20 做 bit-wise or。
2. 輸入的內容再根據年分做 transform，然後跟位於 byte_408008 的字串比較。如果不同就會印出"oops"並跳出迴圈。
3. 輸入的內容再跟 byte_40801C做xor ，最後當成 flag 印出來。

從這幾個迴圈就能猜出來，flag 就是 byte_408008 跟 byte_40801C 兩者的內容做 xor。接下來只需找出這兩者就好：

```
.data:00408008 ; _BYTE byte_408008[20]
.data:00408008 byte_408008     db 1Dh, 13h, 10h, 18h, 51h, 4Ch, 4Fh, 1Ch, 12h, 51h, 0Bh
.data:00408008                                         ; DATA XREF: _main+197↑o
.data:00408008                 db 8, 50h, 51h, 50h, 51h, 50h, 51h, 50h, 0
```

所以繞了一大圈，其實不需要一步一步回推，只要把兩個字串找到並做 xor 就好。

```
str1 =
"\x1d\x13\x10\x18\x51\x4c\x4f\x1c\x12\x51\x0b\x08\x50\x
51\x50\x51\x50\x51\x50"
str2 = "[_Q_*\x1c\nC3\x02TM\x11\x02\t,pqp"
print(''.join(chr(ord(i)^ord(j)) for i, j in zip(str1,
str2)))
# FLAG{PE_!S_EASY}
```

文言CTF

敘述：
給你一段文言文的程式碼，以及其output，
你必須有足夠的文學底蘊，才能從中找出flag。

Output: 217 215 213 191 193 187 207 227 187 203 225 199 191 229

工作分配

- 李子筠：強大技術支援、解PWN, Reverse題目
- 許秉倫：架設文言文網站
- 沈哲瑋：設計文言文題目、講解Reverse題目