

2023 SWING CTF write-up

31기 육은서

[MISC] Nogada

zip파일 안에 또 하나의 zip파일이 있는 구조였고 8740.zip부터 시작하는 파일이었다.

```
import zipfile

i=8740
while i > 0:
    #i.zip인 파일 찾기(i는 8740부터 시작)
    my_zip=zipfile.ZipFile("C:\\Users\\82104\\Downloads\\"+str(i)+".zip")
    #i-1.zip으로 지정된 경로에 저장하기
    #본 코드에서는 다운로드에 저장하여 반복문을 돌 때 다운로드에서 다시 파일 탐색이 가능하게 설정
    my_zip.extract(str(i-1)+".zip", "C:\\Users\\82104\\Downloads\\")
    int(i)
    i-=1
```

[MISC] Click

홈페이지에 클릭 버튼이 있었고 클릭할 때마다 count가 +1되었다.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from webdriver_manager.opera import OperaDriverManager
from selenium.common.exceptions import NoSuchElementException
from time import sleep

#1. 셀레니움으로 원하는 링크 열기
chrome_options = webdriver.ChromeOptions()
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=chrome_options)

driver.get('http://hspace.io:20001/')

for _ in range(10000):
    #2. 버튼 요소 찾기
    button=driver.find_element(By.CSS_SELECTOR, 'input[type="submit"]')

    #3. 버튼 클릭
    button.click()
```

셀레니움을 이용해서 웹 크롤링을 하였다. 사실 웹드라이버를 설치해야 하는데 제대로 작동되지 않아서 다음 링크를 참고하여 웹드라이버 설치 없이 셀레니움 코드를 사용했다.

[파이썬]chrome-driver 설치 안하고 사용하기 <https://seong6496.tistory.com/330>

Pip install webdriver-manager

Pip install selenium

-위 명령어를 이용하여 설치 후에 파이썬 코드를 작성해주면 드라이버 설치 없이 사용할 수 있다.

버튼 요소 찾기는 f12 누른 뒤 ctrl+shift+c, 원하는 요소를 누르면 코드가 나온다. 마우스 우클릭으로 copy>다양한 방식의 코드 선택을 할 수 있는데 처음에는 Xpath 방법을 택했었다.

근데 클릭 작동이 안되어서 나중에 쿼리셀렉터 복사하고 CSS_SELECTOR 요소 찾기로 코드를 수정했더니 정상적으로 작동했다.

[MISC] eagles

<https://github.com/ragibson/Steganography>

위 링크 참고했다. Stego-lsb에서 WavSteg는 .wav 파일에 무언가를 hiding하거나 recovering 할 수 있다. 주어진 eagles.wav 파일을 Recovering Data 예제를 보며 작성했는데 텍스트가 모두 깨져서 나왔다.

나중에 풀이과정을 들어보니 n 옵션의 개수를 조정해가면서 답을 찾는 문제였다. 이때 n은 LSB를 몇번 돌릴건지 설정하는 옵션이다.

```
user1@user1-virtual-machine:~$ stegolsb wavsteg -r -i eagles.wav -o eeag.txt -n 5 -b 1000
Files read          in 0.01s
Recovered 1000 bytes in 0.00s
Written output file in 0.00s
user1@user1-virtual-machine:~$ cat eeag.txt
hspace{I_Am_v3ry_h4ppyyyyyyyyyyyy}user1@user1-virtual-machine:~$
```

[Reversing] readme

문풀 들었을 때 선배님들이 코드가 있으면 아이디어를 우선적으로 돌리는 거 같았다. Readme 파일을 아이디어에 돌렸더니 rdx에 flag1234로 저장되어 있었다.

```

lea    rdx, flag1      ; "hspace{"
mov     rsi, rdx        ; src
mov     rdi, rax        ; dest
call    _strcat
lea     rax, [rbp+s]
lea     rdx, flag2      ; "hey_hey_"
mov     rsi, rdx        ; src
mov     rdi, rax        ; dest
call    _strcat
lea     rax, [rbp+s]
lea     rdx, flag2      ; "hey_hey_"
mov     rsi, rdx        ; src
mov     rdi, rax        ; dest
call    _strcat
lea     rax, [rbp+s]
lea     rdx, flag2      ; "hey_hey_"
mov     rsi, rdx        ; src
mov     rdi, rax        ; dest
call    _strcat
lea     rax, [rbp+s]
lea     rdx, flag3      ; "good_job"
mov     rsi, rdx        ; src
mov     rdi, rax        ; dest
call    _strcat
lea     rax, [rbp+s]
lea     rdx, flag4      ; "_lets_go_:D}"

```

rdx에는 다음과 같이 저장되어 있다.

Flag1+flag2+flag2+flag2+flag3+flag4

정리하면 hspace(hey_hey_hey_hey_hey_hey_good_job_lets_go_:D)가 나온다

[Reversing] Random defense

Seed값이 같으면 random 값도 같고 shuffle규칙이 같다.

먼저 seed값을 설정해주고 32개 랜덤 숫자를 출력하고(a)

셔플 돌린 후를 출력하여(b) 그 두 개 값의 변화를 적었다.(a와 b를 비교, 1번째 숫자가 5번째로 가는 등 규칙 작성)

본 문제에서 주어진 것은 random_table과 xor값(random_table과 flag를 xor연산하고 셔플을 돌린 값)이다. Flag는 미지수이다. 식으로 표현하면 다음과 같다.

1. Random_table^flag==xorbeforeshuffle
2. Xorbeforeshuffle.shuffle==xor

이것을 거꾸로 거꾸로 연산해서 flag를 구해주면 되는데 xor을 shuffle을 이전으로 되돌리는 방법 보다는 다음과 같이 계산했다.

1. Random_table도 shuffle을 돌린다. 씨드값이 있기 때문에 xor이 셔플된 규칙과 같은 방법으로 돌려진다.
2. Random_table_aftershuffle과 xor을 역연산하면 flag_aftershuffle값이 나온다.
3. 위에서 구한 shuffle 규칙을 이용하여 flag_aftershuffle을 flag로 되돌린다.

Xor의 역연산은 xor이므로 다음과 같이 계산한다.

```
import random
random.seed(5952)
xor=b'\x0ea3\x1dq\x98\xab\x9f\xec\x95\xd4#\x8d+-,\x7a\x9d\x85\xd1\xfa0\xbb\x89z\x9c\x83\x19\xb3\x91Z'

random_table = bytearray(random.randrange(1, 255) for _ in range(32))
for v in random_table:
    print(v)

random.shuffle(random_table)
print("\n\n")
for v in random_table:
    print(v)
k=bytearray(random_table[i]^xor[i] for i in range(32))
print(k)
```

프린트된 k를 셔플 규칙의 반대로 맞춰주면 flag가 나온다.

[Reversing] Simple VM

```

def lets_go(flag, code):
    registers = [0] * 0x100
    flag = [ord(x) for x in list(flag)]
    pc = 0

    while True:
        if pc == len(code):
            break

        cur = code[pc].split('.')
        pc += 1

        if cur[0] == 'A':
            dest = int(cur[1])
            src = int(cur[2])
            registers[dest] += registers[src]

        elif cur[0] == 'B':
            dest = int(cur[1])
            src = int(cur[2])
            registers[dest] ^= registers[src]

        elif cur[0] == 'C':
            dest = int(cur[1])
            src = int(cur[2])
            registers[dest] *= registers[src]

        elif cur[0] == 'D':
            dest = int(cur[1])
            value = int(cur[2])
            registers[dest] = value

        elif cur[0] == 'E':
            dest = int(cur[1])
            src = int(cur[2])
            print(char(registers[dest]), end=" ")
            if registers[dest] != flag[src]:
                pc = int(cur[3])

        elif cur[0] == 'F':
            print(cur[1])

        elif cur[0] == 'H':
            break

if __name__ == '__main__':
    flag = input('Your flag : ')
    program = [
        'D.0.104', 'E.0.0.104', 'D.0.115', 'E.0.1.104', 'D.0.112', 'E.0.2.104', 'D.0.97', 'E.0.3.104',
        'D.0.99', 'E.0.4.104', 'D.0.101', 'E.0.5.104', 'D.0.123', 'E.0.6.104',
        'D.1.65', 'D.2.40', 'A.1.2', 'E.1.7.104', 'D.1.31', 'D.2.84', 'A.1.2', 'E.1.8.104',
        'D.1.38', 'D.2.57', 'A.1.2', 'E.1.9.104', 'D.1.35', 'D.2.81', 'A.1.2', 'E.1.10.104',
        'D.1.66', 'D.2.38', 'A.1.2', 'E.1.11.104', 'D.1.30', 'D.2.75', 'A.1.2', 'E.1.12.104',
        'D.1.44', 'D.2.71', 'A.1.2', 'E.1.13.104', 'D.1.70', 'D.2.25', 'A.1.2', 'E.1.14.104',
        'D.1.61', 'D.2.60', 'A.1.2', 'E.1.15.104', 'D.1.65', 'D.2.46', 'A.1.2', 'E.1.16.104',
        'D.6.56', 'D.7.77', 'B.6.7', 'E.6.17.104', 'D.6.56', 'D.7.74', 'B.6.7', 'E.6.18.104',
        'D.6.38', 'D.7.121', 'B.6.7', 'E.6.19.104', 'D.6.32', 'D.7.70', 'B.6.7', 'E.6.20.104',
        'D.6.42', 'D.7.67', 'B.6.7', 'E.6.21.104', 'D.6.31', 'D.7.109', 'B.6.7', 'E.6.22.104',
        'D.6.32', 'D.7.83', 'B.6.7', 'E.6.23.104', 'D.6.36', 'D.7.80', 'B.6.7', 'E.6.24.104',
        'D.6.47', 'D.7.112', 'B.6.7', 'E.6.25.104', 'D.6.45', 'D.7.91', 'B.6.7', 'E.6.26.104',
        'D.6.50', 'D.7.95', 'B.6.7', 'E.6.27.104', 'D.6.54', 'D.7.75', 'B.6.7', 'E.6.28.104',
        'F.Correct!!!', 'H', 'F.Wrong!!!'
    ]
    lets_go(flag, program)

```

문제 코드는 위와 같다.

Pc가 104가 되면 while문이 break된다. Cur[0]이 E가 되면 if문을 충족할 시 다음 반복에서 break

된다. if문은 `registers[dest] != flag[src]`이다. 때문에 종료되지 않게 `flag`를 `registers[dest]`와 같게 설정해줘야 한다.

E: `flag`값 설정

D: `registers[dest]`값 설정

A.1.2: `registers[1]=registers[1]+register[2]`

B.6.7: `registers[6]=registers[6]^registers[7]`

이를 이용해서 program 배열을 계산해주면 `flag`가 나온다. 코드가 생각이 안 나서 다 계산했다.

```
pc5 flag[2]=112
pc6 regi97
pc7 flag[3]=97
pc8 99
pc9 flag[4]=99
pc10 101
pc11 flag[5]=101
pc12 123
pc13 flag[6]=123
pc14 regi[1]=65
pc15 regi[2]=40
pc16 regi[1]=105
pc17 flag[7]=105
pc18 31
pc19 84
pc20 115
pc21 flag[8]=115
38 57 flag[9]=95
35 81 flag[10]=116
66 38 flag[11]=104
flag[12]=105
flag[13]=115
flag[14]=95
flag[15]=121
flag[16]=111
flag[17]=117
flag[18]=114
flag[19]=95
flag[20]=102
flag[21]=105
```

각 `flag[n]`을 ASCII 코드로 변환하면 다음과 같다.

```
hspace{is_this_your_first_vm}
104 h
115 s
112 p
97 a
99 c
101 e
123{
105 i
pc21 flag[8]=115 s
38 57 flag[9]=95 _
35 81 flag[10]=116 t
66 38 flag[11]=104 h
flag[12]=105 i
flag[13]=115 s
flag[14]=95 _
flag[15]=121 y
```

[Web] Magic-2

```
← → ↻ ⚠ 주의 요함 | hspace.io:20100/magic2/?key=null&privateKey=null
YouTube UN 특수문자 이모티콘... 📌 todo mate 📄 레포트 🌐 SWING Home 📁 교육/세미나/보고... 📝 blog

hspace{b03007d3391ec2a96eec1f90b84474bce39baf0f58449620573338852a34897d} <?php
include "./config.php";

extract($_GET);
if($privateKey === $key) echo $flag;
else echo "Nop";

show_source(__FILE__);
?>
```

Extract에 \$_GET 인자로 매개변수와 값을 넣으면

매개변수->코드 내 변수, 값->초기값으로 설정이 가능하다.

\$privateKey와 \$key의 초기값을 같은 값으로 다시 설정해주면

If문 조건이 성립해서 \$flag를 출력한다.

[Web] Magic-3

```
← → ↻ ⚠ 주의 요함 | hspace.io:20100/magic3/?__flag__=1
YouTube UN 특수문자 이모티콘... 📌 todo mate 📄 레포트 🌐 SWING Home 📁 교육/세미나/보고... 📝 blog 참고문헌 표

Warning: preg_match(): No ending delimiter '_' found in /var/www/html/magic3/index.php on line 4
hspace{d59846f65d11283b2ccd4d2c79fcefa8d6f591171e5d66402e550edc7ef11197} <?php
include "./config.php";

if(preg_match('_',$_SERVER['QUERY_STRING'])){
    die('Do not hacking!');
}
if(isset($_GET['__flag__'])){
    echo $flag;
}

show_source(__FILE__);
?>
```

Preg_match 우회를 해야한다. 이 조건문은 _문자를 검열하는 것처럼 보이지만 실제로 쿼리 문자열에서 _를 포함하는지 확인하려면 '_' 말고 '/'로 고쳐야 검열이 된다.

때문에 그냥 isset안에 있는 __flag__ 변수를 지정해주면 풀리는 문제이다.

[Forensic] Hspace Backdoor3

<https://realsung.tistory.com/170>

위링크를 참고했다. 블로그 안에 있는 사진에 png 구조가 잘 설명되어 있다.

크기	항목	
8	Signature (89 50 4E 47 0D 0A 1A 0A)	
4	0x13	IHRD(헤더) Chunk
4	IHDR	
4	Image width	
4	Image height	
1	PNG bit depth	
1	PNG color type	
1	Compression	
1	Filter	
1	Interlace	
4	CRC	
4	3(r,g,b) * palette number	PLTE(팔레트) Chunk
4	PLTE	
3	Palette 0 RGB	
3	Palette 1 RGB	
...	...	
4	CRC	
4	Chunk Length	투명도 Chunk
4	tRNS	
1	Palette 0 alpha value	
1	Palette 1 alpha value	
...	...	
4	CRC	
4	Chunk Length	IDAT(이미지 데이터) Chunk
4	IDAT	
2	Compression method	
1	마지막 블록 여부 플래그	
2	little-endian block size	
2	little-endian block size NOT [~(block size)]	
1	Filter	
1	Row image data	
1	Filter	
1	Row image data	
...	...	
4	IDAT image data block zcrc	
4	CRC	
4	Chunk Length	IEND(PNG 이미지의 마지막) Chunk
4	IEND	
0	Chunk Data	
4	CRC	

Admin_password의 hex값을 png 구조에 맞게 설정해주면 이미지 파일이 열린다. Png 파일에서 맨 첫부분에 나오는 file signature과 맨 아래 footer signature를 먼저 맞춰줬다. footer부분은 수정

할 것이 없었고 file signature은 JPG로 저장되어 있는 부분을 png로 수정해줬다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	89	4A	50	47	0D	0A	1A	0A	00	00	00	0D	57	48	41	54	%JPG.....WHAT
00000010	00	00	01	2C	00	00	01	3F	08	02	00	00	00	73	5D	69	...,...?.....s]i
00000020	17	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00sRGB.©İ.é..
00000030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	..gAMA...±...üa...
00000040	00	09	70	48	59	73	00	00	0E	C3	00	00	0E	C3	01	C7	..pHYs...Ã...Ã.Ç
00000050	6F	A8	64	00	00	1C	FB	49	44	41	54	78	5E	ED	9D	3B	o"d...ûIDATx^i.;
00000060	B2	DD	36	B3	85	8F	03	BB	5C	E5	44	CA	9C	F9	0F	15	*Ý6'.....»\ãDÊæù..
00000070	7A	08	27	D4	10	14	2A	74	A8	19	28	F4	10	1C	7A	08	z.'Ô...*t".(ô..z.

89 4A 50 47 0D 0A 1A 0A -> 89 50 4E 47 0D 0A 1A 0A

그 뒤에 tweakPNG(PNG 파일 구조를 한 눈에 볼 수 있는 툴)에서 파일을 열고 오류 나는대로 고쳐주려고 했는데 알고보니 WHAT이라고 써져 있는 부분을 IHDR로 바꾸면 되는 거였다.



이후에 이미지 파일을 열면 flag가 보인다.