

網路管理與系統管理 HW 11

B13902022 賴昱錡

2025 年 5 月 24 日

2. 果汁店也有洞 (20 pts)

(a)

(i) DOM XSS

Observe that after we search for something, the query keyword we use will be embedded directly in the page. So we can type following text in the searching textbox, the alert function is then successfully triggered.

```
1 <iframe src="javascript:alert(`xss`)">
```

(v) View Basket

I open Burp Suite's browser and intercept its traffic, then I observe that when I click Your Basket, there's a GET request contains something like GET /rest/basket/1 HTTP/1.1 (if we login admin), then try to change it to 2 or 3 and forward the request, the content of basket is someone else's.

(b)

4. TESTING in the FUZZ (41 pts)

(a)

mutation-based fuzzing is about mutating the existing input values but in lack of understanding the format of the data. generation-based fuzzing is about generating input based on specific or expected format/structure. subsection*(b)

5. 敗北協定太多了！(28 pts)

6. 猫物語（赤） (34 pts)

(a)

我得到的 flag 是: NASA_HW11{pseudorandomness_does_not_guarantee_unpredictability}. 可以觀察到 fatcat.py 得到 random number 的公式是 $k = (ak + c) \bmod m$, k 是當前的 state, 其中 a 和 c 是可以透過連續的 3 次 state output 得到的, 假設我們亂猜後得到 3 個正確的 state: k_1, k_2, k_3 . 則他們會有以下關係式:

$$k_2 = (ak_1 + c) \bmod m$$

$$k_3 = (ak_2 + c) \bmod m$$

透過一些簡單的數學, 我們可以得到 $a = (k_1 - k_2)^{-1}(k_2 - k_3)$, $c = (k_2 - ak_1) \bmod m$ (在這裡 $(k_1 - k_2)^{-1}$ 代表其模 m 下的模逆元)。有了 a 和 c 我們就能預測接下來的每個數字, 但 trust 要大於等於 100 才會有 flag1, 所以我們就用 pwntools 猜 100 多次, 再向 server 索取答案。我的 script 如下所示:

```
1 from pwn import *
2
3 server = remote('140.112.91.4', 1234)
4
5 server.sendlineafter(b'Your choice: ', b'1')
6 server.sendlineafter(b'Guess a number: ', b'1')
7 k1 = int(server.recvuntil(b'.').decode().split(' ')[8].rstrip(','))
8
9 server.sendlineafter(b'Your choice: ', b'1')
10 server.sendlineafter(b'Guess a number: ', b'1')
11 k2 = int(server.recvuntil(b'.').decode().split(' ')[8].rstrip(','))
12
13 server.sendlineafter(b'Your choice: ', b'1')
14 server.sendlineafter(b'Guess a number: ', b'1')
15 k3 = int(server.recvuntil(b'.').decode().split(' ')[8].rstrip(','))
16
17 m = ... # too long to fit in, but it's given by fatcat.py
18
19 a = pow(k1-k2, -1, m) * (k2-k3)
20 c = (k2-a*k1)%m
21
22 cur = k3
23
24 for i in range(100):
```

```

25     server.sendlineafter(b'Your choice: ', b'1')
26     cur = (a*cur+c)%m
27     server.sendlineafter(b'Guess a number: ', str(cur))
28
29 server.sendlineafter(b'Your choice: ', b'2')
30 flag = server.recvuntil(b'}')
31 print(flag)

```

(b)

The flag (FLAG2) is NASA_HW11{`Z\iW^b8\$s"f\I[P<"}. Observe that the key is used in a cyclic manner. Since we know the prefix of flag must be NASA_HW11{, and its length is equal to the OTP key (it's 10). By xor the string we got in remote server and the prefix, we can obtain the key, thus the original string.

```

1  import binascii
2
3  tmp = # too long to fit in
4  tmp = binascii.unhexlify(tmp)
5
6  prefix = b'NASA_HW11{'
7  key = bytes([tmp[i] ^ prefix[i] for i in range(10)])
8
9  full_flag = bytes([tmp[i] ^ key[i % 10] for i in range(len(tmp))]).decode()
10 print(full_flag)

```

(c)

The flag (FLAG3) is NASA_HW11{https://youtu.be/1GxwDuV5JMc}. Just brute force since 2^{24} is not that big, so we can store all the answer and its corresponding key in dictionary. (for simplicity, I don't consider the collisions). My script is shown as below:

```

1  from pwn import *
2  import hashlib
3
4  mp = dict()
5  for i in range(2**24):
6      mp[hashlib.md5(str(i).encode()).hexdigest()[0:8]] = i
7
8  server = remote('140.112.91.4', 1234)

```

```
9  server.sendlineafter(b'Your choice: ', b'4')
10
11  for i in range(10):
12      prefix = server.recvuntil(b': ').decode().split('
    ↪  ')[7].strip(' ').rstrip(' ')
13      ans = str(mp[prefix])
14      server.sendline(ans)
15
16  ans1 = server.recvline()
17  ans2 = server.recvline()
18  print(ans1.decode(), ans2.decode())
```

(d)

FLAG4 是 NASA_HW11{y0u_KNOw_r3F13C710n_4774cK}，觀察 fatcat.py 可以發現 verify 的時候名字一點都不重要，所以只要讓 sha256 的結果是對的就行，而同時我們要根據 server 給的 nonce 輸入正確的結果，但似乎需要 shared key 欸？其實只要把 nonce 丟進 prover 然後把 server 輸出的 mac 複製下來，再回到 verify 的過程中，輸入 richardlaiis||<mac>，就可以取得 flag 了，好耶！