KNOWNSEC 知道创宇

知道创宇安全研究团队

针对近期“博全球眼球 OAuth 漏洞”的分析与防范建议

Fooying、Erevus

2014-5-5

鸣谢微博安全团队

1. OAuth 介绍

1.1. OAuth 简介

来自 OAuth 的官网解释：

> An open protocol to allow secure authorization in a simple and standard method from web，mobile and desktop applications.
>
> OAuth is a simple way to publish and interact with protected data. It's also a safer and more secure way for people to give you access. We've kept it simple to save you time.

大概的意思是 OAuth 是一种开放的协议，为桌面程序或者基于 BS 的 Web 应用提供了一种简单的，标准的方式去访问需要用户授权的 API 服务。OAuth 是一个发布并与受保护数据交互的简单方法。这也是一个更安全的访问方式。我们已经保持它的简单来节约您的时间。

简单的说，OAuth 就是第三方的应用可以通过你的授权而不用知道你的帐号密码能够访问你在某网站的你自己的数据或功能。像 weibo、QQ、淘宝等网站都提供了 OAuth 服务，提供 OAuth 服务的网站一般都有很多开放的 API，第三方应用会调用这些 API 来开发他们的应用以让用户拥有更多的功能，但是，当用户在使用这些第三方应用的时候，这些第三方的应用会来访问用户的帐户内的功能和数据，所以，当第三应用要干这些事的时候，我们不能让第三方应用弹出一个对话框来问用户要他的帐号密码，不然第三方的应用就把用户的密码给获取了，所以，OAuth 协议会跳转到一个页面，让用户授权给这个第三方应用以某些权限，然后，这个权限授权的记录保存在提供 OAuth 服务的网站上， 并向第三方应用返回一个授权 token，于是第三方的应用通过这个 token 来操作某用户帐号的功能和数据时，就畅通无阻了。

这其实是一个授权操作的过程，用户注册帐号的网站（如微博等）提供 OAuth 服务，第三方服务想调用用户对应帐号的相关信息做帐号相关的操作，就会请求 OAuth 提供方的授权，当 OAuth 提供方询问用户并得到授权，就会给予第三方服务一些权限做相对应的操作。

1.2. OAuth 2.0

OAuth 分为 OAuth 以及 OAuth 2.0。OAuth Core 1.0 版本发布于 2007 年 12 月 4 日，由于存在可被会话定向攻击(session fixation attack)的缘故，2009 年 6 月 24 日发布了 OAuth Core 1.0 Revision A 版本。最终在 2010 年 4 月，OAuth 成为了 RFC 标准： RFC 5849: The OAuth 1.0 Protocol。

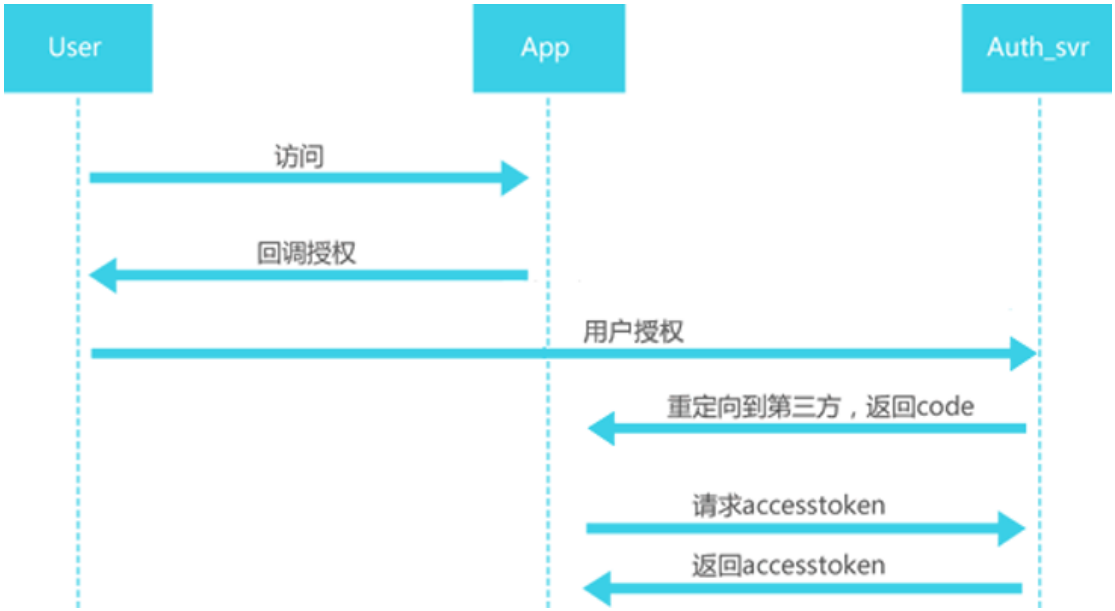
OAuth 2.0 的草案是在 2010 年 5 月初在 IETF 发布的。OAuth 2.0 是 OAuth 协议的下一版本，但不向后兼容 OAuth 1.0。 OAuth 2.0 关注客户端开发者的简易性，同时为 Web 应用，桌面应用和手机，和起居室设备提供专门的认证流程。规范还在 IETF OAuth 工作组的开发中，按照 Eran Hammer-Lahav 的说法，OAuth 于 2010 年末完成。

OAuth 2.0 授权框架允许第三方应用程序获得指定 HTTP 服务的有限的访问权限。

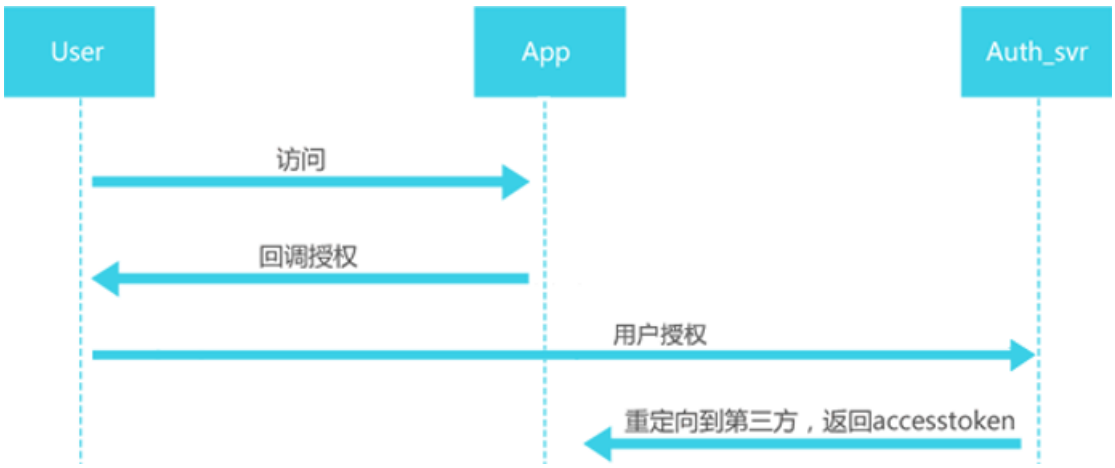
1.3. 授权交互流程

OAuth 的授权有四种方式，主流采用两种方式，分别是 Authorization Code（response_type=code）与 Implicit（response_type=token）。

Authorization Code 方式授权交互：



与 Implicit 方式授权交互：



2. OAuth 与 OpenID 登录工具漏洞

2.1. 漏洞发现与报道

据 Cnet 报道，新加坡南洋理工大学一位名叫 Wang Jing 的博士生，发现了 OAuth 和 OpenID 开源登录工具的“隐蔽重定向”漏洞(Covert Redirect)。

这可导致攻击者创建一个使用真实站点地址的弹出式登录窗口——而不是使用一个假的域名——以引诱上网者输入他们的个人信息。

鉴于 OAuth 和 OpenID 被广泛用于各大公司——如微软、Facebook、Google、以及 LinkedIn——Wang 表示他已经向这些公司已经了汇报。

Wang 声称，微软已经给出了答复，调查并证实该问题出在第三方系统，而不是该公司的自有站点。

Facebook 也表示，“短期内仍无法完成完成这两个问题的修复工作，只得迫使每个应用程序平台采用白名单”。

至于 Google，预计该公司会追踪 OpenID 的问题；而 LinkedIn 则声称它将很快在博客中说明这一问题。

该中文版来自 http://digi.163.com/14/0503/08/9RACJBK900162OUT.html，原英文版 http://www.cnet.com/news/serious-security-flaw-in-oauth-and-openid-discovered/

2.2. 漏洞说明

首先需要明确的一点是，漏洞不是出现在 OAuth 这个协议本身，这个协议本身是没有问题的，之所以存在问题是因为各个厂商没有严格参照官方文档，只是实现了简版。

问题的原因在于 OAuth 的提供方提供 OAuth 授权过程中没有对回调的 URL 进行校验，从而导致可以被赋值为非原定的回调 URL，甚至在对回调 URL 进行了校验的情况可以被绕过。利用这种 URL 跳转或 XSS 漏洞，可以获取到相关授权 token，危害到目标用户的账号权限。具体将在下文进行分析。

微博安全团队 4 月中旬已经率先发现该问题，并联合业务部门进行威胁的评估和落地修复方案的敲定，截止今天中午前，回调 URL 校验和校验绕过漏洞在开放平台已经修复上线。

3. 漏洞成因、利用及危害

3.1. 漏洞利用

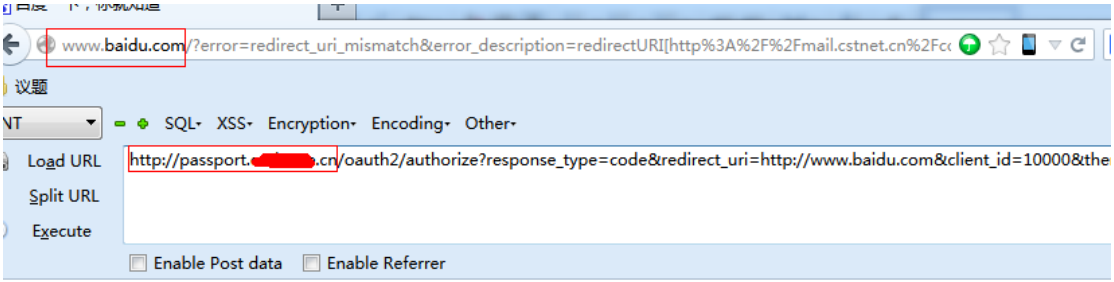
部分 OAuth 2.0 提供未对回调 URL 进行校验甚至校验可以被绕过的情况下，黑客可以通过构造钓鱼页面，用户在访问了黑客构造的页面之后，可以被获取 OAuth 授权中最终返回的 token，通过 token 可以实现登陆该用户的第三方应用或者是调用 OAuth 提供的 API 进行相关操作，包括获取在 OAuth 提供方注册的相关资料等。

3.1.1. 回调 URL 未校验

如果回调 URL 没有进行校验，则黑客可以直接修改回调的 URL 为指定的任意 URL，即可以实现跳转甚至是 XSS。

如：

http://passport.xxxx.cn/oauth2/authorize?response_type=code&redirect_uri=http://www.baidu.com&client_id=10000&theme=coremail



3.1.2. 回调校验绕过

部分 OAuth 提供方在进行的回调 URL 校验后存在被绕过的情况。

如：

https://api.xxx.com/oauth2/authorize?redirect_uri=http%3A%2F%2Fwww.knownsec.om\.www.zhihu.com%2Foauth%2Fauth%2Frequest_token%3Fnext%3D%252Foauth%252Faccount_callback&response_type=code&client_id=30638

未进行绕过修改回调 URL 提示校验失败

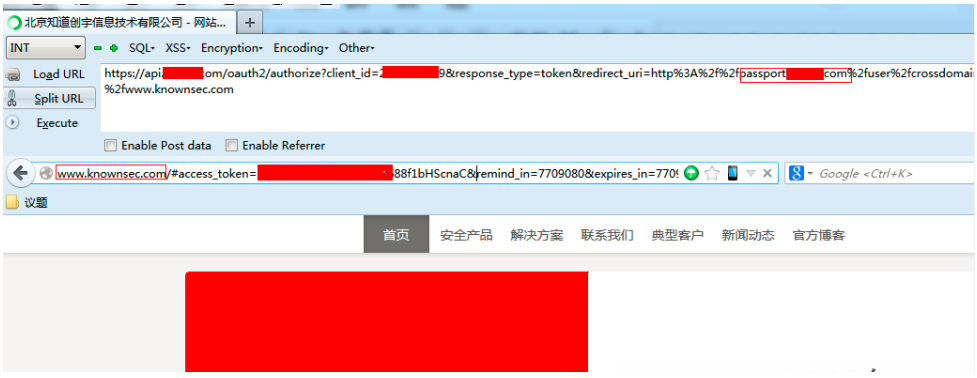
绕过后后修改 URL 校验成功

3.1.3. 利用第三方应用漏洞

这其实也属于校验不完整的而绕过的一种情况，因为 OAuth 提供方只对回调 URL 的根域等进行了校验，当回调的 URL 根域确实是原正常回调 URL 的根域，但实际是该域下的一个存在 URL 跳转漏洞的 URL，就可以构造跳转到钓鱼页面，就可以绕过回调 URL 的校验了。

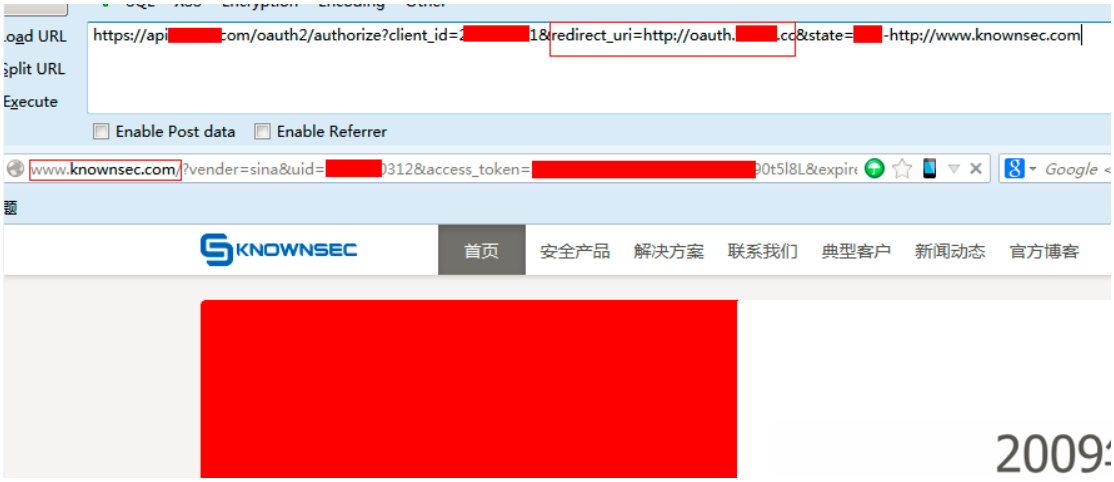
如：

https://api.xxx.com/oauth2/authorize?client_id=204649&response_type=token&redirect_uri=http%3A%2f%2fpassport.xxxx.com%2fuser%2fcrossdomain%3Fact%3Dlogout%26return_url%3Dhttp%3A%2f%2fwww.knownsec.com



3.1.4. 授权验证参数的不正确使用

部分第三方应用在授权过程中采用如 state 里包含 access token 接收的回调 URL，但是因为 OAuth 提供方只对回调 URL，即参数 redirect_uri 的值进行校验，就可以导致黑客可以随意构造回调的 URL，就导致问题的出现。



3.1.5. 绕过方式

1、 redirect_uri=http%3A%2F%2Fwww.a.com?www.b.com

2、 redirect_uri=http%3A%2F%2Fwww.a.com\.www.b.com

3、 redirect_uri=http%3A%2F%2Fwww.a.com:\@www.b.com

其中 www.a.com 为钓鱼或者接收 token 的页面，www.b.com 为实际回调的 URL

3.2. 漏洞危害

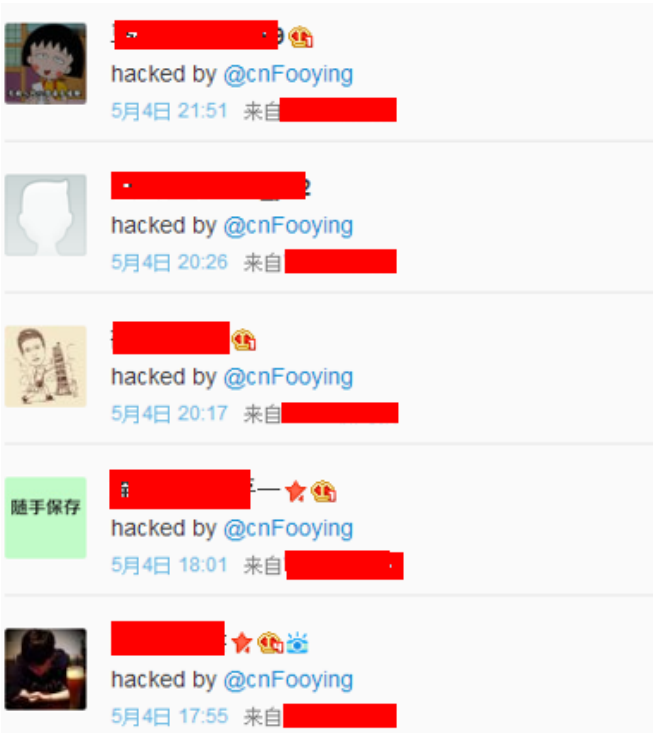
3.2.1. URL 跳转

黑客可以利用该问题构造相关的钓鱼页面，诱使用户访问从而造成用户的帐号被窃取等相关损失等。

3.2.2. API 调用

因为黑客可以通过给问题获取到用户的相关授权 token，可以通过 token 调用 OAuth 提

供方的相关 API 方法进行相关的操作，包括获取用户资料、发表微博等等（如腾讯 API 调用可以参考 http://wiki.open.qq.com/wiki/API%E5%88%97%E8%A1%A8）。



3.2.3. CSRF

利用 CSRF 技巧进行隐蔽攻击，可以获取到用户的 token，然后使用 token 调用相应开放平台的 API 接口，登陆第三方应用并对用户的账户进行相关操作。

这意味着黑客可以通过该问题构造对用户进行钓鱼甚至是获取用户的注册资料、登陆用户帐号进行相关操作等。

4. 漏洞影响

通过对国内部分提供 OAuth 2.0 的网站进行测试和调查，发现均不同程度的存在以上的问题。

OAuth 提供方	回调 URL 校验	可利用第三方应用漏洞	校验绕过
新浪微博	是	是	是（已修复）
百度	是	否	未测试
腾讯	是	是	是（已修复）
360	是	未测试	未测试
开心网	否	是	
人人网	是	是	未测试
淘宝网	是	是	未测试
天涯	否	是	
搜狐	否	是	
网易微博	否	是	

从测试结果可以看出，除了百度绕过未进行测试外，其他都存在问题，而且好几个甚至对回调 URL 都没有进行校验，而对回调 URL 进行校验了的又可以被绕过。

5. 漏洞防范

5.1. OAuth 提供方

4、 对 redirect_uri 进行全路径验证，避免 URL 跳转情况

5、 参数 state 即用即毁

6、 首次授权，强制验证

7、 获取 access_token，验证 App secret

8、 回调 URL 进行跳转校验等

9、 加强 redirect_uri 验证，避免绕过

5.2. 普通用户

对于普通用户来说，其实没有什么好恐慌的，这次问题的利用的前提是对构造 URL 的访问，所以主要是针对 URL 提高警惕和识别，需要注意以下几点：

1、 只授权给可信的第三方应用

2、 不要访问不明来路的链接，正常的应用授权应该是通过页面中的登陆按钮等方式进行的。

附录

1. http://www.cnet.com/news/serious-security-flaw-in-oauth-and-openid-discovered/
2. http://homakov.blogspot.com/2014/05/covert-redirect-faq.html
3. http://dannythorpe.com/2014/05/02/tech-analysis-of-serious-security-flaw-in-oauth-openid-discovered/