Security Analysis Of OAuth

Huang Zheng
Xiong Jiaxi
He Hui
2013.07
Let’s shopping at 1mall
Log in, we have serval choices~~
No longer need to create new account for 1mall
Log in with my QQ account!
Log in successfully, shopping!

1号商城 需要得到您的授权来进行以下操作

访问你的详细资料
昵称、头像、性别等信息

授权后表明您已同意QQ登录使用协议，您也可在授权管理中取消授权。

在QQ空间个人中心添加QQ登录模块？
确认 跳过
After the 3Q war, open is almost the most popular word, and all kinds of applications based on open platform is approaching.

Docking with the web platform, we can:

• No Registering.
• No worrying about the storage and transmission of large amounts of user account information
• Bringing in flow, resources of users
Through the API, platform can get a lot of access, provide users with more and better services. This is a win-win situation for developers and platform providers.

QQ login, the various microblogging login and SNS landing also seems to be a third-party website or application essential button.
What is the Secret?

What’s the big secret when we use the existed account to log in the third website?

http://openapi.qzone.qq.com/oauth/show?which=Login&display=pc&response_type=code&client_id=100304213&redirect_uri=https://passport.1mall.com/qq/callback.do&scope=get_user_info&state=f52cfcf7034102387df70c7d1cac0abc
Agenda

- Introduction of OAuth
- The authorization process
- Security analysis
What is OAuth?

- Oauth is security certification standards of open platform. Allow the client to access the protected resources without user name and password.

- Oauth allows the user provide a token, but not the user name and password. The token is used to authorized a specific website to access specific resources. In this way, the user can authorize the third website to access the protected resources stored in the service provider.
Authorization

Three parts involved in the process:

- **Service Provider**: The web site who stores some protected resources, such as Blog, Twitter.
- **Users**: The owner who saves some private information in the Service Provider, such as photos, articles.
- **Client**: The third part application, who wants to access some private information of the users, including Web application, desktop application, mobile application and so on.
Authorization

- **Oauth Terminology**
  - **Token**
    - Request Token: The temporary token which obtain from the service provider the first time that the client request.
    - Access Token: A token obtained from the service provider after user’s authorization, which can be used to access the protected resources.
  - **URLs in OAuth**
    - Request Token URL: A URL of server to obtain Request Token.
    - User Authorization URL: To obtain user authorized Request Token.
    - Access Token URL: A URL to obtain Access Token with authorized Request Token.
  - **Other Parameters**
    - Oauth_callback: The call back URL
    - Oauth_consumer_key: The identifier portion of the client credentials, which is equivalent to a username. Consumer must register in the Service Provider and then obtain the APP_key.
    - Oauth_nonce: A string generated randomly, which can be used to avoid Replay Attack.
    - Oauth_signature: The signature for the request.
    - Oauth_signature_method: The name of signature method used by the client to sign the request, including HMAC-SHA1, RSA-HAC, PLAINTEXT.
    - Oauth_timestamp: A request timestamp.
    - Oauth_version: The version of OAuth protocol, which is optional.
The process of OAuth

- Obtain the under-authorized Request Token;
- Obtain the user authorized Request Token;
- Obtain Access Token with Access Token.
1. Client asks for Request Token from Tencent Weibo

URI: https://open.t.qq.com/cgibin/request_token?
oauth_consumer_key=0aae7ba5c10d4b939694b0b3b9ffe8ac&
oauth_nonce=b788149731e0ed75da03a6e1e30427c&
oauth_signature=zO5%2FgT%2FP6tbhBBZVuoyfjPwjdxY%3D&
oauth_signature_method=HMAC-SHA1&
oauth_consumer_key=0aae7ba5c10d4b939694b0b3b9ffe8ac&
oauth_nonce=b788149731e0ed75da03a6e1e30427c&
oauth_signature=zO5%2FgT%2FP6tbhBBZVuoyfjPwjdxY%3D&
oauth_signature_method=HMAC-SHA1&
oauth_consumer_key=0aae7ba5c10d4b939694b0b3b9ffe8ac&
oauth_nonce=b788149731e0ed75da03a6e1e30427c&
oauth_signature=zO5%2FgT%2FP6tbhBBZVuoyfjPwjdxY%3D&
oauth_signature_method=HMAC-SHA1&

The responds from the server:
oauth_token=a63f4a0e1242462fb8c11e53159ba294&
oauth_token_secret=40ece707e064128e4f6e92e3c09c692&
oauth_callback_confirmed=true
2. After obtained Request Token, Consumer sends HTTP request, redirecting the user to finish user authorization.

https://open.t.qq.com/cgi-bin/authorize?
oauth_token=80484deccbeb142dfaf5791c9cb9cd256

Then the server constructs the request URI by adding the following required parameters the callback URI query component.

http://qunmgr.qq.com/oauthapi/callback.php?
oauth_token=80484deccbeb142dfaf5791c9cb9cd256&
oauth_verifier=335566
3. Use Request Token to obtain Access Token
Consumer makes HTTP Request:
https://open.t.qq.com/cgi-bin/access_token?
oauth_consumer_key=0aae7ba5c10d4b939694b0b3b9ffe8ac
oauth_nonce=35885b07fe7dc53942d405e552d086fe&
oauth_signature=TXBihj%2FH4avLJaWmjYARDVVCu8M%3D&
oauth_signature_method=HMAC-SHA1&
oauth_timestamp=1288933101&
oauth_token=66ae88faecd14a1f8e826d35d5e857a7&
oauth_verifier=335566&
oauth_version=1.0

The server must verify the request and if valid, respond back:
oauth_token=ffc37464473e48a2b2226350dc98210&
oauth_token_secret=3fe92ec42e936672ceca549d432c237d&
name=starjiang
The security analysis---1st

1. Oauth Signature
   The signature method of Oauth: HMAC-SHA1, RSA-SHA1 and PLAINTEXT
   however only HMAC-SHA1 can be used in Tencent Weibo.

2. All requests should be signed, and the platform verify if the request is valid with the signature. And in HMAC-SHA1 algorithm, Signature Base String and Secret should be combined to generate signature.
1. Signature Base String is consisted of three parts and connected by 
   “&”
   - Http Methods: GET/POST
   - The encoded URL:
   - The original URL:
     https://open.t.qq.com/cgi-bin/request_token
   - The encoded URL:
     https%3A%2F%2Fopen.t.qq.com%2Fcgi-bin%2Frequest_token
   - The encoded parameters:
     oauth_callback%3Dwww.qq.com%26oauth_consumer_key%5D49b0bes73529
     43a1a5609f9e30346201%26oauth_nonce%3D90523669%26oauth_signature_
     method%3DHMAC-
     SHA1%26oauth_timestamp%3D1298513816%26oauth_version%3D1.0

2. The secret is consisted of two parts: the APP secret and Token Secret.
Figure 1: Consumer sends the signature to Server.

```
secret

signature

String

send the string and signature to the server
```
Figure 4 Server receive the signature

密钥

字符串

服务器接收字符串和签名

签名

比较与服务器返回的签名
In 2009, a session fixation attack against the OAuth Request Token approval flow has been discovered. But the attack is based on the OAuth 1.0, and then OAuth 1.0 Revision A specification has been published to address this issue.
To fix this issue, OAuth 1.0a arises at the historic moment, which we discussed above is all about this version. It fixes these details as follows:

- Add the "oauth_callback" to the request for the temporary credentials. In this way, the "oauth_callback" can be a part of the string needed in Signature so that we can avoid attacking.
- When Service Provider obtains the user’s authorization and redirects the user to consumer, it response "oauth_verifier", which will be used when the consumer requests Access Token. However, the attacker can’t guess the value of the "oauth_verifier".
Replay Attack: The Attack sends a received packet to a destinate host to cheat the system, mainly used in the authentication process, undermine the correctness of the certification.

Oauth_nounth: A nonce is a random string, uniquely generated by the client to allow the server to verify that a request has never been made before. The nonce is unique across all requests, but it is not too realistic to store all the nonce.

Oauth_timestamp: Timestamp. To avoid the need to retain an infinite number of nonce values for future checks, servers may choose to restrict the time period after which a request with an old timestamp is rejected.
OAuth development

- OAuth 1.0
- OAuth 1.0a
- OAuth 2.0
THE END!

THANK YOU!