Vulnerabilities in OAuth-based single sign on systems

Wanpeng Li and Chris J Mitchell

Agenda

- Single sign-on and OAuth 2.0
- Motivation
- Two case studies
- Security analyses
- Concluding remarks

Single sign on (SSO)

- An Internet single sign on (SSO) system allows a user to log in to multiple web sites with just one authentication.
- Increasingly widely used, e.g. in form of
  - Facebook Connect (Oauth 2.0);
  - Google (formerly OpenID and now OpenID Connect).

Terminology

- **Identity Provider (IdP)** authenticates user and vouches for **User** identity to …
- **Relying Parties (RPs)**, which rely on IdP and provide online services to …
- **Users**, who employ …
- **User Agents (UAs)** (typically web browsers), to interact with RPs.

OAuth 2.0

- OAuth 2.0, published in 2012 (RFC 6819) is being widely used as the basis of SSO services, e.g. for *Facebook Connect*.
- It is also being very widely used for SSO by a wide range of popular IdPs in China.
- Issues with use of OAuth 2.0 by Facebook and others have already been identified.
- This motivated study of security of Chinese implementations.

OAuth design goals

- Original goal of OAuth (1.0 & 2.0) not SSO.
- OAuth allows a *Client* application to access information (belonging to a *Resource Owner*) held by a *Resource Server*, without knowing the *Resource Owner*'s credentials.
- Also requires an *Authorisation Server*, which, after authenticating the *Resource Owner*, issues an *access token* to the *Client*, which sends it to the *Resource Server* to get access.

Use for SSO

- When used to support SSO:
  - **IdP** = Resource Server + Authorisation Server;
  - **Client** = RP;
  - **User** = Resource Owner;
  - **UA** = web browser.
- Access token used to provide SSO service.
- Of the 4 ways to get access tokens, we focus on **Authorisation Code Grant**.

OAuth 2.0/SSO – data flows

1. User clicks button on RP website, and UA sends HTTP request to RP.
2. RP sends OAuth 2.0 *authorisation request* to UA, optionally including *state* variable (used to maintain state between request and response).
3. UA redirects request to IdP.
4. If necessary, IdP authenticates User.
5. IdP generates *authorisation response* containing *code* (an authorisation code), and the *state* value, and sends it to UA.
6. UA redirects response to RP.
7. RP sends *access token request* to IdP (directly) containing *code* and *client_secret* (shared by IdP and RP).
8. IdP checks request values and responds to RP with *access token*.
9. RP uses *access token* to retrieve user attributes (specifically the IdP user identifier) from IdP.

OAuth 2.0 – identity federation I

- OAuth 2.0 specs do not provide a standardised approach to identity federation.
- Not surprising given OAuth 2.0 not really designed for SSO.
- Commonly used (ad hoc) means of federation involves RP binding the user-RP account with the user-IdP account, using the unique user ID generated by the IdP.
- The IdP account ID is fetched from the IdP in step 9 of previous slide.

OAuth 2.0 – identity federation II

- After receiving the access token (step 8), RP retrieves the user-IdP account ID.
- RP then binds user-RP account ID to user-IdP account ID.
- One method of achieving binding is:
  - user initiates binding after logging in to RP;
  - user required to log in to IdP;
  - user grants permission for binding;
  - RP completes binding process.

Agenda

- Single sign-on and OAuth 2.0
- Motivation
- Two case studies
- Security analyses
- Concluding remarks

Wide use

- In the relatively short time since OAuth 2.0 specifications published, it has become widely used as basis for SSO (e.g. by Facebook).
- Particularly big uptake in China:
  - some Chinese language RPs support as many as eight (OAuth-based) IdPs;
  - at least ten major websites offer Oauth 2.0-based IdP services.

Known issues

- OAuth 2.0 has been critically examined by a number of authors.
  - Frostig & Slack (2011) found a Cross-Site Request Forgery (XSRF) attack in the *Implicit Grant* flow of OAuth 2.0.
  - Wang, Chen & Wang (2012) found a logic flaw in a range of SSO implementations.
  - Sun & Beznosov (2012) found flaws in OAuth 2.0 implementations.
- However, no published studies of real-life security of Chinese-language sites, despite large numbers and wide use of OAuth 2.0.

Attack countermeasures

- OAuth 2.0 specifications recommend use of *state* parameter in authorisation request & response to protect against XSRF attacks.
- For it to work *state* must be non-guessable.
- Otherwise attacker could include guessed value in a XSRF-generated fraudulent authorisation response.
- We observed that many real-world RPs either omit *state* or use it incorrectly.

Scope of attacks

- New attacks we have discovered are more powerful than previously known attacks.
- Attacks using XSRFs enable false identity federations, so that an attacker can log in at will to victim accounts.
- Attacks do not require victim cooperation (except to visit a malicious website at some point prior to attempting a federation).

Agenda

- Single sign-on and OAuth 2.0
- Motivation
- Two case studies
- Security analyses
- Concluding remarks

General approach

- Investigated properties of range of real-world implementations of OAuth 2.0-based SSO.
- Looked at browser-relayed messages (BRMs) between RPs and IdPs.
- Used Fiddler (open source tool) to capture BRMs, and developed Java parser for BRMs.
- Focussed on attacks on the identity federation 'binding' process.

Scope of study

- We looked at 60 Chinese RPs supporting federation-based SSO using OAuth 2.0.
- Of these 14 did not support the vulnerable binding method.
- Of the remaining 46, a total of 21 (i.e. **nearly half**) were found to be vulnerable to XSRF-based false binding attacks.

Renren Network

- Renren is a social networking site with 320 million users – the 'Facebook of China'.
- It supports several OAuth 2.0-based IdPs for SSO, including Baidu and China Mobile (both major sites).
- We examined federation interactions between Renren and both Baidu and China Mobile.

Ctrip

- Ctrip is a China-focused travel agency with 60 million members.
- Ctrip supports eight Oauth 2.0-based SSO IdPs, including Renren, Wangyi, Taobao, MSN and Sina.
- We looked at federation interactions between Ctrip and Renren.

Agenda

- Single sign-on and OAuth 2.0
- Motivation
- Two case studies
- Security analyses
- Concluding remarks

Renren-Baidu binding attack I

- Suppose user logged in to Renren (RP) wants to bind Renren account to Baidu (IdP) account.
- Renren generates an *auth request* and redirects UA (user browser) to Baidu.
- Renren does **not** include *state* in *auth request*, i.e. no means of binding the *auth request* to the subsequent *auth response*.
- After authenticating the user, Baidu returns an *auth response* containing *code* (via the UA) – the UA adds cookies containing session ID.
- Renren uses *code* to get *access token* from Baidu, and then uses the *access token* to retrieve the Baidu account ID.
- Finally Renren binds its account ID to the Baidu account ID.

Renren-Baidu binding attack II

- Because no *state* value, attacker could replace the *code* in the *auth response* with a *code* generated by Baidu for a separate attacker-initiated interaction.
- Then the user ID that Renren later retrieves from Baidu will be attacker's ID not the user's ID.
- This means Renren will bind the attacker's Baidu ID with the user's Renren ID.
- Catastrophe!
- We tested this using a XSRF approach to perform the substitution, and it worked.

Renren-China Mobile binding attack

- In this case, both *auth request* and *auth response* contain a *state* value.
- However, *state* value is the same for multiple requests and responses (always '9').
- Thus an attack almost identical to the Renren-Baidu attack works, enabling binding of attacker's China Mobile account to victim's Renren account.

Generic Ctrip binding attack I

- Looked at Renren-Ctrip binding process (Renren acting as IdP).
- No *state* value in *auth request*.
- However, *code* substitution attack did not work (not sure why).
- We observed that the initial HTTP request contained a *Uid* (a Ctrip-generated user ID).
- We speculated that is we replaced the *Uid* in an attacker-generated request with a victim's *Uid*, then it might be possible to force Ctrip to bind the attacker's IdP account to the victim's Ctrip account.

Generic Ctrip binding attack II

- We tried it and it worked!
- We analysed this further, and found it would work with many IdPs working with Ctrip.
- The Ctrip implementation contained logic flaws.
- Getting *Uid* values for victims is simple using the Ctrip user forum.
- **In all our attacks we used specially created accounts (no 'real' accounts were hacked).**

Agenda

- Single sign-on and OAuth 2.0
- Motivation
- Two case studies
- Security analyses
- Concluding remarks

Disclosures

- We notified all the affected RPs and IdPs earlier this year, several months before publication of our results.
- We got a mixed response – most major sites fixed the problems and thanked us.
- However, some sites denied that our attacks were a problem …

Reasons for problems

- Perhaps the single most important reason that these attacks arise is because of the lack of standards for OAuth 2.0-based SSO and identity federation.
- This is now partly addressed by OpenID Connect, which builds a standardised identity layer on top of OAuth 2.0.

Recommendations

- In absence of clear standards, guidance from IdPs critical.
- Some IdPs did not clarify use of *state*, and did not even include *state* in their sample code.
- Consequences of not using *state* value were not made clear to RPs.
- Have published detailed list of recommendations for IdPs and RPs.

Publication

- The main results of the study have been published at ISC 2014:
  - W. Li and C. J. Mitchell, 'Security issues in OAuth 2.0 SSO implementations', *in: Proceedings of the 17th Information Security Conference, Hong Kong, China, 12-14 October 2014 (ISC 2014)*, Springer-Verlag **LNCS 8783** (2014), pp. 529-541.