
# Addressing threats to real-world identity management systems

Wanpeng Li and Chris J Mitchell

Information Security Group

Royal Holloway, University of London

## Agenda

- Single sign-on and identity management
- OAuth 2.0
- OpenID Connect
- Uni-IdM: a new approach
- Concluding remarks

2

## 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 SSO service (formerly built using OpenID and now employing OpenID Connect).

3

## Identity management

- An SSO system is just a special case of an identity management system.
- In general, in an Internet-based identity management system, one or more third parties manage aspects of a user's identity on behalf of a user, e.g. they
  - store user attributes;
  - authenticate users on behalf of other parties.

4

## Identity management 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.

5

## Agenda

- Single sign-on and identity management
- OAuth 2.0
- OpenID Connect
- Uni-IdM: a new approach
- Concluding remarks

6

## OAuth 2.0

- OAuth 2.0, published in 2012 (RFC 6819), is being widely used as the basis of English-language SSO services, e.g. for *Facebook Connect*.
- It is also being very widely used for SSO by a range of popular IdPs operating in other languages, notably in China.
- Serious practical issues with use of OAuth 2.0 by Facebook and others have been identified.

7

## 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 *Authorization Server*, which, after authenticating the *Resource Owner*, issues an *access token* to the *Client*, which sends it to the *Resource Server* to get access.

8

## Use for SSO

- When used to support SSO:
  - **IdP** = *Resource Server* (stores user attributes) + *Authorization Server* (authenticates user);
  - **RP** = *Client*;
  - **User** = *Resource Owner* (owns user attributes);
  - **UA** = web browser.
- *Access token* used to provide SSO service (not really what it was intended for).
- OAuth supports four separate protocols (*flows*) which allow a Client to get an *access token*.

9

## 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.

10

## 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.
  - Li & Mitchell (2014) found range of flaws in federation process for widely used Chinese language implementations.

11

## Agenda

- Single sign-on and identity management
- OAuth 2.0
- OpenID Connect
- Uni-IdM: a new approach
- Concluding remarks

12

## Building on OAuth 2.0


- OpenID Connect 1.0 is built as an *identity layer* on top of OAuth 2.0.
- Adds extra functionality aimed specifically at SSO, and hence should help to address OAuth probems.
- Adds a new type of token to OAuth 2.0, namely the *id token* [a JSON web token].
- The *id token* contains claims about authentication of end user – generated by entity known as *OpenID Provider (OP)* [=IdP].
- It is digitally signed by the OP.

13

## Four ways to retrieve an *id token*


- OAuth (and hence OpenID Connect) supports four ways for a Client (the RP) to retrieve a token from the Authorization Server (IdP):
  - *hybrid flow* [token sent via the UA, using an RP-provided JavaScript client running on UA];
  - *client-side flow* [very similar to hybrid flow];
  - *authorization code flow* [token sent directly from authorization server (IdP) to client (RP)];
  - *pure server-side flow* [not supported by Google].

14

## A large study

ISSE 2015

- We looked at the GTMetrix top 1000 websites providing an English language service.
- Of these, 103 support Google's SSO service based on OpenID Connect.
- We examined all 103 in detail.
- As in OAuth study, we use Fiddler to capture browser-relayed messages, and developed a Python program to analyse these messages.
- No third party accounts were hacked.

15

## Retrieving the *id token*

ISSE 2015

- As mentioned, OpenID Connect supports four ways for a Client (the RP) to retrieve a token from the Authorization Server (IdP).
- Of the 103 websites we examined:
  - 69 use the authorization code flow;
  - 33 use the hybrid flow;
  - just one uses the client-side flow.

16

## Hybrid server-side flow

- Identified a wide range of serious vulnerabilities in many of the 33 RP sites implementing this approach.
- Issues identified include:
  - using Google ID for authentication (3 out of 33);
  - using an unverified access token instead of the id token (13 out of 33);
  - sending an access token across a cleartext link (4 out of 33);
  - session-swapping vulnerability (24 of 33).

17

## Authorization Code flow

- The authorization code flow (used by 69 of 103 RPs) is inherently more secure than the hybrid flow.
- The tokens never pass through the UA, and hence are not at risk from malware running on the user machine.
- However, we still identified a range of security issues.

18

# Authorization code flow issues

- Issues identified include:
  - sending an *access token* over an non-SSL protected link (4 out of 69);
  - stealing an *access token* using a common XSS vulnerability (possible for all 69);
  - sending user information unprotected across a link (11 out of 69);
  - session-swapping vulnerability (24 of 69);
  - CSRF-based forced login (24 of 69).

19

# Agenda

- Single sign-on and identity management
- OAuth 2.0
- OpenID Connect
- Uni-IdM: a new approach
- Concluding remarks

20

## The phishing threat

- Many identity management systems are susceptible to phishing attacks, in which a malicious (or fake) RP redirects a user browser to a fake IdP.
- The user then reveals to the fake IdP secrets that are shared with a genuine IdP.
- This arises because, in the absence of a system-aware client agent, schemes rely on browser redirects.

21

## Lack of consistency

- One huge problem faced by any user is that the user experience of every identity management system is different.
- We all know that users fail to make good security decisions, even when confronted with relatively simple decisions.
- The lack of consistency is likely to make the situation much worse, with users simply not understanding the complex privacy- and security-relevant decisions they are being asked to make.

22

## Privacy

- When using third party IdPs which provide assertions about user attributes, there is a danger that a user will damage their privacy by revealing attributes unintentionally to an SP.
- This is a threat when using systems like OAuth (e.g. as instantiated by Facebook Connect).
- In general, getting privacy settings right is highly non-trivial.

23

## Another new infrastructure?

- It is tempting to try to devise another new scheme which has the practical advantages of OAuth and OpenID Connect, but yet provides robust protection against phishing and privacy loss.
- However, it seems that a new solution is:
  - unlikely to succeed when others (some with a great deal of inertia and incorporating very nice features) have failed;
  - likely to create yet another different user experience, increasing the likelihood of serious mistakes.
- Thus maybe this is not the right approach.

24

## A new approach?

- We now introduce a new approach to the problem.
- It does not involve proposing any new protocols or infrastructures.
- The goal is to try to make it easier to use existing systems, and also to make their use more secure (less prone to phishing) and privacy-enhancing (consistent interface and explicit consent).

25

## Client-based solution

- The scheme we propose involves a client-based user agent.
- This is a single tool which supports a wide range of ID management systems yet provides a single interface to the user.
- The consistent user interface should maximise user understanding of what is happening (and reduce risk of errors).
- It also avoids the need for passive browser redirects, hence mitigating phishing attacks.

26

## Motivation for scheme


- One motivation for the scheme comes from considering CardSpace (and its open source 'twin', Higgins).
- CardSpace acts as client-based agent, and provides a consistent card-based user interface.
- That is, sets of user credentials (relationships with IdPs) are represented to users as cards.
- CardSpace also defines a set of protocols for interactions between IdPs, Clients (user machines) and SPs.

27

## Uni-IdM and OpenID Connect


- In OpenID Connect, the browser is redirected by a RP to an IdP (and vice versa).
- OpenID Connect works with unmodified browsers.
- A major disadvantage is that a malicious RP can redirect the browser to a fake IdP (e.g. to fraudulently obtain user credentials).

28

## Role of Uni-IdM

- The Uni-IdM browser plugin essentially converts a redirect system into an active-client system.
- Redirects are no longer under the control of the RP (and IdP).
- The Uni-IdM client also manages authentication of the user to the IdP.
- The operation of Uni-IdM is completely transparent to the IdP and RP.

29

## General features

- Regardless of the ID system protocols supported by the RP and IdP, Uni-IdM is transparent to both parties.
- That is, no parties (except the user who installs and uses the software) need to be aware of its presence.
- As long as the RP and IdP share at least one identity system, then Uni-IdM operation is possible.

30

## Agenda

- Single sign-on and identity management
- OAuth 2.0
- OpenID Connect
- Uni-IdM: a new approach
- Concluding remarks



## Uni-IdM works!

- A preliminary prototype of Uni-IdM has recently been built by my co-author (Wanpeng Li), and is still under development.

## Related work

- Copies of published papers on Uni-IdM and analyses of the practical security of real-world identity management schemes can be found on Chris Mitchell's home page:

  www.chrismitchell.net



## Questions?

- For further information please contact:
  - Wanpeng Li
    Wanpeng.Li.2013@live.rhul.ac.uk
  - Chris Mitchell
    me@chrismitchell.net and www.chrismitchell.net
- Address:

  Information Security Group
  Royal Holloway
  University of London
  Egham TW20 0EX
  UK