

Browser Support for the Open Authorization (OAuth) Protocol

http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_32.pdf

Hannes Tschofenig, Barry Leiba,
Blaine Cook, Rob Van Eijk

Agenda

- Authentication Mechanisms
- JavaScript Crypto Library Support
- Authorization Interface
- Moving Crypto Into the Browser

Authentication Mechanisms

- Many identity management protocols treat the authentication exchange out of scope
 - So does OAuth
- Problems:
 - Authentication credentials being used are of poor quality (low entropy secrets)
 - Enrollment of strong credentials not implemented in browsers (e.g. enrollment of OTP mechanisms)
 - Authentication mechanisms being used on the Web today are pretty weak.
 - There is no authentication framework that allows for easily exchangeable authentication methods.
 - Other problem (but not related to lack of standardization): weak identity proofing
- Examples to look at: GSS-API, SASL, EAP, PSKC

What mechanisms should browser support?

Authorization Interface


The image shows a browser window titled "Request for permission" with the address bar displaying "http://www.facebook.com/connect/uisever.". The page content includes a header with the Facebook logo and the text "Request for permission". Below this, a message states: "mit-cfp-test-app is requesting permission to do the following:". A small profile picture of Hannes Tschofenig is shown next to the heading "Access my basic information". The text below the heading reads: "Includes name, profile picture, gender, networks, user ID, list of friends and any other information I've shared with everyone." A large grey arrow points from this text to a blue square icon containing two white gears, labeled "mit-cfp-test-app". At the bottom of the main content area, there is a line of text: "By proceeding, you agree to the mit-cfp-test-app Terms of Service and Privacy policy · Report app". The footer of the browser window shows "Logged in as Hannes Tschofenig (Not you?)" and two buttons: "Allow" and "Don't allow".


Request for permission

http://www.facebook.com/connect/uisever.

Request for permission

mit-cfp-test-app is requesting permission to do the following:

 **Access my basic information**
Includes name, profile picture, gender, networks, user ID, list of friends and any other information I've shared with everyone.


mit-cfp-test-app

By proceeding, you agree to the mit-cfp-test-app [Terms of Service](#) and [Privacy policy](#) · [Report app](#)

Logged in as Hannes Tschofenig (Not you?)

Allow **Don't allow**

Authorization Interface, cont.

CNN Social is requesting permission to do the following:

Access my basic information

- Name: Serge Egelman
- Profile Picture
- Gender: Male
- Networks:
 - National Institute of Standards & Technology
 - Carnegie Mellon
 - UVA
 - Microsoft
- User ID: serge.egelman (767455623)
- List of Friends: [Link](#)
- Any other information I've shared with everyone

[Report App](#)

CNNMoney.com

CNN Social

★★★★★

Logged in as Serge Egelman (Not You?)

Allow **Don't Allow**

Authorization Interface, cont.

Google accounts



Launchpad.37signals.com is asking for some information from your Google Account hannes.tschofenig@gmail.com

- Google profile: [hannes.tschofenig](#)

Sign in

Cancel

What guidance can we provide to developers for a consistent permission dialog experience?

What is the role of the browser in this exchange?

JavaScript Crypto Library Support

- JavaScript is an essential language for the Web eco-system. Increasingly popular also on the server-side.
- Problem: No standardized APIs for access to cryptographic functions and key storage.
- One consequence: Cryptographic material cannot be kept confidential with JavaScript-based clients.
- Example:

https://developer.mozilla.org/en/JavaScript_crypto

Would it be useful to standardize a

JavaScript crypto API?

Moving Crypto Into the Browser

- OAuth protocol design is impacted by the capabilities offered by browsers.
- The security considerations capture this quite well:
<http://datatracker.ietf.org/doc/draft-ietf-oauth-v2/>
<http://tools.ietf.org/html/draft-lodderstedt-oauth-security>
- For example,
<http://datatracker.ietf.org/doc/draft-ietf-oauth-v2-http-mac/> suggests a MAC based authentication mechanism that can also be used to implement a more secure cookie/HTTP state management mechanism.

What functionality can we put into the browser platform?