TWeb

Authentication and Authorization

Bertil Chapuis

Overview of Today's Class

- Quiz about last week's lecture
- Correction of last week's assignment
- Authentication and Authorization

? Quiz

❓ Speakup

You can answer to the following Quiz on Speakup.

http://www.speakup.info/

Room Number: **XXXXX**

Once connected, answer to the first test question.

? Question 1

Parmis les architectures suivantes, la ou lesquelles permettent une communication bi-directionnelle?

- Polling
- Long-Polling
- Server-Sent Events
- **WebSocket**
- Aucune réponse correcte

❓ Question 2

Cochez les affirmations correctes à propos de l'API EventStream (SSE).

- **Lorsque le serveur ferme la connexion, le navigateur essaie de se reconnecter automatiquement.**
- Lorsque le navigateur ferme la connexion, le serveur essaie de se reconnecter automatiquement.
- **Un mechanisme de callback permet d'observer l'état de la connexion ('open', 'close').**
- **Le navigateur transmet les message au serveur à l'aide du 'Chunked transfer encoding'.**
- Aucune affirmation correcte

Question 3

Vous souhaitez implémenter une stratégie de type Long-Polling pour communiquer avec un serveur (`www.example.com`). Parmis les programmes suivant, lequel implémente cette stratégie?

- `var conn = new WebSocket("http://www.example.com");`
- `var conn = new EventStream("http://www.example.com");`
- `setInterval(function () { fetch("http://www.example.com").then(doSomething) }, 1000);`
- **`function get() {fetch("http://www.example.com").then(() => { doSomething(); get(); })};`**

Question 4

Quel header HTTP permet à un serveur d'initializer un cookie dans le navigateur?

- **`Set-Cookie`**
- `Cookie`
- `Secure-Cookie`
- `HttpOnly`
- `document.cookie`
- Aucune affirmation correcte

✋ Questions ?

Correction

✋ Questions ?

Authentication and Authorization

Authentication and Authorization *

In a web application, **authentication** verifies the credentials (login information). **Authorization** typically verifies the right to view, create, edit, or delete some content.

- **Authentication** asserts that someone is who he claim to be.
- **Authorization** asserts that someone has the right to perform a given action.

* source

Authentication Mechanisms

The Web provides a plethora of authentication method:

- Cookie and Session
- `Authentication` Header
    - HTTP basic
    - HMAC token
    - JWT token
    - Bearer token
- `X-API-Key` Header
- OAuth2
- WebAuthN
- etc.

One size does not fit all!

The choice of an authentication method typically varies depending on:

- The kind of entity you authenticate (Human or Machine)
- The kind of service you provide (API or Webapp)
- The kind of web application you devise (SPA or MPA)
- The needs in terms of security (immediate revokation, time-to-live, etc.)
- etc.

Authorization Mechanisms

In Web applications, authorization mechanisms often rely on the notions of:

- Roles (admin, editor, user)
- Ownership (does this resource, object or attribute belong to that user)

In Web applications, authorization mechanisms are often implemented by hand, which gives a lot of flexibility.

Plugins, such as express-acl or express-rbac, implement popular authorization strategies, such as **access control list (ACL)** or **role based access control (RBAC)**.

✋ Learn more about JSON Web Token (JWT) ?

Learn more about JWT:

https://jwt.io/

Try to answer the following questions:

- Can JWT be used for Authentication?
- Can JWT be used for Authorization?
- Why is JWT often refered to as a scalable method?
- Can a JWT token be easily revoked?
- Would you use JWT tokens for authenticating computers that perform API calls?

✋ Learn more about OAuth2 ?

Learn more about OAuth2 on:

https://auth0.com/docs/api-auth/which-oauth-flow-to-use

Try to answer the following questions:

- What kind of Access Token does OAuth2 use?
- Why does OAuth2 introduced the notion of Flow?
- Can OAuth2 be used to authenticate the users of an MPA?
- Can OAuth2 be used to authenticate the users of an SPA?

✋ Configure an Authentication Middleware in Express

Clone the `example-passport` repository from the `tweb-classroom` organization.

It illustrates how:

- Local authentication can be configured in express with Passport
- Github can be used for authentication (via oauth2)
- An attacker can impersonate users with CSRF
- A website can be protected from CSRF attacks

✋ Questions ?