MTAT.03.229
Enterprise System Integration

Lecture 12: REST Security

Marlon Dumas
University of Tartu

Based on slides by Luciano García-Bañuelos
Security is a critical concern to take into account during the development of enterprise software
- Should be considered during the entire application lifecycle
- Concerns not only firewalls, encryption, etc., but should also be embedded throughout the software architecture

We will focus only on application security, relying on the use of Spring Security provided tools
Security

- Security is a family of non-functional requirements
- We need to determine which specific security requirements are relevant to a given project

- **Internal projects:**
  - Authentication
  - Authorization
  - Confidentiality, Data Integrity, Availability

- **External projects:**
  - Message integrity
  - Non-repudiation
  - Legally binding identity assertion and signature
Authentication

• Process of identifying an individual, usually based on a username and password (a.k.a. credentials)

• Authentication object
  ◦ Principal (user name)
  ◦ Authority (role)
How it works?

- If an authenticated request arrives for a restricted resource, server returns HTTP 401: Unauthorized status with WWW-Authenticate header indicating required authentication method (e.g. Basic)

- Client requests username and password. Concatenates them as: username:password

- Encodes them using base64 method for example
  - Note that base64 is reversible, so very insecure
  - Several methods exist to negotiate credentials securely, one of the simpler ones being Digest access authentication

- Encoded credential is sent in an HTTP request header

- Credential is validated on the server side by an authentication provider in the application server
Authorization

• Process of giving individuals access to system objects based on their identity

• Role-based access control
  ◦ Site engineer: Create PHRs, extend rental period, etc.
  ◦ Works engineer: Accept/reject PHRs, etc.
Securing Spring MVC applications
Accessing a restricted resource (1/2)

Web Server

Spring MVC container

Client

GET /pos/123

/\login (form)

View resolver

Controller

Handler mapping

URL

Interc. 1

Dispatcher

View

Auth. Context
Anonymous

Authentication Manager

Access Decision Manager
Accessing a restricted resource (2/2)

**Web Server**

**Spring MVC container**

- Access Decision Manager
- Auth. Context
  - user: luciano
  - auth.: ROLE_SITE_ENG
- Authentication Manager
- Handler mapping
- Controller
- View resolver
- View
- Dispatcher
- Interceptor

**Client**

**POST /login**

{
  "username": "luciano",
  "password": "password"
}
Implementation in Spring Boot

Step-by-step tutorial:


Code of tutorial:

• https://bitbucket.org/lgbanuelos/esi2018-rbac/
Client-Side Authentication (VueJS)