Enterprise System Integration
(MTAT.03.229)

LECTURE 10: SECURITY ASPECTS
OF INTEGRATION - REST SECURITY
(AUTHORISATION AND
AUTHENTICATION)

MOHAMAD GHARIB
UNIVERSITY OF TARTU
1960s - Physical security: security was limited to the access points, where computers were stored.

1970s - Evolution of personal computer and information flow through telephone lines: first hackers appeared, who tries different ways to intercept such information to steal the data.

1980s - Evolution of cyber-crime: hacking and other forms of cyber crimes skyrocketed as hackers find various methods to break into the computer systems benefiting from the absent of strict regulation against hacking/cyber-crimes.

1990s - Hacking becoming a serious concern: As the worldwide web becomes available, people started putting their personal information online; hackers saw this as a potential revenue source. Although firewalls/antivirus programs exists, yet the web was a mostly unsecured.

2000s - Cybercrime becoming a serious issue: hacking was not considered as serious issues but with evolution of hacking methods and the severity of their consequences, governments started taking strong measures against cyber criminals.

2010s - now: Information security nowadays: different protection measures/methods are actively developed but “hackers” are also developing new more sophisticated attack methods to breach such protection measures/methods.
What is Security for our applications?
What is Security for our applications?

Who is asking for the recourse/service?

Is it allowed to “use” the recourse/service?
What is Security for our applications?

Who is asking for the recourse/service?

Authentication

Is it allowed to “use” the recourse/service?

Authorization
Four Core Spring Security Concepts

1. Authentication,
2. Authorization,
3. Granted Authorities/Roles,
1. Authentication

Authentication is the process of verifying that "you are who you say you are"

Authentication factors:

- **Knowledge based authentication** *(something you know)*: e.g., a password, partial password, personal identification number (PIN), security question, etc.

- **Ownership/possession-based authentication** *(something you have)*: e.g., ID card, security token, implanted device, cell phone with a built-in hardware token, software token, etc.

- **Inherence-based authentication** *(something you are)*: e.g., fingerprint, retinal pattern, signature, face, voice, unique bio-electric signals, other biometric identifiers, etc.

**Note.** Research suggests that for an effective authentication, at least two, and preferably all three, factors should be verified.

**Single-factor authentication** is the weakest level of authentication, only a single component from one of the three categories of factors is used to authenticate an individual’s identity.

**Multi-factor authentication** involves two or more authentication factors. Clearly, two-factor authentication is a special case of multi-factor authentication.
2. Authorization

Authorization is the process of verifying that
"you are permitted to do what you are trying to do".

In CS, Authorization is the function/process of specifying access rights/privileges to resources.

Authorization must always follow authentication: users should first prove that their identities are genuine before the system grant them access to the requested resources.
3. Granted Authorities/Roles

A Granted Authority can be seen as an individual privilege. Examples could include (e.g., read, write, execute, etc.).

Granted Authorities restrict/control access in a fine-grained manner.

A Role can be seen as containers of authorities/privileges

A Role restrict/control access in a coarse-grained Granted Authorities.
4. Principal

A **principal**, in computer security, is an entity that can be **authenticated**, and can be assigned rights and privileges/permissions over resources.

A principal can be an individual person, a computer, a service, etc.

A **principal** typically has an associated identifier that allows it to be referenced for identification or assignment of properties and permissions.
How can we secure our Spring Boot App?

1. Configure a Spring Security authentication mechanism,
2. Configure a Spring Security authorization mechanism.
How can we secure our Spring Boot App?

1. Configure a Spring Security authentication mechanism,
2. Configure a Spring Security authorization mechanism.

Specify App users and associate appropriate roles to them, which define their allowed/prevented activities.
How can we secure our Spring Boot App?

1. Configure a Spring Security authentication mechanism,
2. Configure a Spring Security authorization mechanism.

Specify App users and associate appropriate roles to them, which define their allowed/prevented activities.
How can we secure our Spring Boot App?

1. Configure a Spring Security authentication mechanism,

Specify App users and associate appropriate roles to them, which define their allowed/prevented activities.

2. Configure a Spring Security authorization mechanism.
Thank You
for your attention

Mohamad Gharib
mohamad.gharib@ut.ee

unitartu
tartuuniversity