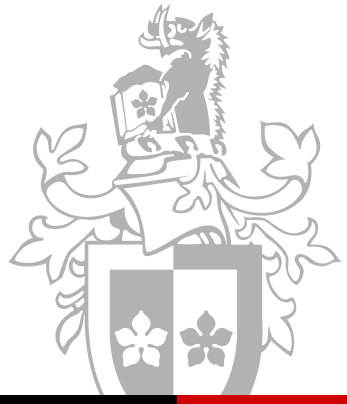
COS30041 Creating Secure and Scalable Software

Lecture 07a Securing Enterprise Applications



SWINBURNE UNIVERSITY OF TECHNOLOGY

Commonwealth of Australia
Copyright Act 1968

Notice for paragraph 135ZXA (a) of the *Copyright Act 1968*

Warning

This material has been reproduced and communicated to you by or on behalf of Swinburne University of Technology under Part VB of the *Copyright Act 1968* (the *Act*).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Learning Objectives

- After studying the lecture material, you will be able to
  - Understand and describe the security issues in Enterprise Application
  - Understand and describe the advantages and disadvantages of using the JavaEE security model
  - Understand the issues involved in implementing Enterprise Security using JavaEE

© Swinburne University of Technology

Pre-requisite

- Basic Java Security concepts

© Swinburne University of Technology

Outline

- Characteristics of Application Security
- Overview of Enterprise Application Security

© Swinburne University of Technology

Roadmap

- **Characteristics of Application Security**
- Overview of Enterprise Application Security

© Swinburne University of Technology

Security Concerns

- Who is accessing the service?
- Are they who they say they are?
- Are they allowed to?
- Can anyone modify the data?
- Can anyone read this private data?

© Swinburne University of Technology

Characteristics of Secure Application

- Authentication – Who you say you are
- Authorization – Do you have access to the resources
- Data integrity – Authorized users can modify the data
- Confidentiality – Only authorized users can view the sensitive data

© Swinburne University of Technology

Other Concerns

- **Auditing** – Capture a tamper-resistant record of security-related events for the purpose of being able to evaluate the effectiveness of security policies and mechanisms
- Non-repudiation – Proving the transaction has happened
- Quality of Service – Provide better service to selected network traffic over various technologies

© Swinburne University of Technology

Roadmap

- Characteristics of Application Security
- **Overview of Enterprise Application Security**

© Swinburne University of Technology

Overview of Enterprise Application Security

- Container provides the security for their software components
- Security structure
  - Authentication requirements (who you say you are)
  - Access control requirements (who can access what)
  - User-based / role-based security
  - Security roles (what role you are with respect to the application)

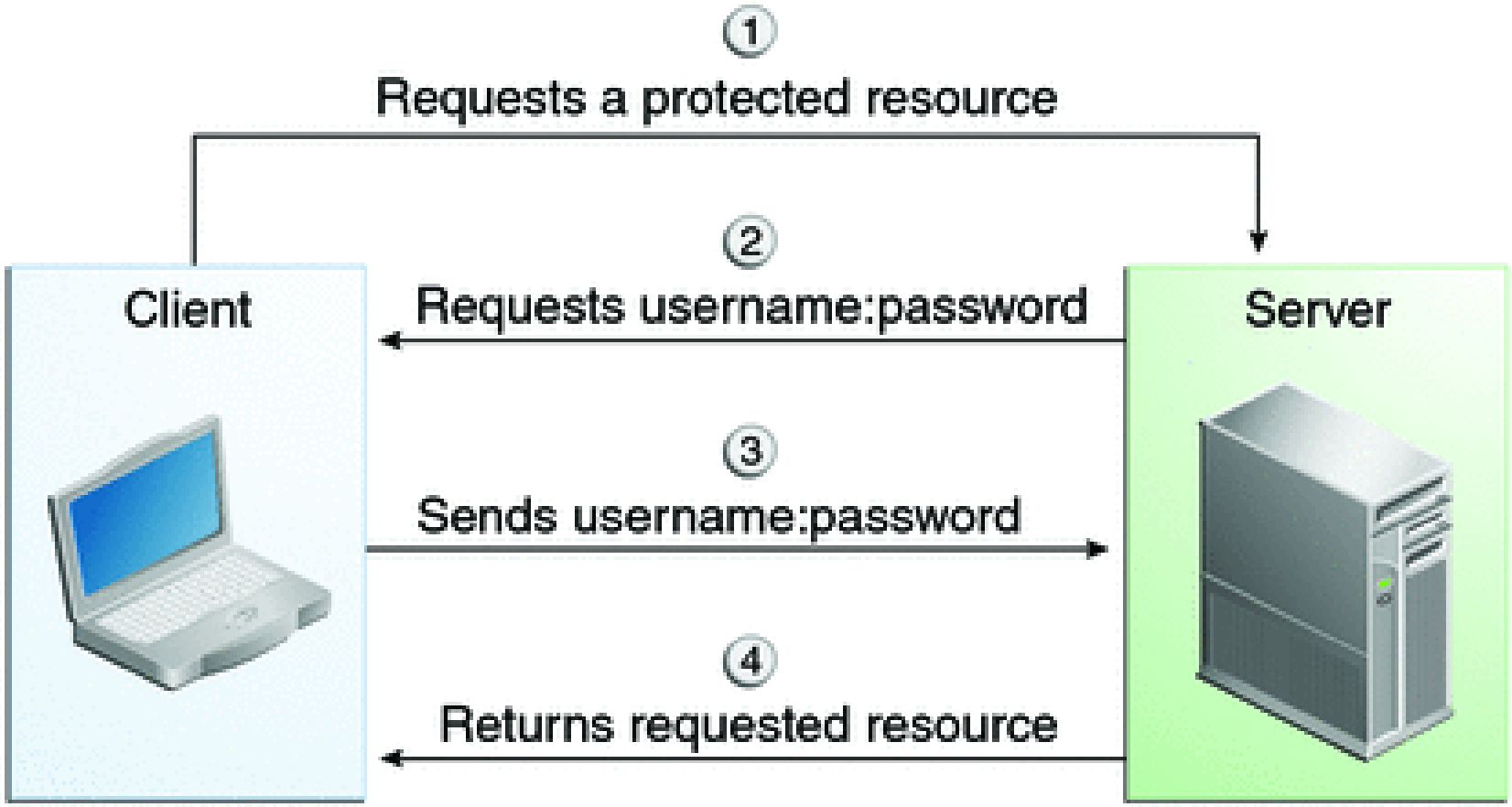
© Swinburne University of Technology

Password Security

- Password Authentication
- Storing passwords in database
  - Plain Text (e.g. “123456”)
  - HASH (e.g. “123456” →hash→ “4982796920”)
  - HASH + SALT (e.g. “123456” → hash → “4982796920” →salt→ “axez49827djfjl96920sljdfsl”)
  - …

© Swinburne University of Technology

Default Security Behaviour in Ent. App.



Adapted from Fig. 40-2 [JEE6T]

© Swinburne University of Technology

Authentication

- Purpose: To identify who you say you are
- Many ways of authentication
  - Entity
    - UserId and Password (simplest, we used this in the subject)
    - …
    - UserId, Password and (Answer to Security Question | Sec Device)
    - …
    - Certificate based authentication (most sophisticated ?)
  - Secure vs Non-secure connection
  - Encrypted vs Plain

Modes of communication

© Swinburne University of Technology

Realm, User and Group [Java EE]

- A Realm = A security policy domain defined for a web or application server
  - Contain a collection of users with or without groups
- A User = An individual that has been defined in the [GlassFish] server
- A Group = A set of authenticated users defined in the [GlassFish] server

© Swinburne University of Technology

Different types of Realm [JavaEE]

- File-realm*
  - Stores user credentials in a file
- Admin-realm
  - A file realm that stores administrators' user credentials
- JDBC-realm [Java EE]
  - Store user credentials in database records
- Certificate
  - Store user credentials in a certificate database

Note: We will only focus on file realm.

© Swinburne University of Technology

Authorization

- Purpose: to verify whether you have access to certain (protected) resources in the application
- Need to define the “access control requirements”
- Approaches
  - Declarative security
  - Programmatic security

© Swinburne University of Technology

Declarative vs Programmatic

Declarative Security

- Specify the application component's security requirements by using either
  - Deployment descriptors, or
  - Annotations
- @DeclareRoles
- @RolesAllowed
- @PermitAll
- See EX-SLSB-SecInfo

Programmatic Security

- Security decisions were programmed in the source code
- Useful when declarative security is not sufficient to express the security model of an application
- SessionContext.getCallerPrincipal()
- SessionContext.isCallerInRole()
- Principal.getName()
- See EX-SLSB-SecInfo

© Swinburne University of Technology

Security Role

- simply, Role
- An abstract name for the permission to access a particular set of resources in an application

© Swinburne University of Technology

Group vs Security Role [Java EE specific]

Group

- Designated for the entire GlassFish Server

Security Role

- Associated only with a specific application in the GlassFish server

© Swinburne University of Technology

Associating Group with Role [Java EE specific]

- GlassFish server has a way to map security roles in an application to the groups
- Default – principal-to-role (group name = role name)
- Principal = an entity that can be authenticated by an authentication protocol in a security service that is deployed in an enterprise

Note*: In case, group names and role names are not the same, you need to use runtime deployment descriptor to specify the mapping

© Swinburne University of Technology

Further Topics (for you to investigate)

- Different authentication mechanisms
- Defining different types of realm
- Configuring Security Using Deployment Descriptors
- Creating your own login module rather than the default provided by the application server

© Swinburne University of Technology

References

- Java EE Tutorial

© Swinburne University of Technology