ITMD 415/515
Advanced Software Development

Week 11 – Web Application Security

Scott Spyrison

Security

- Authentication
- Authorization
  - Declarative (annotations, XML)
  - Programmatic

Java EE Security

- Roles
- Realms
  - Users
  - Groups
- Formerly configured as GF security realm
- JSR375: @DatabaseIdentityStoreDefinition

Java EE Security (web.xml vs JSR375)

- **Declaring Security Roles**

```
<!-- Security roles used by this web application -->
<security-role>
   <role-name>manager</role-name>
</security-role>
<security-role>
   <role-name>employee</role-name>
</security-role>

@DeclareRoles({”manager", ”,”employee"})
```

Java EE Security (web.xml vs JSR375)

- **Security Constraint**
  - Web Resource Collection
    - <web-resource-collection>
      - <url-pattern>
      - <http-method>
  - Authorization Constraint
    - <auth-constraint>
      - <role-name>
  - (User Data Constraint)
- @ServletSecurity(@HttpConstraint(rolesAllowed = ”foo"))

Java EE Security (web.xml)

```
<!-- SECURITY CONSTRAINT #1 -->
<security-constraint>
   <web-resource-collection>
      <web-resource-name>wholesale</web-resource-name>
      <url-pattern>/acme/wholesale/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
      <role-name>PARTNER</role-name>
   </auth-constraint>
   <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
   </user-data-constraint>
</security-constraint>
```

Java EE Security (web.xml)

```
<!-- SECURITY CONSTRAINT #2 -->
<security-constraint>
   <web-resource-collection>
      <web-resource-name>retail</web-resource-name>
      <url-pattern>/acme/retail/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
      <role-name>CLIENT</role-name>
   </auth-constraint>
   <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
   </user-data-constraint>
</security-constraint>
```

Java EE Security (web.xml vs JSR375)

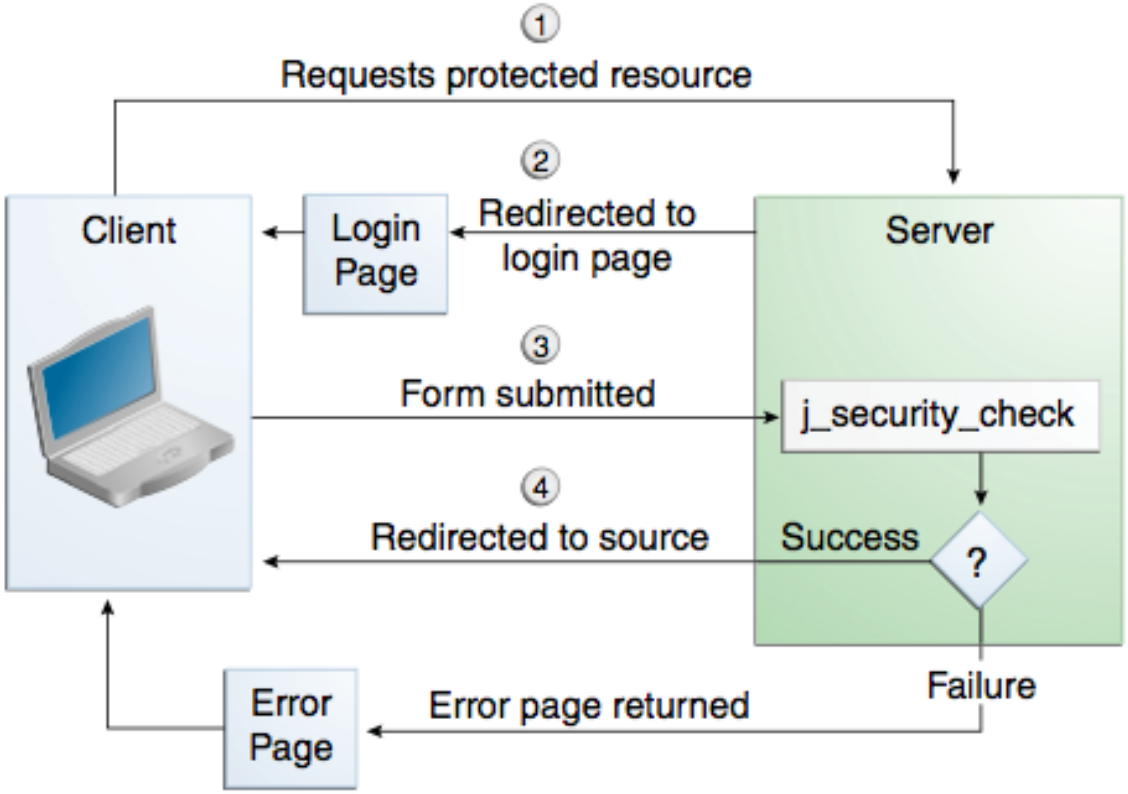
- **Authentication Mechanism**

```
<login-config>
   <auth-method>FORM</auth-method>
   <realm-name>file</realm-name>
   <form-login-config>
      <form-login-page>/login.html</form-login-page>
      <form-error-page>/error.html</form-error-page>
   </form-login-config>
</login-config>
```

```
@CustomFormAuthenticationMechanismDefinition
```

Form-Based Authentication

Figure 45–3 Form-Based Authentication



Role Mapping (payara-web.xml)

- Group to Role Mapping
- Layer of Abstraction for Users and Groups
- In payara-web.xml, **map** users and groups to roles:

```
<security-role-mapping>
    <role-name>BOSS</role-name>
    <principal-name>bob</principal-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>ADMIN</role-name>
    <group-name>it-admins</group-name>
</security-role-mapping>
```

Final Project

- Securing and “webifying” your domain:
  - Enhance Model
  - Stateless EJBs (Db Operations)
  - Singleton EJB (DatabasePopulator) or SQL Load
  - Web Application Security
    - authN
    - authZ
  - Presentation and Functionality using authenticated user in JSF

Important – Java Standards

- Javadoc Analyzer
- Naming Conventions

Sources Used

- The Java EE Tutorial. Retrieved Jan 14, 2019, from https://javaee.github.io/tutorial/
- Juneau, J. (2020). Jakarta EE 8 Recipes. New York, NY: Apress.
- Goncalves, A. (2013). Beginning Java EE 7. New York, NY: Apress.
- Some slides adapted with permission from Marty Hall (www.coreservlets.com – JSP and Servlets)