Java EE Security Explained - examples lab

CTU, Prague 2016

JBoss by Red Hat

Peter Škopek, pskopek@redhat.com, twitter: @pskopek

Nov 4, 2016

Abstract

This lecture will guide you through various aspects of security in Java Enterprise Edition Applications. It will start with plain JAAS and continue with Java EE security concepts and explanation of their usage in your application. Next comes JAAS and its usage in WildFly 10. Then we will finish with login modules in WildFly 10.

Agenda

1 **Introduction**

2 **Examples**

- JAAS
- JEE web container examples

Section 1

Introduction

redhat.

Project

- Project location: `https://github.com/qa/pv243-a4m36jee-2016-security-seminar`
- Each task has its own branch: security-00, security-01 ...
  - initial status for task is in branch with previous number
- download: `http://download.jboss.org/wildfly/10.0.0.Final/wildfly-10.0.0.Final.zip`

redhat.

WildFly Setup

- Copy content of webapp01/src/main/resources/config-samples/security-domain.xml to wildfly/standalone/configuration/standalone.xml at proper section inside security subsystem. It will add security domain "test" to your wildfly configuration. (Don't forget to restart the server).
- Check if your deployment contains WEB-INF/jboss-web.xml file with following content:

```
<?xml version="1.0"?>
<jboss-web>
    <security-domain>test</security-domain>
</jboss-web>
```

Section 2

Examples

redhat.

Task 1: Plain JAAS Example

Start branch: "security-00" Solution branch: "security-01".

1. Explore all parts of jaas-example project (directory: jaas-example/)
2. Execute example using: mvn exec:java
3. Modify "Sample" JAAS configuration which will include sample.module.CardLoginModule with following characteristics:
   - Try to use "Card" authentication if it fails use provided SampleLoginModule.
   - "Card" authentication can fail and SampleLoginModule has to be enough to authenticate user.
   - Use provided sample.module.CardLoginModule class

Hint: change login context configuration file to include "Card" login module.

redhat.

Task 2: Secure access to servlet using annotations

Start branch: "security-01" Solution branch: "security-02".

- Secure SecuredServlet using annotations and security domain "test" configured with UsersRoles login module.
- Only user with role "gooduser" can have access to it using all HTTP methods (verbs).
- The new domain has to use users.properties and roles.properties located at WEB-INF/classes directory of webapp01.war.
- Hint: modify manually standalone/config/standalone.xml to add desired domain.

redhat.

Task 3: Secure static content of the web application

Start branch: “security-03-init” Solution branch: “security-03”.

- Secure static content of the web-application at /static/secured/. All pages there must be readable for “superuser” only.
- All pages located at /static/ should be readable by any authenticated user.
- SecuredServlet from previous task has to stay secured as it was.
- Hint: For details see Java Servlet 3.1 specification.

redhat.

Task 4: Override security annotations using deployment descriptor

Start branch: “security-03” Solution branch: “security-04”.

- Modify security constraint attached to the SecuredServlet so that only members of “superuser” group can run it.
- Act as application assembler, therefore you are not allowed to change code of SecuredServlet.java.
- Hint: the hit is already show at title of this task. You have to define servlet in web.xml.

redhat.

Task 5: Identity propagation

Start branch: "security-05-init" | Solution branch: "security-05".

- Application has added application logic layer in form of EJB called TestBean.
- Modify security settings using annotations to allow users in roles "gooduser" and "superuser" to run SecuredServlet.
- Set security constraints on each method TestBean method to allow users in following roles to call them:
  - echo → all users
  - goodUserEcho → "gooduser" members
  - superUserEcho → "superuser" members
- Hint: Do not forget to add SecurityDomain to TestBean.

redhat.

Task 6: Programmatic Security

Start branch: "security-05" Solution branch: "security-06".

- At the beginning of doGet SecuredServlet method display following information:
  - remote user name
  - user principal
  - information if the user is in "superuser" role
- Create new method in TestBean which displays following information:
  - user principal
  - information if the user is in "gooduser" role
- Call the method at the end of SecuredServlet doGet method.
- Hint: Use @Resource annotation and HttpServlerRequest.