Java EE Security Explained - examples lab

MUNI, Brno 2015

JBoss by Red Hat

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

Apr 27, 2015

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 JEE security concepts and explanation of their usage in your application. Next comes JAAS and its usage in WildFly 8 and JBoss EAP 6. Then we will finish with login modules in WildFly 8 and JBoss EAP 6.

Agenda

Section 1
Introduction

Project

- Project location:
  https://github.com/qa/pv243-2015-security-seminar
- Each task has its own branch: security-01, security-02 ...
  - some branches contain init and complete suffix
- download: `http://download.jboss.org/wildfly/8.2.0.Final/wildfly-8.2.0.Final.zip`

redhat.

Section 2
Examples

Task 1: Plain JAAS Example

Start with security-00 branch.

1. Explore all parts of jaas-example project (directory: jaas-example/)
2. Run example under JBDS (use -Djava.security.auth.login.config==sample_jaas.config)
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.

Task 1: Plain JAAS Example - Maven Configuration

To run example using maven exec plugin change pom.xml.

```
<configuration>
    <mainClass>sample.SampleAcn</mainClass>
    <killAfter>-1</killAfter>
    <systemProperties>
        <systemProperty>
            <key>java.security.auth.login.config</key>
            <value>=sample_jaas.config</value>
        </systemProperty>
    </systemProperties>
</configuration>
```

execute it using: mvn exec:java

Task 2: Secure access to servlet using annotations

Start with security-00 branch. Solution is in security-02 branch.

- Secure SecuredServlet using annotations (programmatic way) using 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 web-application.war.
- Hint: modify manually standalone/config/standalone.xml to add desired domain.

Task 3: Secure static content of the web application

Start with security-03-init branch. Solution is in security-03-complete branch.

- 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.

Task 4: Override security annotations using deployment descriptor

Start with security-03-complete branch. Solution is in security-04 branch.

- 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.

Task 5: Identity propagation

Start with security-05-init branch. Solution is in security-05-complete branch.

- 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 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.

Task 6: Programmatic Security

Start with security-05-complete branch. Solution is in security-06 branch.

- 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.