JEE Security Structure Part 1

Ari Ayvazyan

29.09.2014

Contents

Chapter 1

JEE Security Structure Part 1

1.1 Introduction to Security Architecture

Most web applications have a few things in common:
They need to figure out who is the user that is using the application and what is he allowed to do and see.
According to Oracle, there are two ways to implement such functionality.[4]//

1. Programmatic
2. Declarative (this includes Annotations and XML-Files)

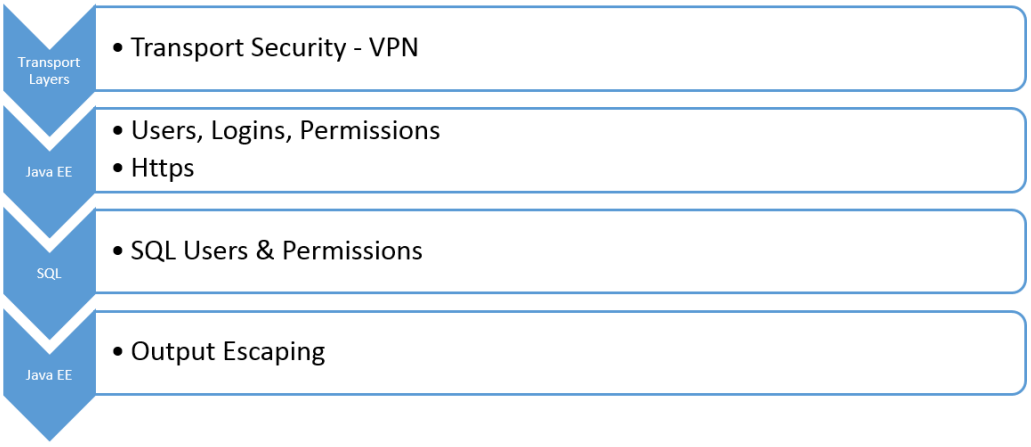

Figure 1.1: Security Layers in a common JEE application

1.2 Authentication

Authentication
Who are you?
Identification

1.3 Authorization

Authorization
What are you allowed to do?
Assignment of Permissions to a Authenticated User

1.4 Deployment Descriptors

Describes how the Application should be Deployed.
Defines Security Constraints

- Protected Information
- Probably SSL
- Specify which user may access them

Deployment Descriptors are XML-Files
Usually located in /WEB-INF/

- web.xml
- Vendor-specific.xml (E.g. Glassfish: glassfish-web.xml)

web.xml

Protected Resources
Security Roles
Authentication methods

(vendor-specific).xml

User – Role mapping
Group – Role mapping

Vendor specific settings

1.5 Principals

A Principal is a identity that can be authenticated.
E.g. a Unique user name

1.6 Credential

A Credential is defined as information that is used to authenticate a Principal.
E.g. a Password

1.7 Groups

Groups and Principals can be mapped to Roles.
Groups are defined in vendor-specific.xml

1.8 Roles

Permissions are granted to Roles.
Roles are defined in the web.xml file

1.9 Realms

aka Security policy domain
Provides information about principals, their Groups and their credentials
May be a Database, File structure, connection...

In other words:
It contains user information
E.g. Username, Password & Permissions

1.10 Implementation sample

https://github.com/aayvazyan-tgm/JavaEESecurityExample

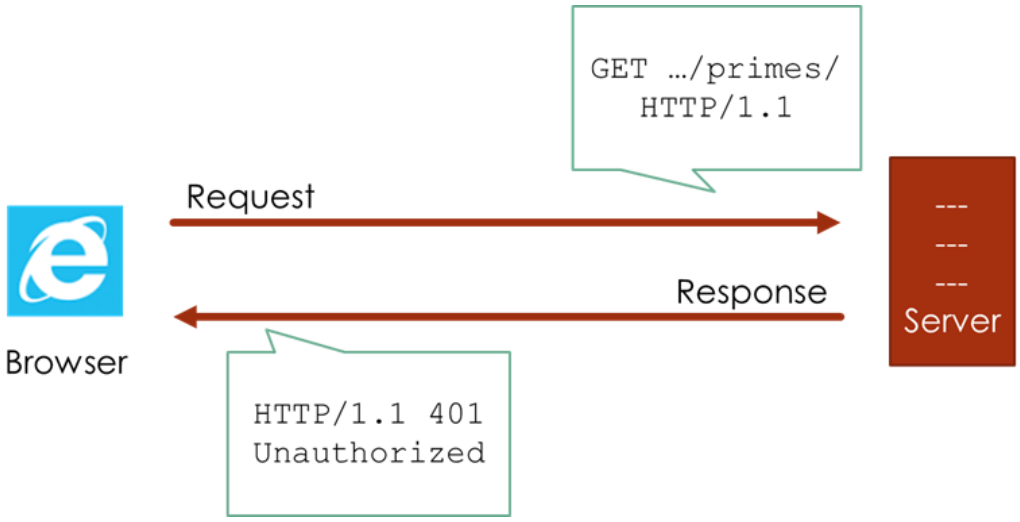

Figure 1.2: The user tries to access a resource without authentication

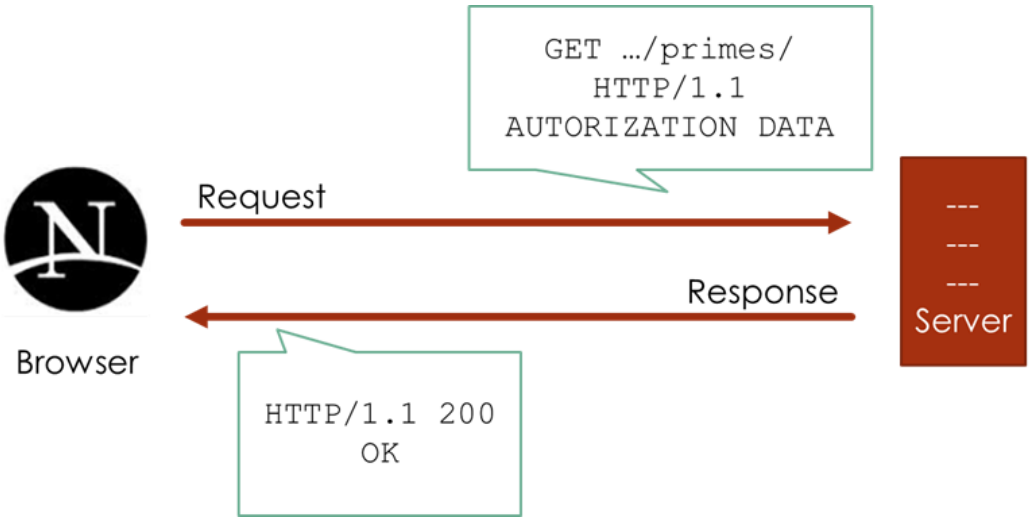

Figure 1.3: The user sends authentication data with his request

1.11 Frameworks

1.11.1 Shiro

Offers: Authentication, Authorization, Cryptography
Simple to use

Advantages/Disadvantages
Implementation Sample

1.11.2 Spring

Offers: Authentication, Authorization, Cryptography
Very structured

Advantages/Disadvantages

1.11.3 JAAS - Java Authentication and Authorization Service

Offers: Authentication, Authorization, Cryptography
Included in Java SE since Java 1.4 (javax.security.auth)

Advantages/Disadvantages

1.12 Output escaping

Escape user input to prevent injections.

Escape the output to add a extra layer of security.
Use a Framework to do so!

1.13 Whats to come in Part 2 (Adrian)

- Working with Digital Certificates

- Securing Application Clients
- Security with Enterprise Beans
- Further Framework Information

Bibliography

[1] JavaOne 2014: The Anatomy of a Secure Web Application Using Java,
Shawn McKinney & John Field, September 29, 2014
San Francisco

[2] Java Security: Sicherheitslücken identifizieren und vermeiden,
Marc Schönefeld, 1. edition 2011
Publisher: Hüthig Jehle Rehm GmbH, Heidelberg.
ISBN/ISSN 978-3-8266-9105-8

[3] Enterprise Java Security: Building Secure J2EE Applications,
Marco Pistoia, Nataraj Nagaratnam, Larry Koved, Anthony Nadalin,
1. edition 2004
Publisher: Addison-Wesley Professional.
ISBN/ISSN: ISBN 0-321-11889-8

[4] Official JavaEE Documentation, Oracle,
29.09.2014 http://docs.oracle.com/javaee/7/tutorial/partsecurity.htm#GIJRP
Java EE 6,
Dirk Weil, 1. edition 2012
Publisher: entwickler.press
ISBN 978-3-86802-077-9

[5] Java EE 6 Cookbook for Securing, Tuning, and Extending Enterprise Applications,
Mick Knutson,
1. edition June 2012
Publisher: Addison-Wesley Professional.
ISBN/ISSN: ISBN 9781849683166