Professional JSP
Security and Web Applications

Security in web applications – not just for authentications, but also for authorization – access to resources.
Security should be addressed at all the different tiers of a web application: presentation, application and data tier.
Securing boxes/code is only part of it, security data over protocols (soap, http, rmi), also needs to be considered, especially with the increase of DoS attacks in recent years. For more info, see http://www.owasp.org/, http://online.securityfocus.com.
For the scope of this chapter/book, we will only cover security and how it affects JSP-based web applications. Specifics include SSL, authentication mechanisms (form, basic, mutual), certificates, protecting secure resources, placing JSPs under WEB-INF, only allowing certain roles to access certain actions (Struts), using EJBs for further security in the middle tier, JAAS, +other alternatives.
Benefits of declarative (container-managed) security (i.e. protecting resources with web.xml and ejb-jar.xml) vs. programmatic security.
Relate story of writing custom programmatic security to LDAP and then using container managed and how much easier it was.
What's new in Servlet spec 2.4.

Authentication

Authentication in web applications.

- Container-managed authentication – it's there to make it easier.
- The level of security is up to the developer (i.e. SSL, SecurID-type smart cards).
- Don't try to re-invent the wheel when it's already been done for you. JAAS is often used by vendors under-the-covers.

Authentication options

HTTP Basic, Form-based, Client-certificate
Most popular is Form-based
Tips and Tricks with Form-based
Configuring a Realm in Tomcat and options

Using SSL

Obtaining a certification from a certificate authority
How to create your own on Tomcat
Configuring your web.xml to require SSL.
How SSL affects performance.

Form-based Authentication Tips and Tricks

Welcome file that redirects to protected resource. Go over configuration in web.xml. Putting login form on a welcome page.
Using same JSP for login and login error (note only tested on Tomcat, does not work on some containers, but there are workarounds (i.e. cookies on iPlanet). **Code example**
Using SSL for login only – show how using Xdoclet to build a servlet that will switch to SSL. **Code example**

Using a Filter to obtain a user's information after authenticating.
Add a switch in your applications code, and make it configurable in web.xml (and your build process), so you can start with a database-based user store, and switch to an LDAP-based user store at a later date. **Code example**
Discuss development process: start with file-based user/pass, move to JDBC, then move to LDAP.

Password encryption

Needed if storing user's passwords in plain text, i.e. Tomcat's tomcat-user.xml or in a JDBC Realm.
User stores, LDAP vs. Database and password encryption – give MySQL example of password encryption and discuss portability options.
Discuss encryption types: Base64, SHA, MD2, and MD5
Programmatic encryption vs. declarative encryption in Tomcat.
Code example.
Provide password lookup facility, or at least plan for one. If using password encyption, a hint is best since password cannot be decrypted. Another common thing is to re-create password – what a pain for the user though!

Servlet 2.4 Security changes

HttpSession.logout + Others?
Deployment description from DTD to XSD, how does that affect security constraints and login-config?

Other Considerations

SecurityFilter: http://securityfilter.sourceforge.net/
SSLExt for Struts to switch to SSL for certain actions.
Single Sign-On
Remembering passwords with cookies – handy, but might not be as needed with modern browsers since they have built-in password memory features.

Authorization

Overview

Revisit difference between authentication and authorization and provide own thoughts.
Discuss how the two are related.

Protecting pages and URLs

Discuss difference between protecting a page and protecting a URL.
JSPs placed under WEB-INF – only available via servlets. Provide snippet from Ant build script. **Code example**
Programmatic protection – using a Filter or Tag Library to protect.
Discuss maintenance problems with this approach.

Available technologies to make authorization easier

Using Struts role attribute in struts-config to protect actions. Make sure and create friendly error message page. **Code example**

EJBs and method level declarative security. Recommend EJBs for webapps that require more security and transaction support. **Show how easy it is to create EJBs with Xdoclet and possibly Middlgen.**

Using Tiles and Struts Menu to show different pages/links for different roles.

Authorization and Databases

Don't forget about web-to-data tier issues. For instance, username/password for database access. You can really lock down security by creating 1-1 mapping for web users and database users. Consider performance and maintenance nightmare.

Logging is important if you have highly secure system, both at the web tier and database level. One method is to add create_user_id, create_date, update_user_id and update_date methods to each table.

Summary

Best practices (in my experience) for security in webapps.

Use container-managed security for ease of setup. Beware of app server implementation issues. Try on Tomcat (RI), then try on your production app server.

Using a good app server can eliminate many headaches – you get to write code, not write workarounds.

Test and test often using Cactus/HttpUnit. Point to **code samples** in sample app.

Open Issues

How has the JSP 2.0 and Servlet 2.4 spec improved security?