Tomcat 3.x und realm Login

Fabian Heusser

2002-06-05

Contents

Abstract

Dieses Dokument beschreibt kurz wie eine Zugriffssteuerung auf bestimmte Verzeichnisse mit Tomcat 3.x realisiert wird. Unter Zugriffssteuerung verstehe ich, dass ein Verzeichniss oder darin enthaltene Dateien nur nach einem Login einsehbar sind, genau wie das bei Apache mit dem .htaccess erreicht wird.

1 Voraussetzung

Es wird Tomcat der Version 3.x genau der Version 3.2 vorausgesetzt. Mit diesem ist das beschriebene Vorgehen getestet worden. Es kann mit anderen Versionen funktionieren, ist jedoch nicht getestet worden.

2 Unterschiede Apache - Tomcat

Zwischen Apache und Tomcat gibt es mehrere unterschiede, Angefangen von der Konfiguration die ganz unterschiedlich ist bis zu den Möglichkeiten. Tomcat ist dabei von den Möglichkeiten dem Apache überlegen. Angefangen damit, dass Benutzer beim Tomcat in sogenannten "Rollen" eingeteilt werden, auf dennen schlussendlich die Zugriffsteuerung funkioniert. Auch gibt es Module für Tomcat die sehr einfach die Benutzer aus einer relationalen Datenbank holen. Desweiteren unterstützt Tomcat auch ein breiteres Spektrum an Login-Shemen wie "BASIC", "DIGEST" oder "SSL". Weiter lässt sich beim Tomcat ein Form basiertes Login ausführen, durch welchen das Look-and-Feel des Login Prozesses angepasst werden kann.

3 Konfiguration

Die nachfolgende Konfigurationshilfe beschreibt das Einrichten der einfachsten Variante, Unverschlüsselte Authentifizierung ("BASIC") mit einer XML Benutzer Datenbank.

3.1 User-Passwort-Rollen konfiguration

In der Datei `conf\tomcat-user.xml` kann man alle Benutzer eintragen. dies sieht so etwa aus

```
<tomcat-users>
  <user name="seminar"   password="" roles="Seminar" />
  <user name="kunde"   password="" roles="Customer" />
  <user name="admin"   password="" roles="Admin" />
  <user name="god"   password="i am" roles="Seminar, Customer, Admin" />
</tomcat-users>
```

Es werden Benutzer und Passwörter und Rollen definiert. Rollen sind ähnlich den UNIX Gruppen. Die Zugriffsteuerung kann man nur auf Rollen-Ebene Vergeben. Aussgenommen davon ist eine Zugriffsteuerung im Servlet von welchem aus der Benutzername abgefragt werden kann. (siehe unten)

3.2 Realm Konfiguration

In `conf\server.xml` muss versichtert werden dass auch die Benutzer aus dem XML - File gelesen werden, dazu muss

```
<Realm className="org.apache.catalina.realm.MemoryRealm" />
```

drin stehen. Hier kämen auch Verbindungen zu Datenbanken zu stehen, mehr dazu in der Tomcat Documentation im "Realm Configuration HOW-TO".

3.3 Spezifische Zugriffs Konfiguration

Sind obige Schritte getan kann man weiter das `web.xml`, vorzugsweise das des Verzeichniss das man schützen will, anpassen.
Folgenden Zeilen schützen alles im assozierten Verzeichnis. Nur Benutzer mit den Rollen 'Admin', 'Seminar' und 'Customer' haben zutritt.

```
<security-constraint>
  <display-name>seminarmanager</display-name>
  <web-resource-collection>
    <web-resource-name>seminarmanager servlet</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
    <http-method>DELETE</http-method>
    <http-method>HEAD</http-method>
    <http-method>OPTIONS</http-method>
    <http-method>TRACE</http-method>
```

```
    </web-resource-collection>
    <auth-constraint>
      <role-name>Admin</role-name>
      <role-name>Seminar</role-name>
      <role-name>Customer</role-name>
    </auth-constraint>
  </security-constraint>


  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>SeminarManager</realm-name>
  </login-config>


  <security-role>
    <description>Administrator</description>
    <role-name>Admin</role-name>
  </security-role>

  <security-role>
    <description>Kunden Sachbearbeiter</description>
    <role-name>Customer</role-name>
  </security-role>

  <security-role>
    <description>Seminar Sachbearbeiter</description>
    <role-name>Seminar</role-name>
  </security-role>
```

Im '`<security-constraint>`' Tag wird festgelegt für welchen bereich '`\*`' für welche Request Art ('GET', 'POST', 'PUT' ...) die Beschrenkung gilt. Desweiteren die zugelassenen Rollen.
Im '`<login-config>`' Tag wird die Authentifikations Methode festgelegt. '`<realm-name>`' ist die Beschreibung die welche im Login Fenster erscheint. Hier müsste der Hebel angesetzt werden wenn man eine Formular basierten Login Mechanismus wünscht. Dann wäre die '`<auth-method>`' 'FORM' .
Zu guter letzt werden noch die zu verwendenden Rollen definiert.

4 Benutzer, Rollen und Servlets

Wird nun in einem geschützten verzeichnis ein Servlet aufgerufen, sind gewisse Parameter durch das Servlet verfügbar. Dies sind folgende Methoden der Klasse HttpServletRequest.

getAuthType() Liefert die Authentifikations-Methode zurück

getRemoteUser() Liefert den Loginname des Eingeloggten Users zurück.

isUserInRole(java.lang.String role) Liefert true zurück fals der User sich in der übergebenen Rolle befindet

getUserPrincipal() Liefert ein java.security.Principal Objekt mit dem Loginname des Authentifizierten Benutzers zurück.