

Seguridad en Tomcat

Antonio Espín Herranz

Conceptos

- **Autenticación:**
 - Es el proceso de verificar si alguien es quien dice ser. Típicamente mediante login y password.
 - La clave puede estar en muchos formatos
 - Una vez autenticado, al usuario se le asigna un rol.
- **Autorización:**
 - Una vez autenticado el componente, se procede por un mecanismo de autorización, basado en roles
 - En J2EE las políticas se aplican sobre roles y no sobre usuarios.
- **Realms (reinos):**
 - También conocidos como security policy domains or security domains.
 - Son ámbitos sobre los cuales se definen entidades de seguridad similares.

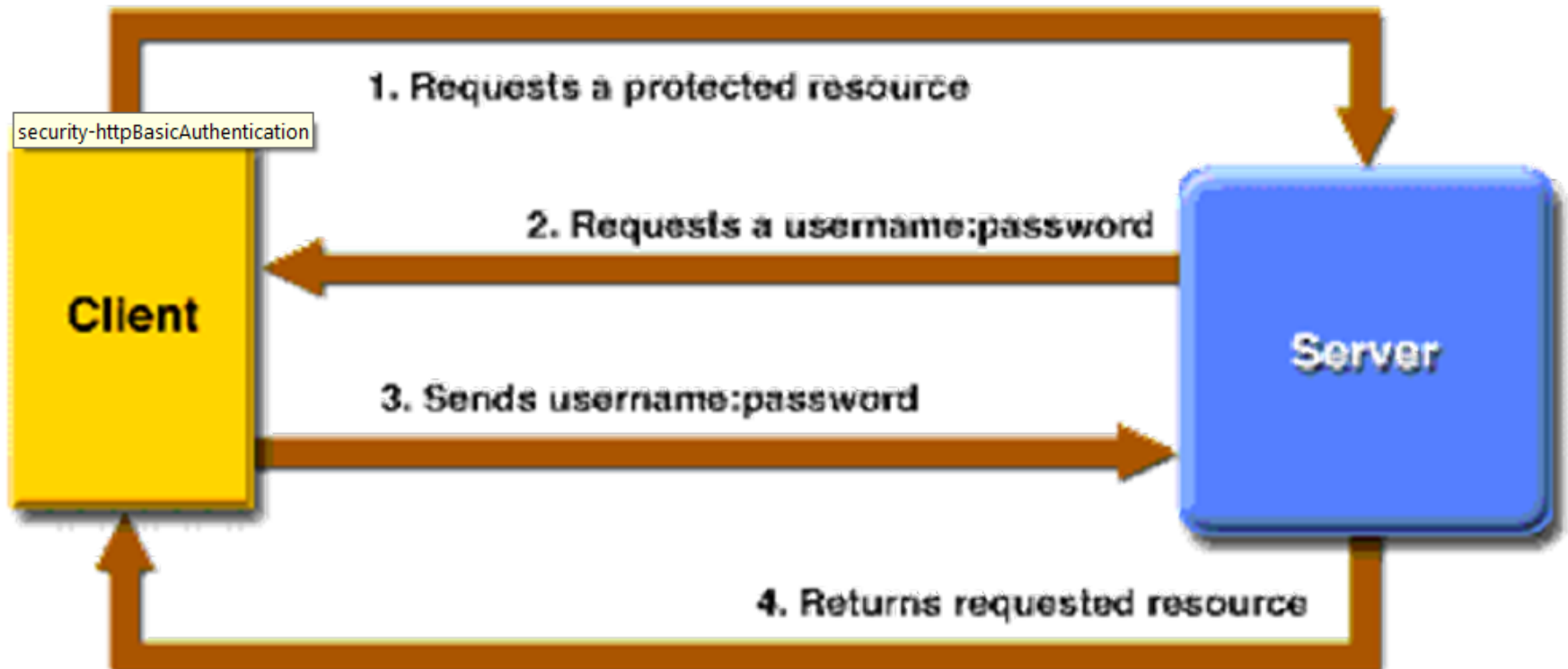
Conceptos

- **Usuario:**
 - Identidad de un usuario. Ejemplo: Pablo Sanchez.
- **Grupo:**
 - Entidad colectiva de un grupo de usuarios
Ejemplo: Profesor de la asignatura de J2EE.
- **Rol:**
 - Desde el punto de vista lógico de la aplicación es un perfil Ejemplo: profesor.

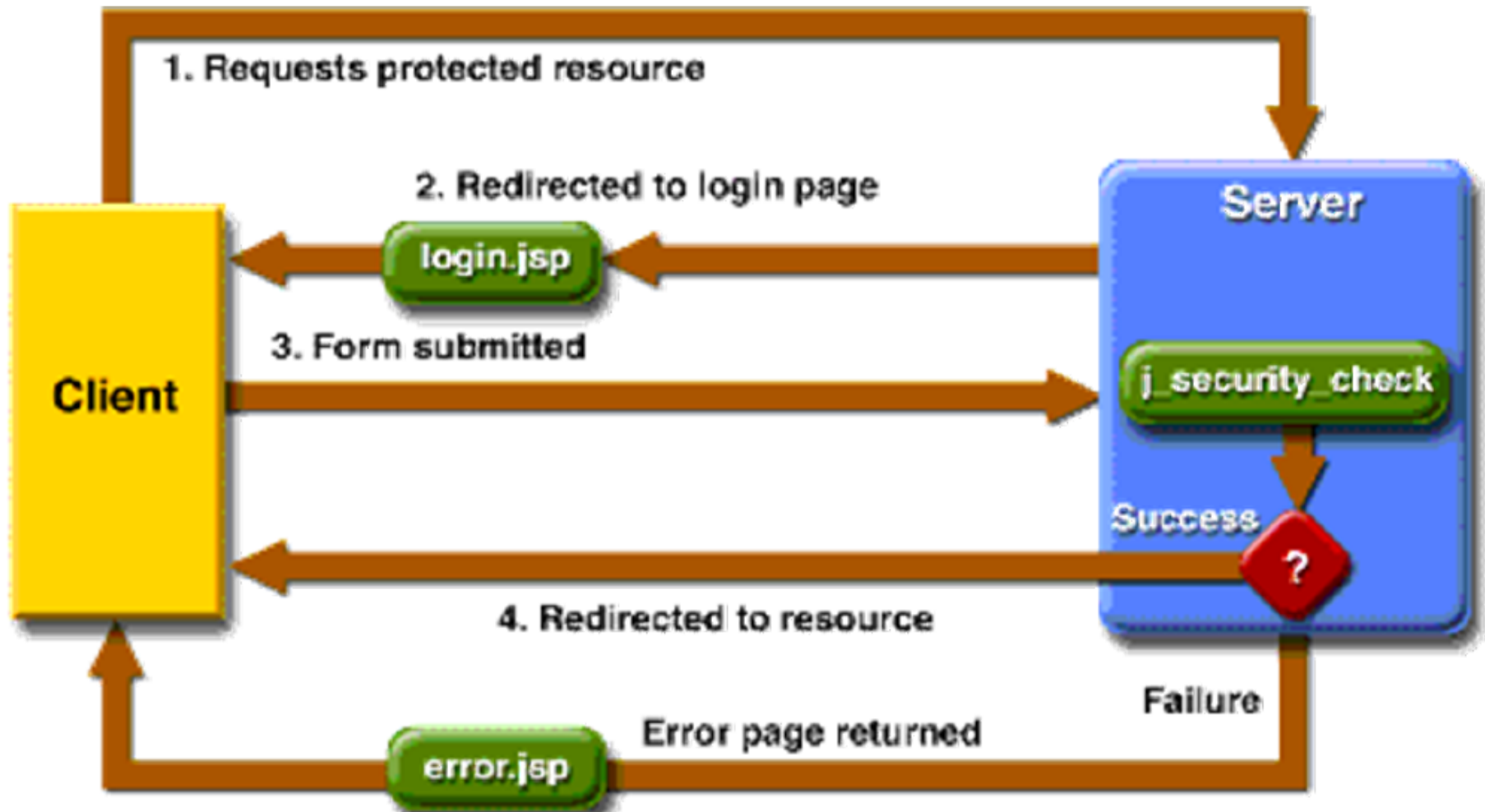
Tipos de Autenticación

- Los clientes (webapps clients) se autentican antes de poder entrar.
- En aplicaciones web se soportan varios tipos de autenticaciones:
 - **Basic, Form, Digest, Client Certificate.**
- Mientras que para **clientes de consola** se suelen preferir otros mecanismos **JAAS** (Java Authorization and Authentication System).

HTTP BASIC



Basada en Form



Digest

- Es bastante parecida a la http basic pero más segura. HTTP-basic utiliza 64-base encoding.
- Pero sin embargo no es muy utilizada, aunque J2EE la mantiene por universalidad.

Basada en Certificado

- Utiliza HTTP sobre **SSL** (Secure Server Layer), por lo que es bastante segura
- SSL ofrece tanto encriptación, como autenticación del servidor, como integridad de mensaje como identificación opcional del cliente(opcional) para conexiones TCP/IP.
- En caso de hacerse la autenticación del cliente se utiliza un certificado X.509.
- Antes de usar esta configuración se ha de habilitar dicho tipo de seguridad en el servidor.

Autenticación

- Se establece en el descriptor: **web.xml**
- Por un lado definimos recursos a los que se asignan roles.
- Por otro lado en Tomcat diremos que usuarios tenemos y a los roles a los que pertenecen.
- En este caso los usuarios se dan de alta en un fichero de configuración en Tomcat.
- Es mejor realizar la validación mediante una BD.

Autenticación de Usuarios: BASIC

- Se realiza de forma declarativa (configuración mediante XML) en el contenedor.
- Será el contenedor el que nos libera de las comprobaciones necesarias en cuanto a seguridad.

Claúsula security-constraint

<security-constraint>

Definimos una colección con los recursos que queremos aplicar seguridad.

Nombre del recurso y la url con la que se mapea.

<web-resource-collection>

<web-resource-name>MiServlet</web-resource-name>

<url-pattern>/ServletProtegido</url-pattern>

</web-resource-collection>

<auth-constraint>

<role-name>curso_java</role-name>

</auth-constraint>

</security-constraint>

Clausula login-config

<login-config>

 <auth-method>BASIC</auth-method>

 <realm-name>CURSO DE JAVA</realm-name>

</login-config>

- Indicamos el método de autenticación, en este caso básica.
- El nombre que aparecerá en el cuadro de diálogo emergente.

En Tomcat

- Tendremos que añadir una entrada por cada usuario en el fichero:

- %CATALINA_HOME%\conf\tomcat-users.xml

```
<role rolename="curso_java"/>
```

```
<user name="login" password="contraseña"  
      roles="curso_java" />
```

- Indicamos el login y pass de cada usuario y a los roles que pertenece.
 - Separados por , si queremos poner alguno mas.

En el Servlet

- Podemos capturar el método de autenticación que estamos utilizando y el nombre del usuario.
 - `String tipo = request.getAuthType();`
 - `Principal principal = request.getUserPrincipal();`
 - `String nombre = principal.getName();`
 - Esta clase se encuentra `java.security`
- Pero no tenemos que programar nada, en cuanto a comprobaciones.


Autenticación de Usuarios: FORM

- En este caso podemos diseñar nosotros nuestro propio formulario para pedir las contraseñas al usuario.
- Tenemos que cambiar el web.xml:

```
<login-config>  
  <auth-method>FORM</auth-method>  
  <form-login-config>  
    <form-login-page>/login.jsp</form-login-page>  
    <form-error-page>/error.jsp</form-error-page>  
  </form-login-config>  
</login-config>
```

Autenticación de Usuarios: FORM

En el formulario se deben indicar estos nombres para el usuario, el password y donde se va a enviar el form.



```
<body>  
  <form action="j_security_check">  
    Enter your user name: <input type="text" name="j_username"/>  
  
    Enter your password: <input type="text" name="j_password"/>  
  </form>
```


Autenticación con la BD

- La autenticación en Tomcat también se puede realizar a partir de la BD.
- Manteniendo en la BD tablas para almacenar los usuarios y los roles.
- También habría que modificar el fichero de configuración conf/server.xml.

Autenticación con la BD: PASOS

- Tenemos quedar una serie de pasos para configurar el acceso mediante BD.
 - Copiar el driver (fichero jar) de mysql en la carpeta lib de Tomcat.
 - Crear en la BD las tablas para almacenar usuarios, roles.
 - `CREATE TABLE usuarios (USERNAME varchar(30) NOT NULL default "", PASSWORD varchar(30) NOT NULL default "", PRIMARY KEY (USERNAME));`
 - `INSERT INTO usuarios (USERNAME, PASSWORD) VALUES ('admin', '12345');`
 - `CREATE TABLE usuarios_rols (ROLE_NAME varchar(30) NOT NULL, USERNAME varchar(30) NOT NULL);`
 - `INSERT INTO usuarios_rols (USERNAME, ROLE_NAME) VALUES ('admin', 'administrador');`

Autenticación con la BD: **PASOS**

- Modificamos el fichero **conf/server.xml** de Tomcat. Añadiendo la información del Realm, básicamente los parámetros de conexión a la BD, nombres de tablas y campos que almacenan los usuarios y los roles.

Autenticación con la BD: PASOS

```
<Realm
  debug="9"
  className="org.apache.catalina.realm.JDBCRealm"
  driverName="org.gjt.mm.mysql.Driver"
  connectionURL="jdbc:mysql://localhost/usuarios"
  connectionName="root"
  connectionPassword="antonio"
  userTable="vista_user"
  userNameCol="username"
  userCredCol="password"
  userRoleTable="vista_user"
  roleNameCol="role" />
```

Autenticación con la BD: PASOS

- En el web.xml, por ejemplo:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>All Page</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>Administrador</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>

  <realm-name>Tomcat-Realm</realm-name>
</login-config>

<security-role>
  <role-name>administrador</role-name>
</security-role>
```

Indicamos:

- Para una colección de recursos que role necesitamos para poder acceder.
- /* Para todos los recursos es necesario estar autenticado y tener asignado el role de Administrador.
- Tipo de Acceso.
- Declaración del role.