# Servidores Web

## Sesión 6: Seguridad y autentificación en JBoss. LDAP

# Puntos a tratar

- Autentificación en JBoss
  - Dominios
  - Autentificación contra una BD
- LDAP
- Seguridad del servidor

# Dominios de seguridad

- Equivalentes a los “realm” de Tomcat.
  - También se conocen como “login modules”
- Al igual que en Tomcat, hay implementaciones alternativas. Por ejemplo:
  - `UsersRolesLoginModule`: Almacena los usuarios en un par de ficheros .properties (uno para login/password, otro para login/rol)
  - `DatabaseServerLoginModule`: almacena los usuarios en una base de datos accesible mediante JDBC

# Autentificar contra una BD

- La información de logins/passwords/roles puede estar en la BD como queramos, siempre que podamos extraerla mediante SQL en el formato esperado por JBoss
- A JBoss debemos darle 2 consultas SQL:
  - Autentificación: debe devolver un registro con un único campo, que será el password
  - Autorización: debe devolver un conjunto de registros cada uno con dos campos
    - El primero debe ser el rol del usuario
    - El segundo debe ser el valor literal “Roles” (para el caso más común, en realidad depende de  nuestra config. JAAS)

# Ejemplo de configuración

- Añadir al archivo **conf/login-config.xml**
- Suponemos 2 tablas
  - **USUARIOS**: cada registro tiene campos *login*,*password*
  - **USU_ROLES**: cada registro relaciona un *login* con un *rol*

```
<application-policy name="seguridadBD">
 <authentication>
   <login-module
      code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
    <module-option name="dsJndiName">java:/DefaultDS</module-option>
    <module-option name="principalsQuery">
      select password from USUARIOS where login=?
    </module-option>
    <module-option name="rolesQuery">
      select rol, 'Roles' from USU_ROLES where login=?
    </module-option>
   </login-module>
 </authentication>
</application-policy>
```

# Ejemplo de configuración (II)

- **flag=”required”** indica que el intento de autentificación no puede fallar
  - Podríamos definir mecanismos alternativos si falla este (si no se encuentra en la BD, que se tome de un fichero, por ejemplo)
- Obsérvese que necesitamos un DataSource para conectar con la BD
- Obsérvese también que le damos un nombre arbitrario al dominio: **seguridadBD**

```
<application-policy name="seguridadBD">
 <authentication>
   <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule"
                 flag="required">
     <module-option name="dsJndiName">java:/DefaultDS</module-option>
     <module-option name="principalsQuery">
       select password from USUARIOS where login=?
     </module-option>
     <module-option name="rolesQuery">
       select rol, 'Roles' from USU_ROLES where login=?
     </module-option>
   </login-module>
 </authentication>
</application-policy>
```

# Enlazar el *dominio* con la aplicación

- ¿Cómo sabe JBoss en qué aplicación/es usar el *login module*?
  - A través de un descriptor de despliegue propio de JBoss (**WEB-INF/jboss-web.xml**)
  - En él se usa el nombre JNDI del dominio (el dado al "application policy" precedido de **java:/jaas/**

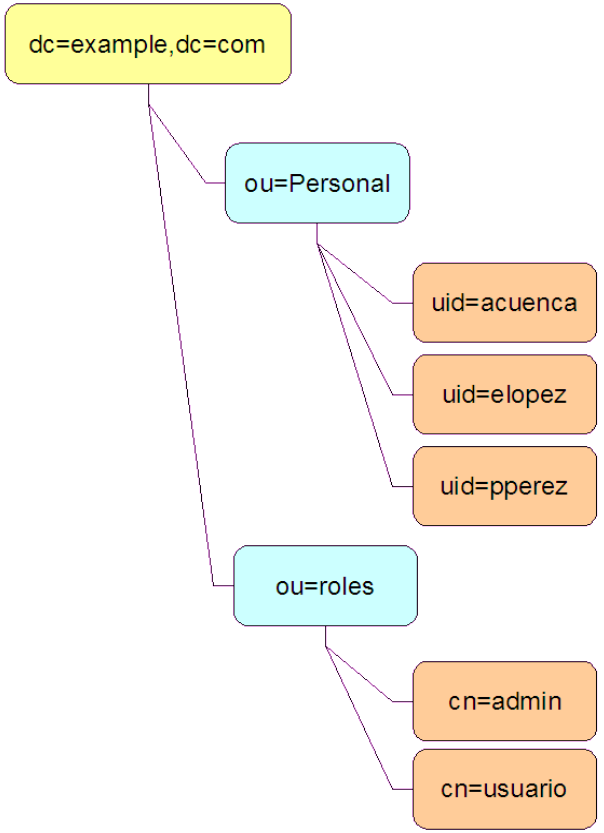
```
<!DOCTYPE jboss-web PUBLIC
        "-//JBoss//DTD Web Application 2.4//EN"
        "http://www.jboss.org/j2ee/dtd/jboss-web_4_0.dtd">
<jboss-web>
   <security-domain>java:/jaas/seguridadBD</security-domain>
</jboss-web>
```

# Definir un dominio basado en BD: resumen

1) Definir un DataSource para conectar con la BD
   - Recordemos que era en **deploy/XXX-ds.xml**
- En **conf/login-config.xml** definir un <application-policy>
  - Asignarle un nombre arbitrario
  - Poner las *queries* SQL para sacar password y roles
- Crear el descriptor de despliegue **WEB-INF/jboss-web.xml**

# LDAP

- Resumen LDAP
  - Cada nodo es un objeto que hereda de 1 o más clases
    - LDAP ofrece gran número de clases predefinidas
  - En cada nodo hay un campo que lo distingue en ese nivel (**rdn** o *relatively distinguished name* (!)) (se muestra en la figura)
  - El identif. único de un nodo (**dn** o *distinguished name*) se forma concatenando los *rdn* desde el nodo hasta la raíz




```
dn: uid=acuenca,ou=Personal,
    dc=example,dc=com
objectClass: person
objectClass: uidObject
objectClass: top
cn: Alfonso
sn: Cuenca Richardson
uid: acuenca
userpassword:: Y3VlbmNhYQ==
```

# LDAP y Jboss (I)

- Hay un *login module* especial para LDAP

```
<application-policy name="seguridadLDAP">
   <authentication>
      <login-module code="org.jboss.security.auth.spi.LdapLoginModule"
                    flag="required">
         <module-option name="java.naming.factory.initial">
            com.sun.jndi.ldap.LdapCtxFactory
         </module-option>
         <!-- URL del servidor. Ponemos el puerto que usa ApacheDS -->
         <module-option name="java.naming.provider.url">
            ldap://localhost:10389/
         </module-option>
         <module-option name="java.naming.security.authentication">
            simple
         </module-option>
```

*(continúa en la traspa siguiente...)*

# LDAP y Jboss (II)

```
        <!--Cómo se forma el identificador completo (DN) del usuario -->
        <module-option name="principalDNPrefix">uid= </module-option>
        <module-option name="principalDNSuffix">
           ,ou=Personal,dc=example,dc=com
        </module-option>
        <!-- Dónde están los roles -->
        <module-option name="rolesCtxDN">
           ou=roles,dc=example,dc=com
        </module-option>
        <!-- cómo se identifica un usuario concreto en un rol -->
        <module-option name="uidAttributeID">member</module-option>
        <module-option name="matchOnUserDN">true</module-option>
        <!-- dónde se define el nombre del rol -->
        <module-option name="roleAttributeID">cn</module-option>
        <module-option name="roleAttributeIsDN">false</module-option>
      </login-module>
   </authentication>
</application-policy>
```

# Combinar módulos

- ***Password stacking***: se puede autentificar usando un módulo (p.ej. LDAP) y autorizar con otro (p.ej. BD)
  - **useFirstPass**: usar la primera autentificación que tenga éxito

```
<application-policy name="LDAPconBD">
  <authentication>
     <login-module code="org.jboss.security.auth.spi.LdapLoginModule"
                   flag="required">
       ...
       <module-option name="password-stacking">useFirstPass</module-option>

     </login-module>
     <login-module
        code="org.jboss.security.auth.spi.DatabaseServerLoginModule"
        flag="required">
         ...
       <module-option name="password-stacking">useFirstPass</module-option>
     </login-module>
  </authentication>
</application-policy>
```