

Tema 2 : Aspectos de seguridad en los servidores de aplicaciones



Ciclo Superior DAW

Asignatura: Despliegue de aplicaciones web

Curso 20/21



Introducción

- En este capítulo veremos aspectos como:
 - Configuración de la autenticación.
 - Configuración de control de acceso.
 - Empleo de Valves para el control de acceso mediante IP.
 - Comunicaciones aseguradas entre cliente y servidor.



Autenticación y autorización

- La **autenticación** es el **proceso para verificar que alguien es quien realmente dice ser**
- La **autorización**, por su parte, es el **proceso por el que se le permite hacer a alguien autenticado lo que solicita.**



Actividad 1

Poned un ejemplo de autenticación y de autorización en la vida real



Autenticación y autorización

- Para llevar a cabo estas tareas en Tomcat, se emplean los **Realms**.
- Un **Realm** es un **archivo, base de datos o servicio de directorio que contiene una colección de usuarios y contraseñas y roles.**



Autenticación y autorización

- Se emplea la clase **org.apache.catalina.Realm**
- Existen varios tipos de Realms, aunque nos centraremos en MemoryRealm, en el que se accede a la información almacenada en un fichero (normalmente tomcat-users.xml).



Autenticación y autorización

- La localización del Realm depende del ámbito del que queremos que controle la autenticación y autorización:
 - Engine
 - Host
 - Context



Autenticación y autorización

- **Engine:**
 - Todo el servidor.
 - Debe estar definida en **<RUTA_TOMCAT>/conf/server.xml** dentro del elemento <Engine>.



Autenticación y autorización

- **Host:**

- Todo el servidor virtual.
- Debe estar definida en el archivo de configuración **<RUTA_TOMCAT>/conf/server.xml** dentro del elemento <Host> que se corresponda con el sitio virtual a lo que queremos que se aplique.



Autenticación y autorización

- **Context:**
 - Una aplicación concreta.
 - Debe estar definida en el fichero **context.xml** (en WebContent/META-INF) de la aplicación correspondiente, dentro del elemento <Context>.



Autenticación y autorización

- La información de usuarios y roles está almacenada en un fichero que se carga en memoria al iniciar Tomcat
- Este fichero, por defecto, es **tomcat-users.xml**.



Actividad 2

*Encuentra los ficheros tomcat-users.xml en las máquinas servidor Windows y
Linux*



Configuración

- A continuación, veremos cómo **definir el fichero de configuración de usuarios y roles** (por defecto, tomcat-users.xml)
- En este fichero definiremos:
 - **Roles**
 - **Usuarios** pertenecientes a un rol



Configuración. Roles

- Para definir un rol:

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



Configuración. Usuarios

- Para definir un usuario:

```
<user username="nombreUsuario" password="contraseña" roles="rol1 rol2..."/>
```



Actividad 3

Crea un rol y un usuario en cada uno de los servidores



Configuración. Realm

- Veremos cómo **configurar un Realm** en el ámbito que se considere más adecuado
- Debemos configurarlo en el fichero **context.xml**:

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



Configuración. Protección del recurso

- Vemos cómo proteger el recurso, es decir, **configurar los permisos necesarios**
- Debemos configurarlo en el descriptor de despliegue **web.xml** de la aplicación



Configuración. Protección del recurso

```
<security-constraint>
<web-resource-collection>
    <web-resource-name>ServletPrivado</web-resource-name>
    <url-pattern>/*</url-pattern>
</web-resource-collection>
<auth-constraint>
    <role-name>usuariosAutenticados</role-name>
</auth-constraint>
</security-constraint>
```



Configuración. Tipo de autenticación

- Veremos cómo **configurar el tipo de autenticación**
- Debemos configurarlo en el fichero **web.xml**:

```
<login-config>  
    <auth-method>BASIC</auth-method>  
    <realm-name>Acceso a la aplicación</realm-name>  
</login-config>
```



Actividad 4

- *Accede a los ficheros context.xml y web.xml en los dos servidores*
 - *Configura un Realm*
 - *Protege el recurso*
- *Configura el tipo de autenticación*



Configuración de SSL

- El protocolo HTTP envía todos los datos, incluidos los usuarios y contraseñas, en descubierto.
- **Para asegurar la conexión entre cliente y servidor, podemos cifrar dicha comunicación empleando el protocolo SSL**



Configuración de SSL

- Echaremos un vistazo a este vídeo donde un experto nos explica muy bien las diferencias entre protocolo HTTP y HTTPS:

<https://www.youtube.com/watch?v=AqWEjTECQqM>



Configuración de SSL

- En primer lugar, será necesario **crear un almacén de claves con un certificado SSL:**
- La sintaxis es:

```
keytool -genkey -alias tomcat -keyalg RSA -keystore RUTA_ALMACEN_CLAVES
```




Configuración de SSL

- Por ejemplo, en Windows sería:

```
keytool.exe -genkey -alias tomcat -keyalg RSA -keystore C:\claves
```

- Y en Linux:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keystore /var/lib/tomcat8/claves
```



Configuración de SSL

- Durante la generación del certificado se tienen que introducir dos claves:
 - La del almacén de claves
 - La de las claves asociadas al alias creado.
- Las claves pueden ser las mismas.



Actividad 5

En esta actividad habilitaremos SSL en los dos servidores



Configuración de SSL

- Una vez hecho esto, hay que crear un conector SSL en Tomcat:
 - Editamos el fichero **<RUTA_TOMCAT>/conf/server.xml**
 - Añadimos las siguientes líneas dentro del elemento raíz <Server>:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
maxThreads="150" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="/var/lib/tomcat8/almacen_claves"  
keystorePass="tomcat"  
keyAlias="tomcat" keyPass="tomcat"/>
```



Actividad 6

En esta actividad crearemos un conector SSL en Tomcat en la máquina virtual



Configuración de SSL

- Por último, vamos a **configurar la aplicación para que sólo acepte conexiones HTTPS**
- Para esto, hay que añadir en el descriptor de despliegue (web.xml) los siguientes elementos, dentro del <security-constraint>



Configuración de SSL

```
<user-data-constraint>  
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
</user-data-constraint>
```



Configuración de SSL

- Como se ha modificado el descriptor de despliegue hay que reiniciar el servidor y desplegar la aplicación.
- Las conexiones HTTP se redireccionarán a HTTPS



Actividad 7

En esta actividad configuraremos la aplicación para que sólo acepte conexiones

HTTPS



Actividad 8

¿Qué sucede al acceder a la dirección <http://localhost:8080/Aplicacion> ?



Valves

- Las Valves **permiten interceptar las peticiones al servidor de aplicaciones y procesarlas previamente**
- Para esto se asocia la clase Java de la Valve a:
 - Engine (todo el servidor)
 - Host (un sitio virtual concreto dentro del servidor)
 - Context (una aplicación).



Valves

- Se emplean, por ejemplo, para realizar tareas de:
 - Filtrado por IP.
 - Generación de logs.
 - Compresión de datos.
 - Identificación de la localización de las peticiones (idioma, país, etc.) para responder en consecuencia



Valves

- Su sintaxis es:

```
<Valve className="Clase_que_define_la_valve" parámetros/>
```



Valves

- Su localización depende del ámbito de peticiones que queramos que intercepte:
 - **Engine.** Todas las peticiones del servidor. **En el fichero server.xml, dentro de la etiqueta <ENGINE>**
 - **Host.** Todas las peticiones al servidor virtual. **En el fichero server.xml , dentro de la etiqueta <HOST>**
 - **Context.** Todas las peticiones a una aplicación concreta. **En el fichero context.xml**



Ejemplo práctico

- A continuación veremos un ejemplo práctico del uso de la Valve **RemoteAddrValve**
- Su sintaxis es:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"  
allow="direcciones_a_permitir" deny="direcciones_a_denegar"/>
```



Ejemplo práctico

- En el siguiente ejemplo definimos tres Valves:
 - Permitimos el acceso a la máquina con IP 192.168.0.4 y se lo denegamos a 192.168.0.6
 - Denegamos el acceso a las máquinas con IP 192.168.0.6 y 192.168.10.7
 - Denegamos el acceso a las máquinas con IP entre 192.168.0.1 y 192.168.0.254

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="192.168.0.4" deny="192.168.0.6"/>  
<Valve className="org.apache.catalina.valves.RemoteAddrValve" deny="192.168.0.6,192.168.0.7"/>  
<Valve className="org.apache.catalina.valves.RemoteAddrValve" deny="192.168.0.*"/>
```




Actividad 9

*Configurar una Valve en el servidor Windows para permitir el acceso desde la
máquina cliente*

Tema 2 : Aspectos de seguridad en los servidores de aplicaciones



Ciclo Superior DAW

Asignatura: Despliegue de aplicaciones web

Curso 20/21