Informe de configuración https

1. Introducción

No es objetivo de este documento explicar como funciona el protocolo HTTPS, sino simplemente explicar como llevar a cabo su configuración en Tomcat. Por este motivo no vamos a entrar en mucho detalle sobre los elementos necesarios, pero si los veremos por encima.

2. Resumen de la configuración

Junto a este documento se adjunta una carpeta llamada "cfg", que contiene dos archivos encapsulados en varias carpetas. El contenido de la carpeta cfg está preparado para ser copiado en el directorio C:\ de la máquina de Pre-Producción. Al copiar ambas carpetas se creará un nuevo directorio llamado "Certificados" que contiene el archivo sample.jks y se remplazará un archivo de configuración de Tomcat en la ruta C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf. A continuación se da más información de ambos ficheros, y los cambios que habría que hacer si se eligiera otro nombre, directorio o clave para el certificado.

3. Certificado

Para establecer una comunicación segura necesitamos implementar SSL/TLS, y con tal fin necesitamos un certificado que nos identifique cuando se establezca dicha conexión. De esta manera cuando el cliente se conecte y vaya a enviar al servidor información sensible, como sus datos personales, el cliente puede asegurarse de que el receptor es, efectivamente, el servidor al que quiere enviar dicha información.

Hay dos formas de obtener un certificado, obteniendolo de un tercero de confianza llamado *Certificate Authorities*, el cual estará firmado tanto por el tercero como por el propietario, o creándolo uno mismo. Mientras que para la fase de producción deberíamos obtener el certificado de un tercero, de forma que los navegadores de los visitantes validen el certificado sin mostrar un aviso de seguridad, para la fase de desarrollo nos vale con un certificado generado por nosotros mismos.

Tomcat admite tres tipos de certificados: JKS, PKCS11 o PKCS12. El JDK que tenemos instalado en ambas configuraciones incorpora una herramienta para crear certificados del tipo JKS (Java KeyStore). Para crearlo basta con abrir un terminal, cambiar al directorio donde se encuentra dicha herramienta y ejecutarla con ciertos parámetros. Una vez hecho esto nos preguntará por una

contraseña para el archivo JKS, cierta información de la empresa que se refleja en el certificado, y por último otra contraseña para el certificado concreto que acabamos de crear. Esta última contraseña es por defecto la misma que la primera. La clave que usaremos nosotros será samplePassword:

```
>cd "C:\Program Files\Java\jdk1.7.0_13\bin"
>keytool -genkey -alias tomcat -keyalg RSA -keystore C:\Certificados\sample.jks
```

Antes de crear el archivo JKS debe existir el directorio, en nuestro caso C:\Certificados.

4. Configuración de Tomcat

Ahora que tenemos el certificado tenemos que configurar Tomcat para que acepte conexiones seguras. Para ello debemos abrir el archivo server.xml que se encuentra en el directorio C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf. Si nos fijamos, ya existen unas lineas comentadas para dar soporte a una conexión https. Sin embargo, por un lado dejar la variable protocol="HTTP/1.1" hace que tomcat elija automáticamente la implementación SSL, algo que según la documentación se debe evitar, y por otro lado, le falta la información sobre el certificado que se va a usar. Eligiendo una de las posibles implementaciones que se dan en la documentación, añadimos el siguiente código al archivo:

```
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
port="443" maxThreads="200" scheme="https" secure="true"
SSLEnabled="true" keystoreFile="C:\Certificados\sample.jks"
keystorePass="samplePassword" clientAuth="false" sslProtocol="TLS"/>
```

Nótese que tanto el directorio indicado en keystoreFile como la contraseña usada en keystorePass son las que le hemos dado a nuestro certificado. Por otro lado, hemos cambiado el puerto por defecto 8433 al 433, que es el puerto por defecto de https, de forma que en el navegador podamos entrar a través de "https://www.acme.com", ya que tenemos esa dirección puenteada en nuestro archivo de DNS.

Con estas configuraciones, y una vez desplegada nuestra aplicación como de costumbre, ya podemos entrar en ella através del protocolo https.

5. Otras consideraciones

Ya que lo único que hemos modificado o añadido son archivos que no pertenecen al proyecto o la plantilla, la plantilla expuesta en este item, 1.10, y su proyecto correspondiente, apenas sufre modificaciones con respecto a los de otros items. Lo único que se ha modificado en estos son los términos y condiciones, para exponer que se usa conexión segura.

Por otro lado teníamos la posibilidad de quitar la entrada por http, pero hemos decidido dejarla por simplicidad. Otra opción algo más compleja hubiera sido redirigirla siempre a la https, y lo óptimo hubiera sido que se usara https en ciertas páginas, como cada vez que se establece una conexión con otro usuario, y http en páginas que no requirieran una conexión segura. No hemos explorado estas opciones por falta de tiempo, ya que hemos decidido invertir el tiempo que

nos ha sobrado al terminar el proyecto en mejorar este en vez de mejorar este apartado.

6. Bibliografía

http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html