Aplicación segura

Jeisson Geovanny Sanchez Ramos

Octubre 2020

1 Introducción

En el presente trabajo vamos a generar y utilizar certificados como un mecanismo para la protección de la comunicación del servicio. También vamos a tener algunas otras consideraciones en cuenta como por ejemplo proteger el acceso a vistas, para restringir la comunicación entre el cliente y los respectivos servicios tanto desde back como front.

2 Vistaso al problema

El objetivo del presente trabajo se encuentra en generar certificados mediante los cuales vamos a generar una comunicación segura https.

La arquitectura desde una perspectiva general debe contar con 2 nodos (Máquinas virtuales), las cuales van a proveer 2 servicios una el login y otra un servicio cualquiera el cual se decidió seria de notas, el cliente se comunica con el servicio de login mediante https, y este servicio de notas.

3 Administración de certificados

Primero que todo tuvimos que generar las llaves para nuestros 2 servicios y esto es relativamente sensillo, luego generamos los TrustStore, Pero aquí tuvimos que analizar a quien le correspondía conocer algún Truststore, en base a la descripción de la arquitectura propuesta por el profesor, se determinó que el nodo de autenticación debería conocer el TrustStore del servicio de notas, ya que, sin este, el servicio de autenticación no podría consumir las notas.

4 Diseño del Backend

Para el servicio de notas se crearon los métodos GET y post para poder publicar y consultar estos registros, para la interacción con los datos se creó una entidad nota, y se convirtió mediante la librería GSON, para el almacenamiento de registros se utilizó una colección concurrente de Java.

Para el servicio de autenticación, se quemó un usuario para realizar el login, aunque esto es una mala práctica, por la simplesa del problema se determino era lo más optimo, se creó una capa de servicios que sería quien permitiría la comunicación mediante https.

Como buena práctica, y por facilidad para la migración a los servicios de aws se decidió que la url del servicio de notas dentro del login fuera suministrado como una variable de entorno.

Aunque vale aclarar que en nuestra aplicación se quemó una contraseña cifrada insegura (test), es importante al momento de desarrollar solicitar un patrón fuerte, ya que estos tienen menor probabilidad de estar en un diccionario de claves y asi lograr garantizar que un algoritmo de fuerza bruta no pueda vulnerar nuestras claves, asi que una de las consideraciones más importantes aunque salen del alcance del presente trabajo, es concientizar a los usuarios de las buenas prácticas en seguridad, ya que nuestro sistema puede verse vulnerado inclusive por que las mismas personas revelen información confidencial (Ingeniería social).

5 Servicios Backend

- Login: Recibe un objeto JSON con el usuario y la contraseña, valida las credenciales y actualiza los valores de la sesión correspondiente.
- IsLogged: Da un booleano que corresponde si existe una sesión activa en el servidor o no.
- logout: Cambia la variable en la sesión del servidor para cerrarla.
- Registrar nota: Agrega una nota siempre que el usuario este autenticado.
- Consultar notas: Muestra las notas siempre que el usuario este autenticado.

6 Diseño Frontend

En el diseño del front end se decidió utilizar JavaScript para establecer la comunicación entre el cliente y el servidor, asi que desde javascript restringimos el acceso a la vistas, asi que si alguno se preguntaba por qué en back necesitábamos un servicio isLogged, la respuesta es muy sencilla, este es el mecanismo mediante el cual podemos conocer si el usuario esta autenticado y por lo tanto restringir el acceso a las vistas desde JavaScript, asi cuando tenga la sesión iniciada me muestre la vista de notas, y en caso contrario me muestre el login.

De igual manera en el Backend también se encuentran las restricciones de acceso a la información.

7 Consideración en el back

El servicio que es completamente seguro es el login, ya que este administra si se encuentra autenticado el usuario, asi que nuestro servicio externo aunque maneja https, es inseguro ya que no requiere de autenticación alguna para acceder la información, en un despliegue a producción, este servicio no se expondría mediante una IP publica, ya que representa un riesgo de seguridad, mientras que el módulo del login, el cual en base la arquitectura presentada es con el cual el cliente se comunica, posee las validaciones necesarias para garantizar la integridad de la información.

8 Puesta en producción

Para desplegar este servicio se decidió realizarlo mediante Docker, aunque también se hubiera podido ejecutar directamente la aplicación sobre la máquina virtual, esta elección se dio a motivo de practicar el uso de Docker.

Como se mencionó anteriormente la aplicación requiere variables de entorno asi que es importante al momento de ejecutarlo no olvidar pasar el puerto porque vamos a exponer el servicio, y host:port por el cual se está ofreciendo nuestro servicio externo de notas.

9 Conclusión

Al momento de desarrollar aplicaciones web, es importante pensar en la manera en que vamos a garantizar la integridad, disponibilidad y confidencialidad de la aplicación, en el presente trabajo abordamos exclusivamente integridad y confidencialidad, ya que por simpleza utilizamos una sesión del lado del servidor, lo cual al ser un estado del servidor no nos permite escalar fácilmente la aplicación, por lo regular este problema se soluciona generando token's que el servidor válida para cualquier petición.

Esta es un enfoque muy sencillo para garantizar que el nodo de autenticación no ofrezca un canal de comunicación seguro y integridad sobre la informacion.