Lección 5 - Mensajería de Última Milla II

Tecnologías del sector financiero

Asier Sampietro Alberdi

Práctica 1: Preparación

Se ha generado un certificado con el siguiente comando:

```
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$ keytool -genkey -alias practicefive -keyalg RSA -validity 365 -keystore practicefive -keypass 123456 -storepass 123456
¿Cuáles son su nombre y su apellido?
  [Unknown]:  Asier Sampietro
¿Cuál es el nombre de su unidad de organización?
  [Unknown]:
¿Cuál es el nombre de su organización?
  [Unknown]:  Umbrella Corporation
¿Cuál es el nombre de su ciudad o localidad?
  [Unknown]:  Raccoon City
¿Cuál es el nombre de su estado o provincia?
  [Unknown]:  Montana
¿Cuál es el código de país de dos letras de la unidad?
  [Unknown]:  US
¿Es correcto CN=Asier Sampietro, OU=Unknown, O=Umbrella Corporation, L=Raccoon City, ST=Montana, C=US?
  [no]:  si
```

Este certificado se ha almacenado localmente en un almacenamiento local llamado “practicefice”, y para poder usarlo se ha exportado el certificado con el siguiente comando:

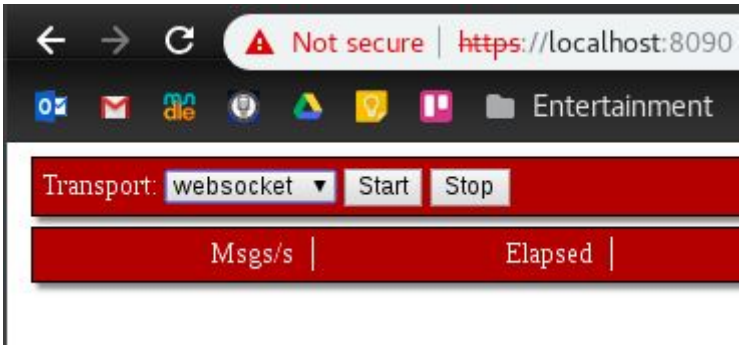
```
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$ keytool -export -alias practicefive -keystore practicefive -rfc -file PracticeFivePublica.cer
Introduzca la contraseña del almacén de claves:
Certificado almacenado en el archivo <PracticeFivePublica.cer>
```

Para poder usarlo, se ha añadido el certificado “PracticeFivePublica.cef” en la raiz del proyecto, y se ha importado al repositorio de certificados del sistema con los siguientes comandos:

```
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$ cp PracticeFivePublica.cer /usr/local/share/ca-certificates/
cp: no se puede crear el fichero regular '/usr/local/share/ca-certificates/PracticeFivePublica.cer': Permiso denegado
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$ sudo !!
sudo cp PracticeFivePublica.cer /usr/local/share/ca-certificates/
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$ sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
Updating Mono key store
/etc/ca-certificates/update.d/mono-keystore: 10: /etc/ca-certificates/update.d/mono-keystore: /usr/bin/cert-sync: not found
Done
done.
sampru@debian:/media/sampru/Datuak/eskola/tec-sec-fin/MasterUC3Practices/Lesson5$
```

Al hacer esto, se ha podido acceder a la aplicación con el certificado creado:



Práctica 2: Medición de mensajes por segundo

1. Websocket

Usando los zócalos web y el protocolo sin SSL, obtenemos los siguientes resultados. A medida que sube el sleepTime disminuye el flujo de mensajes, como se ve en la siguiente tabla:

sleepTime	Mensajes/segundo	Tiempo	Mensajes
0	660	30	19797
1	280	30	8394
10	86	30	2596

Al usar SSL para realizar una conexión segura, los tiempos incrementan hasta cierto punto que observamos una bajada en el rendimiento propio del sacrificio por usar algoritmos más seguros:

sleepTime	Mensajes/segundo	Tiempo	Mensajes
0	319	30	9577
1	207	30	6217
10	85	30	2555

2. Long-polling

Usando el long-polling, se vuelve a repetir el patrón extraño del ejercicio anterior, ambas veces.

sleepTime	Mensajes/segundo	Tiempo	Mensajes
0	41	30	1242
1	55	30	1640
10	49	30	1463

Tanto para el 10 como el 0, son los mismos resultados, y el sleepTime de 1 destaca un poco.
A la hora de ejecutarlo usando una conexión SSL, los tiempos de mensaje aumentan, disminuyendo la frecuencia, pero este también invierte el anterior grafico: cuanto más tiempo de espera tiene mas mensajes manda, lo que confirma la hipótesis de que evita que el proceso se colapse.

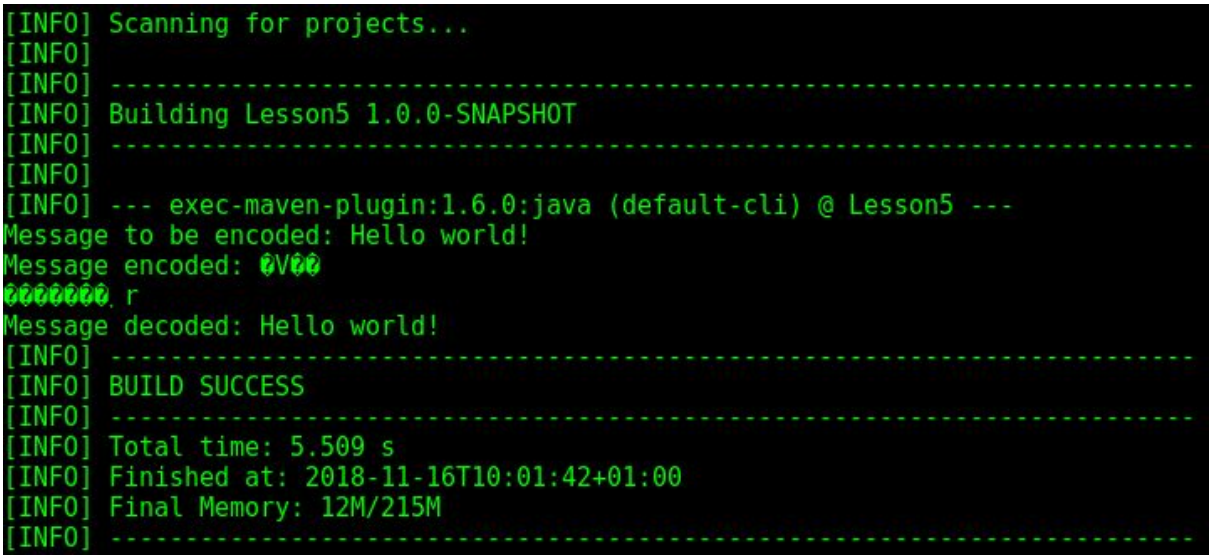
sleepTime	Mensajes/segundo	Tiempo	Mensajes
0	30	30	899
1	36	30	1120
10	39	30	1186

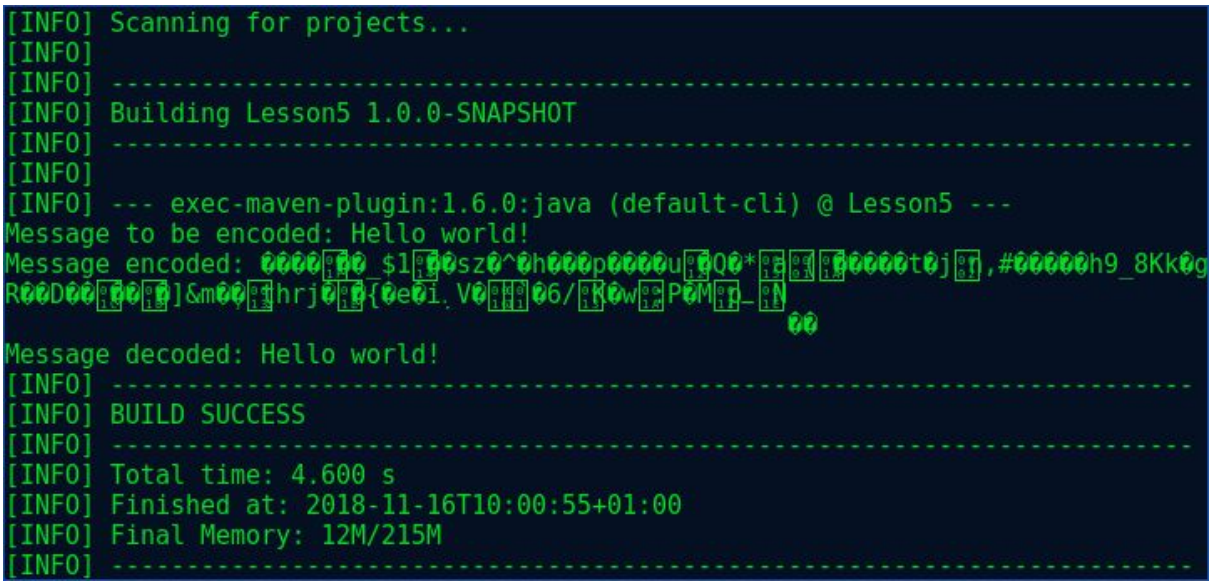
3. Conclusiones

Al hacer la comparativa entre los protocolos HTTP y HTTPS queda claro que una certificación y cifrado de punto a punto afecta al rendimiento de la aplicación. El cifrado suele ser una herramienta muy útil, pero el coste en tiempo puede no compensar en algunos casos, por lo que un análisis previo para determinar la necesidad es necesario.

Práctica 3: Cifrado y descifrado con claves simetricas y asimetricas

Usando las dos clases para cifrado AES y RSA, se obtienen los siguientes outputs:





En ambos se observa que se cifra el texto y se pasa a binario, dejando una cadena de caracteres difícil de interpretar. Después, con el algoritmo de vuelta se consigue el texto de origen, habilitando la lectura de nuevo.