Øving 2 – Nettverksprogrammering

Oppgave 1

Løsningen består av vedlagt *Klient.java*, *Tjener.java* og *Main.java*. Koden har kommentarer som burde forklare hva som skjer.

Oppgave 2

Vi opprettet JavaSSLServer.java og JavaSSLClient.java (Ukjent, 2021). Deretter opprettet vi en jar-fil til hver av disse med følgende steg:

```
olineamundsen@Olines-MBP server % javac ../../src/JavaSSLServer.java -d .
```

Lager .class-fil av java-fila.

```
1    Main-Class: JavaSSLServer
2
```

Lager MANIFEST.MF-fil, og linker main-klassen og legger den i samme mappe som class-fila.

```
olineamundsen@Olines-MBP server % jar cvmf MANIFEST.MF JavaSSLServer.jar *
added manifest
adding: JavaSSLServer.class(in = 1954) (out= 1060)(deflated 45%)
adding: MANIFEST.MF(in = 26) (out= 28)(deflated -7%)
```

Lager jar-fil i samme mappe. Gjør dette for både server og client.

```
olineamundsen@Olines-MBP øving 2 % mkdir mykeystore
olineamundsen@Olines-MBP øving 2 % cd mykeystore
olineamundsen@Olines-MBP mykeystore % /usr/bin/keytool -genkey -alias signFiles
-keystore examplestore -keyalg RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  k
What is the name of your organizational unit?
  [Unknown]:  k
What is the name of your organization?
  [Unknown]:  k
What is the name of your City or Locality?
  [Unknown]:  k
What is the name of your State or Province?
  [Unknown]:  k
What is the two-letter country code for this unit?
  [Unknown]:  k
Is CN=k, OU=k, O=k, L=k, ST=k, C=k correct?
  [no]:  y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) wi
th a validity of 90 days
        for: CN=k, OU=k, O=k, L=k, ST=k, C=k
```

Oppretter en keystore i en egen mappe i prosjektet.

```
olineamundsen@Olines-MBP øving 2 % java -jar -Djavax.net.ssl.keyStore=ex
amplestore -Djavax.net.ssl.keyStorePassword=123456 "bin/server/JavaSSLSe
rver.jar"
SSL ServerSocket started
[SSL: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8000]]
```

Kjører deretter komandoen:
java -jar -Djavax.net.ssl.keyStore=examplestore -Djavax.net.ssl.keyStorePassword=123456 "bin/server/JavaSSLServer.jar"

Som kjører jar-fila og bruker keystore. Starter da server og deretter client, og har derav kryptert meldinger gjennom TLS/SSL.

```
olineamundsen@Olines-MBP øving 2 % java -jar -Djavax.net.ssl.trustStore=examples
tore -Djavax.net.ssl.trustStorePassword=123456 "bin/client/JavaSSLClient.jar"
Enter something:
1 + 3
1 + 3
Enter something:
dette funker
dette funker
Enter something:
q
```

Figure 1: Client side

```
olineamundsen@Olines-MBP øving 2 % java -jar -Djavax.net.ssl.keyStore=ex
amplestore -Djavax.net.ssl.keyStorePassword=123456 "bin/server/JavaSSLSe
rver.jar"
SSL ServerSocket started
[SSL: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8000]]
ServerSocket accepted
1 + 3
dette funker
Closed
```

Figur 1: Server side

Vi kjørte Wireshark, og observerte på loopback under kommunikasjonen. Vi observerte at innholdet var kryptert med TLS. Datafangsten ligger ved som vedlegg.

```
> Frame 13: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface lo0, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 8000, Dst Port: 49524, Seq: 134, Ack: 392, Len: 70
v Transport Layer Security
  v TLSv1.3 Record Layer: Application Data Protocol: Application Data
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 65
      Encrypted Application Data: 5ec261522476e809b262131e7f28d7b45ffa11cebbf0093bac5ae6a75ead3f194c172fe6…
```

Ser at port 8000 er i bruk som vi spesifikt satte opp i koden, og at dataen er kryptert.

Works Cited

Ukjent. (2021, Feb 1). *Java example of SSL Server and Client, and how to generate keystore*. Retrieved from java-buddy.blogspot.com: http://java-buddy.blogspot.com/2016/07/java-example-of-ssl-server-and-client.html