DAT159 – CRYPTOGRAPHY OBLIG

By Arne Engelsen Flatekval (181194)

Innholdsfortegnelse

Using no encryption

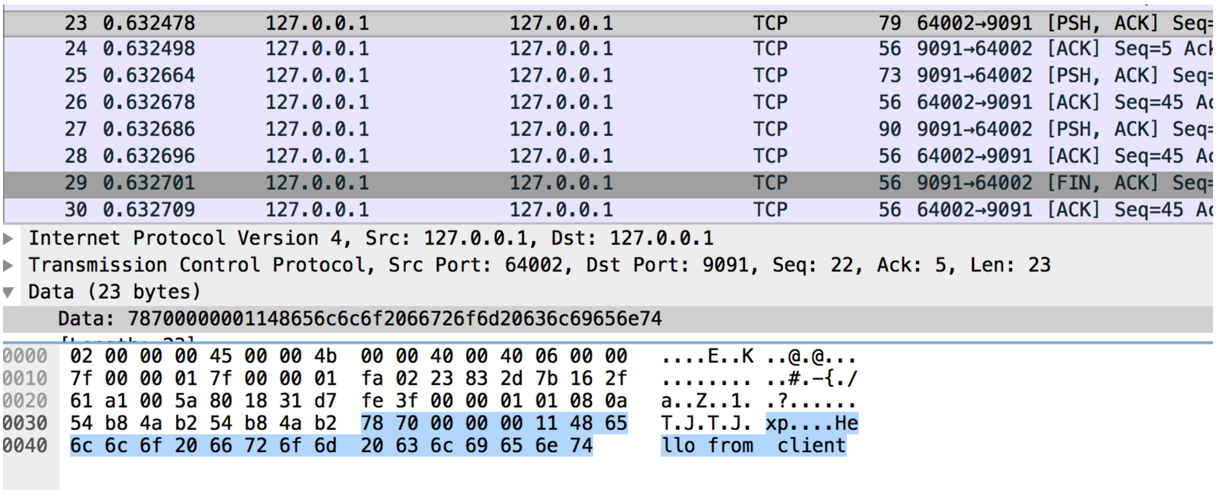
The client console:

```
Connected to Server on localhost/127.0.0.1
Response from server: Message received from client
```

The server console:

```
Waiting for requests from client...
Connected to client at the address: /127.0.0.1
Message from Client: Hello from client
Waiting for requests from client...
```

Wireshark



```
    23 0.632478     127.0.0.1       127.0.0.1       TCP      79 64002→9091 [PSH, ACK] Seq=
    24 0.632498     127.0.0.1       127.0.0.1       TCP      56 9091→64002 [ACK] Seq=5 Ack
    25 0.632664     127.0.0.1       127.0.0.1       TCP      73 9091→64002 [PSH, ACK] Seq=
    26 0.632678     127.0.0.1       127.0.0.1       TCP      56 64002→9091 [ACK] Seq=45 Ac
    27 0.632686     127.0.0.1       127.0.0.1       TCP      90 9091→64002 [PSH, ACK] Seq=
    28 0.632696     127.0.0.1       127.0.0.1       TCP      56 64002→9091 [ACK] Seq=45 Ac
    29 0.632701     127.0.0.1       127.0.0.1       TCP      56 9091→64002 [FIN, ACK] Seq=
    30 0.632709     127.0.0.1       127.0.0.1       TCP      56 64002→9091 [ACK] Seq=45 Ac
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 64002, Dst Port: 9091, Seq: 22, Ack: 5, Len: 23
▼ Data (23 bytes)
    Data: 78700000001148656c6c6f2066726f6d20636c69656e74
0000  02 00 00 00 45 00 00 4b  00 00 40 00 40 06 00 00   ....E..K ..@.@...
0010  7f 00 00 01 7f 00 00 01  fa 02 23 83 2d 7b 16 2f   ........ ..#.-{./
0020  61 a1 00 5a 80 18 31 d7  fe 3f 00 00 01 01 08 0a   a..Z..1. .?......
0030  54 b8 4a b2 54 b8 4a b2  78 70 00 00 00 11 48 65   T.J.T.J. xp....He
0040  6c 6c 6f 20 66 72 6f 6d  20 63 6c 69 65 6e 74      llo from  client
```

We can clearly see in plaintext what the client sent to the server.

Using DES encryption

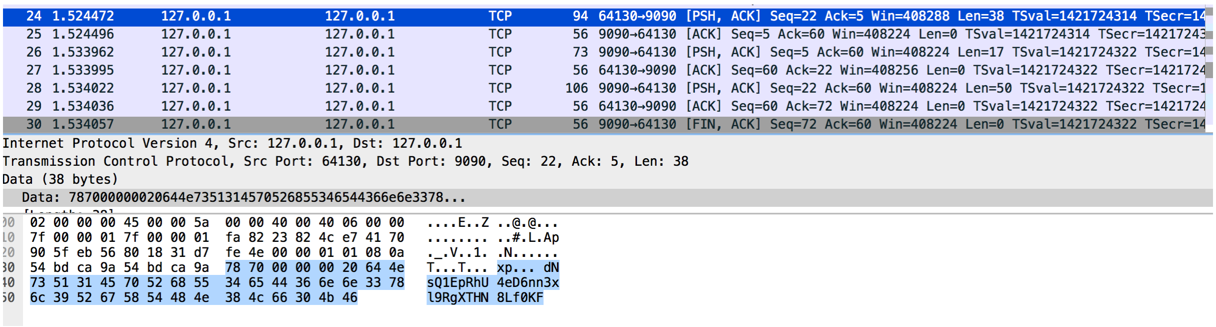
The client console:

```
Connected to Server on localhost/127.0.0.1
Response from server: Message received from client
```

The server console:

```
Waiting for requests from client...
Connected to client at the address: /127.0.0.1
Message from Client: Hello from client
Waiting for requests from client...
```

Wireshark



```
 24 1.524472    127.0.0.1      127.0.0.1      TCP     94 64130→9090 [PSH, ACK] Seq=22 Ack=5 Win=408288 Len=38 TSval=1421724314 TSecr=14
 25 1.524496    127.0.0.1      127.0.0.1      TCP     56 9090→64130 [ACK] Seq=5 Ack=60 Win=408224 Len=0 TSval=1421724314 TSecr=14217243
 26 1.533962    127.0.0.1      127.0.0.1      TCP     73 9090→64130 [PSH, ACK] Seq=5 Ack=60 Win=408224 Len=17 TSval=1421724322 TSecr=14
 27 1.533995    127.0.0.1      127.0.0.1      TCP     56 64130→9090 [ACK] Seq=60 Ack=22 Win=408256 Len=0 TSval=1421724322 TSecr=1421724
 28 1.534022    127.0.0.1      127.0.0.1      TCP    106 9090→64130 [PSH, ACK] Seq=22 Ack=60 Win=408224 Len=50 TSval=1421724322 TSecr=1
 29 1.534036    127.0.0.1      127.0.0.1      TCP     56 64130→9090 [ACK] Seq=60 Ack=72 Win=408224 Len=0 TSval=1421724322 TSecr=1421724
 30 1.534057    127.0.0.1      127.0.0.1      TCP     56 9090→64130 [FIN, ACK] Seq=72 Ack=60 Win=408224 Len=0 TSval=1421724322 TSecr=14
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 64130, Dst Port: 9090, Seq: 22, Ack: 5, Len: 38
Data (38 bytes)
  Data: 787000000020644e735131457052685534654436 6e6e3378...
```

```
00  02 00 00 00 45 00 00 5a  00 00 40 00 40 06 00 00   ....E..Z ..@.@...
10  7f 00 00 01 7f 00 00 01  fa 82 23 82 4c e7 41 70   ........ ..#.L.Ap
20  90 5f eb 56 80 18 31 d7  fe 4e 00 00 01 01 08 0a   ._.V..1. .N......
30  54 bd ca 9a 54 bd ca 9a  78 70 00 00 00 20 64 4e   T...T... xp... dN
40  73 51 31 45 70 52 68 55  34 65 44 36 6e 6e 33 78   sQ1EpRhU 4eD6nn3x
50  6c 39 52 67 58 54 48 4e  38 4c 66 30 4b 46         l9RgXTHN 8Lf0KF
```

This time we can see that the message sent to the server is encrypted.