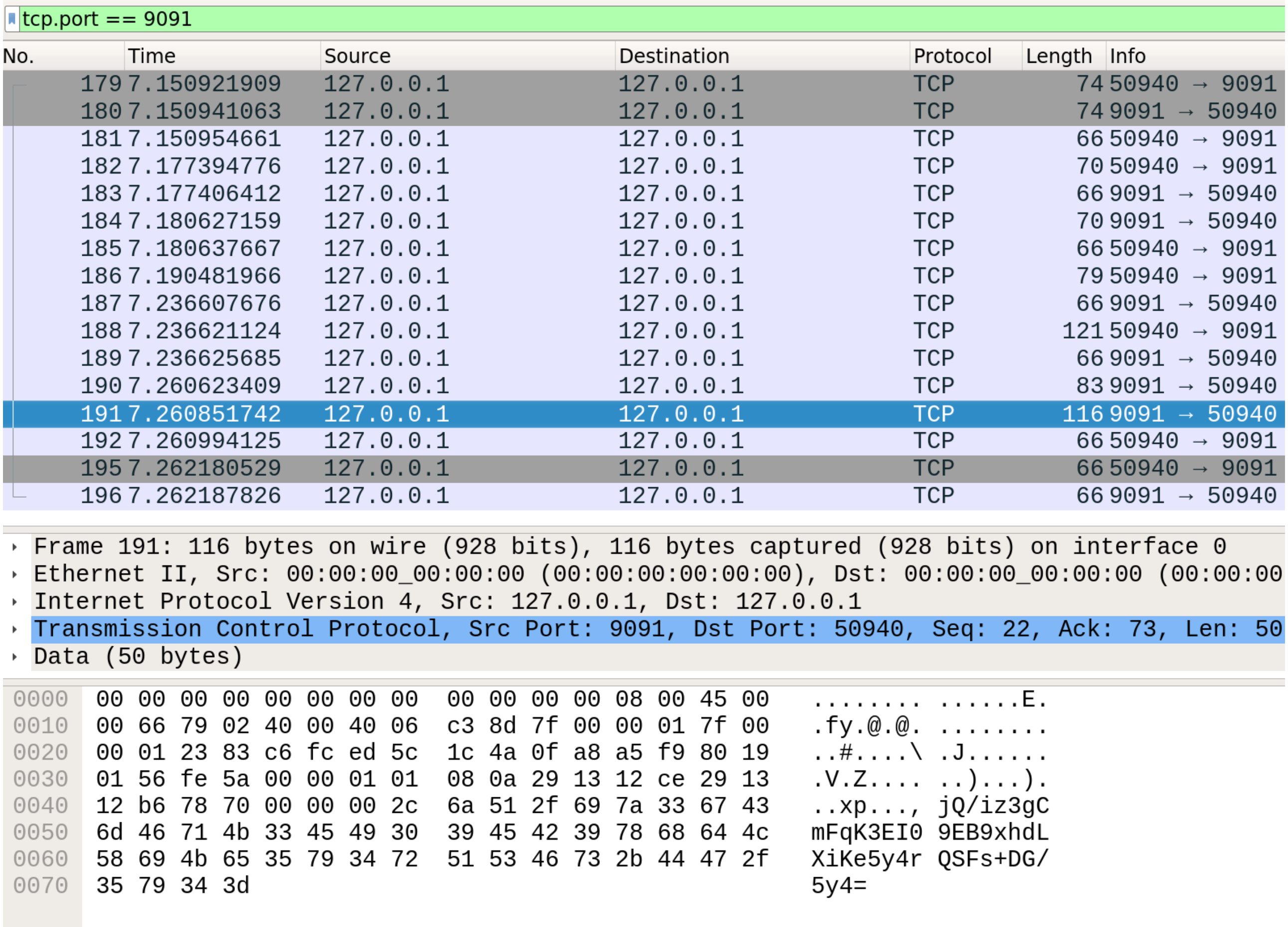
# DAT159 - Oblig 2

## by: Kristian Åsnes, Fredrik Mathisen, Sindre Steinsvik and Preben Haukebøe

We tried to find a way to send the key as plaintext, so that we could “sniff” It up and use to decrypt the message, but we couldn’t see it through. That’s why we didn’t try to decrypt the message and why the code ended up being unnecessarily complicated and messy where we send and receive the key. (In the java files.)

1.



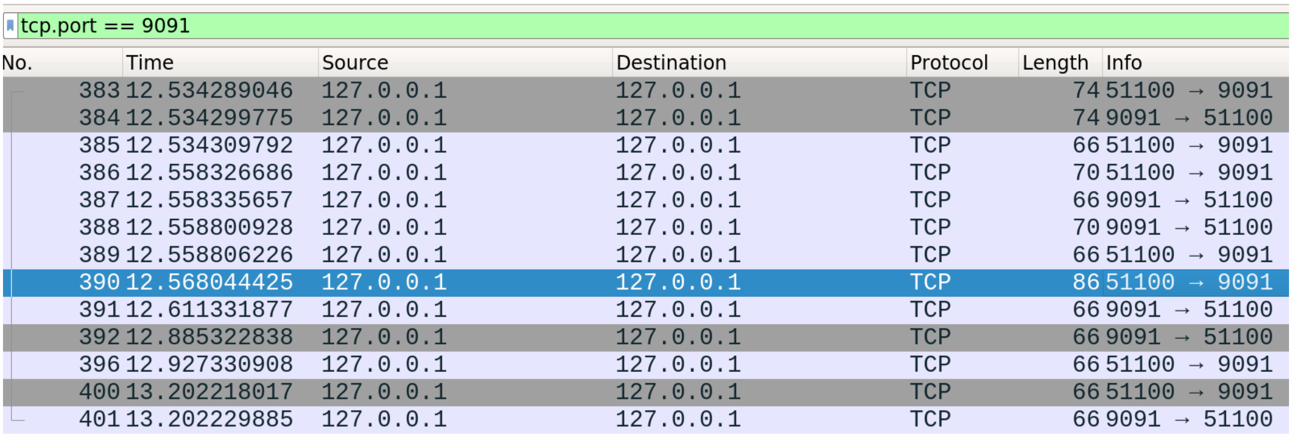
tcp.port == 9091

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 179 | 7.150921909 | 127.0.0.1 | 127.0.0.1 | TCP | 74 | 50940 → 9091 |
| 180 | 7.150941063 | 127.0.0.1 | 127.0.0.1 | TCP | 74 | 9091 → 50940 |
| 181 | 7.150954661 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 50940 → 9091 |
| 182 | 7.177394776 | 127.0.0.1 | 127.0.0.1 | TCP | 70 | 50940 → 9091 |
| 183 | 7.177406412 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 9091 → 50940 |
| 184 | 7.180627159 | 127.0.0.1 | 127.0.0.1 | TCP | 70 | 9091 → 50940 |
| 185 | 7.180637667 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 50940 → 9091 |
| 186 | 7.190481966 | 127.0.0.1 | 127.0.0.1 | TCP | 79 | 50940 → 9091 |
| 187 | 7.236607676 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 9091 → 50940 |
| 188 | 7.236621124 | 127.0.0.1 | 127.0.0.1 | TCP | 121 | 50940 → 9091 |
| 189 | 7.236625685 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 9091 → 50940 |
| 190 | 7.260623409 | 127.0.0.1 | 127.0.0.1 | TCP | 83 | 9091 → 50940 |
| 191 | 7.260851742 | 127.0.0.1 | 127.0.0.1 | TCP | 116 | 9091 → 50940 |
| 192 | 7.260994125 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 50940 → 9091 |
| 195 | 7.262180529 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 50940 → 9091 |
| 196 | 7.262187826 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 9091 → 50940 |

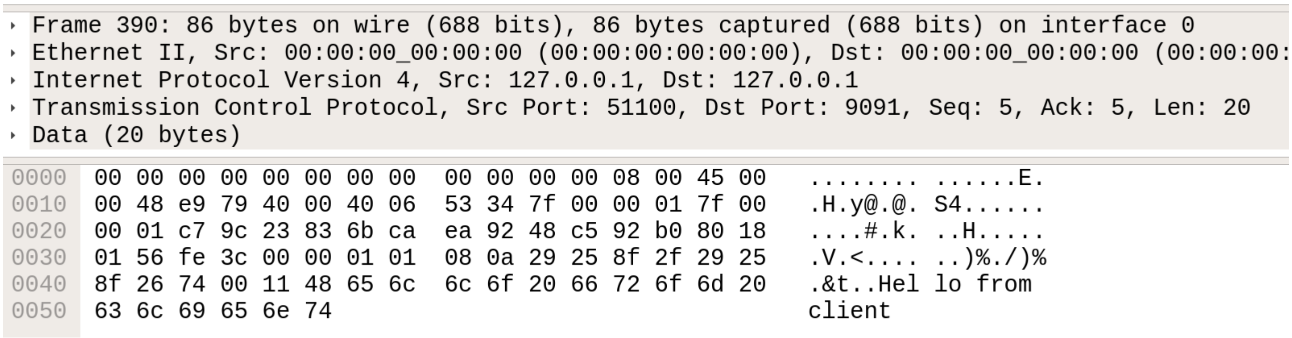
```
▸ Frame 191: 116 bytes on wire (928 bits), 116 bytes captured (928 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 9091, Dst Port: 50940, Seq: 22, Ack: 73, Len: 50
▸ Data (50 bytes)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 66 79 02 40 00 40 06  c3 8d 7f 00 00 01 7f 00   .fy.@.@. ........
0020  00 01 23 83 c6 fc ed 5c  1c 4a 0f a8 a5 f9 80 19   ..#....\ .J......
0030  01 56 fe 5a 00 00 01 01  08 0a 29 13 12 ce 29 13   .V.Z.... ..)...).
0040  12 b6 78 70 00 00 00 2c  6a 51 2f 69 7a 33 67 43   ..xp..., jQ/iz3gC
0050  6d 46 71 4b 33 45 49 30  39 45 42 39 78 68 64 4c   mFqK3EI0 9EB9xhdL
0060  58 69 4b 65 35 79 34 72  51 53 46 73 2b 44 47 2f   XiKe5y4r QSFs+DG/
0070  35 79 34 3d                                        5y4=
```

On the picture above, you can see the encrypted message in the down-left corner.



```
tcp.port == 9091
No.    Time           Source      Destination   Protocol  Length  Info
  383  12.534289046   127.0.0.1   127.0.0.1     TCP       74      51100 → 9091
  384  12.534299775   127.0.0.1   127.0.0.1     TCP       74      9091 → 51100
  385  12.534309792   127.0.0.1   127.0.0.1     TCP       66      51100 → 9091
  386  12.558326686   127.0.0.1   127.0.0.1     TCP       70      51100 → 9091
  387  12.558335657   127.0.0.1   127.0.0.1     TCP       66      9091 → 51100
  388  12.558800928   127.0.0.1   127.0.0.1     TCP       70      9091 → 51100
  389  12.558806226   127.0.0.1   127.0.0.1     TCP       66      51100 → 9091
  390  12.568044425   127.0.0.1   127.0.0.1     TCP       86      51100 → 9091
  391  12.611331877   127.0.0.1   127.0.0.1     TCP       66      9091 → 51100
  392  12.885322838   127.0.0.1   127.0.0.1     TCP       66      9091 → 51100
  396  12.927330908   127.0.0.1   127.0.0.1     TCP       66      51100 → 9091
  400  13.202218017   127.0.0.1   127.0.0.1     TCP       66      51100 → 9091
  401  13.202229885   127.0.0.1   127.0.0.1     TCP       66      9091 → 51100
```



```
› Frame 390: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
› Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 51100, Dst Port: 9091, Seq: 5, Ack: 5, Len: 20
› Data (20 bytes)

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 48 e9 79 40 00 40 06  53 34 7f 00 00 01 7f 00   .H.y@.@. S4......
0020  00 01 c7 9c 23 83 6b ca  ea 92 48 c5 92 b0 80 18   ....#.k. ..H.....
0030  01 56 fe 3c 00 00 01 01  08 0a 29 25 8f 2f 29 25   .V.<.... ..)%./)%
0040  8f 26 74 00 11 48 65 6c  6c 6f 20 66 72 6f 6d 20   .&t..Hel lo from 
0050  63 6c 69 65 6e 74                                  client
```

On the picture above, you can see the text sent in plaintext in the down-right corner.