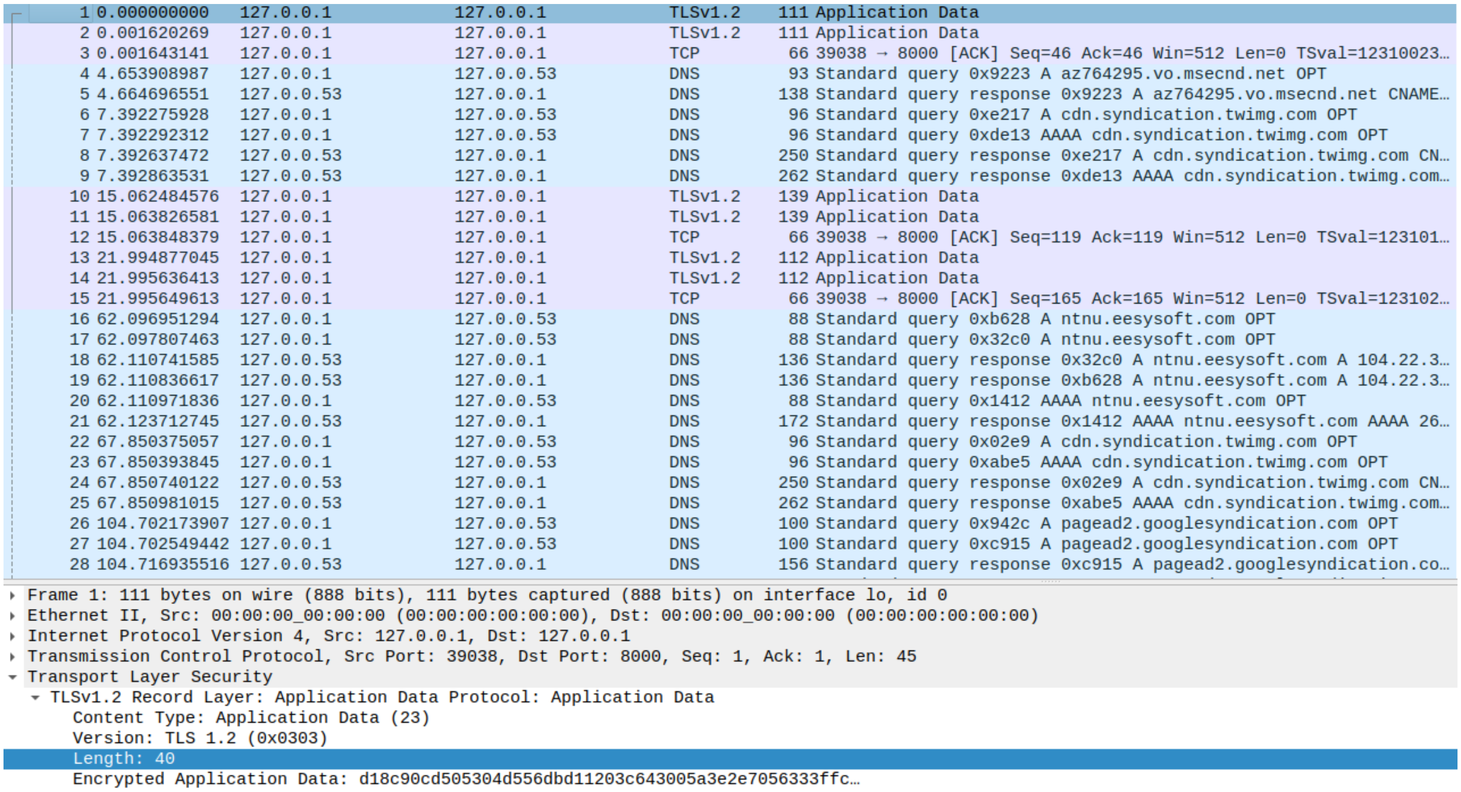
Oblig 2 - Oppgave 2 - TSL

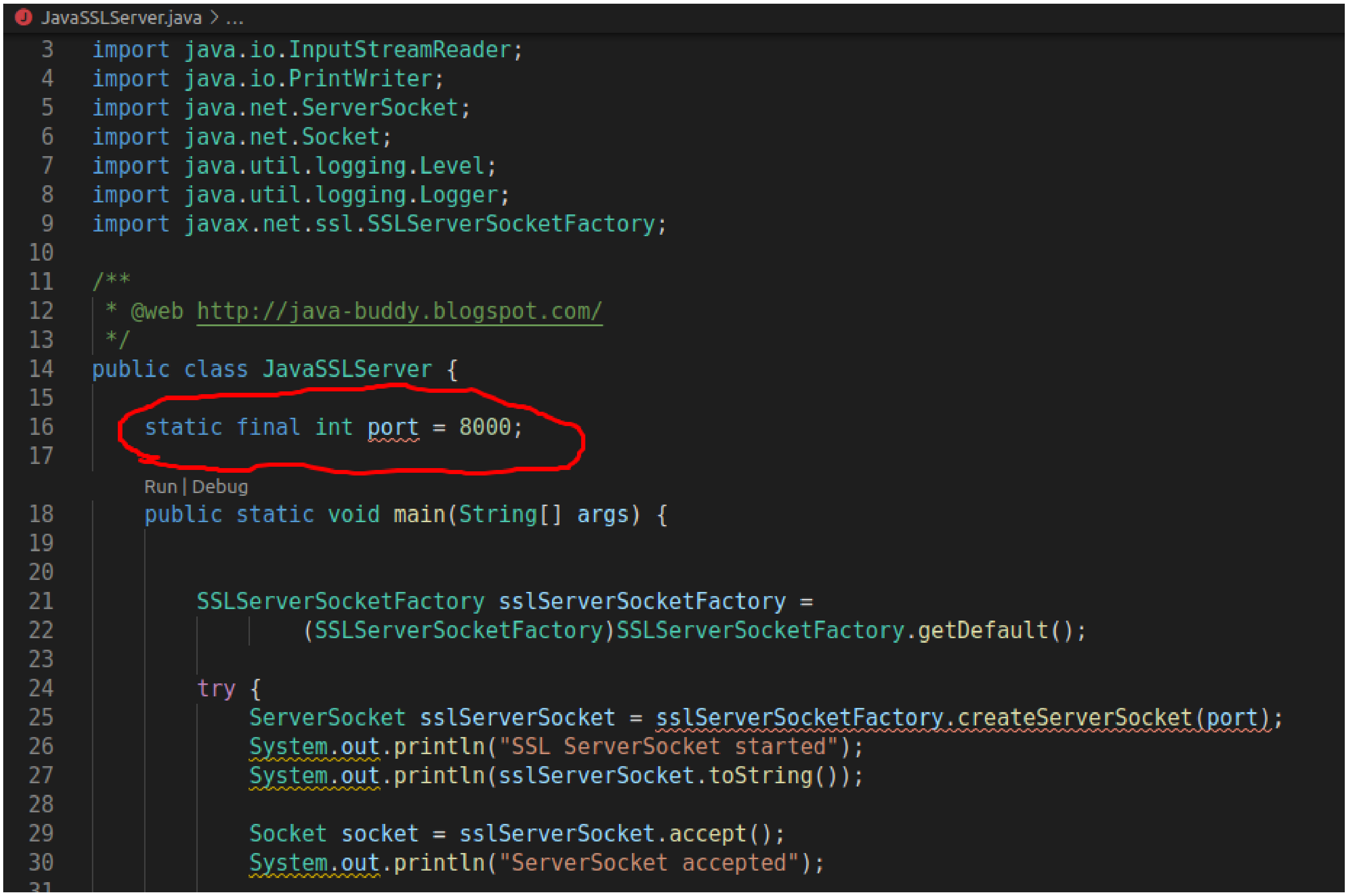
Wireshark capture when I use the loopback interface:



```
  1 0.000000000   127.0.0.1     127.0.0.1     TLSv1.2   111 Application Data
  2 0.001620269   127.0.0.1     127.0.0.1     TLSv1.2   111 Application Data
  3 0.001643141   127.0.0.1     127.0.0.1     TCP        66 39038 → 8000 [ACK] Seq=46 Ack=46 Win=512 Len=0 TSval=12310023…
  4 4.653908987   127.0.0.1     127.0.0.53    DNS        93 Standard query 0x9223 A az764295.vo.msecnd.net OPT
  5 4.664696551   127.0.0.53    127.0.0.1     DNS       138 Standard query response 0x9223 A az764295.vo.msecnd.net CNAME…
  6 7.392275928   127.0.0.1     127.0.0.53    DNS        96 Standard query 0xe217 A cdn.syndication.twimg.com OPT
  7 7.392292312   127.0.0.1     127.0.0.53    DNS        96 Standard query 0xde13 AAAA cdn.syndication.twimg.com OPT
  8 7.392637472   127.0.0.53    127.0.0.1     DNS       250 Standard query response 0xe217 A cdn.syndication.twimg.com CN…
  9 7.392863531   127.0.0.53    127.0.0.1     DNS       262 Standard query response 0xde13 AAAA cdn.syndication.twimg.com…
 10 15.062484576  127.0.0.1     127.0.0.1     TLSv1.2   139 Application Data
 11 15.063826581  127.0.0.1     127.0.0.1     TLSv1.2   139 Application Data
 12 15.063848379  127.0.0.1     127.0.0.1     TCP        66 39038 → 8000 [ACK] Seq=119 Ack=119 Win=512 Len=0 TSval=123101…
 13 21.994877045  127.0.0.1     127.0.0.1     TLSv1.2   112 Application Data
 14 21.995636413  127.0.0.1     127.0.0.1     TLSv1.2   112 Application Data
 15 21.995649613  127.0.0.1     127.0.0.1     TCP        66 39038 → 8000 [ACK] Seq=165 Ack=165 Win=512 Len=0 TSval=123102…
 16 62.096951294  127.0.0.1     127.0.0.53    DNS        88 Standard query 0xb628 A ntnu.eesysoft.com OPT
 17 62.097807463  127.0.0.1     127.0.0.53    DNS        88 Standard query 0x32c0 A ntnu.eesysoft.com OPT
 18 62.110741585  127.0.0.53    127.0.0.1     DNS       136 Standard query response 0x32c0 A ntnu.eesysoft.com A 104.22.3…
 19 62.110836617  127.0.0.53    127.0.0.1     DNS       136 Standard query response 0xb628 A ntnu.eesysoft.com A 104.22.3…
 20 62.110971836  127.0.0.1     127.0.0.53    DNS        88 Standard query 0x1412 AAAA ntnu.eesysoft.com OPT
 21 62.123712745  127.0.0.53    127.0.0.1     DNS       172 Standard query response 0x1412 AAAA ntnu.eesysoft.com AAAA 26…
 22 67.850375057  127.0.0.1     127.0.0.53    DNS        96 Standard query 0x02e9 A cdn.syndication.twimg.com OPT
 23 67.850393845  127.0.0.1     127.0.0.53    DNS        96 Standard query 0xabe5 AAAA cdn.syndication.twimg.com OPT
 24 67.850740122  127.0.0.53    127.0.0.1     DNS       250 Standard query response 0x02e9 A cdn.syndication.twimg.com CN…
 25 67.850981015  127.0.0.53    127.0.0.1     DNS       262 Standard query response 0xabe5 AAAA cdn.syndication.twimg.com…
 26 104.702173907 127.0.0.1     127.0.0.53    DNS       100 Standard query 0x942c A pagead2.googlesyndication.com OPT
 27 104.702549442 127.0.0.1     127.0.0.53    DNS       100 Standard query 0xc915 A pagead2.googlesyndication.com OPT
 28 104.716935516 127.0.0.53    127.0.0.1     DNS       156 Standard query response 0xc915 A pagead2.googlesyndication.co…
```

```
▸ Frame 1: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 39038, Dst Port: 8000, Seq: 1, Ack: 1, Len: 45
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 40
      Encrypted Application Data: d18c90cd505304d556dbd11203c643005a3e2e7056333ffc…
```

A snippet from the JavaSSLServer code showing that the server is using port 8000:



```java
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.net.ssl.SSLServerSocketFactory;

/**
 * @web http://java-buddy.blogspot.com/
 */
public class JavaSSLServer {

    static final int port = 8000;

    public static void main(String[] args) {


        SSLServerSocketFactory sslServerSocketFactory =
                (SSLServerSocketFactory)SSLServerSocketFactory.getDefault();

        try {
            ServerSocket sslServerSocket = sslServerSocketFactory.createServerSocket(port);
            System.out.println("SSL ServerSocket started");
            System.out.println(sslServerSocket.toString());

            Socket socket = sslServerSocket.accept();
            System.out.println("ServerSocket accepted");
```

In the wireshark picture you can see in the column to the right most, there is traffic going on between port 8000, the server, and port 39038, the client. We can also see that the application data is enctrypted using TLSv1.2. In other words we can't see what is written from the server to the client and the other way around and the transport layer security is working!