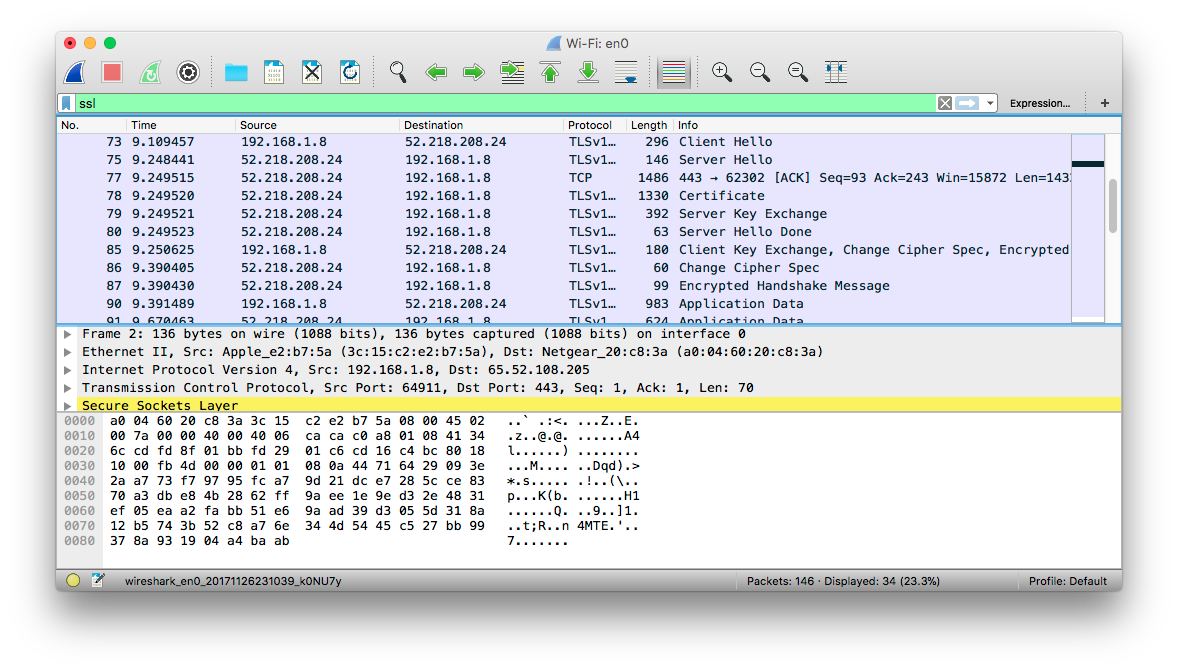
CSC 265 Lab Assignment 5

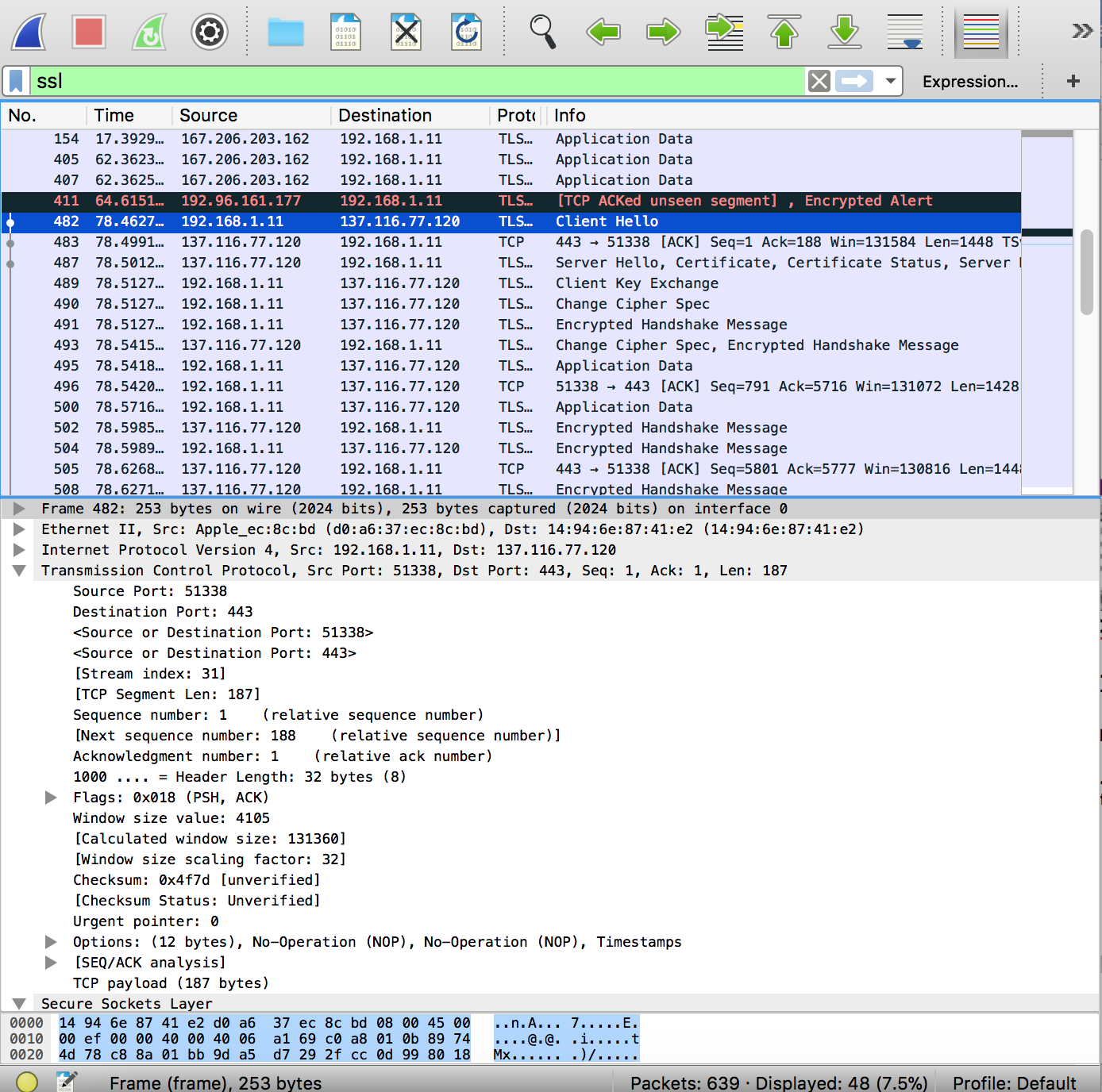
Due on 12/09/2017 11:59pm

In this lab, we’ll investigate the Secure Sockets Layer (SSL) protocol, focusing on the SSL records sent over a TCP connection. We’ll do so by analyzing a trace of the SSL records sent between your host and an e-commerce server. We’ll investigate the various SSL record types as well as the fields in the SSL messages.



1. Capturing packets in an SSL session

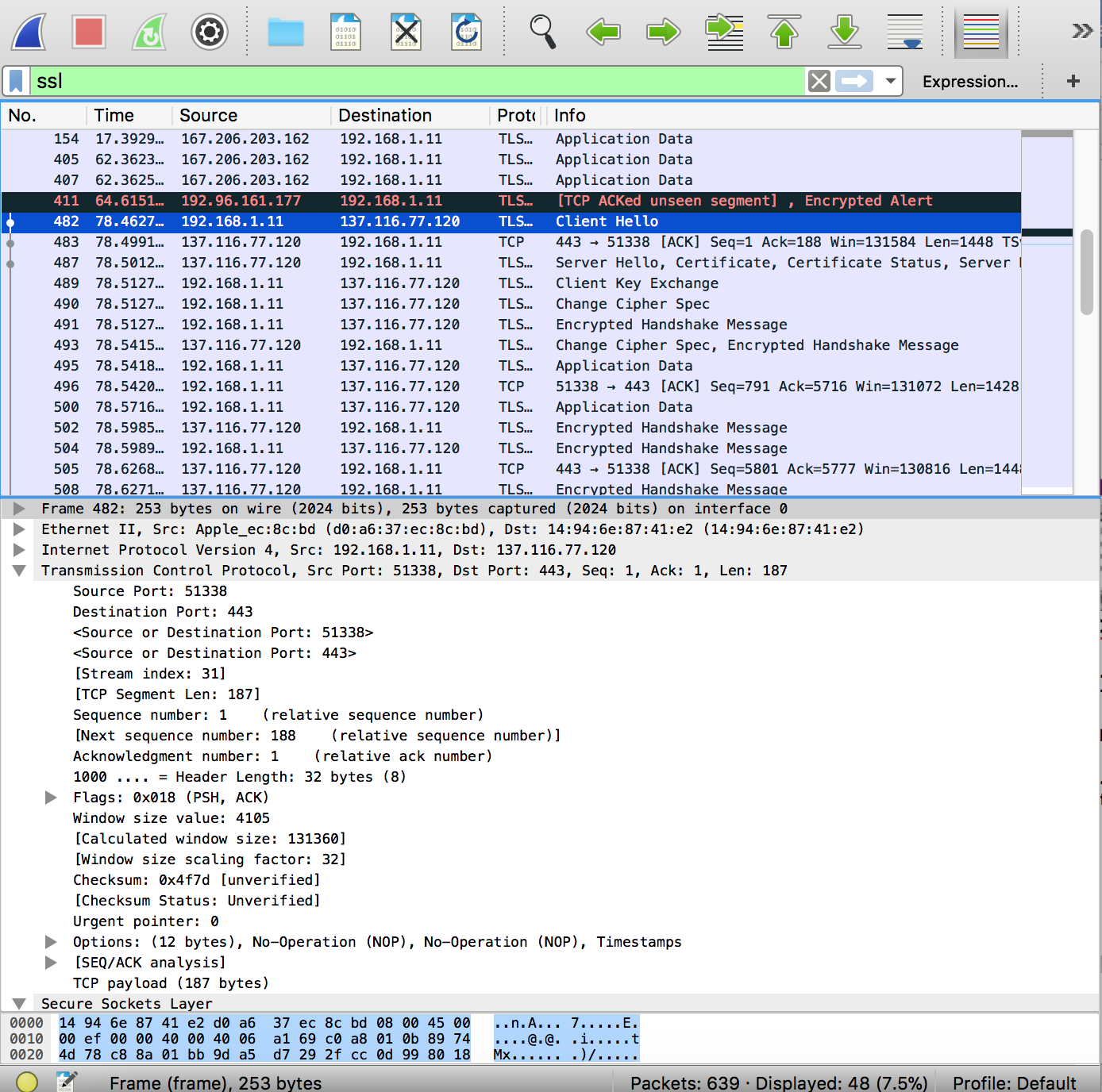
The first step is to capture the packets in an SSL session. To do this, you should go to your favorite e-commerce site and begin the process of purchasing an item (but terminating before making the actual purpose!). After capturing the packets with Wireshark, you should set the filter so that it displays only the Ethernet frames that contain SSL records sent from and received by your host. (An SSL record is the same thing as an SSL message.) You should obtain something like screenshot on the previous page.

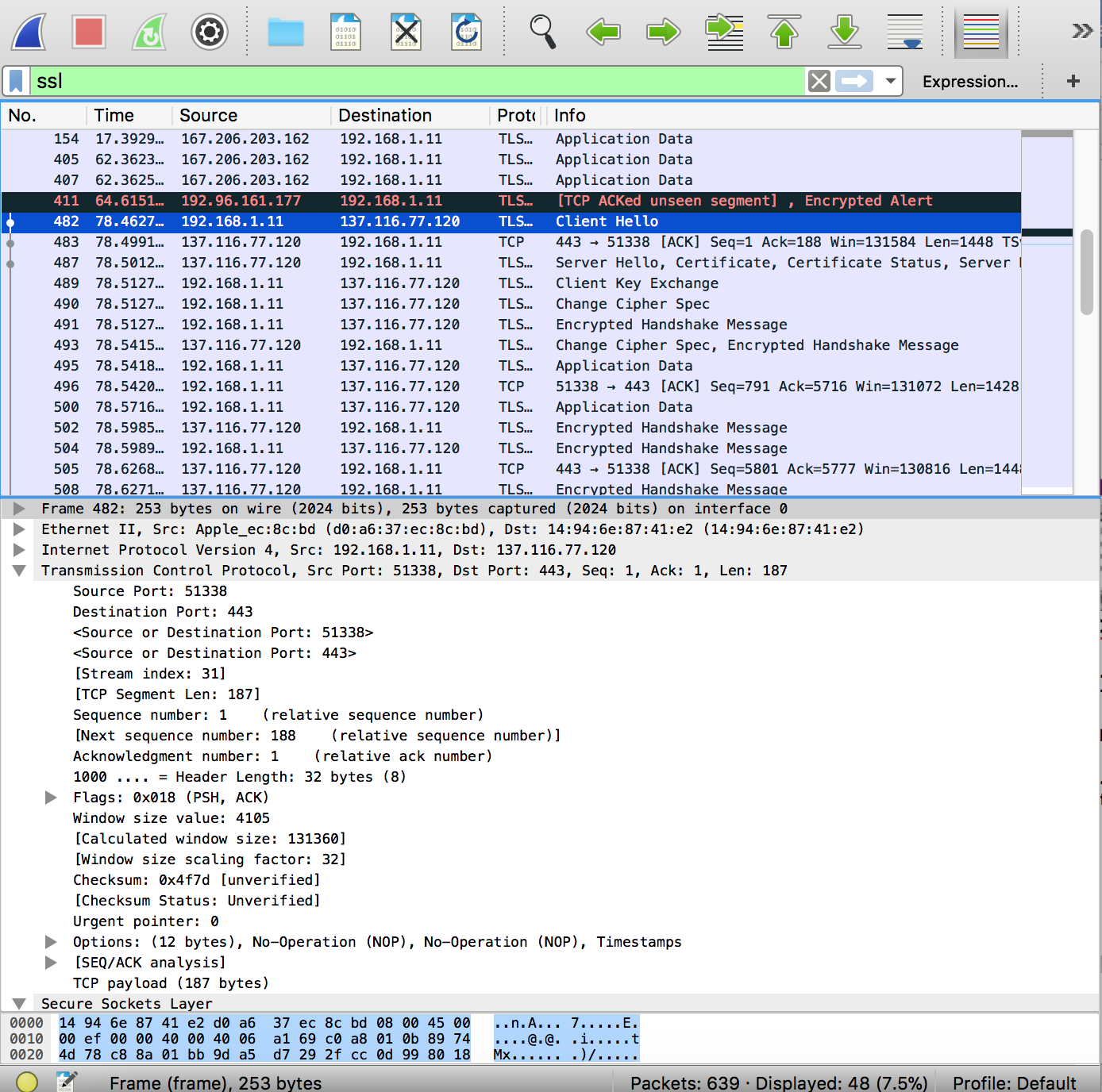


2. A look at the captured trace

Your Wireshark GUI should be displaying only the Ethernet frames that have SSL records. It is important to keep in mind that an Ethernet frame may contain one or more SSL records. (This is very different from HTTP, for which each frame contains either one complete HTTP message or a portion of a HTTP message.) Also, an SSL record may not completely fit into an Ethernet frame, in which case multiple frames will be needed to carry the record.

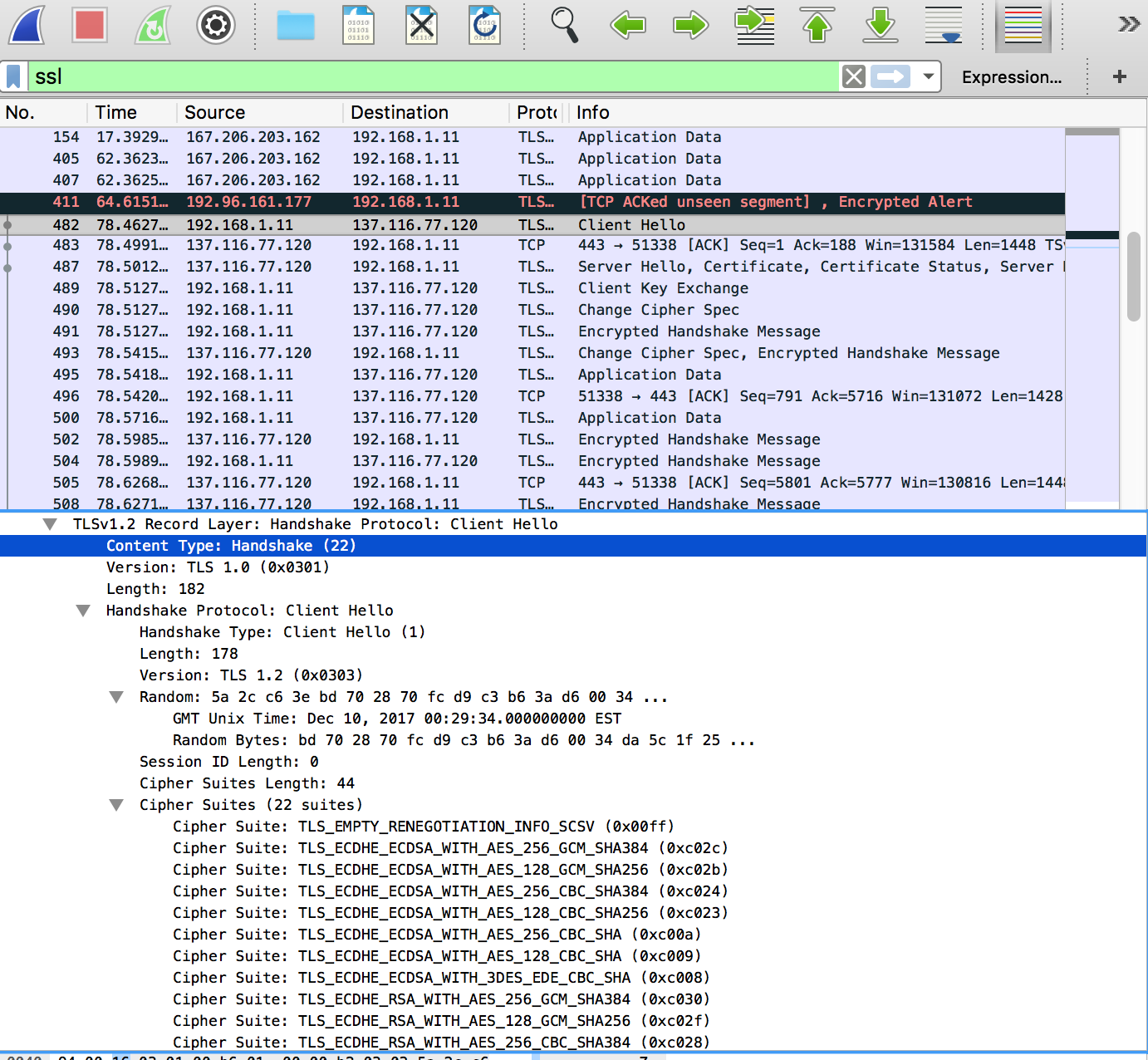
Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout[[1]](#footnote-1) to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question



1. Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is “content type” and has length of one byte. List all three fields and their lengths. 
   1. Content Type: 1 byte
   2. Version: 2 byte
   3. Length: 3 byte

ClientHello Record:

1. Does the ClientHello record contain a nonce (also known as a “challenge”)? If so, what is the value of the challenge in hexadecimal notation?
   1. The client hello challenge is c0 28 c0 27 c0 14 c0 13 c0 12 00 9d 00 9c 00 3d.
2. Does the ClientHello record advertise the cyber suites it supports? If so, in the first listed suite, what are the public-key algorithm, the symmetric-key algorithm, and the hash algorithm?
   1. First use of RSA for public key crypto, that the RC4 are for the symmetric key that is cipher and uses the Mds hashing aligothrm.

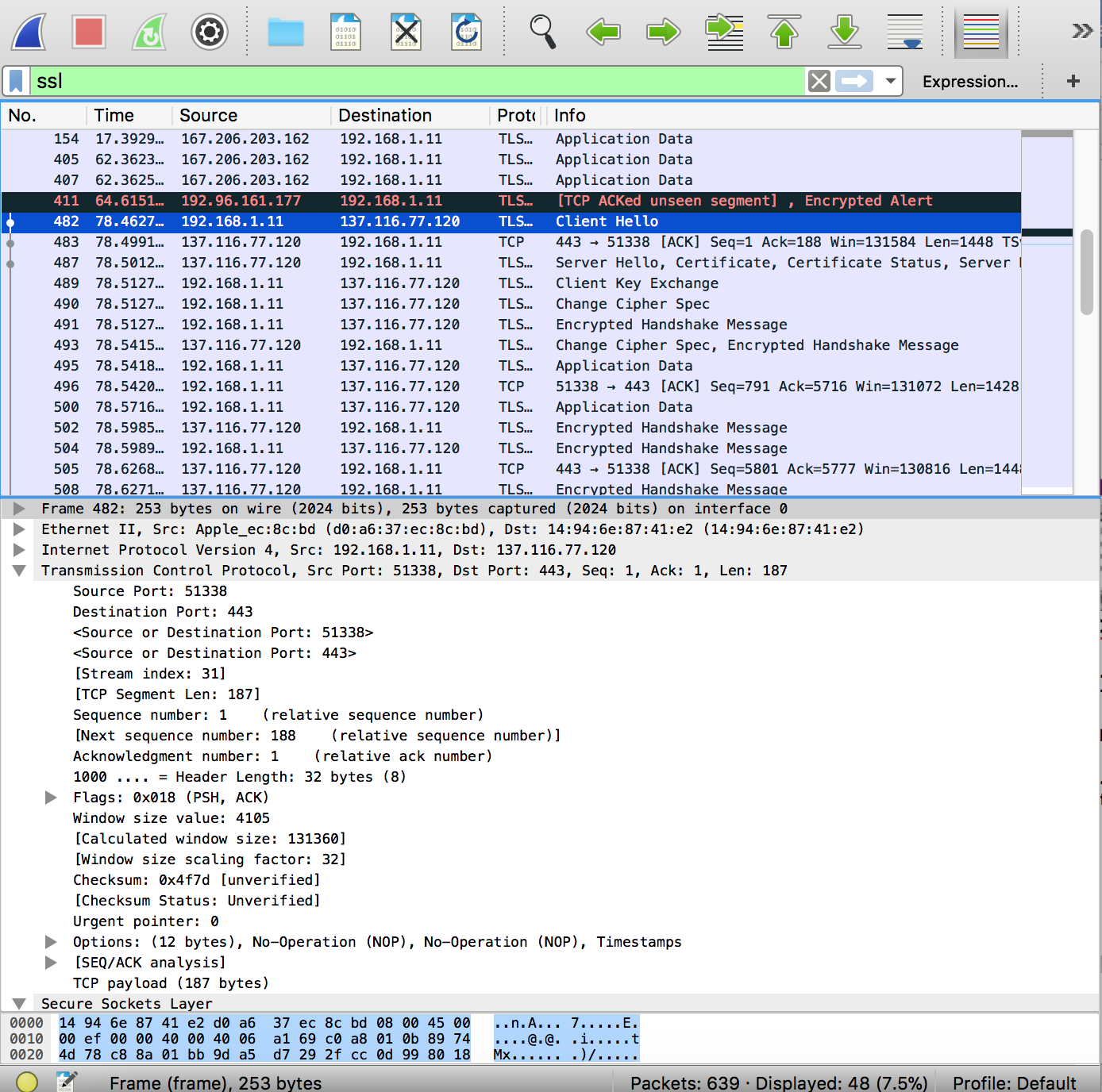


ServerHello Record:

1. Does this record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL?
   1. The record does include a nonce, which is under random. It is long 32 bits long and 28 data and 4 for the time and the purpose of the client and server nonces in SSL is to prevent replay attacks.
2. Does this record include a session ID? What is the purpose of the session ID?
   1. Yes, this session does include session ID. That provides a unique and persistent identifier for the SSL session, this sends in the clear to go. Clients has to resume to the same session in which tis used by the server provided session ID that’s when it sends Client Hello.

1. Does this record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?
   1. This is no record that contains a certificate or hold a certificate included in a separate record. Yes, certificate fit into Ethernet frame.

Client Key Exchange Record:



1. Locate the client key exchange record. Does this record contain a pre-master secret? What is this secret used for? Is the secret encrypted? If so, how? How long is the encrypted secret?
   1. The pre-master secret does contain a record. The secret use for this is that that both server and client makes a secret which uses a generated session key for the Mac and encryption. Server public key is use to get the secret encryption. An encrypted secret is 128 bytes.

Change Cipher Spec Record (sent by client) and Encrypted Handshake Record:

1. In the encrypted handshake record, what is being encrypted? How?
   1. In an encrypted handshake record indicate that the content of the following ssl records everything sent by the client is and will be encrypted.
2. Does the server also send a change cipher record and an encrypted handshake record to the client? How are those records different from those sent by the client?
   1. Yes, the server also sends a change cipher record and encrypted handshake record to the client. Those records are different from what is sent by client because it contains the concatenation of all the handshake messages sent. The messages sent from the server instead of from the client. The record would be the same at the end.

Application Data

1. How is the application data being encrypted? Do the records containing application data include a MAC? Does Wireshark distinguish between the encrypted application data and the MAC?
   1. The application data being encrypted is using handshake phase. Is chosen by the algorithm. A symmetric key encryption algorithm that is chosen in the key generated by the pre-master and nonce.
   2. Wireshark does not distinguish it between the encrypted application data and the Mac.

1. [↑](#footnote-ref-1)