Transport Layer Security (TLS)

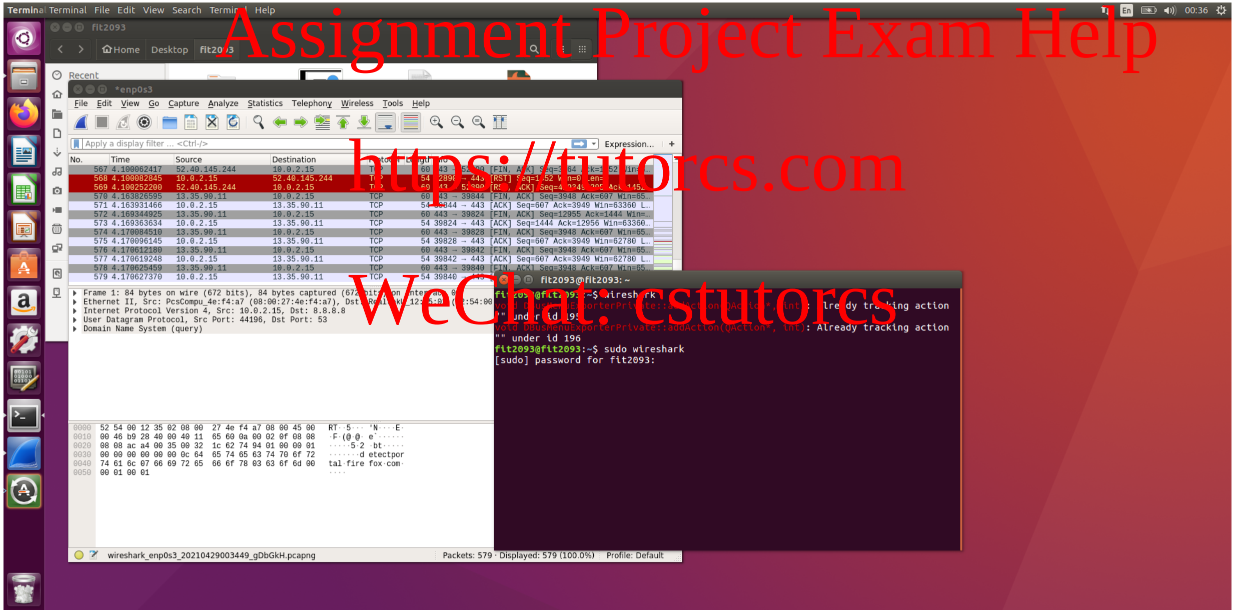
IMPORTANT NOTES:
Study lecture materials at least 1 hour and prepare Lab Task 3.1 prior to the lab session. Prepared questions will be discussed in the lab session.

1 Overview

The learning objective of this lab is for students to get familiar with TLS protocol.

2 Lab Environment

In this lab, we will use wireshark preinstalled in the cloud VM to analyse three captured packets files. Click "Applications–>Internet–>Wireshark" from the desktop to start the Wireshark. Alternatively, click any captured file in folder `/srv/fit2093files/fit2093lab/` such as `Example1.pcap` to open Wireshark. You may also choose to download and install the Wireshark on your own devices. More information can be found from `https://www.wireshark.org/`


Terminal Terminal File Edit View Search Terminal Help
00:36
fit2093
Home Desktop fit2093
Recent
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.  Time         Source          Destination     Protocol Length Info
 567 4.100062417  52.40.145.244   10.0.2.15       TCP      60 443 → 52890 [FIN, ACK] Seq=3?64 Ack=1452 Win=…
 568 4.100082845  10.0.2.15       52.40.145.244   TCP      54 52890 → 443 [RST] Seq=1452 Win=0 Len=
 569 4.100252200  52.40.145.244   10.0.2.15       TCP      60 443 → 52890 [RST, ACK] Seq=4…
 570 4.163826595  13.35.90.11     10.0.2.15       TCP      60 443 → 39844 [FIN, ACK] Seq=3948 Ack=607 Win=65…
 571 4.163931466  10.0.2.15       13.35.90.11     TCP      54 39844 → 443 [ACK] Seq=607 Ack=3949 Win=63360 L…
 572 4.169344925  13.35.90.11     10.0.2.15       TCP      60 443 → 39824 [FIN, ACK] Seq=12955 Ack=1444 Win=…
 573 4.169363634  10.0.2.15       13.35.90.11     TCP      54 39824 → 443 [ACK] Seq=1444 Ack=12956 Win=63360…
 574 4.170084510  13.35.90.11     10.0.2.15       TCP      60 443 → 39828 [FIN, ACK] Seq=3948 Ack=607 Win=65…
 575 4.170096145  10.0.2.15       13.35.90.11     TCP      54 39828 → 443 [ACK] Seq=607 Ack=3949 Win=62780 L…
 576 4.170612180  13.35.90.11     10.0.2.15       TCP      60 443 → 39842 [FIN, ACK] Seq=3948 Ack=607 Win=65…
 577 4.170619248  10.0.2.15       13.35.90.11     TCP      54 39842 → 443 [ACK] Seq=607 Ack=3949 Win=62780 L…
 578 4.170625459  13.35.90.11     10.0.2.15       TCP      60 443 → 39840 [FIN, ACK] Seq=3948 Ack=607 Win=65…
 579 4.170627370  10.0.2.15       13.35.90.11     TCP      54 39840 → 443 …

▸ Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_4e:f4:a7 (08:00:27:4e:f4:a7), Dst: RealtekU_12:35:02 (52:54:00…
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
▸ User Datagram Protocol, Src Port: 44196, Dst Port: 53
▸ Domain Name System (query)

0000  52 54 00 12 35 02 08 00  27 4e f4 a7 08 00 45 00   RT··5··· 'N····E·
0010  00 46 b9 28 40 00 40 11  65 60 0a 00 02 0f 08 08   ·F·(@·@· e`······
0020  08 08 ac a4 00 35 00 32  1c 62 74 94 01 00 00 01   ·····5·2 ·bt·····
0030  00 00 00 00 00 00 0c 64  65 74 65 63 74 70 6f 72   ·······d etectpor
0040  74 61 6c 07 66 69 72 65  66 6f 78 03 63 6f 6d 00   tal·fire fox·com·
0050  00 01 00 01                                        ····

wireshark_enp0s3_20210429003449_gDbGkH.pcapng      Packets: 579 · Displayed: 579 (100.0%)      Profile: Default
```

```
fit2093@fit2093: ~
fit2093@fit2093:~$ wireshark
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action
"" under id 195
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action
"" under id 196
fit2093@fit2093:~$ sudo wireshark
[sudo] password for fit2093:
```

3 Lab Tasks

3.1 TLS, HTTP, HTTPS

For this task you need to use Wireshark in order to look at three different examples of recorded network traffic. All three examples show parts of the communication between a client and a webserver.

1. Start Wireshark and open `/srv/fit2093files/fit2093lab/Example1.pcap`.

   (a) Can you identify the domain name of the server?

   (b) Which protocols are used on application layer?

(c) Can you get information on the location of destination and source?

The wireshark file just shows an extract with HTTP messages. Students should look at the different layers and see what kind of information they can get.

The address is `http://www.bendigobank.com.au`. This page just uses HTTP. No authenticity, no encryption. Location for Bendigo bank and Monash University can be found.

2. Open `/srv/fit2093files/fit2093lab/Example2.pcap` in Wireshark.

   (a) Can you identify the domain name of the server? It might be somewhere within the packet.

   The server is the same, but this time with HTTPS: `https://www.bendigobank.com.au/`. This can be seen under `Client Hello` message → Secure Socket Layer → TLSv1 Record Layer: Handshake Protocol: Client Hello → Handshake Protocol: Client Hello → Extension: server_name

   (b) Which protocols are used on application layer?

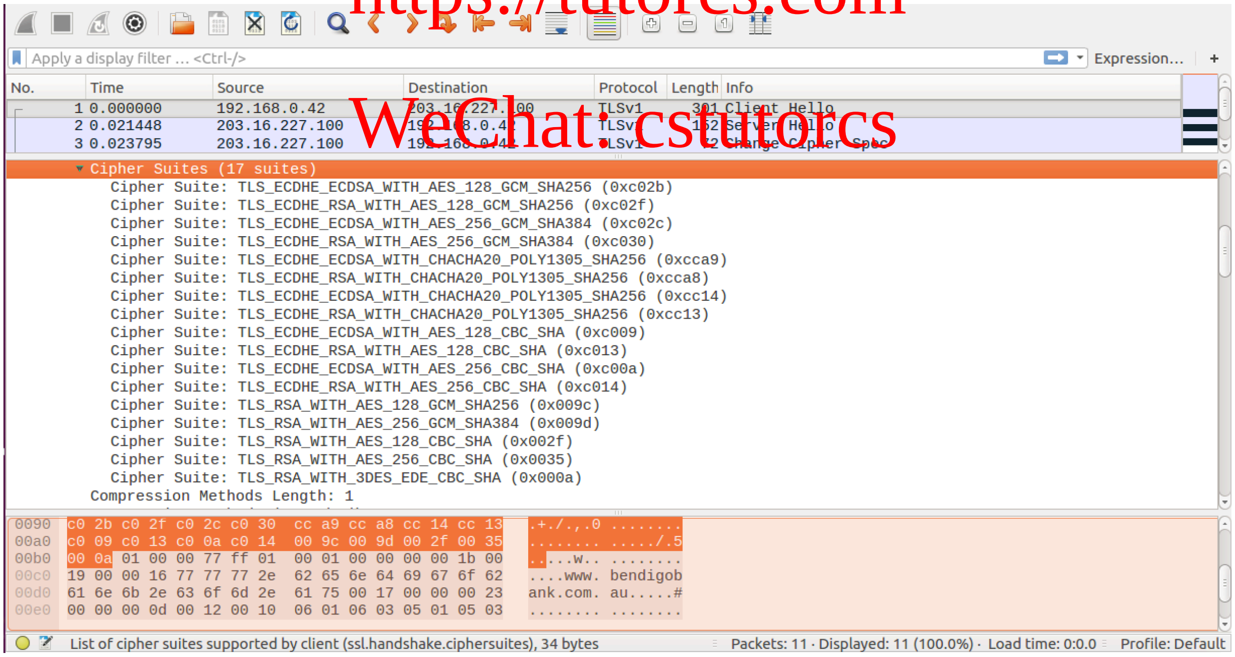
   Based on the TCP port used, 443, the traffic is HTTPS.

   (c) Identify which version of the security protocol is used. Is this considered to be a secure version?

   It uses TLSv1.0. If you look into the packets, you only find encrypted content. However, students should try to get some information on TLS 1.0 on the Internet and they will find that it is outdated and should no longer be used.

   (d) Find the `Client Hello` packet sent from client. What cryptographic functions are supported by the client?

   The supported `Cipher Suite` by the client.



```
No.  Time        Source            Destination       Protocol  Length  Info
  1  0.000000    192.168.0.42      203.16.227.100    TLSv1     301     Client Hello
  2  0.021448    203.16.227.100    192.168.0.42      TLSv1     152     Server Hello
  3  0.023795    203.16.227.100    192.168.0.42      TLSv1     72      Change Cipher Spec
```

```
 ▾ Cipher Suites (17 suites)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
     Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14)
     Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
     Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
     Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
     Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
     Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
     Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
   Compression Methods Length: 1
```

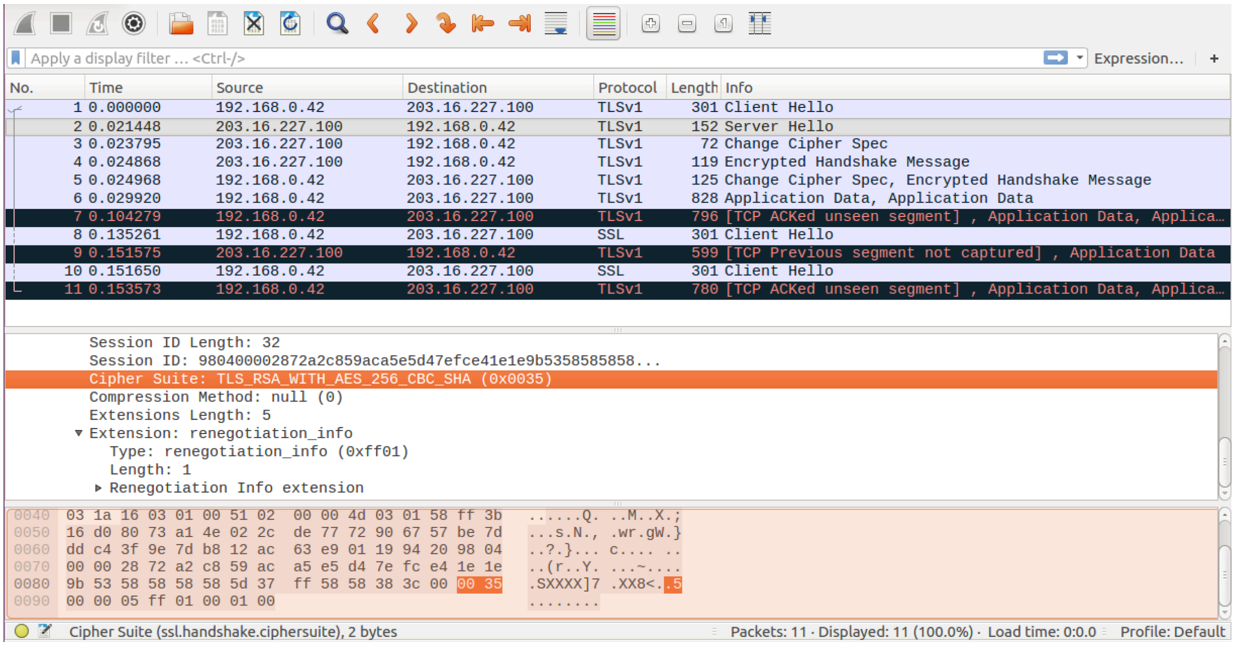
```
0090  c0 2b c0 2f c0 2c c0 30  cc a9 cc a8 cc 14 cc 13   .+./.,.0 ........
00a0  c0 09 c0 13 c0 0a c0 14  00 9c 00 9d 00 2f 00 35   ........ ...../.5
00b0  00 0a 01 00 00 77 ff 01  00 01 00 00 00 00 1b 00   .....w.. ........
00c0  19 00 00 16 77 77 77 2e  62 65 6e 64 69 67 6f 62   ....www. bendigob
00d0  61 6e 6b 2e 63 6f 6d 2e  61 75 00 17 00 00 00 23   ank.com. au.....#
00e0  00 00 00 0d 00 12 00 10  06 01 06 03 05 01 05 03   ........ ........
```

List of cipher suites supported by client (ssl.handshake.ciphersuites), 34 bytes

Packets: 11 · Displayed: 11 (100.0%) · Load time: 0:0.0 Profile: Default

   (e) Find the `Server Hello` packet sent from server in response. What `Cipher Suite` the server agrees to use?

   The server agrees to



```
No. Time       Source          Destination     Protocol Length Info
  1 0.000000   192.168.0.42    203.16.227.100  TLSv1       301 Client Hello
  2 0.021448   203.16.227.100  192.168.0.42    TLSv1       152 Server Hello
  3 0.023795   203.16.227.100  192.168.0.42    TLSv1        72 Change Cipher Spec
  4 0.024868   203.16.227.100  192.168.0.42    TLSv1       119 Encrypted Handshake Message
  5 0.024968   192.168.0.42    203.16.227.100  TLSv1       125 Change Cipher Spec, Encrypted Handshake Message
  6 0.029920   192.168.0.42    203.16.227.100  TLSv1       828 Application Data, Application Data
  7 0.104279   192.168.0.42    203.16.227.100  TLSv1       796 [TCP ACKed unseen segment] , Application Data, Applica…
  8 0.135261   192.168.0.42    203.16.227.100  SSL         301 Client Hello
  9 0.151575   203.16.227.100  192.168.0.42    TLSv1       599 [TCP Previous segment not captured] , Application Data
 10 0.151650   192.168.0.42    203.16.227.100  SSL         301 Client Hello
 11 0.153573   192.168.0.42    203.16.227.100  TLSv1       780 [TCP ACKed unseen segment] , Application Data, Applica…
```

```
        Session ID Length: 32
        Session ID: 980400002872a2c859aca5e5d47efce41e1e9b5358585858...
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Compression Method: null (0)
        Extensions Length: 5
      ▾ Extension: renegotiation_info
          Type: renegotiation_info (0xff01)
          Length: 1
        ▸ Renegotiation Info extension
```

```
0040  03 1a 16 03 01 00 51 02  00 00 4d 03 01 58 ff 3b   ......Q. ..M..X.;
0050  16 d0 80 73 a1 4e 02 2c  de 77 72 90 67 57 be 7d   ...s.N., .wr.gW.}
0060  dd c4 3f 9e 7d b8 12 ac  63 e9 01 19 94 20 98 04   ..?.}... c..... ..
0070  00 00 28 72 a2 c8 59 ac  a5 e5 d4 7e fc e4 1e 1e   ..(r..Y. ...~....
0080  9b 53 58 58 58 58 5d 37  ff 58 58 38 3c 00 00 35   .SXXXX]7 .XX8<..5
0090  00 00 05 ff 01 00 01 00                            ........
```

Cipher Suite (ssl.handshake.ciphersuite), 2 bytes   Packets: 11 · Displayed: 11 (100.0%) · Load time: 0:0.0   Profile: Default

(f) What is the purpose of the Change Cipher Spec?

The Change Cipher Spec message triggers the TLS protocol to start using the negotiated cryptographic algorithms. Both sides must send this message for TLS to start protecting the traffic. The client sends its Change Cipher Spec message in packet 5.

3. Open /srv/fit2093files/fit2093lab/Example3.pcap in Wireshark.

(a) Can you identify the domain name of the server?

This time it is another server, but also using HTTP: http://commbank.com.au.

(b) What is different to the other two examples?

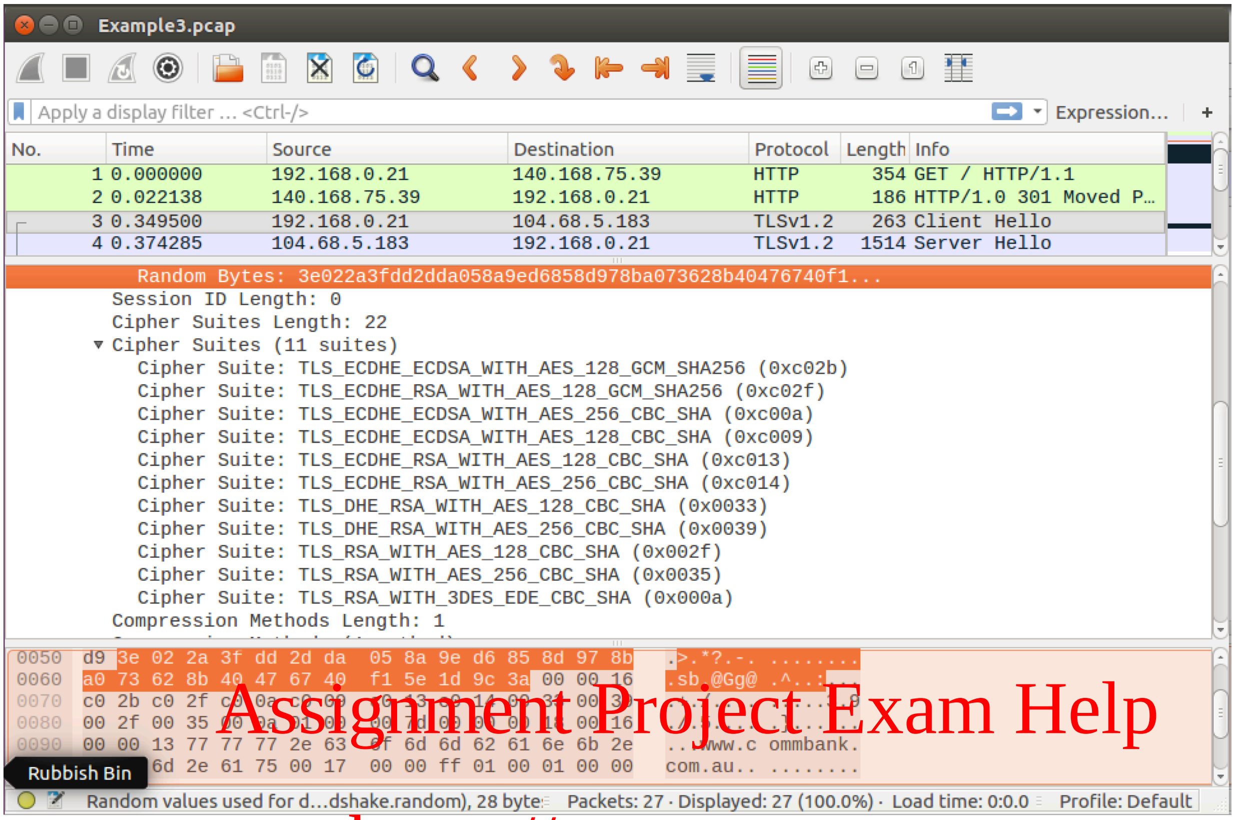
However, you will first see an error and then see that the get request was diverted to HTTPS. Thus, the traffic automatically switches from HTTP to HTTPS (the server forcefully redirect the traffic from HTTP to HTTPS).

(c) Which protocols are used? Are these considered to be secure?

It uses TLS version 1.2, which is state of the art and considered to be secure. (Recently TLS v1.3 was released but it is not widely adapted yet)

(d) Compare the supported client Cipher Suite in Client Hello in Example3.pcap with the supported Cipher Suite in Client Hello in Example3.pcap in Example2.pcap. What is different?

The supported Cipher Suite by the client.



```
No.  Time      Source         Destination    Protocol Length Info
   1 0.000000  192.168.0.21   140.168.75.39  HTTP      354 GET / HTTP/1.1
   2 0.022138  140.168.75.39  192.168.0.21   HTTP      186 HTTP/1.0 301 Moved P…
   3 0.349500  192.168.0.21   104.68.5.183   TLSv1.2   263 Client Hello
   4 0.374285  104.68.5.183   192.168.0.21   TLSv1.2  1514 Server Hello
```

```
        Random Bytes: 3e022a3fdd2dda058a9ed6858d978ba073628b40476740f1...
        Session ID Length: 0
        Cipher Suites Length: 22
      ▾ Cipher Suites (11 suites)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
        Compression Methods Length: 1
```

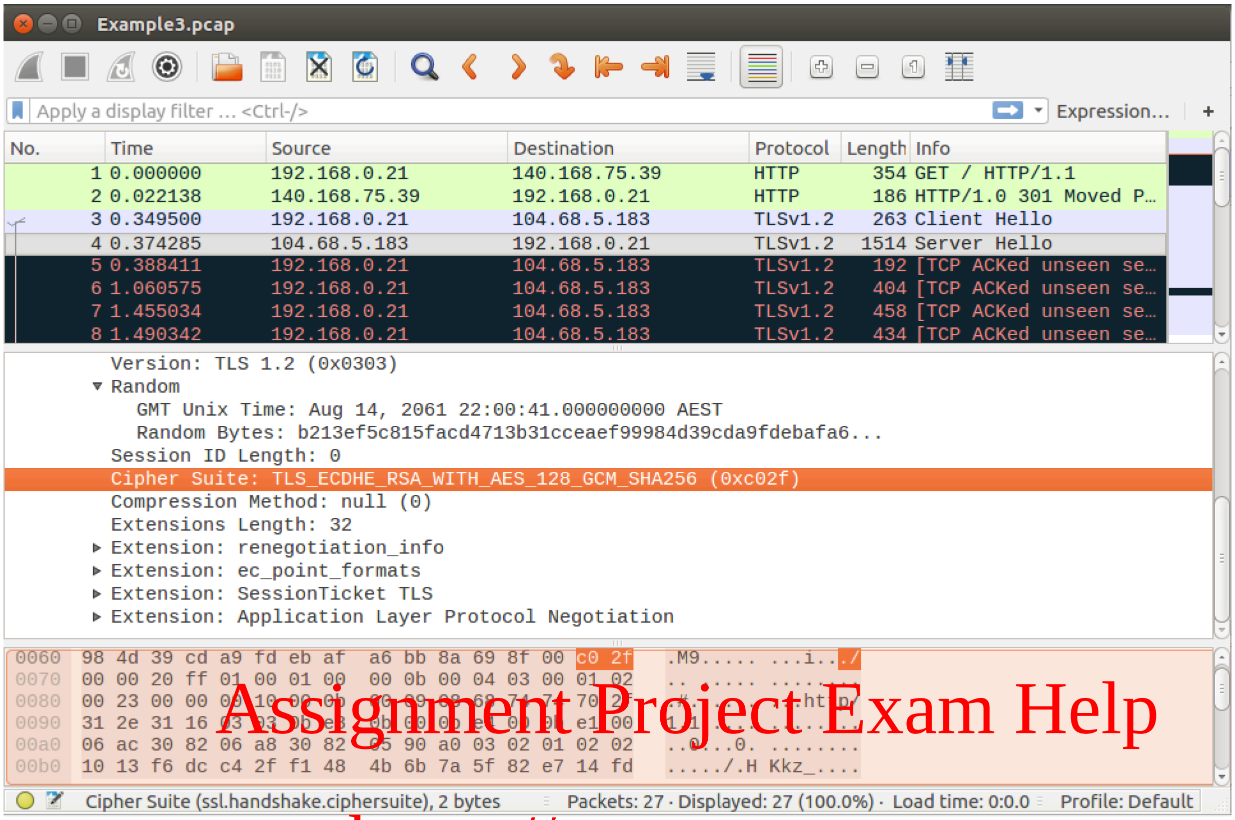
```
0050  d9 3e 02 2a 3f dd 2d da  05 8a 9e d6 85 8d 97 8b   .>.*?.-. ........
0060  a0 73 62 8b 40 47 67 40  f1 5e 1d 9c 3a 00 00 16   .sb.@Gg@ .^..:...
0070  c0 2b c0 2f c0 0a c0 09  c0 13 c0 14 00 33 00 39   .+./.... .....3.9
0080  00 2f 00 35 00 0a 01 00  00 7d 00 00 00 18 00 16   ./.5.... .}......
0090  00 00 13 77 77 77 2e 63  6f 6d 6d 62 61 6e 6b 2e   ...www.c ommbank.
         6d 2e 61 75 00 17  00 00 ff 01 00 01 00 00   com.au.. ........
```

```
Random values used for d…dshake.random), 28 byte   Packets: 27 · Displayed: 27 (100.0%) · Load time: 0:0.0   Profile: Default
```

This client supports 11 Cipher Suite compared to 17 in previous example. The noticeable differences are the lack of support for CHACHA20 symmetric cipher and SHA384 hash function.

(e) What Cipher Suite the server agrees to use?

The server agrees to use:



Example3.pcap

```
No.  Time        Source          Destination     Protocol  Length Info
   1 0.000000    192.168.0.21    140.168.75.39   HTTP        354 GET / HTTP/1.1
   2 0.022138    140.168.75.39   192.168.0.21    HTTP        186 HTTP/1.0 301 Moved P…
   3 0.349500    192.168.0.21    104.68.5.183    TLSv1.2     263 Client Hello
   4 0.374285    104.68.5.183    192.168.0.21    TLSv1.2    1514 Server Hello
   5 0.388411    192.168.0.21    104.68.5.183    TLSv1.2     192 [TCP ACKed unseen se…
   6 1.060575    192.168.0.21    104.68.5.183    TLSv1.2     404 [TCP ACKed unseen se…
   7 1.455034    192.168.0.21    104.68.5.183    TLSv1.2     458 [TCP ACKed unseen se…
   8 1.490342    192.168.0.21    104.68.5.183    TLSv1.2     434 [TCP ACKed unseen se…
```

```
        Version: TLS 1.2 (0x0303)
      ▾ Random
          GMT Unix Time: Aug 14, 2061 22:00:41.000000000 AEST
          Random Bytes: b213ef5c815facd4713b31cceaef99984d39cda9fdebafa6...
        Session ID Length: 0
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
        Compression Method: null (0)
        Extensions Length: 32
      ▸ Extension: renegotiation_info
      ▸ Extension: ec_point_formats
      ▸ Extension: SessionTicket TLS
      ▸ Extension: Application Layer Protocol Negotiation
```

```
0060  98 4d 39 cd a9 fd eb af  a6 bb 8a 69 8f 00 c0 2f   .M9.....  ...i../
0070  00 00 20 ff 01 00 01 00  00 0b 00 04 03 00 01 02   .. .....  ........
0080  00 23 00 00 00 10 00 0b  00 09 08 68 74 74 70 2f   .#......  ...http/
0090  31 2e 31 16 03 03 0b e8  0b 00 0b e4 00 0b e1 00   1.1.....  ........
00a0  06 ac 30 82 06 a8 30 82  05 90 a0 03 02 01 02 02   ..0...0.  ........
00b0  10 13 f6 dc c4 2f f1 48  4b 6b 7a 5f 82 e7 14 fd   ...../.H  Kkz_....
```

Cipher Suite (ssl.handshake.ciphersuite), 2 bytes   Packets: 27 · Displayed: 27 (100.0%) · Load time: 0:0.0   Profile: Default

(f) Using the RFC document for TLSv1.2 (RFC5246) explain what cryptographic algorithms are used in the agreed `Cipher Suite`.

The `Cipher Suite` is discussed in A.5 section of the document (Appendix 5).
`https://tools.ietf.org/html/rfc5246#appendix-A.5`
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 with value c02f.

The Eliptic Curve algorithms for TLS is defined in RFC4492. The algorithm ECDHE_RSA is discussed in section 2.4:
`https://tools.ietf.org/html/rfc4492#section-2.4`

This key exchange algorithm is the same as ECDHE_ECDSA except that the server's certificate MUST contain an RSA public key authorized for signing, and that the signature in the ServerKeyExchange message must be computed with the corresponding RSA private key. The server certificate MUST be signed with RSA.

The section 3 of RFC5289 contains the code for Eliptic Curve cipher suites that support AES in GCM mode.
`https://tools.ietf.org/html/rfc5289#section-3`

RFC 4492 describes elliptic curve cipher suites for Transport Layer Security (TLS). However, all those cipher suites use HMAC-SHA-1 as their Message Authentication Code (MAC) algorithm. This document describes sixteen new cipher suites for TLS that specify stronger MAC algorithms. Eight use Hashed Message Authentication Code (HMAC) with SHA-256 or SHA-384, and eight use AES in Galois Counter Mode (GCM).

3.2 Certificates for HTTPS/TLS

1. Use a web browser on your **own device** (**not** in the VM) to open a webpage that supports TLS. For example `https://commbank.com.au/` Click on the lock shown on the left from the address bar.

   (a) Who is the issuer of the certificate and how long is it valid?

   (b) What is used for key exchange and which cipher suite is used during transport?

   Entrust, Inc. has issued the certificate. Expires on May 26, 2022. TLS 1.2 Key Exchange: `ECDHE_RSA` This is Elliptic Curve Diffie-Hellman, signed with RSA. Cipher Suite: `AES_256_GCM` This is 256 bit AES used in Galois/Counter Mode.

2. Can you find the list of all certification authorities that are installed in your web browser? Can you find some revoked certificates? (Hint: Look in settings under advanced settings)

   Just search "Certificate" in the search box of the web browser settings and open the dialog of certificates. You may find a few revoked certificates (in newer versions of Chrome or Firefox those certificates may already get removed). If someone is interested in the story behind this, google for UTN-USERFirst-Hardware.

3. This article shows a few of the main issues with certificates:

   `https://arstechnica.com/information-technology/2017/03/google-takes-symantec-to-the-woodshed-for-mis-issuing-30000-https-certs/`

   (a) Read the article.

   (b) What are the different entities (companies, software, etc.) that need to be trusted to actually trust a certificate?

   (c) Draw a diagram showing the process of certificate issuing and checking in the browser. It should contain entities (companies, devices, software) used for producing the different certificates and checking it. Assume that the server's certificate is directly signed with the issuer's root certificate.

   Entities are the issuer of the certificate (the owner of the root certificate), software/hardware needed to produce the root certificate and the server's certificate, the company deciding which root certificates to bundle with the browser, the browser for checking, the server storing the secret key, the owner of the server, the client's PC.

3.3 Additional Task: Packet Capturing

Use Wireshark and try to capture the HTTPS haneshake messages on your own devices.