Practical Networking

Practical TLS – Lab Guide

A deep dive into SSL and TLS – the protocols that secure the Internet

Lab 6.3 – Inspecting TLS Handshake Extensions

In this lab you will be provided with a packet capture of three different TLS sessions, and you will explore the TLS Handshake Extensions in each.

Create working directory and acquire lab files

1. Navigate to the folder you created in Lab 0.0 named Practical-TLS
   - `$ cd Practical-TLS`

2. Create and navigate to a new directory for Lab 6.3 files:
   - `$ mkdir “LAB6.3 – Inspecting TLS Extensions”`
   - `$ cd “LAB6.3 – Inspecting TLS Extensions”`

3. Download *one* of the following lab files and place it in the newly created folder:
   - `LAB 6.3 Files - Inspecting TLS Extensions - Practical TLS.tar.gz`
   - `LAB 6.3 Files - Inspecting TLS Extensions - Practical TLS.zip`

   The lab files are in `.zip` or `.tar.gz` format – the contents within are identical, use whichever file is easier for you

4. Unzip or Untar the file you downloaded into your LAB 6.3 working directory:
   - For the TAR.GZ file, you will use the Linux command: `tar -xvzf “LAB 6.3 ...`
   - For the ZIP file, it should be something like right clicking the file and selecting “Extract All…”

5. When finished, you should have the following **1 file** in the LAB 6.3 directory:

```
Practical TLS – Handshake Extensions.pcap
```

Inspecting TLS Handshakes Extensions

The packet capture file you downloaded (`Practical TLS – Handshake Extensions.pcap`) includes three TLS sessions:

1- Full TLS Handshake which includes OCSP Stapling and SNI extensions
2- Full TLS Handshake which includes TLS Session Tickets and SNI extensions
3- Resumed TLS Handshake using TLS Session Tickets

Your task is to explore each of these TLS sessions in Wireshark and validate what you learned in the course lessons. Some guiding questions have been included.

Consider using Wireshark's conversation colorizing feature to distinguish the three sessions. Click a packet and press `CTRL+[1-9]` (`CMD+[1-9]` for MAC). Each number 1-9 is a different color you can use. `CTRL/CMD + Space` will reset the colorizing. Use a different color for each of the three sessions in this packet capture.

Also consider applying the display filter of `tls` to limit the display to just TLS packets.

Write down your answers to the questions below and compare them against the answer key at the end of this lab guide.

TLS Session #1 (Packets 1 – 22)

1. Find the extension in which the Client requested a specific website domain name
   - What domain name certificate is being requested?
2. Find the extension in which the Server confirmed support for Server Name Indication (SNI)
   - Does the server support SNI?
3. Find the extension in which the Client requested the Certificate Status for this TLS Session
4. Find the extension in which the Server confirmed its ability to provide the Certificate Status
5. Find the Record in which the Server provided the Certificate Status
   - What was the Certificate Status?
   - What was the time stamp for when this Certificate Status was acquired?
   - Did the Certificate Status include a Signature?
   - Who signed the Certificate?

TLS Session #2 (Packets 23 – 44)

6. Find the extension in which the Client requested a specific website domain name
   - What domain name certificate is being requested?
7. Find the extension in which the Server confirmed support for Server Name Indication (SNI)
   - Does the server support SNI?
8. What was the Session ID sent by the Client?
9. What was the Session ID sent by the Server?
10. Find the extension in which the Client indicated support for TLS Session Tickets
    - Was anything included inside this Extension?
11. Find the extension in which the Server indicated support for TLS Session Tickets
    - Was anything included inside this Extension?
12. Find the Record in which the Server provided the actual TLS Session Ticket
    - What are the first few digits of the included Session Ticket?
    - What key was used to encrypt this Session Ticket?
    - Who has access to that Key?
13. How many Handshake records were sent for this TLS negotiation?
14. How many Round Trips were necessary to complete this TLS Handshake?

TLS Session #3 (Packets 45 – 58)

15. Find the extension in which the Client indicated support for TLS Session Tickets
    - What was included inside this Extension?
    - What are the first few digits of the included data in this Extension?
      (note, you will have to look at the Packet Bytes pane to answer this question)
16. Find the extension in which the Server indicated support for TLS Session Tickets
    - Was it included in the Server Hello?
17. What was the Session ID sent by the Client?
18. What was the Session ID sent by the Server?
19. How many Handshake records were sent for this TLS negotiation?
20. How many Round Trips were necessary to complete this TLS Handshake?

This page intentionally left blank

Answers for the questions above are on the next page. Check your answers.

If you are unclear about a question and answer, ask about it in the Discord server.

Answer Key

Answers – TLS Session #1 (Packets 1 – 22)

1. Find the extension in which the Client requested a specific website domain name
   Client Hello, Packet #4, Extension: server_name
   - What domain name certificate is being requested?
     example.org
2. Find the extension in which the Server confirmed support for Server Name Indication (SNI)
   It would be in the Server Hello, but this server does not support SNI
   - Does the server support SNI?
     No
3. Find the extension in which the Client requested the Certificate Status for this TLS Session
   Client Hello, Packet #4, Extension: status_request
4. Find the extension in which the Server confirmed its ability to provide the Certificate Status
   Server Hello, Packet #6, Extension: status_request
5. Find the Record in which the Server provided the Certificate Status
   Certificate Status, Packet #10
   - What was the Certificate Status?
     good
   - What was the time stamp for when this Certificate Status was acquired?
     2021-02-23 06:15:02 (UTC)
   - Did the Certificate Status include a Signature?
     Yes
   - Who signed the Certificate?
     The CA which provided the Certificate Status to the Server. Note: the actual CA is not obviously identified in the Certificate Status. The Client would use the field `issuerKeyHash` and compare it with the Certificate chain's x509v3 Authority Key Identifier extension to identify exactly which CA, and therefore which Public Key to use to validate this signature.

Answers – TLS Session #2 (Packets 23 – 44)

6. Find the extension in which the Client requested a specific website domain name
   Client Hello, Packet #26, Extension: server_name
   - What domain name certificate is being requested?
     facebook.com

7. Find the extension in which the Server confirmed support for Server Name Indication (SNI)
   Server Hello, Packet #28, Extension: server_name
   - Does the server support SNI?
     Yes

8. What was the Session ID sent by the Client?
   No Session ID was provided – this is identical to a Session ID of all zeros

9. What was the Session ID sent by the Server?
   No Session ID was provided – this is identical to a Session ID of all zeros

10. Find the extension in which the Client indicated support for TLS Session Tickets
    Client Hello, Packet #26, Extension: session_ticket
    - Was anything included inside this Extension?
      No, it merely indicates support for session tickets

11. Find the extension in which the Server indicated support for TLS Session Tickets
    Server Hello, Packet #28, Extension: session_ticket
    - Was anything included inside this Extension?
      No, it merely indicates support for session tickets

12. Find the Record in which the Server provided the actual TLS Session Ticket
    New Session Ticket, Packet #35
    - What are the first few digits of the included Session Ticket?
      bbd4ff32545315b1f686a4c4145ce550082c586dafdf9b2419c90de3b4b5e4d092d …
    - What key was used to encrypt this Session Ticket?
      Session Ticket Encryption Key (STEK)
    - Who has access to that Key?
      Only the Server or Servers responsible for hosting this website

13. How many Handshake records were sent for this TLS negotiation?
    9 – remember, Change Cipher Spec *is not* a Handshake record, and Encrypted Handshake Message is the Finished record, which *is* a Handshake record

14. How many Round Trips were necessary to complete this TLS Handshake?
    2

Answers – TLS Session #3 (Packets 45 – 58)

15. Find the extension in which the Client indicated support for TLS Session Tickets
    Client Hello, Packet #48, Extension: session_ticket
    - What was included inside this Extension?
      The actual Session Ticket sent by the Server in Packet # 35
    - What are the first few digits of the included data in this Extension?
      (note, you will have to look at the Packet Bytes pane to answer this question)
      bbd4ff32545315b1 …
      See image at the end of this section to understand how this was extracted

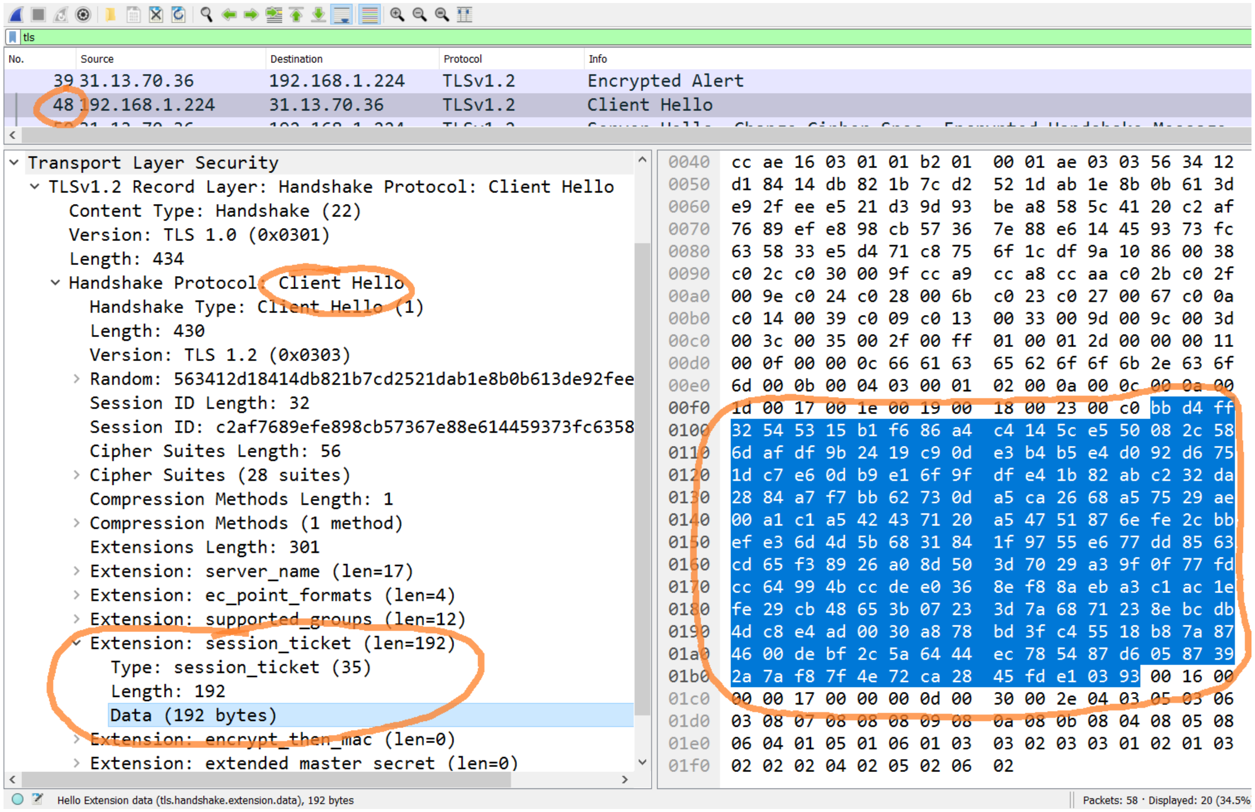
16. Find the extension in which the Server indicated support for TLS Session Tickets
    It would have been in the Server Hello, Packet #50
    - Was it included in the Server Hello?
      No – but none the less, the Session Ticket was used to perform an abbreviated session. This is verified because the Session ID's for this session from the Client and Server are identical.

17. What was the Session ID sent by the Client?
    c2af7689efe898cb57367e88e614459373fc635833e5d471c8756f1cdf9a1086

18. What was the Session ID sent by the Server?
    c2af7689efe898cb57367e88e614459373fc635833e5d471c8756f1cdf9a1086

19. How many Handshake records were sent for this TLS negotiation?
    4

20. How many Round Trips were necessary to complete this TLS Handshake?
    1



```
tls

No.   Source          Destination     Protocol  Info
   39 31.13.70.36     192.168.1.224   TLSv1.2   Encrypted Alert
   48 192.168.1.224   31.13.70.36     TLSv1.2   Client Hello
   50 31.13.70.36     192.168.1.224   TLSv1.2   Server Hello, Change Cipher Spec, Encrypted Handshake Message
```

```
v Transport Layer Security
  v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 434
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 430
        Version: TLS 1.2 (0x0303)
      > Random: 563412d18414db821b7cd2521dab1e8b0b613de92fee
        Session ID Length: 32
        Session ID: c2af7689efe898cb57367e88e614459373fc6358
        Cipher Suites Length: 56
      > Cipher Suites (28 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 301
      > Extension: server_name (len=17)
      > Extension: ec_point_formats (len=4)
      > Extension: supported_groups (len=12)
      v Extension: session_ticket (len=192)
          Type: session_ticket (35)
          Length: 192
          Data (192 bytes)
      > Extension: encrypt_then_mac (len=0)
      > Extension: extended master secret (len=0)
```

```
0040  cc ae 16 03 01 01 b2 01  00 01 ae 03 03 56 34 12
0050  d1 84 14 db 82 1b 7c d2  52 1d ab 1e 8b 0b 61 3d
0060  e9 2f ee e5 21 d3 9d 93  be a8 58 5c 41 20 c2 af
0070  76 89 ef e8 98 cb 57 36  7e 88 e6 14 45 93 73 fc
0080  63 58 33 e5 d4 71 c8 75  6f 1c df 9a 10 86 00 38
0090  c0 2c c0 30 00 9f cc a9  cc a8 cc aa c0 2b c0 2f
00a0  00 9e c0 24 c0 28 00 6b  c0 23 c0 27 00 67 c0 0a
00b0  c0 14 00 39 c0 09 c0 13  00 33 00 9d 00 9c 00 3d
00c0  00 3c 00 35 00 2f 00 ff  01 00 01 2d 00 00 00 11
00d0  00 0f 00 00 0c 66 61 63  65 62 6f 6f 6b 2e 63 6f
00e0  6d 00 0b 00 04 03 00 01  02 00 0a 00 0c 00 0a 00
00f0  1d 00 17 00 1e 00 19 00  18 00 23 00 c0 bb d4 ff
0100  32 54 53 15 b1 f6 86 a4  c4 14 5c e5 50 08 2c 58
0110  6d af df 9b 24 19 c9 0d  e3 b4 b5 e4 d0 92 d6 75
0120  1d c7 e6 0d b9 e1 6f 9f  df e4 1b 82 ab c2 32 da
0130  28 84 a7 f7 bb 62 73 0d  a5 ca 26 68 a5 75 29 ae
0140  00 a1 c1 a5 42 43 71 20  a5 47 51 87 6e fe 2c bb
0150  ef e3 6d 4d 5b 68 31 84  1f 97 55 e6 77 dd 85 63
0160  cd 65 f3 89 26 a0 8d 50  3d 70 29 a3 9f 0f 77 fd
0170  cc 64 99 4b cc de e0 36  8e f8 8a eb a3 c1 ac 1e
0180  fe 29 cb 48 65 3b 07 23  3d 7a 68 71 23 8e bc db
0190  4d c8 e4 ad 00 30 a8 78  bd 3f c4 55 18 b8 7a 87
01a0  46 00 de bf 2c 5a 64 44  ec 78 54 87 d6 05 87 39
01b0  2a 7a f8 7f 4e 72 ca 28  45 fd e1 03 93 00 16 00
01c0  00 00 17 00 00 00 0d 00  30 00 2e 04 03 05 03 06
01d0  03 08 07 08 08 08 09 08  0a 08 0b 08 04 08 05 08
01e0  06 04 01 05 01 06 01 03  03 02 03 03 01 02 01 03
01f0  02 02 02 04 02 05 02 06  02
```

```
Hello Extension data (tls.handshake.extension.data), 192 bytes        Packets: 58 · Displayed: 20 (34.5%
```

Image to assist with Question #15