Practical Networking

Practical TLS – Lab Guide

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

Lab 6.2 – Inspecting TLS Handshake Variants

In this lab you will be inspecting a few different TLS handshakes in Wireshark. Each of these handshakes are a slight variation to the Handshake we explored in LAB 6.1.

This lab uses freely available software known as Wireshark. Wireshark lets you capture packets sent to or from your computer, and view saved packet captures.

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 6.2 files:
   - `$ mkdir "LAB6.2 – Inspecting TLS Handshake Variants"`
   - `$ cd "LAB6.2 – Inspecting TLS Handshake Variants"`

3. Download *one* of the following lab files and place it in the newly created folder:
   - **LAB 6.2 Files - Inspecting TLS Handshake Variants - Practical TLS.tar.gz**
   - **LAB 6.2 Files - Inspecting TLS Handshake Variants - 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 6.2 working directory:
   - For the TAR.GZ file, you will use the Linux command: `tar -xvzf "LAB 6.2 ...`
   - 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 **3 files** in the LAB 6.2 directory:

```
Practical TLS - DHE KX.pcap
Practical TLS - Session Resumption - Session ID.pcap
Practical TLS - Mutual Authentication.pcap
```

Inspecting an Ephemeral Diffie-Hellman TLS Handshake

The basic handshake we explored thoroughly in LAB 6.1 used RSA as its Key Exchange (KX) protocol. The TLS handshake for the RSA KX is different from a TLS handshake with an Ephemeral DH KX. The primary differences are as follows:

In an RSA Key Exchange:
- The Client generates the Pre Master Secret without any contribution from the Server
- The Client encrypts this value with the Public Key in the Server Certificate
    - Only the true owner of the Certificate can extract the real Pre Master Secret

In an Ephemeral Diffie-Hellman Key Exchange (ECDHE, DHE):
- Both the Client and the Server contribute DH Public Values to create the Shared Secret
- The Server signs their DH Public Value with the Private Key from the Server Certificate
    - Only the true owner can create a Signature that can be verified with the Certificate's Public Key

These concepts are what you will prove to yourself by exploring the packet capture of an Ephemeral Diffie-Hellman TLS Handshake.

1. Open the file **`Practical TLS - DHE KX.pcap`** in Wireshark
    - (optional) If you are familiar with TCP, try to locate the TCP three-way-handshake
2. Filter the display to only show you the TLS conversation
    - In the display filter, type **`tls`** and press enter
3. Answer the questions below using what you learned from LAB 6.1.
    - It might be helpful to open the RSA handshake packet capture you downloaded in LAB 6.1 for comparison purposes (file: **`Practical TLS - RSA KX.pcap`**)

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

1. Map out the Sequence of Records that are exchanged by the Client and the Server
2. What was the negotiated Cipher Suite selected by the Client and Server?
3. What Record exists in this DHE KX Handshake that did not exist in the RSA KX handshake you explored in LAB 6.1?
4. What value is sent in this Record?
5. The Diffie-Hellman Key exchange starts with both parties agreeing upon a Prime number and a Generator (P and G). Can you find where the P and G values are established?
    - What are the first few characters of the P and G values from the Diffie-Hellman exchange?
6. What is the DH Public Key sent by the Server? (first few characters)
7. What additional value is sent along with the P, G, and DH Public Key values?
    - What key was used to create this value?

8. How is the Client Key Exchange Record in the DHE capture different from the same record in the RSA capture?
9. What is the DH Public Key sent by the Client? (first few characters)
10. Knowing the DH Public Keys sent by the Client and the Server, as well as the P and G value, do you have everything you need to recreate the Shared Secret?
    - If yes, how would you combine the values you know to create the Shared Secret?
    - If no, what value(s) are you missing in order to create the Shared Secret?
11. Packet #17 is an Alert
    - Can you determine the Severity or Description of this Alert?
    - Why or why not?
12. Does this TLS Handshake provide Forward Secrecy?
    - What are the implications of providing or not providing Forward Secrecy?

Inspecting a Session Resumption TLS Handshake

The full TLS handshake involves:

- Exchanging at least 9 different Records
- Two full Round Trips before data can be sent (2RTT)
- Asymmetric math
- Sending the complete certificate chain.

TLS allows a Client and Server who have recently communicated to perform an abbreviated handshake by reusing values from a previous session. This spares the Client and Server from doing all the additional processing the full handshake requires. This is known as Session Resumption.

The abbreviated TLS Handshake for Session Resumption involves:

- Exchanging only 6 records
- One Round Trip before data can be sent (1RTT)
- No Asymmetric Math
- No sending of any certificates

In this task you will be studying a TLS Handshake using Session Resumption and comparing it to a full TLS handshake.

1. Open the file **`Practical TLS - Session Resumption - Session ID.pcap`** in Wireshark
   - Two conversations are occurring in this packet capture
   - (optional) If you are familiar with TCP, try to locate *both* TCP three-way-handshake
2. Filter the display to only show you the TLS conversation
   - In the display filter, type **`tls`** and press enter
3. Wireshark lets you apply a temporary color to network conversations:
   - Select the first Client Hello, then press CTRL+1 (CMD+1 for Mac)
   - Select the second Client Hello, then press CTRL+2 (CMD+2 for Mac)
   - This will colorize the two conversations that are occurring in this capture
     - CTRL+Space / CMD+Space will reset the color to default.

   This step is optional, it will simply make it easier to visually see both session.
4. Answer the questions below using what you learned from LAB 6.1.
   - It might be helpful to open the RSA handshake packet capture you downloaded in LAB 6.1 for comparison purposes (file: **`Practical TLS – RSA KX.pcap`**)

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

1. In the Original session, excluding Application Data and Alert Records…
   - How many Records were sent from Client to Server?
   - How many Records were sent from Server to Client?
2. In the Resumed session, excluding Application Data and Alert Records…
   - How many Records were sent from Client to Server?
   - How many Records were sent from Server to Client?
3. What records existed in the Original session, that did NOT exist in the Resumed session?
4. What was the Record Length for each of the records from the previous question?
5. In the Original session, how many Round Trips were necessary before the Client could send Application Data?
6. In the Resumed session, how many Round Trips were necessary before the Client could send Application Data?
7. List the first few digits of the Session ID for the…
   - Original session Client Hello
   - Original session Server Hello
   - Resumed session Client Hello
   - Resumed session Server Hello
8. What Cipher Suite was selected by the Server in both sessions?
9. Does either TLS Handshake provide Forward Secrecy?
   - What are the implications of providing or not providing Forward Secrecy?

Inspecting a Mutual Authentication TLS Handshake

In a typical TLS / SSL Session, only the Server is authenticated – which is to say, only the Server presents a Certificate to prove its identity. Clients are validated through other means external to TLS itself (such as a user name or password).

TLS/SSL does provide a process which requires both the Client and the Server to provide an Identity Certificate. This process is referred to as Mutual Authentication.

1. Open the file **`Practical TLS - Mutual Authentication.pcap`** in Wireshark
   - (optional) If you are familiar with TCP, try to locate the TCP three-way-handshake
2. Filter the display to only show you the TLS conversation
   - In the display filter, type **`tls`** and press enter
3. Answer the questions below using what you learned from LAB 6.1.
   - It might be helpful to open the RSA handshake packet capture you downloaded in LAB 6.1 for comparison purposes (file: **`Practical TLS - RSA KX.pcap`**)

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

1. Compared to the packet capture from LAB 6.1, which additional records exist in the Mutual Authentication capture?
2. In the Certificate Request record...
   - What Certificate Types did the Server Request?
   - How many Signature Hash Algorithms does the Server accept?
   - What are the Common Name(s) of the Certificate Authorities which the Server accepts?
3. In the Client's Certificate record...
   - How many Certificates did the Client send?
   - What is the Common Name in the Subject fields for each Certificate sent by the Client?
4. What value is being sent in the Client Key Exchange record?
   - What is the ultimate purpose of this value?
5. What value is being sent in the Certificate Verify record?
   - What is the ultimate purpose of this value?
6. Does this TLS Handshake provide Forward Secrecy?
   - What are the implications of providing or not providing Forward Secrecy?

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 for Ephemeral Diffie Hellman packet capture Questions

1. Map out the Sequence of Records that are exchanged by the Client and the Server

Client		Server
Client Hello	-->	
		<-- Server Hello
		<-- Certificate
		<-- Server Key Exchange
		<-- Server Hello Done
Client Key Exchange	-->	
Change Cipher Spec	-->	
Encrypted Handshake Message (Finished)	-->	
		<-- Change Cipher Spec
		<-- Encrypted Handshake Message (Finished)
Application Data	-->	
		<-- Application Data
Encrypted Alert	-->	

2. What was the negotiated Cipher Suite selected by the Client and Server?
   TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x0067)
3. What Record exists in this DHE KX Handshake that did not exist in the RSA KX handshake you explored in LAB 6.1?
   Server Key Exchange
4. What value is sent in this Record?
   Server's DH Public Key (and other DH parameters, see next question)
5. The Diffie-Hellman Key exchange starts with both parties agreeing upon a Prime number and a Generator (P and G). Can you find where the P and G values are established?
   The server sends the P and G value in the Server Key Exchange
   - What are the first few characters of the P and G values from the Diffie-Hellman exchange?
     P: ad107e1e9123a9d0d660faa795597…
     G: ac4032ef4f2d9ae39df30b5c8ffdac5 …
6. What is the DH Public Key sent by the Server? (first few characters)
   Server's DH Public Key: 34dd5a6f78eff5c94d9f5006e96472558bb64cfdf23ea0 …
7. What additional value is sent along with the P, G, and DH Public Key values?
   Signature of Server Key Exchange
   - What key was used to create this value?
     Server's Private Key
8. How is the Client Key Exchange Record in the DHE capture different from the same record in the RSA capture?
   In the RSA Capture, the Client Key Exchange included the (encrypted) Pre-Master-Secret, generated by the Client. In the DHE capture, the Client Key Exchange includes the client's DH Public Key, sent in plain text.

9. What is the DH Public Key sent by the Client? (first few characters)
   Client's DH Public Key: a6b94ddbd50ccdf860c5b03462d83a4197585789be61 ...
10. Knowing the DH Public Keys sent by the Client and the Server, as well as the P and G value, do you have everything you need to recreate the Shared Secret?
    No
    - ~~If yes, how would you combine the values you know to create the Shared Secret?~~
    - If no, what value(s) are you missing in order to create the Shared Secret?
      We would need one DH *Private* Key of either the Client or the Server
11. Packet #17 is an Alert
    - Can you determine the Severity or Description of this Alert?
      No
    - Why or why not?
      It is encrypted since it was sent after the Change Cipher Spec
12. Does this TLS Handshake provide Forward Secrecy?
    Yes!
    - What are the implications of providing or not providing Forward Secrecy?
      Even if you were to acquire the Private Key that correlates to the Server's Certificate, you would not be able to re-create the DH Shared Secret, and therefore the Session Keys used to protect this session. This session was encrypted once, and remains always encrypted.

Answer Key for Session Resumption packet capture Questions

1. In the Original session, excluding Application Data and Alert Records…
   - How many Records were sent from Client to Server?
     4 – Client Hello, Client Key Exchange, Change Cipher Spec, Encrypted Handshake (Finished)
   - How many Records were sent from Server to Client?
     5 – Server Hello, Certificate, Server Hello Done, Change Cipher Spec, Encrypted Handshake Message (Finished)
2. In the Resumed session, excluding Application Data and Alert Records…
   - How many Records were sent from Client to Server?
     3 – Client Hello, Change Cipher Spec, Encrypted Handshake Message (Finished)
   - How many Records were sent from Server to Client?
     3 – Server Hello, Change Cipher Spec, Encrypted Handshake Message (Finished)
3. What records existed in the Original session, that did NOT exist in the Resumed session?
   Client Key Exchange, Certificate, Server Hello Done
4. What was the Record Length for each of the records from the previous question?
   Client Key Exchange – 262 bytes
   Certificate – 2646 bytes
   Server Hello Done – 4 bytes
5. In the Original session, how many Round Trips were necessary before the Client could send Application Data?
   2 full round trips (packets 4 and 6, and packets 8 and 10)
6. In the Resumed session, how many Round Trips were necessary before the Client could send Application Data?
   1 full round trip (packets 24 and 26)
7. List the first few digits of the Session ID for the…
   - Original session Client Hello a4ed4c5f12c187a72dfa573e0b34b59783c1c34c257 …
   - Original session Server Hello 746222175ee47cf4f14f6fe51ad352da4512f542cecf …
   - Resumed session Client Hello 746222175ee47cf4f14f6fe51ad352da4512f542cecf …
   - Resumed session Server Hello 746222175ee47cf4f14f6fe51ad352da4512f542cecf …
8. What Cipher Suite was selected by the Server in both sessions?
   TLS_RSA_WITH_AES_256_CBC_SHA256
9. Does either TLS Handshake provide Forward Secrecy?
   No
   - What are the implications of providing or not providing Forward Secrecy?
     If you were to acquire the Server's Private Key, you could apply it to the *original* Handshake to extract the Pre-Master-Secret, calculate the Master Secret, calculate the Session Keys, and then decrypt all the Application Data *from both sessions*.

Answer Key for Mutual Authentication packet capture Questions

1. Compared to the packet capture from LAB 6.1, which additional records exist in the Mutual Authentication capture?
   From the Server – Certificate Request
   From the Client – Certificate, Certificate Verify
2. In the Certificate Request record…
   - What Certificate Types did the Server Request? RSA Signed, DSS Signed, ECDSA Signed
   - How many Signature Hash Algorithms does the Server accept? 23
   - What are the Common Name(s) of the Certificate Authorities which the Server accepts?
     orangeCA.com
3. In the Client's Certificate record…
   - How many Certificates did the Client send?
     2
   - What is the Common Name in the Subject fields for each Certificate sent by the Client?
     greenClient
     orangeCA.com
4. What value is being sent in the Client Key Exchange record?
   Pre-Master-Secret, generated by the Client
   - What is the ultimate purpose of this value?
     This is the Seed value which will ultimately be converted into a Master Secret, and then into the Session Keys which are protecting the Application Data for this session.
5. What value is being sent in the Certificate Verify record?
   A Hash of all the Handshake Records seen thus far, signed by the Client's Private Key
   - What is the ultimate purpose of this value?
     To prove the Client is indeed the true owner of the Certificate it presented
6. Does this TLS Handshake provide Forward Secrecy?
   No
   - What are the implications of providing or not providing Forward Secrecy?
     If you were to acquire the Server's Private Key, you could apply it this Handshake to extract the Pre-Master-Secret, calculate the Master Secret, calculate the Session Keys, and then decrypt all the Application Data from this session.