



# Criptografia - SSL/TLS

## Trabalho Prático

Autor:

Vasco Barnabé - nº42819

Setembro de 2021



# 1 Introdução

Como exercício prático relativamente ao trabalho teórico realizado, será efetuada uma pequena análise de tráfego na rede utilizando o **Wireshark**, programa que analisa este tráfego e o organiza por protocolos. Deste modo, será apresentado um exemplo do processo de **Handshake com TLS**.

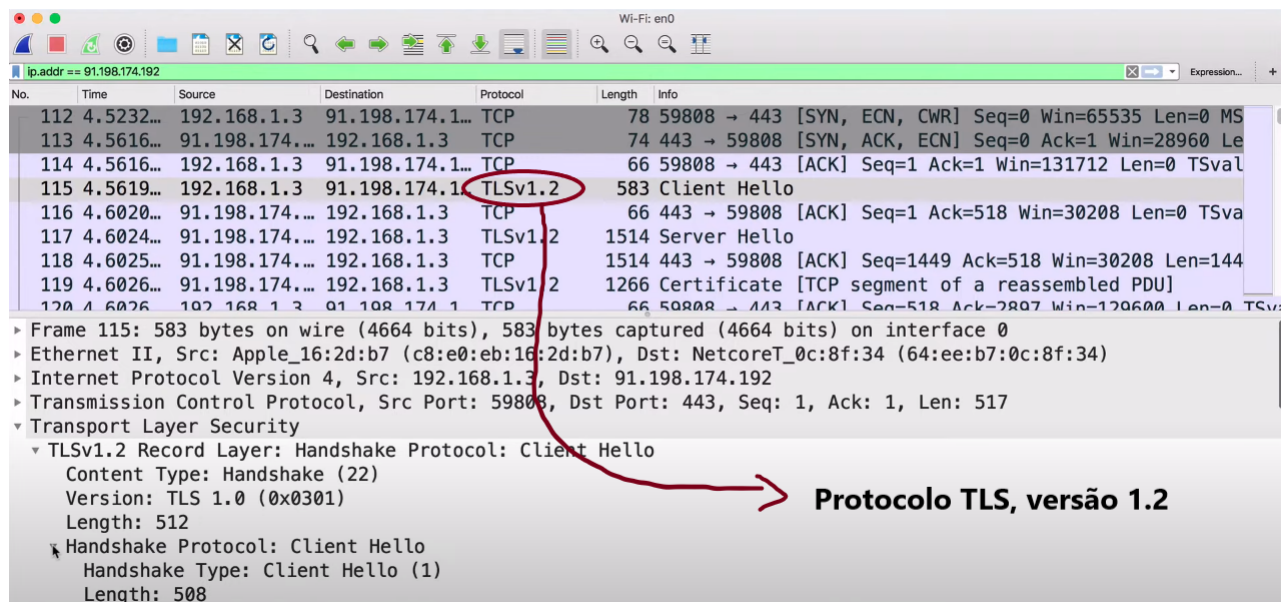
# 2 Requisitos

Para a realização deste exercício basta o uso do programa Wireshark. A sua instalação pode ser feita a partir de um terminal, em Linux, do seguinte modo:

**\$ sudo apt install wireshark.**

# 3 Análise de Tráfego utilizando o WireShark

Como é possível verificar na imagem seguinte, a versão do protocolo TLS em execução neste exemplo é a **versão 1.2**:



Início do processo de **Handshake** entre cliente e servidor:

Wireshark packet capture showing the start of a TLS handshake. The packet list shows a 'Client Hello' message (Frame 115) from the client (192.168.1.3) to the server (91.198.174.1). The packet details pane shows the 'Handshake Protocol: Client Hello' message, which is highlighted with a green arrow pointing to a text box explaining its purpose.

Handshake Protocol: Client Hello  
Content Type: Handshake (22)  
Version: TLS 1.0 (0x0301)  
Length: 512  
Handshake Protocol: Client Hello  
Handshake Type: Client Hello (1)  
Length: 508

Início do Handshake, onde cliente e servidor acordam que se reconhecem e podem estabelecer uma ligação segura entre si

O cliente é o primeiro a enviar uma mensagem. Envia uma mensagem "olá" para o servidor, por um canal inseguro, de modo a dar a conhecer a este que quer estabelecer ligação segura:

Wireshark packet capture showing the start of a TLS handshake. The packet list shows a 'Client Hello' message (Frame 115) from the client (192.168.1.3) to the server (91.198.174.1). The packet details pane shows the 'Handshake Protocol: Client Hello' message, which is highlighted with a red circle. Red and yellow arrows point from specific fields in the message to text boxes explaining their roles.

Handshake Protocol: Client Hello  
Content Type: Handshake (22)  
Version: TLS 1.0 (0x0301)  
Length: 512  
Handshake Protocol: Client Hello  
Handshake Type: Client Hello (1)  
Length: 508

Endereço que envia a mensagem, **cliente**  
Endereço de destino, **servidor**  
Mensagem enviada do cliente para o servidor

De seguida, é o servidor a mandar a mensagem "olá" ao cliente, como se pode verificar comparando com a imagem anterior, houve uma troca dos endereços de envio e de destino, pois agora a mensagem vai no sentido servidor  $\Rightarrow$  cliente :

No.	Time	Source	Destination	Protocol	Length	Info
112	4.5232...	192.168.1.3	91.198.174.1...	TCP	78	59808 → 443 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MS
113	4.5616...	91.198.174.1...	192.168.1.3	TCP	74	443 → 59808 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Le
114	4.5616...	192.168.1.3	91.198.174.1...	TCP	66	59808 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval
115	4.5619...	192.168.1.3	91.198.174.1...	TLSv1.2	583	Client Hello
116	4.6020...	91.198.174.1...	192.168.1.3	TCP	66	443 → 59808 [ACK] Seq=1 Ack=518 Win=30208 Len=0 TSva
117	4.6024...	91.198.174.1...	192.168.1.3	TLSv1.2	1514	Server Hello
118	4.6025...	91.198.174.1...	192.168.1.3	TCP	1514	443 → 59808 [ACK] Seq=1449 Ack=518 Win=30208 Len=144
119	4.6026...	91.198.174.1...	192.168.1.3	TLSv1.2	1266	Certificate [TCP segment of a reassembled PDU]

▶ Frame 115: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0  
 ▶ Ethernet II, Src: Apple\_16:2d:b7 (c8:e0:eb:16:2d:b7), Dst: NetcoreT\_0c:8f:34 (64:ee:b7:0c:8f:34)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.3, Dst: 91.198.174.192  
 ▶ Transmission Control Protocol, Src Port: 59808, Dst Port: 443, Seq: 1, Ack: 1, Len: 517  
 ▶ Transport Layer Security  
   ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello  
     Content Type: Handshake (22)  
     Version: TLS 1.0 (0x0301)  
     Length: 512  
     Handshake Protocol: Client Hello  
       Handshake Type: Client Hello (1)  
       Length: 508

Endereço que envia a mensagem, **servidor**

Endereço de destino, **cliente**

Mensagem enviada do servidor para o cliente

Após as primeiras mensagens de reconhecimento entre cliente e servidor, é altura do servidor enviar o certificado SSL ao cliente, para que se possa dar início à ligação entre ambos por um canal seguro, protegido de possíveis ataques e roubo de dados em tráfego:

No.	Time	Source	Destination	Protocol	Length	Info
112	4.5232...	192.168.1.3	91.198.174.1...	TCP	78	59808 → 443 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MS
113	4.5616...	91.198.174.1...	192.168.1.3	TCP	74	443 → 59808 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Le
114	4.5616...	192.168.1.3	91.198.174.1...	TCP	66	59808 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval
115	4.5619...	192.168.1.3	91.198.174.1...	TLSv1.2	583	Client Hello
116	4.6020...	91.198.174.1...	192.168.1.3	TCP	66	443 → 59808 [ACK] Seq=1 Ack=518 Win=30208 Len=0 TSva
117	4.6024...	91.198.174.1...	192.168.1.3	TLSv1.2	1514	Server Hello
118	4.6025...	91.198.174.1...	192.168.1.3	TCP	1514	443 → 59808 [ACK] Seq=1449 Ack=518 Win=30208 Len=144
119	4.6026...	91.198.174.1...	192.168.1.3	TLSv1.2	1266	Certificate [TCP segment of a reassembled PDU]

▶ Frame 115: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0  
 ▶ Ethernet II, Src: Apple\_16:2d:b7 (c8:e0:eb:16:2d:b7), Dst: NetcoreT\_0c:8f:34 (64:ee:b7:0c:8f:34)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.3, Dst: 91.198.174.192  
 ▶ Transmission Control Protocol, Src Port: 59808, Dst Port: 443, Seq: 1, Ack: 1, Len: 517  
 ▶ Transport Layer Security  
   ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello  
     Content Type: Handshake (22)  
     Version: TLS 1.0 (0x0301)  
     Length: 512  
     Handshake Protocol: Client Hello  
       Handshake Type: Client Hello (1)  
       Length: 508

Certificado enviado pelo servidor ao cliente, para que se possa dar início à conexão por um canal seguro



O último passo deste Handshake entre cliente e servidor, é então a troca de chaves públicas e privadas de forma a ser criada uma chave de sessão para o canal seguro por onde será feita a transferência de dados. Esta chave será utilizada para encriptar e descriptar os dados transferidos nesta ligação.

Wireshark packet capture showing TLS handshake steps. The following table summarizes the highlighted packets:

Time	Source	Destination	Protocol	Length	Info
22	91.198.174.1...	192.168.1.3	TLSv1.2	1078	Certificate Status, Server Key Exchange, Server Hello Done
24	192.168.1.3	91.198.174.1...	TLSv1.2	151	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
25	192.168.1.3	91.198.174.1...	TLSv1.2	151	Application Data
26	192.168.1.3	91.198.174.1...	TLSv1.2	379	Application Data
27	91.198.174.1...	192.168.1.3	TLSv1.2	109	Change Cipher Spec, Encrypted Handshake Message
28	91.198.174.1...	192.168.1.3	TLSv1.2	127	Application Data

Below the packet list, the details of packet 22 are shown:

- Length: 114
- Handshake Protocol: Server Key Exchange
  - Handshake Type: Server Key Exchange (12)
  - Length: 110
  - EC Diffie-Hellman Server Params
- TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 4
  - Handshake Protocol: Server Hello Done

Troca de chaves entre cliente e servidor e estabelecimento de um canal seguro, seguindo-se de troca de dados em mensagens

Após a criação de um canal seguro, dá-se início à troca de dados entre cliente e servidor, como se pode ver na imagem anterior, "Application Data".