How to setup Wireshark in order to decrypt SSL/TLS connections

Wireshark allows us to look at the traffic flowing across our network and dissect it, getting a peek inside of frames at the raw data.

SSL and TLS are two encryption protocols which operate on the Transport layer of the OSI model. They use various encryption methods to secure data as it moves across networks.

SSL/TLS encryption makes using Wireshark more challenging because it prevents administrators from viewing the data that each relevant packet carries. When Wireshark is set up properly, it can decrypt SSL/TLS and restore your ability to view the raw data

Using a **pre-master secret key** to decrypt SSL in Wireshark is the recommended method.

A **pre-master secret key** is generated by the client and used by the server to derive a master key that encrypts the session traffic. It's the current standard in cryptography and is usually implemented via Diffie-Hellman.

Step 1. Store the pre-master secret key

In order to correctly decrypt SSL/TLS connection, we need to store the decryption key. The key is automatically generated from the client side when it has to connect to a server.

In order to view and save the pre-master secret key in Windows/Linux/MacOS we need to set a valid user path to the SSLKEYLOGFILE environment variable of our operating system.

As an example, on Linux and MacOS we can simply open a terminal e type the following string:

```
export SSLKEYLOGFILE="/home/<account_name>/sslkeyfile.key"
```

Similarly, we can export the same environment variable also on Windows using the **set** command of the windows command prompt **(cmd)** or via **powershell**:

```
set SSLKEYLOGFILE=%USERPROFILE%\Desktop\keylogfile.txt
```

Finally, we have to start a browser from the terminal.

Step 2. Configure Wireshark to decrypt connections

1. Start Wireshark
2. Go to: preferences → protocols → TLS
3. Click on **Browse** under **(Pre)-Master-Secret log filename**
4. Specify the pre-master-secret key path chose at step 1
5. Save and run the capture

Hints

You can automate step 1 by running one of the following script depending on your operating system

Windows (start-chrome.cmd)

```
@echo off
set SSLKEYLOGFILE=%USERPROFILE%\Desktop\key.log
start chrome
```

MacOS (start-chrome.sh)

```
export SSLKEYLOGFILE=~/Desktop/key.log
open -n /Applications/Google\ Chrome.app
```

Linux (start-chrome.sh)

```
export SSLKEYLOGFILE=$HOME/Scrivania/key.log
google-chrome &
```