Securing the Web with TLS

COMP30040

November 12, 2018

This lab aims to show how to secure a Web communication by using the TLS protocol. The network topology is illustrated as in Figure

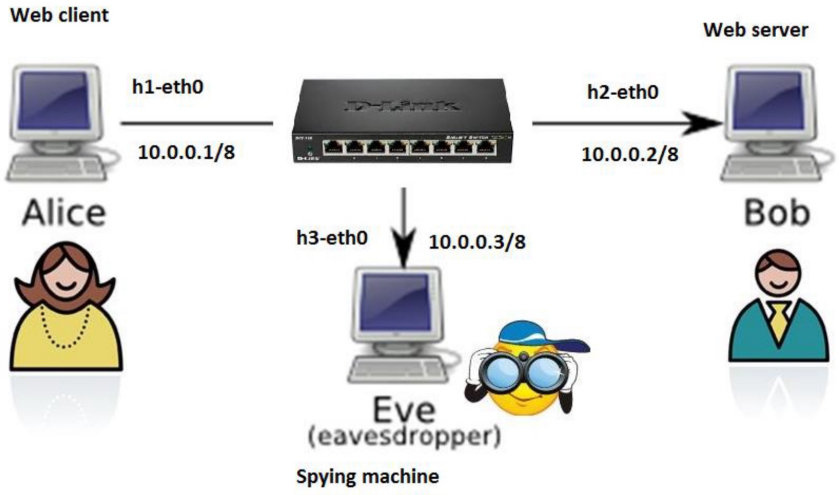

Figure 1: Network topology

Through this exercise, we will show that Eve can spy a regular HTTP traffic while she cannot spy a secured HTTPs one.

Download Week8-TLS from CS Moodle and place it in your home folder so that it is in `/home/comp30040/week8` folder

Launch the virtual network by typing the following commands:

- `sudo ./install_script.sh`
- `./tls.py`

Step 1. Spying an HTTP communication

1. Take a look at the `index.html` file in the `week8/` folder and modify if you wish. Then launch the HTTP server on Bob's node `h2` by typing command:

   `./http-server.py`

2. Use wireshark on node `h3` to capture the traffic between Alice and Bob

   ```
   wireshark -i h3-eth0 &
   ```

3. Close any firefox instance and open a new firefox in h1. Go to the URL `http://10.0.0.2`. Can you see any traffic?

4. Setup a man-in-the-middle (MITM) attack from Eve to Alice and Bob with the following `arpspoof` command on node `h3`:

   ```
   arpspoof -i h3-eth0 -t 10.0.0.1 10.0.0.2 -r
   ```

5. Use wireshark on node `h3` to capture the traffic between Alice and Bob. Can you see the massive emission of ARP packets?

6. In wireshark, select an HTTP packet and right-click on Follow TCP Stream. Can you see the content of the Web page?

7. In node `h2`, shutdown the HTTP server with `Ctrl+C`.

   Step 2. Setting up a certificate for the Web server

8. Open a new terminal in the COMP30040 VM, navigate to `week8/` folder and create a folder named as `certs/`

   ```
   cd week8
   ```

   ```
   mkdir certs
   ```

   ```
   cd certs
   ```

9. Generate the public/private key couple for the Web server:

   ```
   openssl genrsa -out site.key 1024
   ```

10. Generate the certificate signing request for the Web server (common name shall be set to 10.0.0.2)

    ```
    openssl req -new -key site.key -out site.csr
    ```

11. Let's define Bob as a CA himself and generate the CA's key couple

    ```
    openssl genrsa -out ca.key 2048
    ```

12. Make the CA's certificate. The information given in the CA's certificate must be different from the one given in the Web server certificate above (common name shall be set to `UCD CA`

    ```
    openssl req -new -x509 -days 730 -key ca.key -out ca.crt
    ```

13. Sign the Web server's public key with the private key of the CA

    ```
    openssl x509 -req -in site.csr -out site.crt -sha1 -CA ca.crt -CAkey ca.key
    -CAcreateserial -days 365
    ```

14. Look at the CA's certificate

    `cat ca.crt`

15. It is hard to read, lets transform it in a human readable format

    `openssl x509 -in ca.crt -text`

16. Create a pem file for the web server. This file contains the servers certificate as well as its private key

    `cat site.key site.crt > site.pem`

    Step 3. Spying an HTTPs communication

17. Launch the HTTPs Web server on `h2`

    `./https-server.py`

18. Try to access the Web server of Bob from Alice with HTTPS by using firefox from node `h1` and going to the URL `https://10.0.0.2`. What is the problem?

19. Correct it by adding an exception in firefox. The list of valid CAs certificates can be seen in

    `Edit / Preferences / Advanced / Certificates / View Certificates / Authorities`

20. Retry to access the web site of Bob from Alice in HTTPS. Does it work?

21. If you stopped your MITM attack from Eve, start it again

    `arpspoof -i h3-eth0 -t 10.0.0.1 10.0.0.2 -r`

22. In wireshark, restart the capture to get the new HTTPS traffic between Alice and Bob

23. Retry to access the Web site of Bob from Alice by using HTTPSs as explained in point 18

24. From Eve, try to spy the content of the Web communication between Alice and Bob as in point 6). What do you see?