20HT-2DV702 - Internet Security
Assignment 1 - Certificates

Authentication, Confidentiality and SSL/TLS

Faculty of Technology
Department of Computer Science

This document builds on the work by:
Uraz Odyurt

Instructor: Paolo Molinaro
pm222py@student.lnu.se

18-03-2023

INTRODUCTION

The first assignment is based around digital certificates, playing an important role in network security to establish the identity of a server, a client, or an application, to achieve safe communication. Uses of certificates are numerous and can be found in TLS, Virtual Private Networks (VPN), Secure Electronic Transactions (SET), secure electronic mail (PEM, S/MIME), as well as authentication of servers/clients. In this lab, we will look at several situations where digital certificates are used and also learn how to create our own public key certificate to verify ourselves on the Internet or within an intranet. Additionally, we investigate the TLS protocol, where we work with popular network utilities such as `OpenSSL`, `s_client` and `Wireshark`.

We expect you to use a GNU/Linux system for this lab, as it will be most convenient when working with required tools. If you are a Windows user, there are several options such as installing Linux virtually on a hypervisor, e.g., VirtualBox, using a LiveCD of a GNU/Linux distribution of choice, e.g., Ubuntu, or connecting to https://cscloud.lnu.se, where you can run a virtual machine directly from your web browser.

DEADLINE & SUBMISSION

The deadline for the first assignment is **Monday the 17th of April at 23:59**. The individual report should be **limited to 10 pages** and should clearly describe the steps you have taken to solve problems for each section and answer the questions related to each task. It is advisable to include screenshots for the important steps in your report. To support your answers to the questions, you may need to use references, i.e., to state where you got the information from. Examples of this has been given in the document *"Introduction to Practical Work"* and in the *"Report Template",* which must be used for the report.

Submission of the report is done on `MyMoodle`; **observe that you must upload the report before the deadline as the system will not accept your file after this time**. Do not forget to put your name and student id on the front page of the Report Template and to convert it to `.pdf` format before sending it in.

Part 1. AUTHENTICATION - WEB BROWSERS AND CERTIFICATES

1. Start by directing your favourite web browser to the following two sites below. State what happens as you visit the web pages and explain the reason for the messages you are receiving.
   (a) https://tinyurl.com/jfxtkg4
   (b) https://tinyurl.com/y7mpzjz2

2. Read the information found here (and possibly more information you must search for) and install their root certificate into your browser (or your system, depending on your browser and setup) as an authoritative certificate. Explain who is certifying this certificate and who the subject is.

3. Questions (answer with a couple of sentences, 3-5 lines should be sufficient):
   (a) What is a chain of trust and how is this related to certificates?
   (b) Why do we need SSL/TLS when shopping online? Can we not simply use the Internet as it is? How is this connected to authentication? Confidentiality?
   (c) What is a self-signed certificate and when is it used? How would your browser respond if you directed it towards a site using a self-signed certificate?
   (d) A certain site "site.net" has a `https` connection to its web-shop and thereby, uses a SSL/TLS certificate that is validated with a signature from some CA. Does the CA's signature mean that you can trust the site holding the certificate? What does the CA's signature guarantee?
   (e) What are the different types of certificates available, apart from different versions, that are commonly used for securing websites and online transactions? Provide a detailed description of each type of certificate and their specific use cases.

Part 2. OBTAINING A PUBLIC-KEY CERTIFICATE

1. You should now get your own client certificate. The previous source can be used for this. CAcert offers free client certificates, which can be obtained here. Follow the instructions and go through the registration process. Make sure you get the certificate installed in your system, or web browser.

2. Export the certificate from your browser and convert it (probably from PKCS12 format) into a `PEM` key file and a `cert` file. You might use separate software like `OpenSSL` for this operation. (See this resource for help). Use `OpenSSL` to view your PEM `cert` and note the details in the certificate, e.g., the encryption algorithm and key length, validity dates, issuer, and subject. When you have been able to view your `usercert.pem` file, you should also paste its *portable* content as plain text in your report.

3. Consider the CSR in the file `MyAwesomeCo_.csr`. Explain what is it used for? Include the organisational structure of this company in your report. You should use `OpenSSL` to observe the content.

4. Now try to generate your own CSR. Use your current city, university, or company and other meaningful information when generating it. Again, you should use `OpenSSL`. You should also paste its *portable* content as plain text in your report.

Part 3. THE SSL/TLS HANDSHAKE - OPENSSL, S_CLIENT AND WIRESHARK

1. Familiarise yourself with the `OpenSSL` toolkit by first investigating the supported ciphers. (Hint: use the command *`openssl ciphers -v`* to display all cipher options). Which one seems to be the most common standard used for encryption? RSA is an algorithm used for public-key cryptography and is often used for authentication but not for encryption, why?

2. Get to know `Wireshark` and how to capture the network traffic on an interface. `Wireshark` is available here. Try to capture a TLS handshake by directing your browser at `www.google.se`, which uses an `https` connection. You will capture A LOT of packets so use the filter option and specify that you only want to see packets using TLS (check out Wireshark's wiki). Still, there can be too many packets, so make sure you also filter based on IP address (you can use ping to find the IP of `www.google.se`).

   First, we will use `s_client`, a test utility belonging to `OpenSSL`. You can find out more about `s_client` by looking in the man-pages. You will use `s_client` to connect to `www.google.se`, which is done by issuing the command:

   ```
   $ openssl s_client -connect hostname:port-number
   ```

   You may assume that the standard TCP port number for TLS is being used. Now, using the connection you just established, send the message GET to `www.google.se`. Explain in your report what you see in your console. Before using `s_client`, you should start `Wireshark` and capture the TLS packets. When you are done, go to `Wireshark` and stop the packet capturing. Which packet in `Wireshark` will contain your GET message? Where exactly can you read it?

3. Lastly, we will try to decrypt TLS communication. You need to configure an environment variable, `SSLKEYLOGFILE`. You can do this with the command:

   ```
   $ export SSLKEYLOGFILE=/home/<username>/<some_path>/.sslkey.log
   ```

   The path and the filename are your choice. You can also set the variable in the file, `.bashrc`. Wireshark allows you to set this file under SSL/TLS protocol preferences, as *"(Pre)-Master-Secret log filename"*. You can find more details about this functionality of Wireshark and how to set it up, online. Next, install Chromium browser and start it from the same bash session. Use Wireshark to capture traffic while visiting the website, `https://www.eff.org/about/history`.

   Explain the effects of TLS decryption capability. What can you see that you were not able before? Give an example of a packet that you can read now.