Laborator 1 – Digital Certificates

1. What are the digital certificates: <https://www.youtube.com/watch?v=LRMBZhdFjDI>
2. Using Putty to generate SSH keys

We will next use Putty to generate SSH keys.

1. Download puttygen.exe from <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> (for Windows) or follow the instructions at <https://numato.com/blog/how-to-install-putty-on-linux/> (for Linux-based OS).

2. Generate a public key pair - RSA private key on 2048 bits. Click on Generate.

3. Add a PassPhrase (what is it used for?).

4. Export the public key to the publick\_key.pub file, the secret key to the private\_key.ppk file. To do this, use Save public key and Save private key, respectively.

5. Export key in openssh format. To do this, access the Conversions and Export OpenSSH key.

6. Open and see what all the generated files contain.

1. OpenSSL

We will next learn how to generate certificates with OpenSSL. More about OpenSSL:

<https://www.openssl.org/>

You can run the commands directly on your computer (if accepted), or using an online terminal:

<https://bellard.org/jslinux> (use Buildroot (Linux) Console) <http://www.tutorialspoint.com/unix_terminal_online.php>

To get started, generate an RSA key, using:

openssl genrsa -out ca.key 2048

Info about genrsa: <https://www.openssl.org/docs/man1.0.2/man1/genrsa.html>

Check the generated file with:

openssl rsa -in ca.key -check

Several useful commands that may be useful to you:

<https://www.sslshopper.com/article-most-common-openssl-commands.html>

Next, use the previously generated key in a "self-signed" certificate, valid for 120 days, stored as ca.crt.

Use the following information:

Country: RO

Province: Muntenia

City: Bucharest

Name of organization: CA\_SLA

Unit name: CA\_SLA\_Lab

Common Name: CA\_your name (e.g .: CA\_Rux)

E-mail: test@test.ro

To do this, use: openssl req -new -x509 -days <days> -key <key> -out ca.crt

Info about req: <https://www.openssl.org/docs/man1.1.0/man1/req.html>

Read about what an x509 digital certificate is: <https://www.techopedia.com/definition/29751/x509-certificate>

View the created digital certificate:

openssl x509 -text -noout -in ca.crt

Observe the values entered when generating the certificate.

You will use this CA certificate to sign / issue another certificate of a SUB\_SLA subordinate entity.

Use the following information:

Country: RO

Province: Muntenia

City: Bucharest

Name of organization: SUB\_SLA

Unit name: SUB\_SLA\_Lab

Common Name: SUB\_your name (e.g .: SUB\_Rux)

E-mail: test\_sub@test.ro

To do this, first generate a new 2048-bit SUB\_SLA key in the sub.key file (hint: use the same command from generating the key for CA openssl genrsa ...).

Then initiate a Certificate Signing Request (CSR) sub.csr:

openssl req -new -key sub.key -out sub.csr

Finally create a certificate for SUB\_SLA sub.crt signed by the CA authority, valid for 60 days, with serial number 02

openssl x509 -req -days <days> -in sub.csr -CA <certified\_CA> -CAkey <ca\_key> -set\_serial <serial\_no> -out sub.crt

View the created digital certificate:

openssl x509 -text -noout -in sub.crt

Observe the values in the certificate.

The final step is to turn this digital certificate into PKCS #12:

openssl pkcs12 -export -out sub.p12 -inkey sub.key -in sub.crt -chain -CAfile ca.crt

Info about PKCS #12: <https://www.openssl.org/docs/man1.0.2/man1/pkcs12.html>

Check the contents of the sub.p12 file using:

openssl pkcs12 -info -in sub.p12

Enter the correct passwords. What does pkcs #12 file contain?

Show the generated files to the teacher in the laboratory room.

Read more about x509 extensions here: <https://aboutssl.org/cer-vs-crt/>