

Lab1 – Certificates

You can use this lab manual for your personal documentation. Use screenshots for your own documentation, there will be questions later on that may point to this lab manual. Take care to check if you need to collect some information from the lab for the answers.

\ at the end of the line is used to mark that the command needs to be on one line. Replace **student-id** with your own student-id and **x** or **y** as your VMs correct IP in the labs.

The labs use the following topology:



W7-VM
DHCP



CA
192.168.44.50+x



www.student-id.zz
192.168.44.100+x

All VMs in this lab are in VirtualBox **Bridged** network. The machines that have static IP need to have an offset, check the topology image for reference. USE YOUR WIN7 WORKSTATION IP ADDRESS AS **x**.

All templates for VMs can be found in <\\ghost.labranet.jamk.fi\\virtuaalikoneet\\TTKS>

- **Install subCA**

Using the Centos7_k2019 template from [\\ghost.labranet.jamk.fi\\virtuaalikoneet\\TTKS\\](https://ghost.labranet.jamk.fi/virtuaalikoneet/TTKS/), clone another VM with the name CA. Remember to set “Reinitialize the MAC address...” tickbox in the import wizard. Set VM interface as *Bridged*.

Boot up the VMs shown in the topology and login to the new subCA VM (**root/root66**). Check that it has got an IP. First, lets create the CA certificate and other required files:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.44.80 netmask 255.255.255.0 broadcast 192.168.44.255
    inet6 fe80::a00:27ff:fe5f:8946 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:5f:89:46 txqueuelen 1000 (Ethernet)
```

```
mkdir /root/ca
cd /root/ca
wget https://student.labranet.jamk.fi/~jojuh/ttks/ca.cnf
wget https://student.labranet.jamk.fi/~jojuh/ttks/usr.cnf
echo 01 > serial
touch index.txt
touch index.txt.attr
openssl req -new -newkey rsa:4096 -keyout ca.key -config ca.cnf -days 365
    -extensions v3_ca -x509 -out ca.pem"
```

```
[root@localhost.localdomain ~]# mkdir /root/ca
[root@localhost.localdomain ~]# cd /root/ca/
[root@localhost.localdomain ca]# wget https://student.labranet.jamk.fi/~jojuh/ttks/ca.cnf
--2020-01-09 11:07:36-- https://student.labranet.jamk.fi/~jojuh/ttks/ca.cnf
Resolving student.labranet.jamk.fi (student.labranet.jamk.fi)... 195.148.26.130
Connecting to student.labranet.jamk.fi (student.labranet.jamk.fi):195.148.26.130
! :443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4593 (4.5K) [text/plain]
Saving to: 'ca.cnf'

100%[=====>] 4,593      --.-K/s   in 0s

2020-01-09 11:07:41 (196 MB/s) - 'ca.cnf' saved [4593/4593]

[root@localhost.localdomain ca]#
```

When asked for a passphrase for the key, use **root66**. Fill in the information, set both CN (Common Name) as *your-student-id-CA* and O (Organisation) as *your-student-id*. (Example: CN=e6210-ca and O=e6210).

Check the content of the new CA certificate:

```
openssl x509 -text -noout -in ca.pem
```

```
[root@localhost.localdomain ca]# openssl x509 -text -noout -in ca.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            ff:88:6a:45:21:b2:35:af
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=FI, L=JKL, O=M3426, CN=M3426-CA
        Validity
            Not Before: Jan  9 11:16:59 2020 GMT
            Not After : Jan  8 11:16:59 2021 GMT
        Subject: C=FI, L=JKL, O=M3426, CN=M3426-CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (4096 bit)
            Modulus:
                00:e2:3b:1f:e3:75:15:9e:30:37:bf:3c:0b:2f:ee:
                44:e8:6c:ce:37:2d:f5:0f:64:34:5a:29:a9:05:d9:
                05:6e:44:d8:59:db:25:c3:e5:49:8a:70:1f:39:fe:
```

You should see the info you typed, public key is 4096bits and the most important part: CA:TRUE in Basic constraints. Without this, your CA cert will not be trusted by the clients.

```
X509v3 Basic Constraints: critical
    CA:TRUE
X509v3 Key Usage:
    Certificate Sign, CRL Sign
Signature Algorithm: sha256WithRSAEncryption
```

• Creating a CSR for the web server

Clone and boot up the Webserver VM. Create a new RSA key and CSR for the webserver. To use subjectAltNames correctly, we need a local copy of the openssl.conf:

```
cp /etc/pki/tls/openssl.cnf /root/openssl_san.cnf
```

```
[root@localhost.localdomain ~]# cp /etc/pki/tls/openssl.cnf /root/openssl_san.cnf
[root@localhost.localdomain ~]# nano openssl_san.cnf
```

Now modify this new file and add the following lines to the bottom of the config:

```
[alt_names]
DNS.1 = www.your-student-id.zz
DNS.2 = your-student-id.zz
IP.1 = your-webserver-ip-here
[alt_names]
DNS.1 = www.M3426.zz
DNS.2 = M3426.zz
IP.1 = 192.168.44.130
```

Then find the [v3_req] section and add:

```
subjectAltName = @alt_names
```

```
[ v3_req ]

# Extensions to add to a cert
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature
subjectAltName = @alt_names
```

Also uncomment the following under [req] –section:

```
req_extensions = v3_req
req_extensions = v3_req # The
```

BONUS: As there is no way to give these subjectAltNames as parameters to openssl command-line, several different ways of doing this have been engineered. If you want a challenge, find a way to add SANs in a oneliner command.

Now we can use the new config file and correct SANs should appear in the certificate request:

```
openssl req -new -newkey rsa:2048 -nodes -keyout www.key -out www.csr
-config /root/openssl_san.cnf
```

```
[root@localhost.localdomain ~]# openssl req -new -newkey rsa:2048 -nodes -keyout
www.key -out www.csr -config /root/openssl_san.cnf
```

Set CN as www.your-student-id.zz, other fields like you did with the CA. **Be very precise with the command-line above to avoid errors! It is only a single line!** Double-check the contents of the CSR, it absolutely should show the Subject Alternative Name fields too:

```
Country Name (2 letter code) [XX]:FI
State or Province Name (full name) []:JKL
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:M3426
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:M3426.zz
Email Address []:
```

```
Data:
  Version: 0 (0x0)
  Subject: C=FI, ST=JKL, L=Default City, O=M3426, CN=M3426.zz
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
```

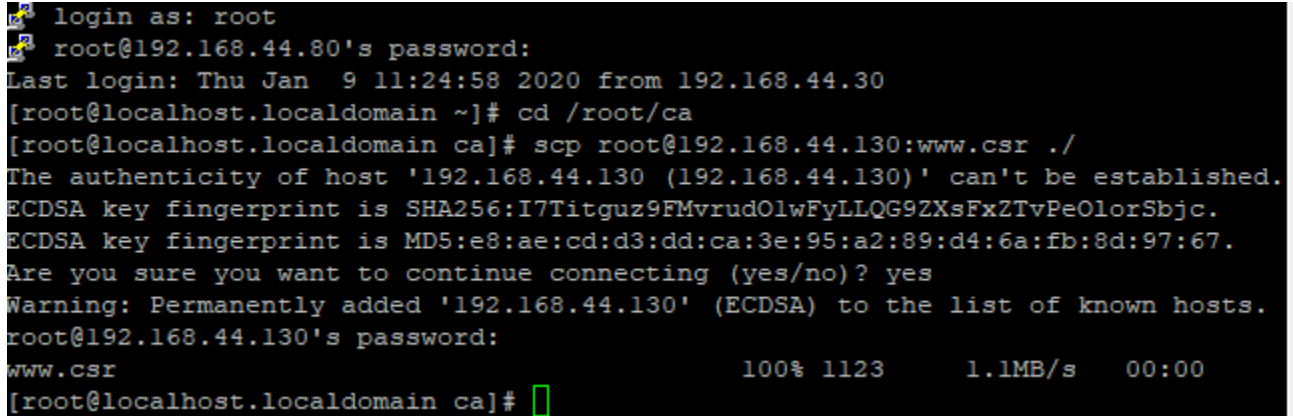
```
Attributes:
  challengePassword      :root66
Requested Extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment
  X509v3 Subject Alternative Name:
    DNS:www.M3426.zz, DNS:M3426.zz, IP Address:192.168.44.130
```

```
openssl req -noout -text -in www.Csr
```

On the CA VM, copy the csr from the webserver to the CA machine (Change the IP to point to Web server):

```
cd /root/ca
```

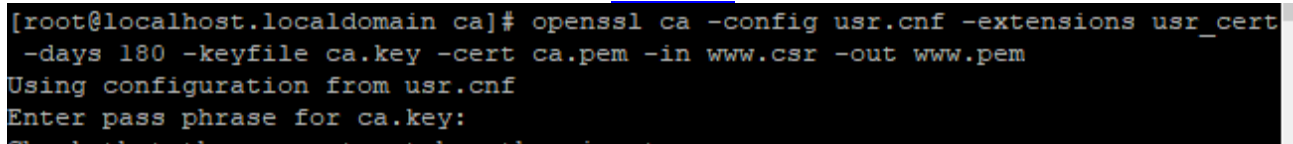
```
scp root@webserver-ip-here:www.csr ./
```



```
login as: root
root@192.168.44.80's password:
Last login: Thu Jan  9 11:24:58 2020 from 192.168.44.30
[root@localhost.localdomain ~]# cd /root/ca
[root@localhost.localdomain ca]# scp root@192.168.44.130:www.csr ./
The authenticity of host '192.168.44.130 (192.168.44.130)' can't be established.
ECDSA key fingerprint is SHA256:I7Titguz9FMvrudOlwFyLLQG92XsFxZTvPeOlorSbjc.
ECDSA key fingerprint is MD5:e8:ae:cd:d3:dd:ca:3e:95:a2:89:d4:6a:fb:8d:97:67.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.44.130' (ECDSA) to the list of known hosts.
root@192.168.44.130's password:
www.csr                                100% 1123      1.1MB/s   00:00
[root@localhost.localdomain ca]#
```

If everything seems to be right, sign the CSR with the CA key:

```
openssl ca -config usr.cnf -extensions usr_cert -days 180 -keyfile ca.key
-cert ca.pem -in www.csr -out www.pem
```



```
[root@localhost.localdomain ca]# openssl ca -config usr.cnf -extensions usr_cert
-days 180 -keyfile ca.key -cert ca.pem -in www.csr -out www.pem
Using configuration from usr.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
```

Before answering yes, take time to check that the certificate info is correct. Especially check that Basic Constraints has CA:FALSE as we do not want our webserver to be a CA. Examine the contents of the new CRT file:

```

Validity
    Not Before: Jan  9 11:59:16 2020 GMT
    Not After : Jul  7 11:59:16 2020 GMT
Subject:
    countryName           = FI
    stateOrProvinceName   = JKL
    localityName          = Default City
    organizationName       = M3426
    commonName             = M3426.zz
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment
    Netscape Comment:
        OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
        9B:8F:0D:92:2C:E7:03:4A:DF:FB:B7:9E:F7:1F:A1:FE:FD:C9:78:0A
    X509v3 Authority Key Identifier:
        keyid:2B:97:93:B9:B2:37:54:B2:70:DC:94:AA:EF:30:B6:D6:A9:02:A5:0
5

    X509v3 Subject Alternative Name:
        DNS:www.M3426.zz, DNS:M3426.zz, IP Address:192.168.44.130

```

QUESTIONNAIRE: What are the correct key usage values for a generic Web server?

```
openssl x509 -noout -text -in www.pem
```

And finally, copy it back to the Webserver VM.

```
scp www.pem root@webserver-ip-here:www.pem
```

```

[root@localhost.localdomain ca]# scp www.pem root@192.168.44.130:www.pem
root@192.168.44.130's password:
www.pem                                100% 5915      5.8MB/s   00:00
[root@localhost.localdomain ca]#

```

- **Configure SSL**

In the Webserver, you have to do two things. First, install Apache and mod_ssl and add firewall rule:

```
yum install httpd mod_ssl
```

```

Installed:
  mod_ssl.x86_64 1:2.4.6-90.el7.centos

Updated:
  httpd.x86_64 0:2.4.6-90.el7.centos

Dependency Updated:
  httpd-tools.x86_64 0:2.4.6-90.el7.centos

Complete!
[root@localhost.localdomain ~]#

```

```

firewall-cmd --permanent --add-service=http --add-service=https
firewall-cmd --reload

```

```
[root@localhost.localdomain ~]# systemctl start firewalld.service
[root@localhost.localdomain ~]# firewall-cmd --permanent --add-service=http --add-service=https
success
[root@localhost.localdomain ~]# firewall-cmd --reload
success
[root@localhost.localdomain ~]#
```

Then copy the key and certificate to the correct paths:

```
cp www.key /etc/pki/tls/private/
cp www.pem /etc/pki/tls/certs/
```

In those folders should exist also a default self-signed certificate (localhost.key and localhost.crt). Check their permissions and set the same permissions to the www.key and [www.pem](#).

```
-rw-----. 1 root root 1468 Jan  9 12:19 localhost.crt
-rwxr-xr-x. 1 root root  610 Aug  4 2017 make-dummy-cert
-rw-r--r--. 1 root root 2516 Aug  4 2017 Makefile
-rwxr-xr-x. 1 root root  829 Aug  4 2017 renew-dummy-cert
-rw-----. 1 root root 5915 Jan  9 12:22 www.pem
[root@localhost.localdomain certs]#
```

```
[root@localhost.localdomain private]# ls -l
total 8
-rw-----. 1 root root 1675 Jan  9 12:19 localhost.key
-rw-----. 1 root root 1704 Jan  9 12:22 www.key
[root@localhost.localdomain private]#
```

Last thing you need to do is edit /etc/httpd/conf.d/ssl.conf and change Apache to use your certificates. Find the following lines:

```
SSLCertificateFile ...
SSLCertificateKeyFile ...
SSLCertificateChainFile ...
```

And set them to point to your files. Reload apache (*systemctl restart httpd*).

```
SSLCertificateFile /etc/pki/tls/certs/www.pem

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/pki/tls/private/www.key
```

```
[root@localhost.localdomain ~]# systemctl restart httpd
[root@localhost.localdomain ~]#
```

QUESTIONNAIRE: When and how is the SSLCertificateChainFile option used?

- **Trusted root**

Your Workstation VM needs to trust the **CA certificate**. Copy the ca.pem to the Windows VM. You can use WinSCP or copy/paste via PuTTY. There are several places where the ca.pem needs to be put

In a Windows-based PC, you can add the certificate to trusted roots in MMC console using the Local Computer Certificate tool. Open mmc.exe from Start->Run, and add *Certificates Snap-in* for the *Computer Account*. Import the certificate to *Trusted Root Certification Authorities*.

M3426-CA	M3426-CA	8.1.2021	<All>	<None>
Microsoft Authenticode(tm) Root...	Microsoft Authenticode(tm) Root...	1.1.2000	Secure Email, Code ...	Microsoft Authenti...

However, Firefox does not use the system certificates so we must add it to your Firefox profile Certificates. Open Firefox, select *Options -> Advanced -> Certificates -> View Certificates*. Select the *Authorities* tab, Click *Import* and select the certificate file. Add trusts for all purposes.

<ul style="list-style-type: none"> M3426 <ul style="list-style-type: none"> M3426-CA Microsec Ltd. 	Software Security Device
--	--------------------------

*BONUS: Copy the ca.pem to the webserver also and add it to the trusted root certificates of the Centos. Test with `wget https://localhost/`. Check the man page for **update-ca-trust** command for help/more information.*

• DNS name and testing

Final step is to add a DNS name for the webserver. If you added the correct IP address to the SubjectAltNames, this is not strictly necessary as browsing with IP works also.

Browse to <https://zz.labranet.jamk.fi/> and login with your LabraNet account. You need to add an DNS A record for www.student-id.zz like in the picture. After this, you should be able to access the webserver using the DNS name also.

Name	Type	Content	Priority	TTL
www.teppo.zz	IN A	192.168.x.y		

Add record

If you are working at home, find out how to add a hosts-entry to the Workstation VM. The correct path for Windows 7 VM is `C:\Windows\System32\drivers\etc\hosts`. Point `www.student-id.zz` in the Windows hosts-file to the correct IP address. This way you can test the server even without a DNS entry.

When you are finished, take a screenshot of the Certificate Hierarchy path (shown in View Certificate - Details)

This server is target.ttkk.local at 192.168.44.130

You are trying to access host **www.m3426.zz** from IP **192.168.44.165**

The connection to the server is via HTTPS

Your browser user-agent is Mozilla/5.0 (Windows NT 6.1; WOW64; rv:57.0) Gecko/20100101 Firefox/57.0

For more information, see <http://php.net/manual/en/reserved.variables.server.php>

The server identification string is: apache/2.4.6 (centos) openssl/1.0.2k-fips php/5.4.16

