Network and System Security ET2595

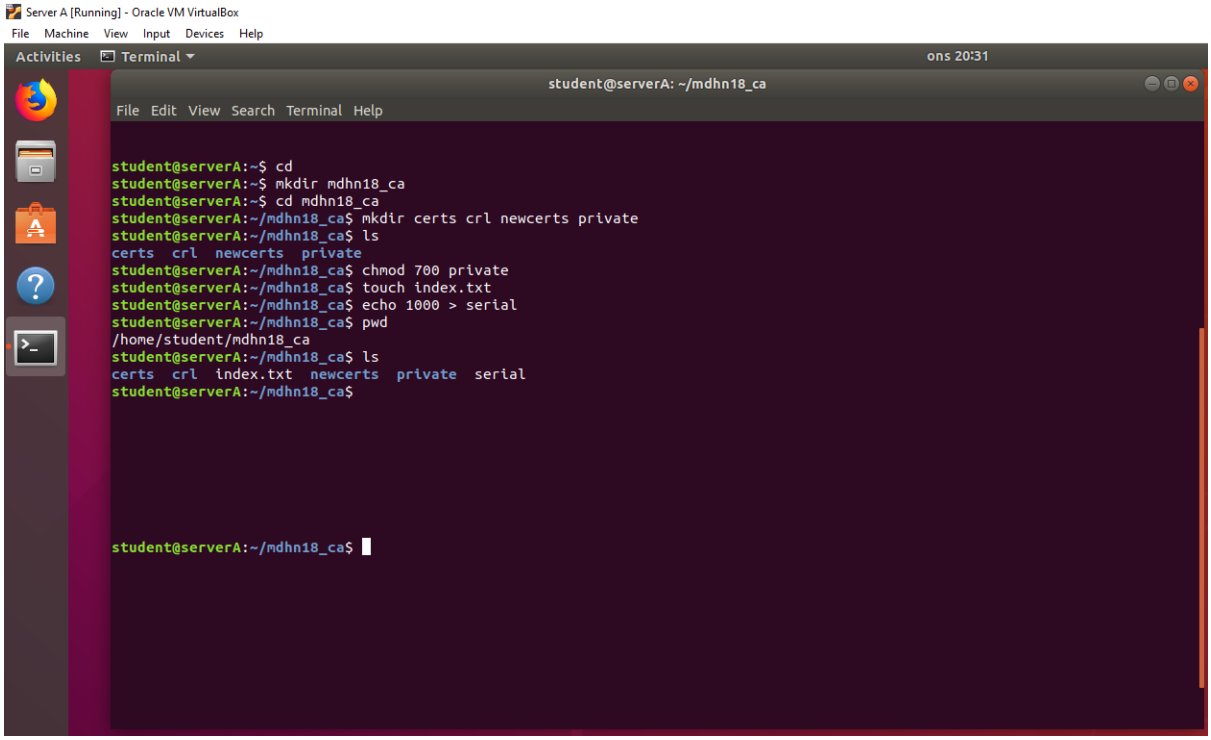
Lab 2 Report

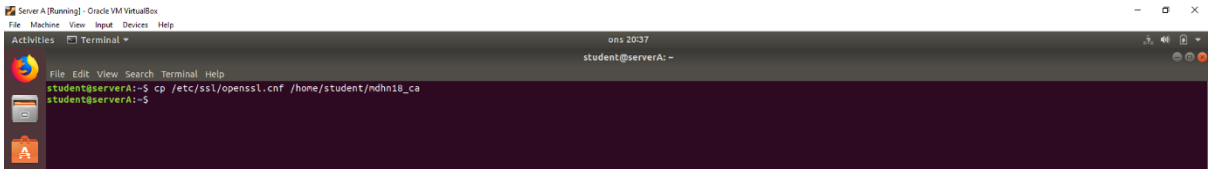
Name: Md Rabiul Ahamed Bin Hanif

P. no. 19920212-2637

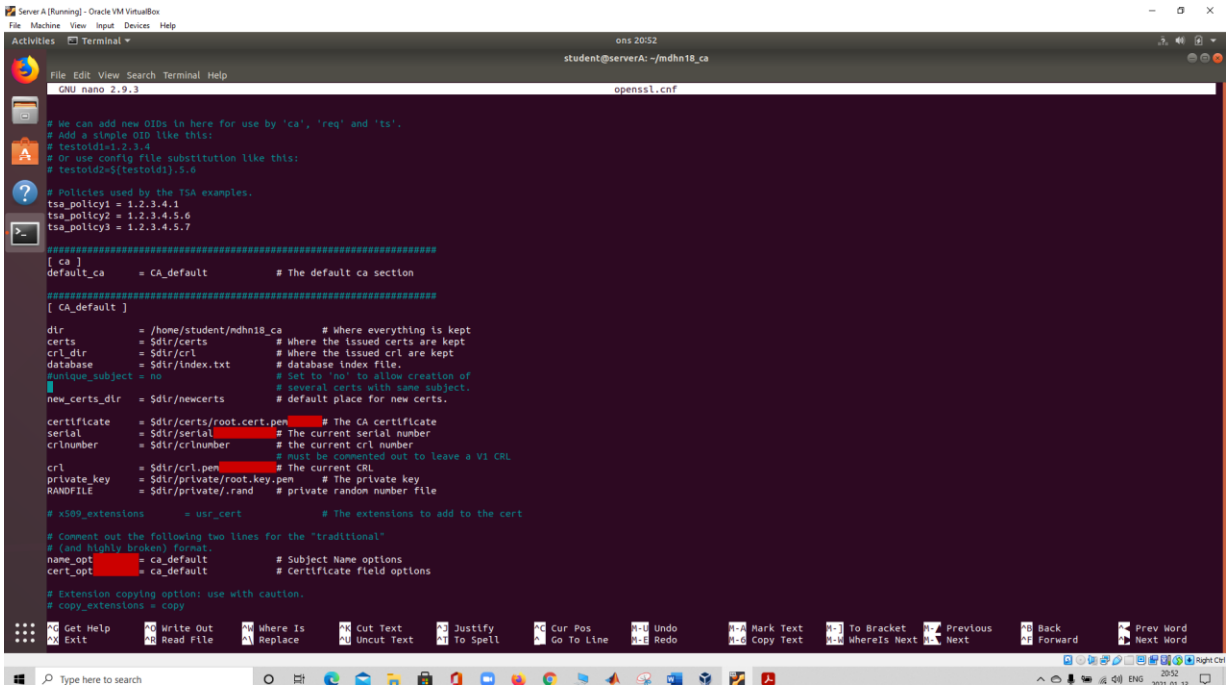
Email: mdhn18@student.bth.se

Environments create:





- openssl.cnf file was changed according to the instruction. All screen short are given below.



Task 1: [v3_ca]

subjectKeyIdentifier = hash

Identifies certificates which contain a specific public key.

authorityKeyIdentifier = keyid:always,issuer

Identifies the public key corresponding to the private key which was used to sign a certificate. For the **keyid** option, if it is present an attempt is made to copy the subject key identifier from the parent certificate.

For the **issuer** option, it copies the issuer and serial number from the issuer certificate. This is applicable when the keyid option fails or is not included unless the "always" flag will always include the value.

basicConstraints = CA:true

This is a multi valued extension which indicates whether a certificate is a CA certificate. The first (mandatory) name is CA followed by TRUE or FALSE. If CA is TRUE then an optional "pathlen" is used followed by an non-negative integer value can be included.

keyUsage = critical, digitalSignature, cRLSign, keyCertSign
Key usage is a multi valued extension consisting of a list of names of the permitted key usages. The digitalSignature bit is used when the subject public key is used for verifying digital signatures, other than signatures on certificates (bit 5) and CRLs (bit 6), such as those used in an entity authentication service, a data origin authentication service, and/or an integrity service.

The cRLSign bit is used when the subject public key is used for verifying signatures on certificate revocation lists (for example, CRLs, delta CRLs, or ARLs).

The keyCertSign bit is used when the subject public key is used for verifying signatures on public key certificates.

Task 2: [[v3_intermediate_ca]]

```
[ v3_intermediate_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
```

subjectKeyIdentifier=hash
Identifies certificates which contain a specific public key.

authorityKeyIdentifier = keyid:always,issuer
Identifies the public key corresponding to the private key which was used to sign a certificate.

For the **keyid** option, if it is present an attempt is made to copy the subject key identifier from the parent certificate.

For the **issuer** option, it copies the issuer and serial number from the issuer certificate. This is applicable when the keyid option fails or is not included unless the "always" flag will always include the value.

basicConstraints = critical, CA:true, pathlen:0
Indicates whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate.
If CA is TRUE then an optional pathlen name followed by an non-negative integer value can be included.

The pathlen parameter indicates the maximum number of CAs that can appear below this one in a chain. So, a CA with a pathlen of zero can only be used to sign end user certificates and not fur-ther CAs.

keyUsage = critical, digitalSignature, cRLSign, keyCertSign
Key usage is a multi valued extension consisting of a list of names of the permitted key usages. The digitalSignature bit is used when the subject public key is used for verifying digital signatures, other than signatures on certificates (bit 5) and CRLs (bit 6), such as those used in an entity authentication service, a data origin authentication service, and/or an integrity service.

The cRLSign bit is used when the subject public key is used for verifying signatures on certificate revocation lists (for example, CRLs, delta CRLs, or ARLs).

The keyCertSign bit is used when the subject public key is used for verifying signatures on public key certificates.

Difference between [v3 _ca] and [v3_intermediate_ca] is the values of basicConstraints. The values are **CA:true** and **critical, CA:true, pathlen:0** respectively.

Task 3: [usr_cert]

```
[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection

# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
```

subjectKeyIdentifier=hash

Identifies certificates which contain a specific public key.

authorityKeyIdentifier = keyid,issuer

Identifies the public key corresponding to the private key which was used to sign a certificate. For the **keyid** option, if it is present an attempt is made to copy the subject key identifier from the parent certificate.

For the **issuer** option, it copies the issuer and serial number from the issuer certificate. This will only be done if the keyid option fails or is not included unless the "always" flag will always in-clude the value.

basicConstraints=CA:FALSE

An end user certificate must either set CA to FALSE or exclude the extension entirely. Some software may require the inclusion of basic Constraints with CA to be set to FALSE for end entity certificates.

nsComment = "OpenSSL Generated Certificate"

This message will be displayed in Netscape's comment listbox.
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment

The **nonRepudiation** bit is used when the subject public key is used to verify digital signatures, other than signatures on certificates (bit 5) and CRLs (bit 6), used to provide a non-repudiation service that protects against the signing entity falsely denying some action.

The **digitalSignature** bit is used when the subject public key is used for verifying digital signatures, other than signatures on certificates (bit 5) and CRLs (bit 6), such as those used in an entity authentication service, a data origin authentication service, and/or an integrity service.

The **keyEncipherment** bit is used when the subject public key is used for enciphering private or secret keys, i.e., for key trransport.

extendedKeyUsage = clientAuth, emailProtection

This extension consists of a list of usages indicating purposes of which the certificate public key could be used for.

The value **clientAuth** represents SSL/TLS Web Client Authentication.
The value **emailProtection** represents E-mail Protection (S/MIME).

Task 4: [server_cert]

```
[ server_cert ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
```

basicConstraints=CA:FALSE

An end user certificate must either set CA to FALSE or exclude the extension entirely. Some software may require the inclusion of basic Constraints with CA to be set to FALSE for end entity certificates.

subjectKeyIdentifier=hash

Identifies certificates which contain a specific public key.

authorityKeyIdentifier = keyid,issuer:always

Identifies the public key corresponding to the private key which was used to sign a certificate. For the **keyid** option, if it is present an attempt is made to copy the subject key identifier from the parent certificate.

For the issuer option, it copies the issuer and serial number from the issuer certificate. This will only be done if the keyid option fails or is not included unless the "always" flag will always in-clude the value.

keyUsage = critical, digitalSignature, keyEncipherment

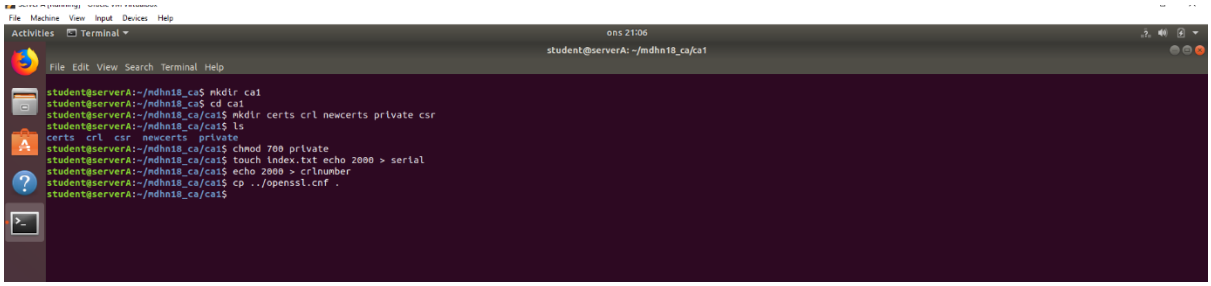
The digitalSignature bit is used when the subject public key isused for verifying digital signa-tures, other than signatures on certificates (bit 5) and CRLs (bit 6), such as those used in an entity authentication service, a data origin authentication service, and/or an integrity service. The keyEncipherment bit is asserted when the subject public key is used for enciphering private or secret keys, i.e., for key transport.

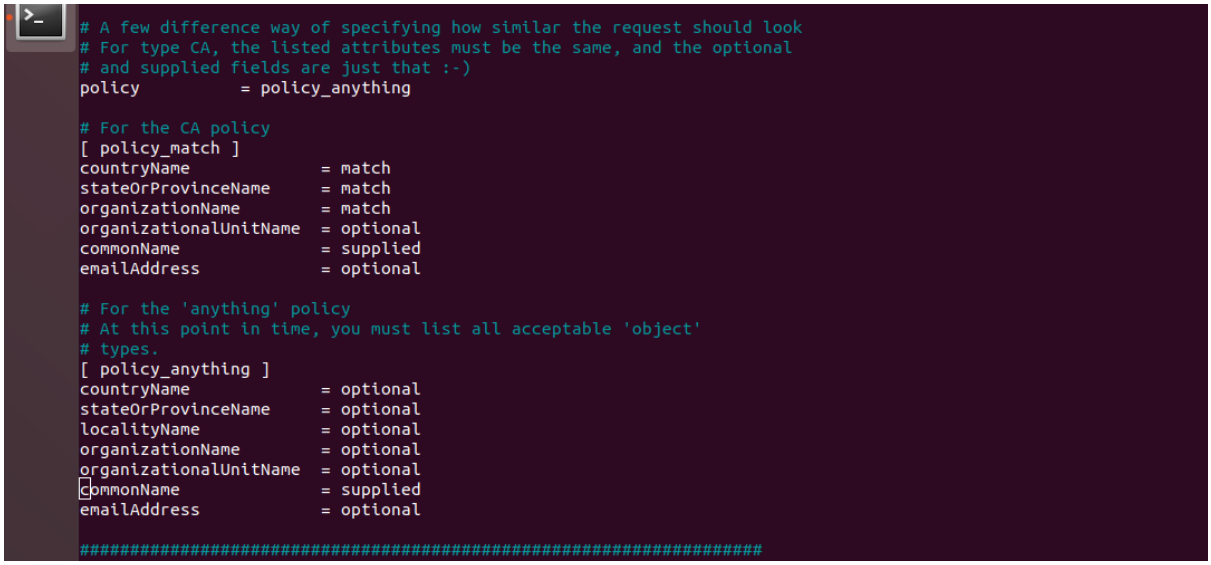
extendedKeyUsage = serverAuth

This extension consists of a list of usages indicating purposes of which the certificate public key could be used for.
The value **serverAuth** represents SSL/TLS Web Server Authentication

Task 5: Policies

- Before checking the policy, I create ca1 directory which is shown I below screen short.



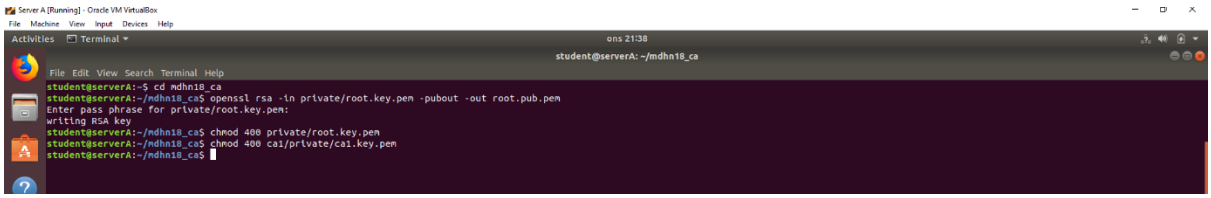


If the value of an attribute is "match", then it should be matched with the CA's DN. All fields listed as "supplied" must be present. If a field is listed as "optional", then it is allowed and not even required to be there.
In **policy_match** section, the values of countryName, stateOrProvinceName, organizationName must be present as the CA for all certificates it signs and commonName will be supplied by the issuer.
In **policy_anything** section, only commonName needs to be supplied and all the other values are optional.

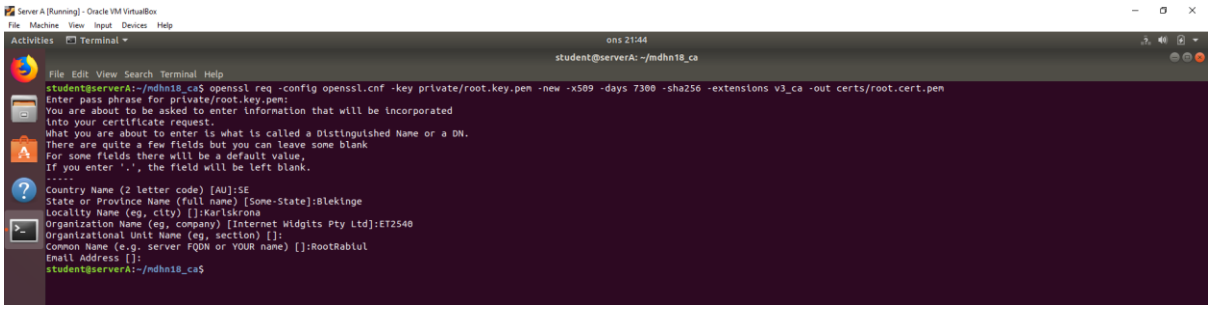
Task 6: Options for the root certificate

- create the private RSA key for *root* and *ca1*, respectively.

- To obtain the public key (e.g., for root.ca.pem) using OpenSSH you can use the following command.
- For additional protection, the filesystem access rights to private keys should be restricted to the owner.



- To generate the self-signed certificate entering the following command.



Command

openssl req -config openssl.cnf -key private/root.key.pem -new -x509 -days 7300 -sha256 -extensions v3_ca -out certs/root.cert.pem

Description

-config filename

this allows an alternative configuration file to be specified, this overrides the compile time filename or any specified in the OPENSSL_CONF environment variable.

-key filename

This specifies the file to read the private key from. It also accepts PKCS#8 format private keys for PEM format files.

-new

this option generates a new certificate request. It will prompt the user for the relevant field values. The actual fields prompted for and their maximum and minimum sizes are specified in the configuration file and any requested extensions. If the -key option is not used it will generate a new RSA private key using information specified in the configuration file.

-x509

this option outputs a self signed certificate instead of a certificate request. This is typically used to generate a test certificate or a self signed root CA. The extensions added to the certificate (if any) are specified in the configuration file. Unless specified using the set_serial option, a large random number will be used for the serial number.

-days n

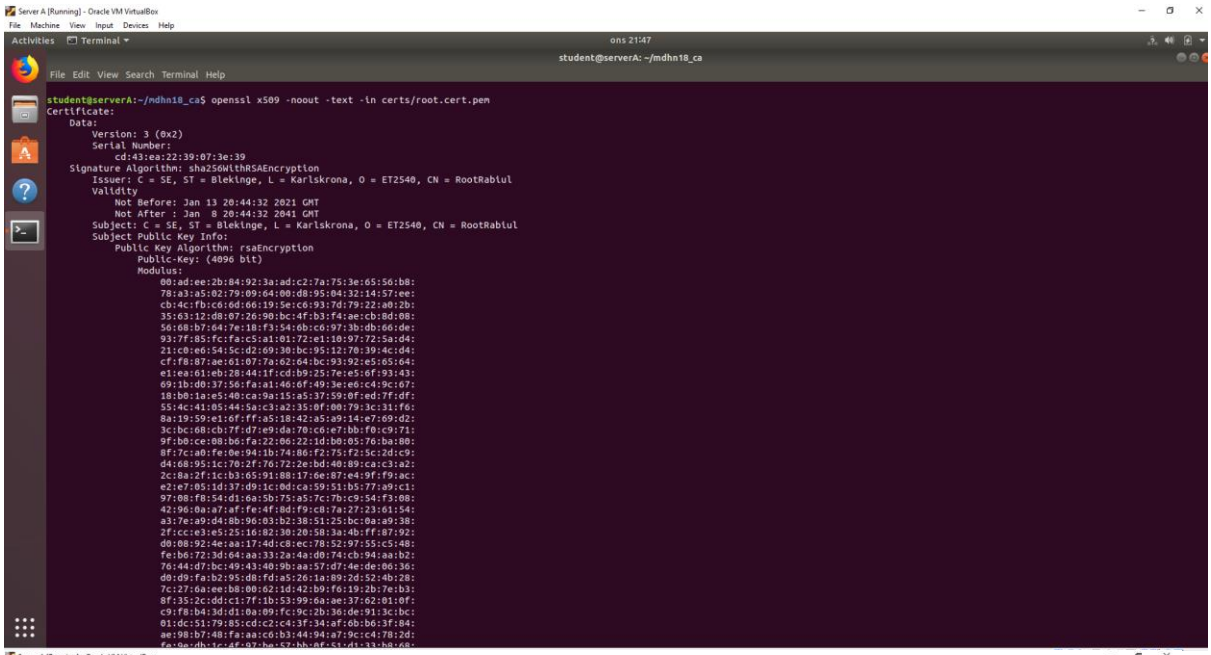
when the -x509 option is being used this specifies the number of days to certify the certificate for. The default is 30 days.

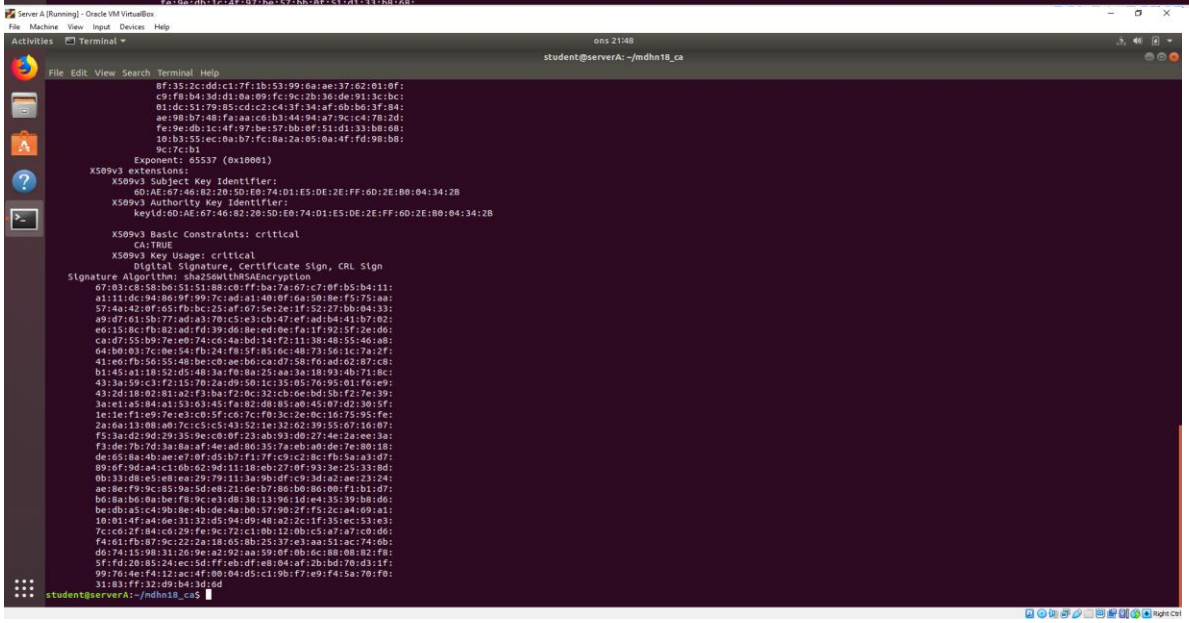
-extensions section
this option specifies alternative sections to include certificate extensions (if the -x509 option is present) or certificate request extensions. This allows several different sections to be used in the same configuration file to specify requests for a variety of purposes.

-out filename
This specifies the output filename to write to or standard output by default.

Task 7: Verify the root certificate

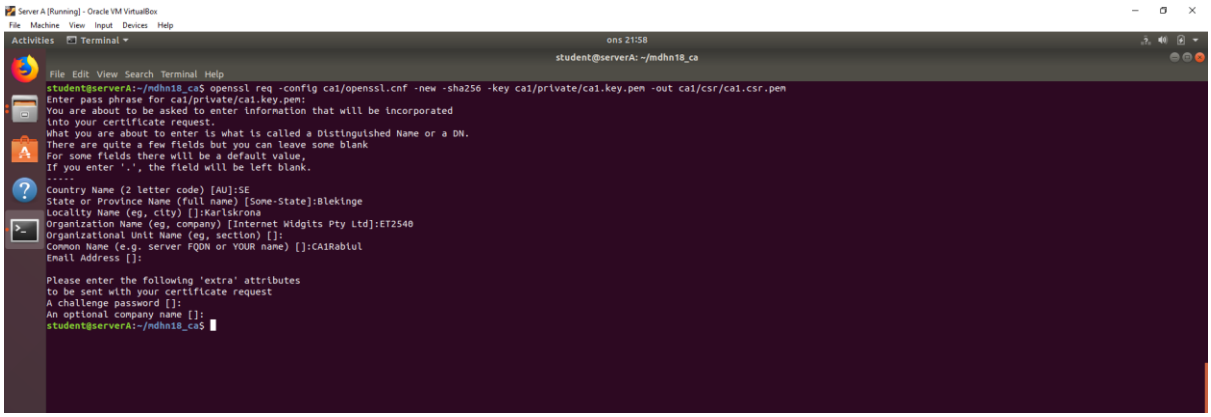
Command:
openssl x509 -noout -text -in certs/root.cert.pem



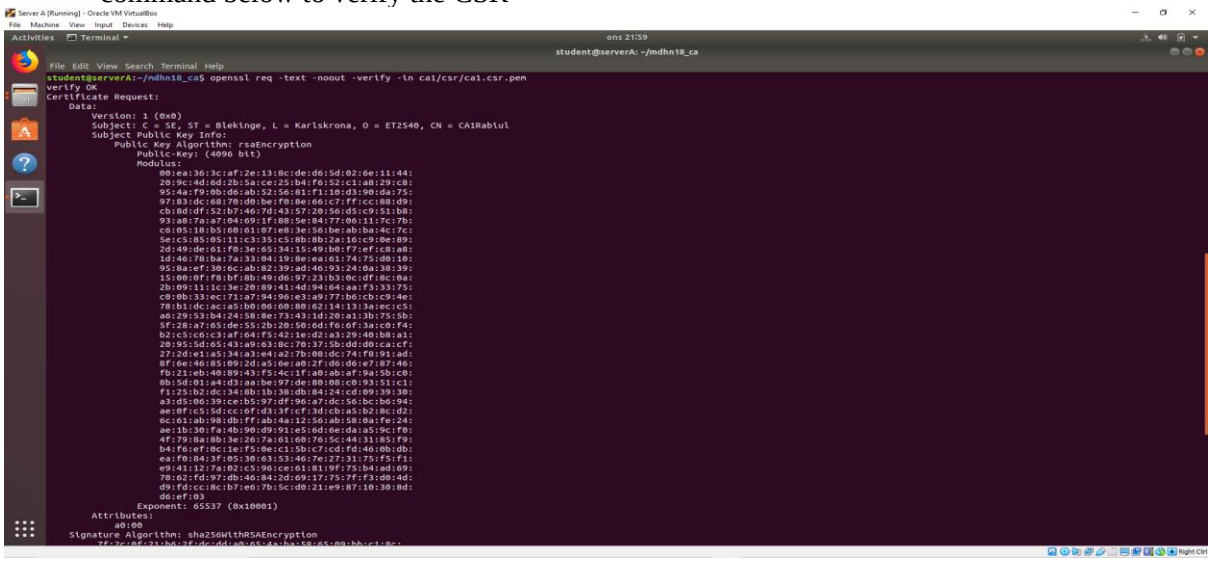


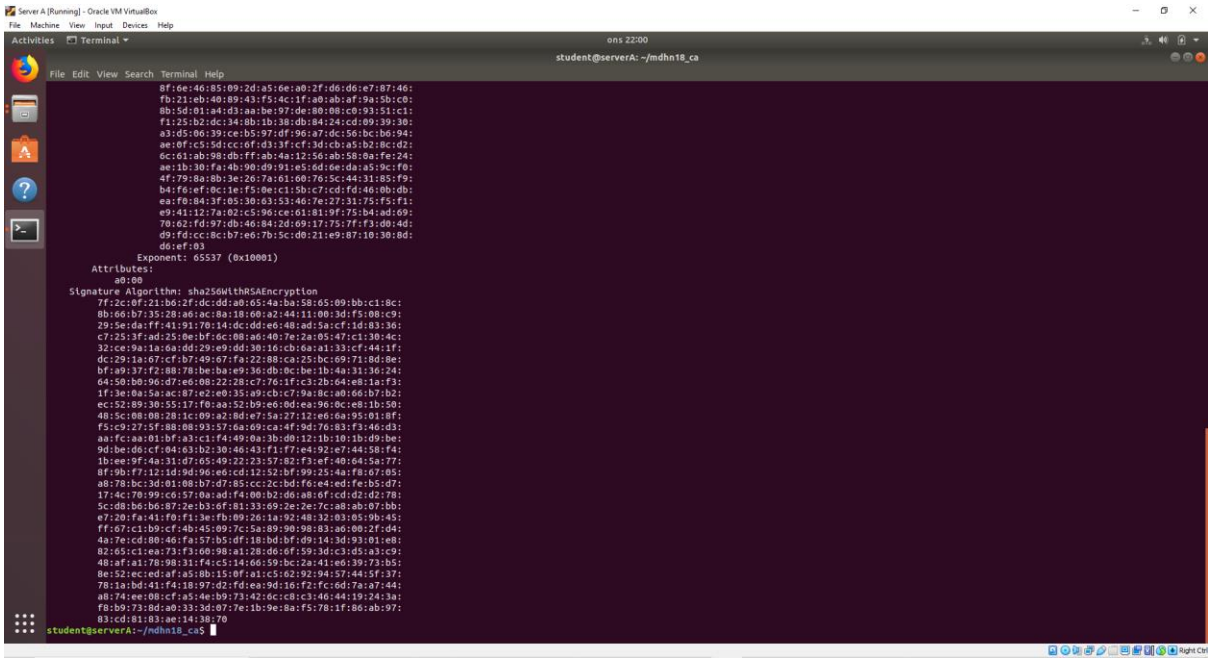
Task 8: Verify the CSR

- Following command below to create the CSR



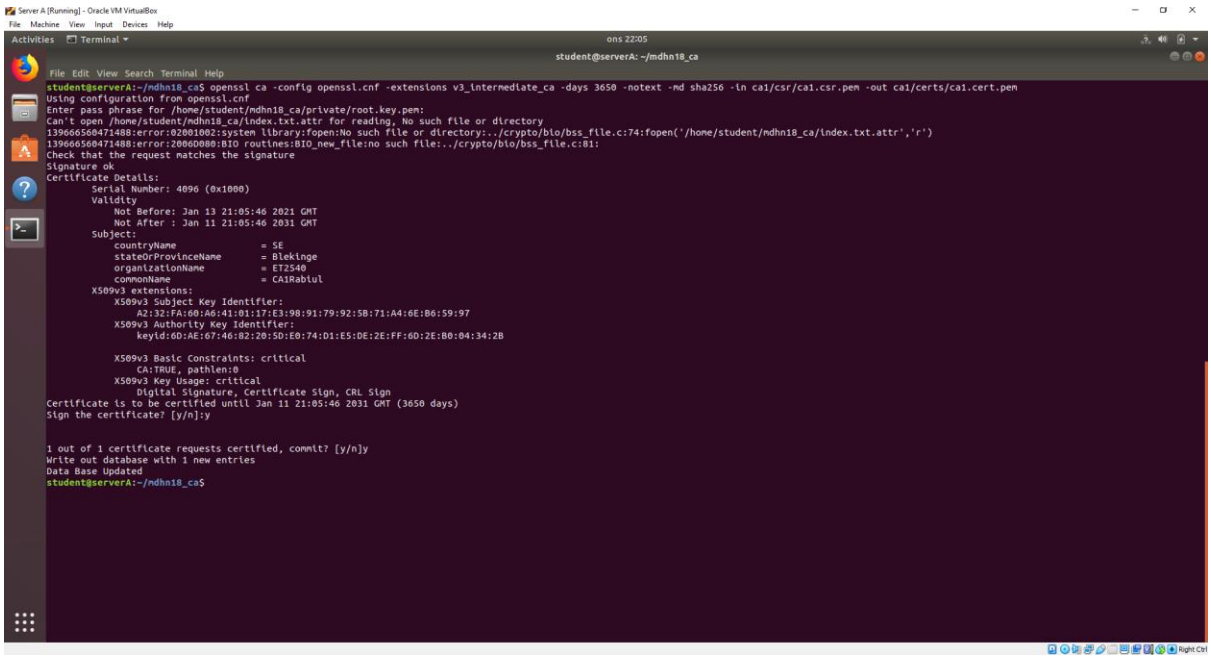
- command below to verify the CSR





Task 9: Options for intermediate CA certificate

- Create the certificate for CA1 using the CSR and the root CA is signing the certificate it is important to use *root's* OpenSSL configuration file.



Command
openssl ca -config openssl.cnf -extensions v3_intermediate_ca -days 3650 -notext -md sha256 -in ca1/csr/ca1.csr.pem -out ca1/certs/ca1.cert.pem

Description

-config openssl.cnf
specifies the configuration file to use.

-extensions v3_intermediate_ca
the section of the configuration file containing certificate extensions to be added when a certificate is issued (defaults to x509_extensions unless the -extfile option is used). If no extension section is present then, a V1 certificate is created. If the extension section is present (even if it is empty), then a V3 certificate is created.

-days 3650
the number of days to certify the certificate for.

-notext
don't output the text form of a certificate to the output file.

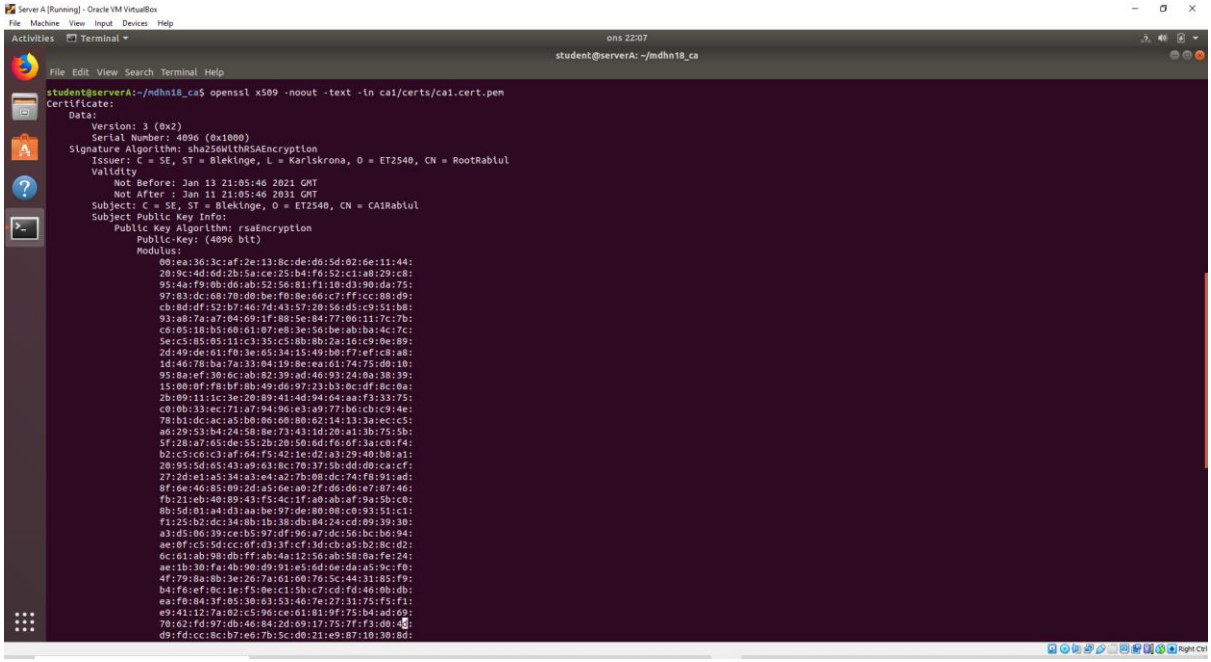
-md sha256
the message digest to use. Possible values include md5, sha1 and mdc2. This option also applies to CRLs.

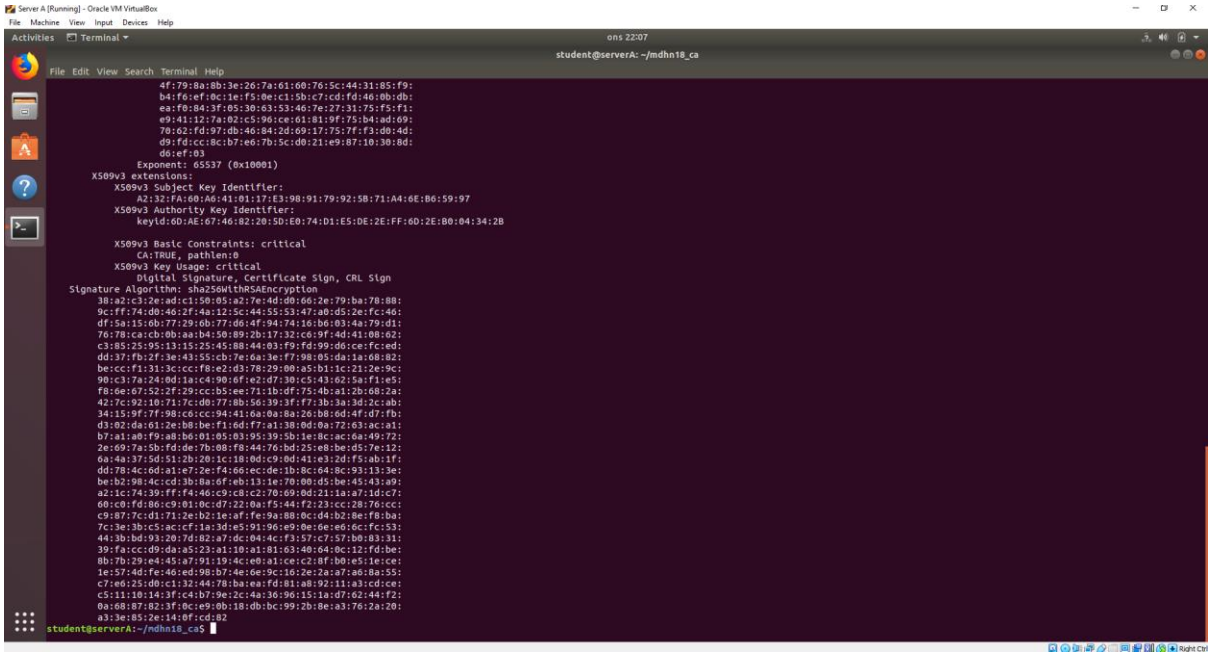
-in ca1/csr/ca1.csr.pem
an input filename containing a single certificate request to be signed by the CA.

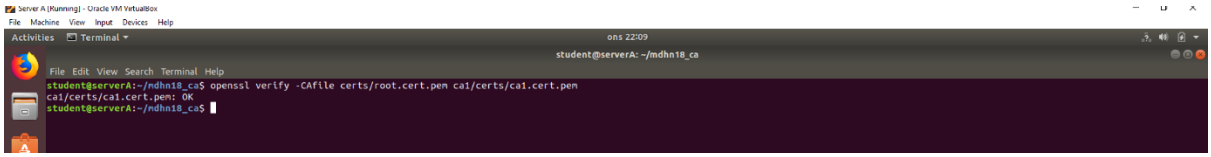
-out ca1/certs/ca1.cert.pem
the output file to output certificates to. The default is standard output. The certificate details will also be printed out to this file in PEM format (except that -spkac outputs DER format).

Task 10: Verify the certificate for CA1

- Verify the certificate for CA1, first command prints the cert and the second verifies the CA1's certificate authenticity against the root certificate.

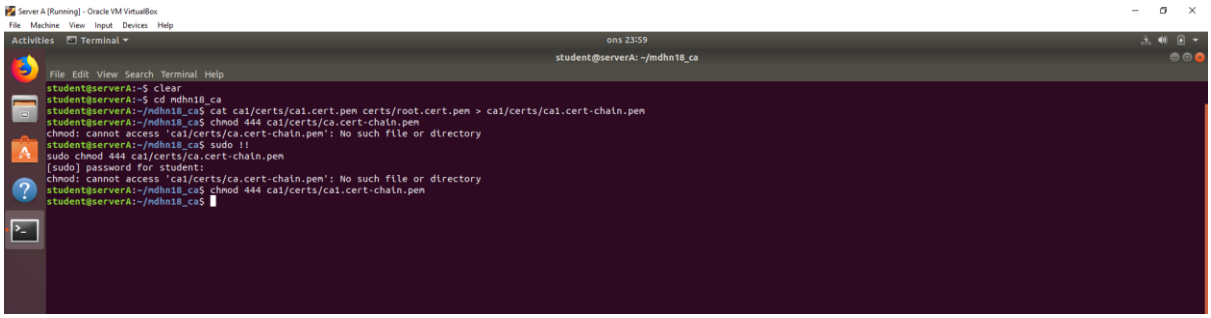






Task 11: Create server certificate

Create a certificate chain.
Ensure the certificate chain is accessible to all users on Server A



Content of the CSR

To solve this task, I follow those steps. All screen shorts are the evidence to show, did this task successfully.

Steps

1. Create an RSA private key for the server.

openssl genrsa -out ca1/private/ca1server.key.pem 2048

2. Generate a CSR using the RSA private key from the previous step.

openssl req -config ca1/openssl.cnf -new -key ca1/private/ca1server.key.pem -out ca1/csr/ca1server.csr.pem

3. Content of the CSR

openssl req -noout -text -in ca1/csr/ca1server.csr.*pem*

4. Use CA1's private key to sign the CSR and create a certificate for the server.

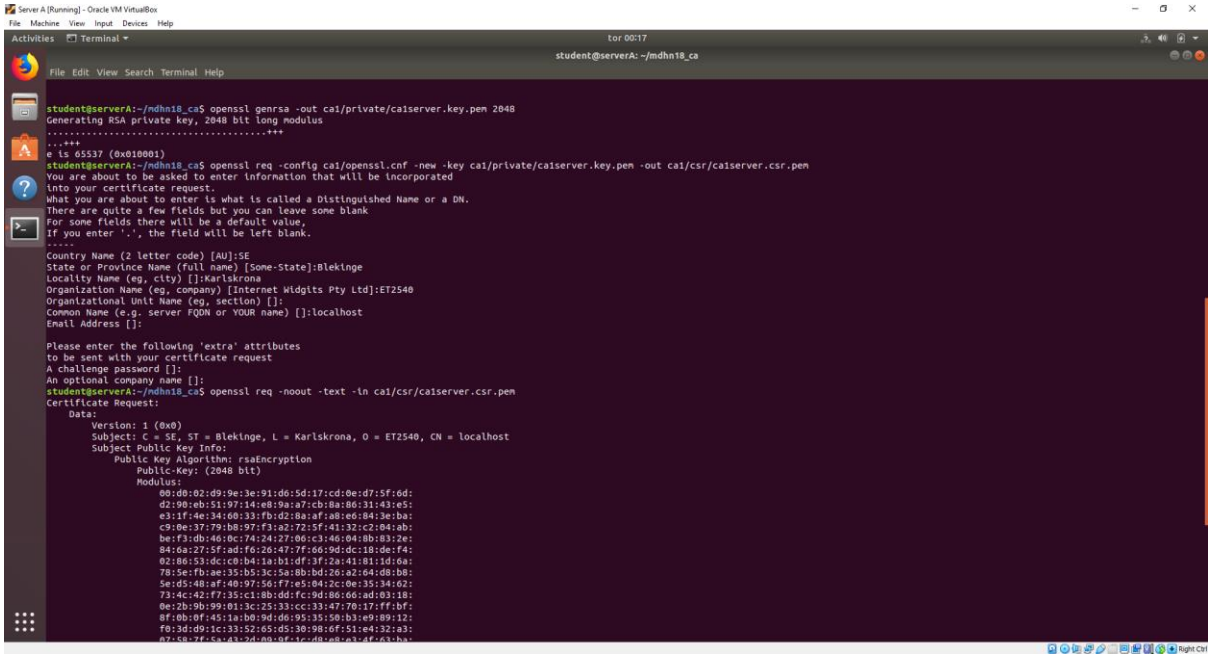
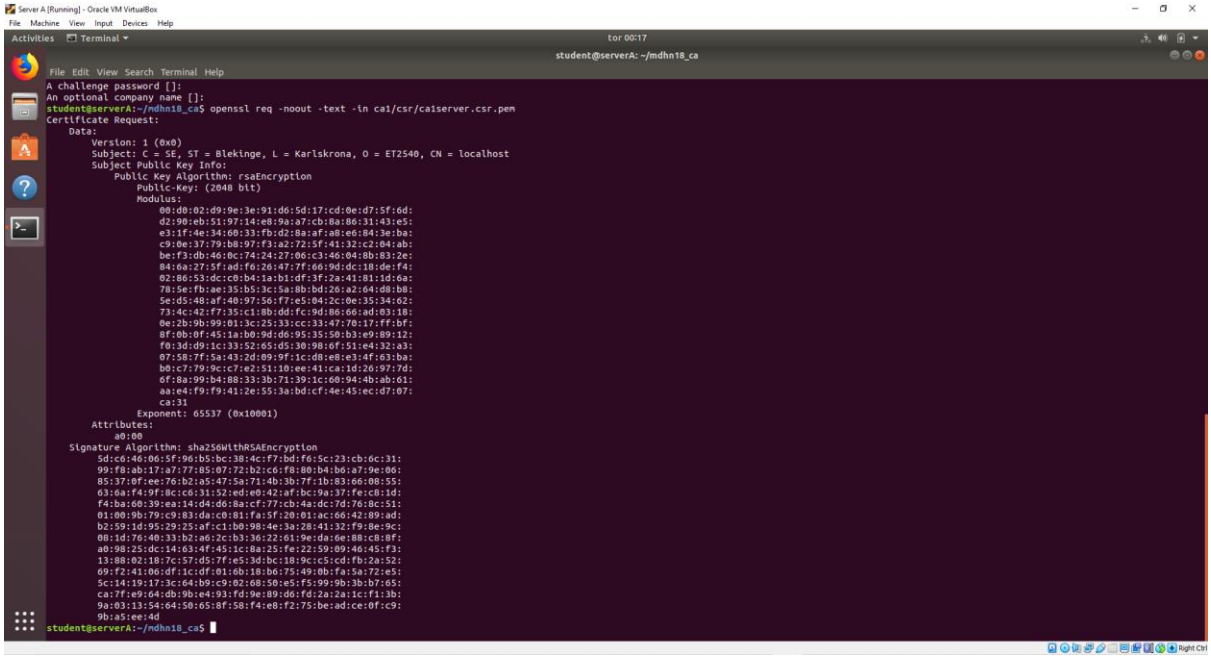
openssl ca -config ca1/openssl.cnf -extensions server_cert -days 3650 -notext -in ca1/csr/ca1server.csr.pem -out ca1/certs/ca1server.cert.pem

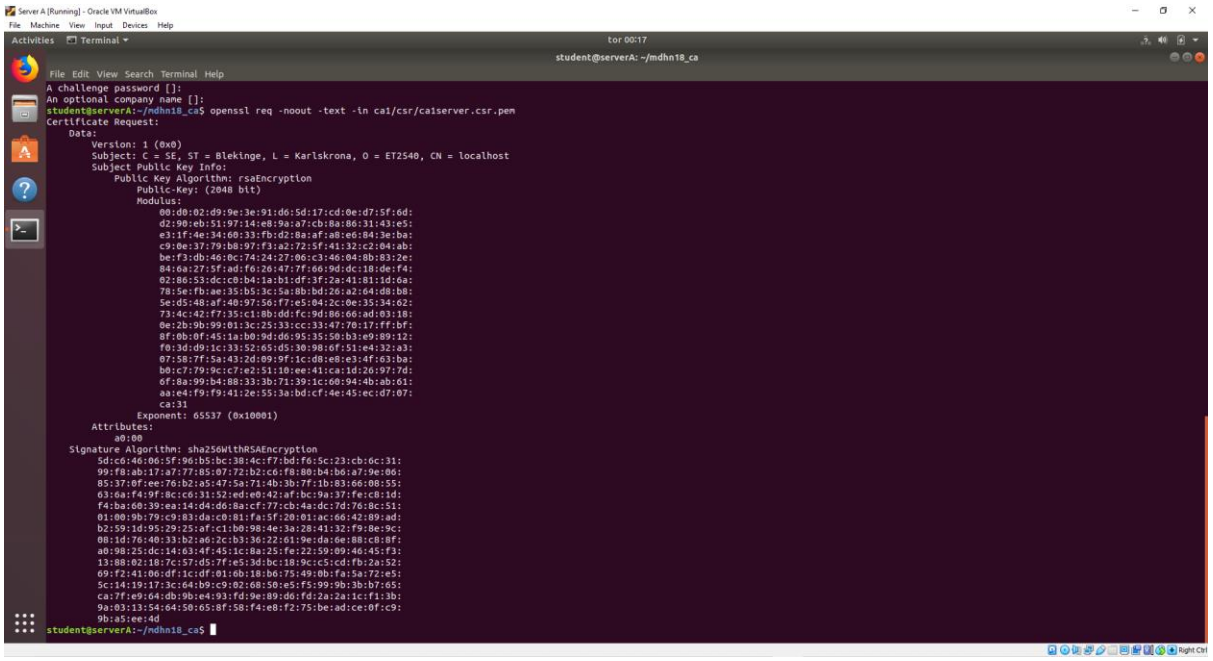
5. Content of the signed certificate.

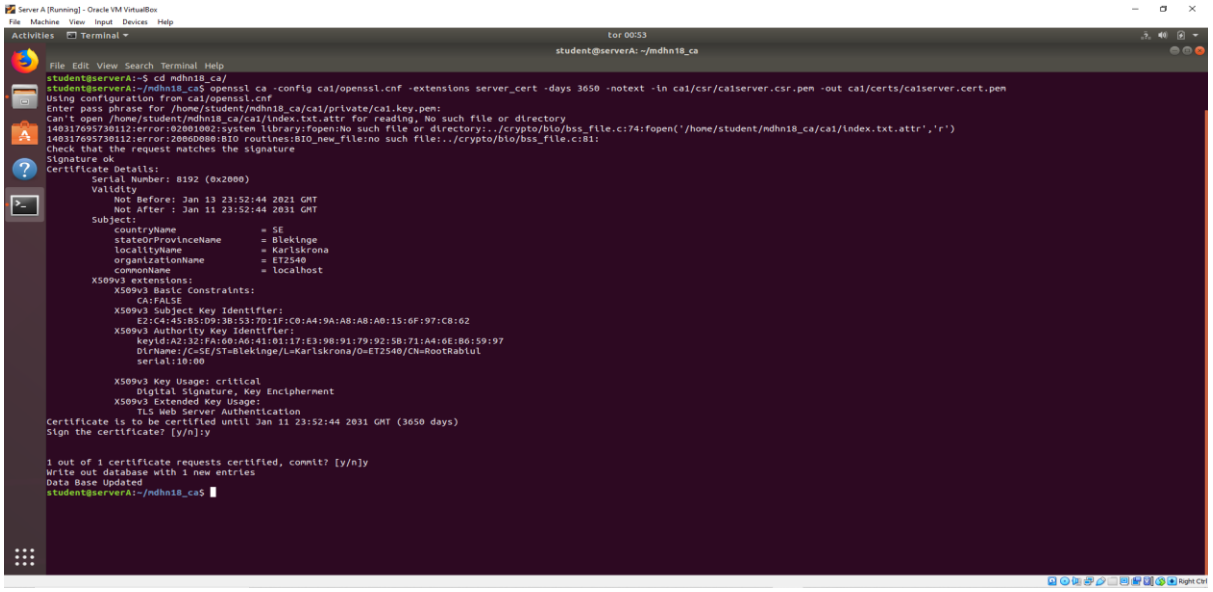
openssl x509 -noout -text -in ca1/certs/ca1server.cert.pem

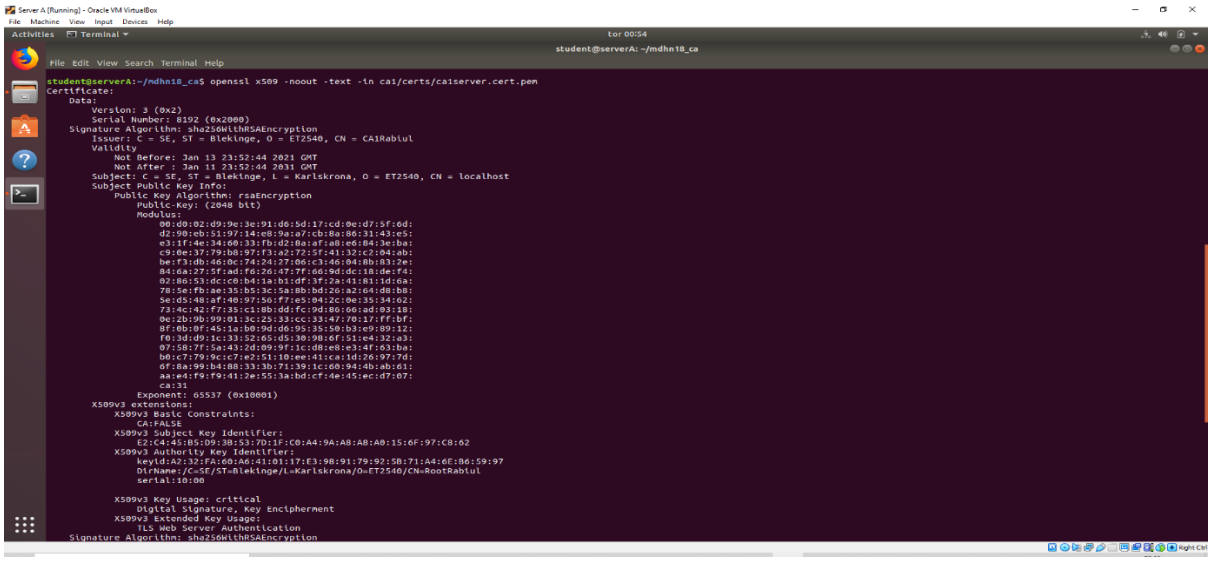
6. Verify the server certificate against certificate chain.

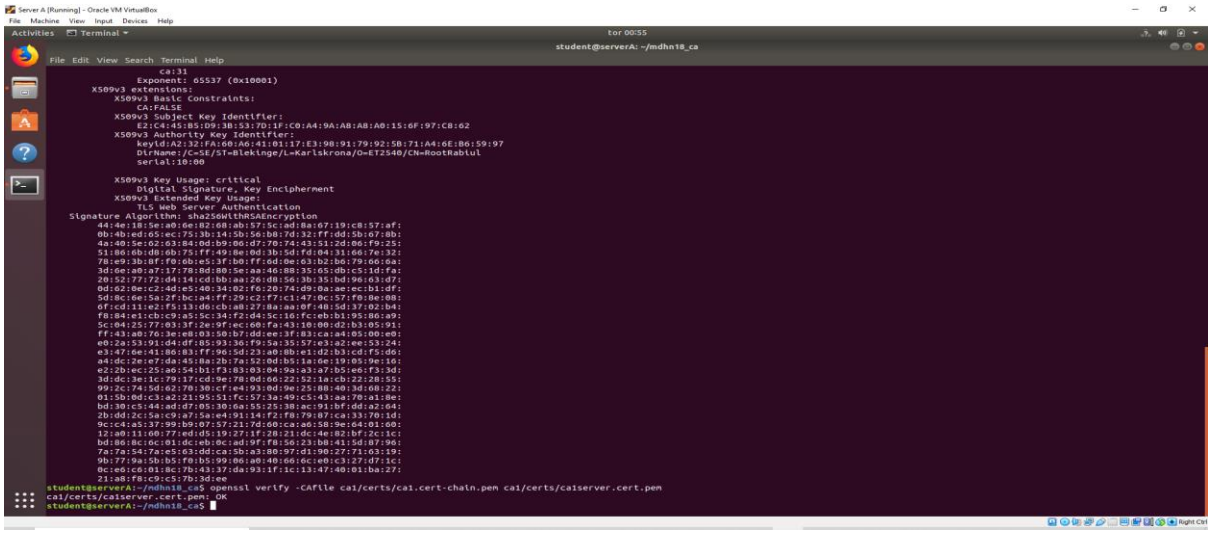
openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/ca1server.cert.pem


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 00:17
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
student@serverA:~/mdhn18_ca$ openssl genrsa -out ca1/private/ca1server.key.pem 2048
Generating RSA private key, 2048 bit long modulus
e is 65537 (0x010001)
student@serverA:~/mdhn18_ca$ openssl req -config ca1/openssl.cnf -new -key ca1/private/ca1server.key.pem -out ca1/csr/ca1server.csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:SE
State or Province Name (full name) [Some-State]:Blekinge
Locality Name (eg, city) []:Karlskrona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ET2540
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/ca1server.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = localhost
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d0:02:d9:9e:3e:91:d6:5d:17:cd:0e:d7:5f:6d:
d2:90:eb:51:97:14:e8:9a:a7:cb:8a:86:31:43:e5:
e3:1f:4e:34:60:33:fb:d2:8a:af:a8:e6:84:3e:ba:
c9:0e:37:79:b8:97:f3:a2:72:5f:41:32:c2:04:ab:
be:f3:db:46:0c:74:24:27:06:c3:46:04:8b:83:2e:
84:6a:27:5f:ad:f6:26:47:7f:66:9d:dc:18:de:f4:
02:86:53:dc:c0:b4:1a:b1:df:3f:2a:41:81:1d:6a:
78:5e:fb:ae:35:b5:3c:5a:8b:bd:26:a2:64:d8:b8:
5e:d5:48:af:40:97:56:f7:e5:04:2c:0e:35:34:62:
73:4c:42:f7:35:c1:8b:dd:fc:9d:86:66:ad:03:18:
0e:2b:9b:99:01:3c:25:33:cc:33:47:70:17:ff:bf:
8f:0b:0f:45:1a:b0:9d:d6:95:35:50:b3:e9:89:12:
f0:3d:d9:1c:33:52:65:d5:30:98:6f:51:e4:32:a3:

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 00:17
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/ca1server.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = localhost
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d0:02:d9:9e:3e:91:d6:5d:17:cd:0e:d7:5f:6d:
d2:90:eb:51:97:14:e8:9a:a7:cb:8a:86:31:43:e5:
e3:1f:4e:34:60:33:fb:d2:8a:af:a8:e6:84:3e:ba:
c9:0e:37:79:b8:97:f3:a2:72:5f:41:32:c2:04:ab:
be:f3:db:46:0c:74:24:27:06:c3:46:04:8b:83:2e:
84:6a:27:5f:ad:f6:26:47:7f:66:9d:dc:18:de:f4:
02:86:53:dc:c0:b4:1a:b1:df:3f:2a:41:81:1d:6a:
78:5e:fb:ae:35:b5:3c:5a:8b:bd:26:a2:64:d8:b8:
5e:d5:48:af:40:97:56:f7:e5:04:2c:0e:35:34:62:
73:4c:42:f7:35:c1:8b:dd:fc:9d:86:66:ad:03:18:
0e:2b:9b:99:01:3c:25:33:cc:33:47:70:17:ff:bf:
8f:0b:0f:45:1a:b0:9d:d6:95:35:50:b3:e9:89:12:
f0:3d:d9:1c:33:52:65:d5:30:98:6f:51:e4:32:a3:
07:58:7f:5a:43:2d:09:9f:1c:d8:e8:e3:4f:63:ba:
b0:c7:79:9c:c7:e2:51:10:ee:41:ca:1d:26:97:7d:
6f:8a:99:b4:88:33:3b:71:39:ic:60:94:4b:ab:61:
aa:e4:f9:f9:41:2e:55:3a:bd:cf:4e:45:ec:d7:07:
ca:31
Exponent: 65537 (0x10001)
Attributes:
a0:00
Signature Algorithm: sha256WithRSAEncryption
5d:c6:46:06:5f:96:b5:bc:38:4c:f7:bd:f6:5c:23:cb:6c:31:
99:f8:ab:17:a7:77:85:07:72:b2:c6:f8:80:b4:b6:a7:9e:06:
85:37:0f:ee:76:b2:a5:47:5a:71:4b:3b:7f:1b:83:66:08:55:
63:6a:f4:9f:8c:c6:31:52:ed:e0:42:af:bc:9a:37:fe:c8:1d:
f4:ba:60:39:ea:14:d4:d6:8a:cf:77:cb:4a:dc:7d:76:8c:51:
01:00:9b:79:c9:83:da:c0:81:fa:5f:20:01:ac:66:42:89:ad:
b2:59:1d:95:29:25:af:c1:b0:98:4e:3a:28:41:32:f9:8e:9c:
08:1d:76:40:33:b2:a6:2c:b3:36:22:61:9e:da:6e:88:c8:8f:
a0:98:25:dc:14:63:4f:45:1c:8a:25:fe:22:59:09:46:45:f3:
13:88:02:18:7c:57:d5:7f:e5:3d:bc:18:9c:c5:cd:fb:2a:52:
69:f2:41:06:df:1c:df:01:6b:18:b6:75:49:0b:fa:5a:72:e5:
5c:14:19:17:3c:64:b9:c9:02:68:50:e5:f5:99:9b:3b:b7:65:
ca:7f:e9:64:db:9b:e4:93:fd:9e:89:d6:fd:2a:2a:1c:f1:3b:
9a:03:13:54:64:50:65:8f:58:f4:e8:f2:75:be:ad:ce:0f:c9:
9b:a5:ee:4d
student@serverA:~/mdhn18_ca$


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 00:17
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/ca1server.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = localhost
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d0:02:d9:9e:3e:91:d6:5d:17:cd:0e:d7:5f:6d:
d2:90:eb:51:97:14:e8:9a:a7:cb:8a:86:31:43:e5:
e3:1f:4e:34:60:33:fb:d2:8a:af:a8:e6:84:3e:ba:
c9:0e:37:79:b8:97:f3:a2:72:5f:41:32:c2:04:ab:
be:f3:db:46:8c:74:24:27:06:c3:46:04:8b:83:2e:
84:6a:27:5f:ad:f6:26:47:7f:66:9d:dc:18:de:f4:
02:86:53:dc:c0:b4:1a:b1:df:3f:2a:41:81:1d:6a:
78:5e:fb:ae:35:b5:3c:5a:8b:bd:26:a2:64:d8:b8:
5e:d5:48:af:40:97:5e:f7:e5:04:2c:0e:35:34:02:
73:4c:42:f7:35:c1:8b:dd:fc:9d:86:66:ad:03:18:
0e:2b:9b:99:01:3c:25:33:cc:33:47:70:17:ff:bf:
8f:0b:0f:45:1a:b0:9d:d6:95:35:50:b3:e9:89:12:
f0:3d:d9:1c:33:52:65:d5:30:98:6f:51:e4:32:a3:
07:58:7f:5a:43:2d:09:9f:1c:d8:e8:e3:4f:63:ba:
b0:c7:79:9c:c7:e2:51:10:ee:41:ca:1d:26:97:7d:
6f:8a:99:b4:88:33:3b:71:39:1c:60:94:4b:ab:61:
aa:e4:f9:f9:41:2e:55:3a:bd:cf:4e:45:ec:d7:07:
ca:31
Exponent: 65537 (0x10001)
Attributes:
a0:00
Signature Algorithm: sha256WithRSAEncryption
5d:c6:46:06:5f:96:b5:bc:38:4c:f7:bd:f6:5c:23:cb:6c:31:
99:f8:ab:17:a7:77:85:07:72:b2:c6:f8:80:b4:b6:a7:9e:06:
85:37:0f:ee:76:b2:a5:47:5a:71:4b:3b:7f:1b:83:66:08:55:
63:6a:f4:9f:8c:c6:31:52:ed:e0:42:af:bc:9a:37:fe:c8:1d:
f4:ba:60:39:ea:14:d4:d6:8a:cf:77:cb:4a:dc:7d:76:8c:51:
01:00:9b:79:c9:83:da:c0:81:fa:5f:20:01:ac:66:42:89:ad:
b2:59:1d:95:29:25:af:c1:b0:98:4e:3a:28:41:32:f9:8e:9c:
08:1d:76:40:33:b2:a6:2c:b3:36:22:61:9e:da:0e:88:c8:8f:
a0:98:25:dc:14:63:4f:45:1c:8a:25:fe:22:59:09:46:45:f3:
13:88:02:18:7c:57:d5:7f:e5:3d:bc:18:9c:c5:cd:fb:2a:52:
69:f2:41:06:df:1c:df:01:6b:18:b6:75:49:0b:fa:5a:72:e5:
5c:14:19:17:3c:64:b9:c9:02:68:50:e5:f5:99:9b:3b:b7:65:
ca:7f:e9:64:db:9b:e4:93:fd:9e:89:d6:fd:2a:2a:1c:f1:3b:
9a:03:13:54:64:50:65:8f:58:f4:e8:f2:75:be:ad:ce:0f:c9:
9b:a5:ee:4d
student@serverA:~/mdhn18_ca$


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 00:53
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
student@serverA:~$ cd mdhn18_ca/
student@serverA:~/mdhn18_ca$ openssl ca -config ca1/openssl.cnf -extensions server_cert -days 3650 -notext -in ca1/csr/ca1server.csr.pem -out ca1/certs/ca1server.cert.pem
Using configuration from ca1/openssl.cnf
Enter pass phrase for /home/student/mdhn18_ca/ca1/private/ca1.key.pem:
Can't open /home/student/mdhn18_ca/ca1/index.txt.attr for reading, No such file or directory
140317695730112:error:02001002:system library:fopen:No such file or directory:../crypto/bio/bss_file.c:74:fopen('/home/student/mdhn18_ca/ca1/index.txt.attr','r')
140317695730112:error:2006D080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:81:
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 8192 (0x2000)
Validity
Not Before: Jan 13 23:52:44 2021 GMT
Not After : Jan 11 23:52:44 2031 GMT
Subject:
countryName = SE
stateOrProvinceName = Blekinge
localityName = Karlskrona
organizationName = ET2540
commonName = localhost
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
E2:C4:45:B5:D9:3B:53:7D:1F:C0:A4:9A:A8:A8:A0:15:6F:97:C8:62
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/CN=RootRablul
serial:10:00
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
Certificate is to be certified until Jan 11 23:52:44 2031 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
student@serverA:~/mdhn18_ca$



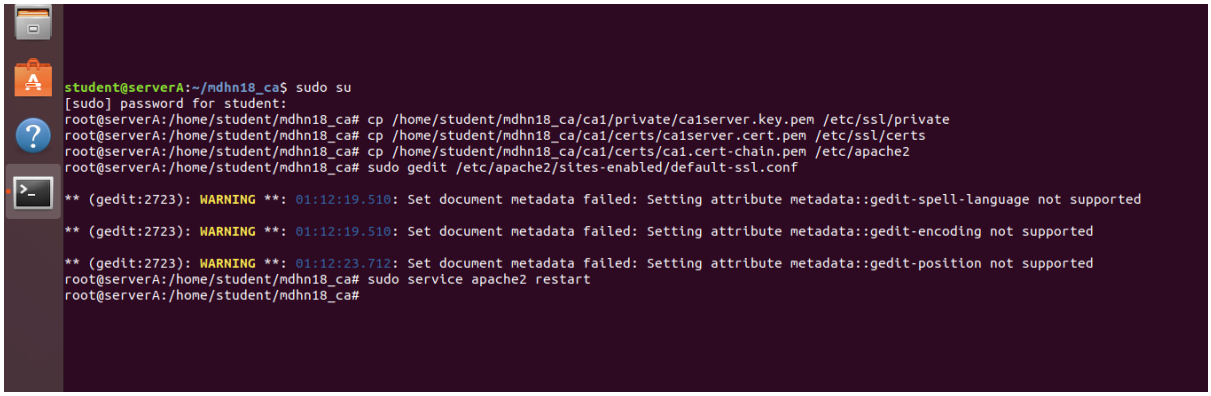


Task 12: Show your certificate in Firefox

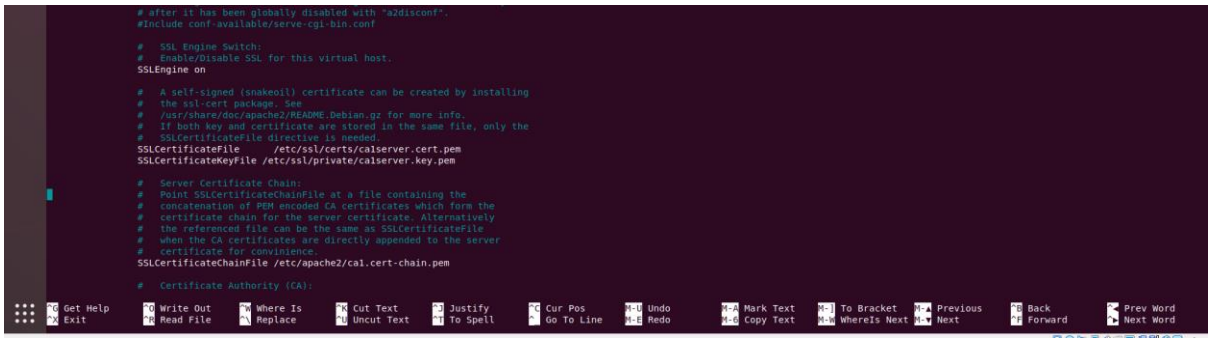
Before that task start, I need.

Configure Apache web server to use the server certificate.

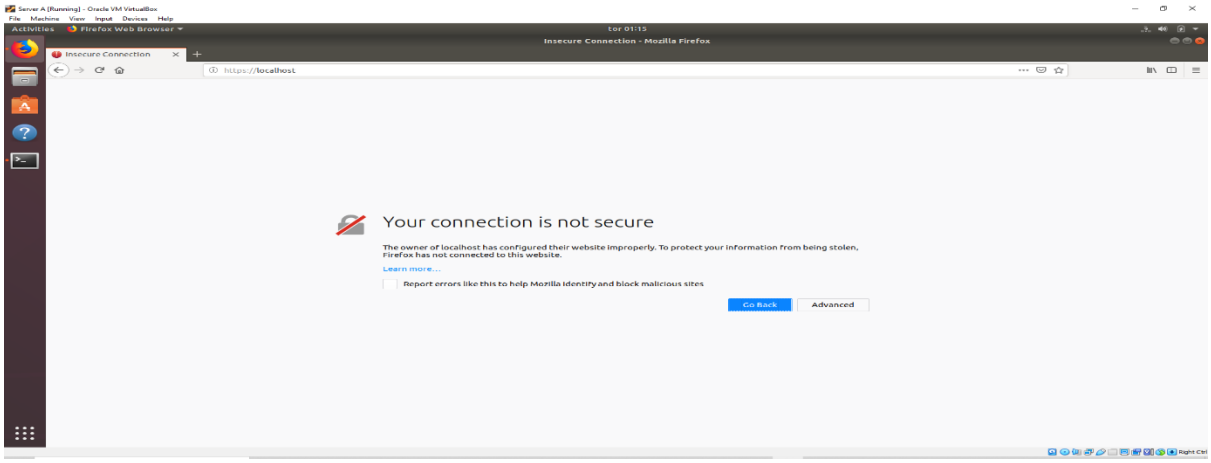
- Copy the certificate to the location, shown in a screen short.

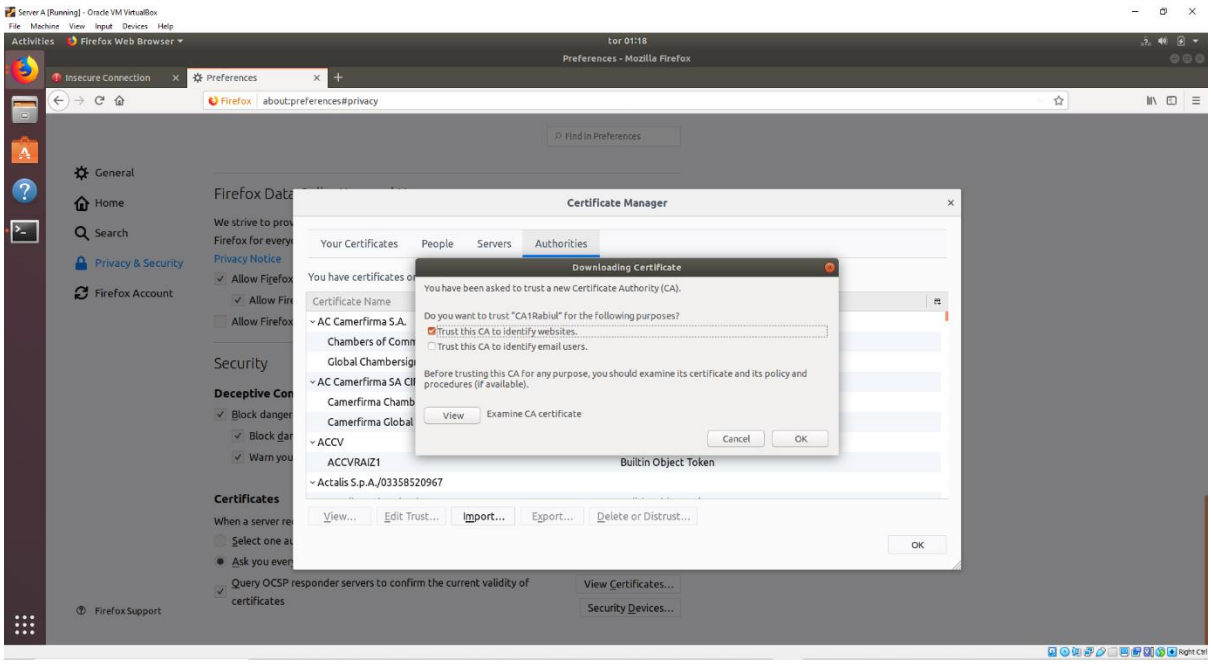


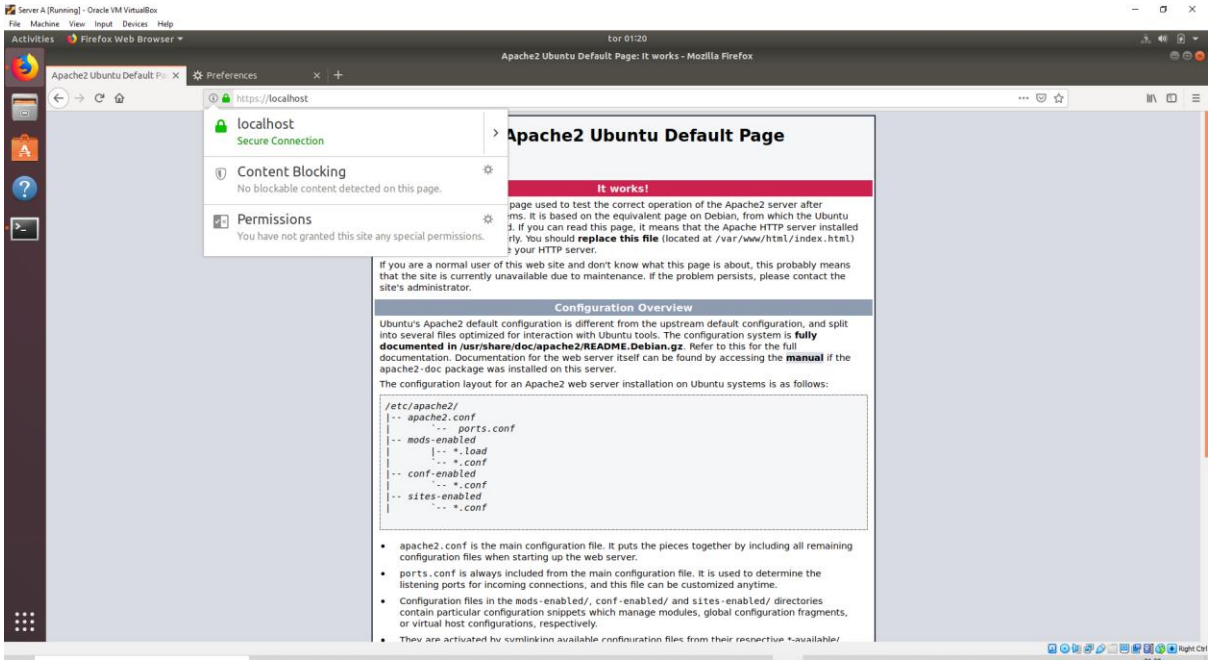
- Edit the file /etc/apache2/sites-enabled/default-ssl.conf

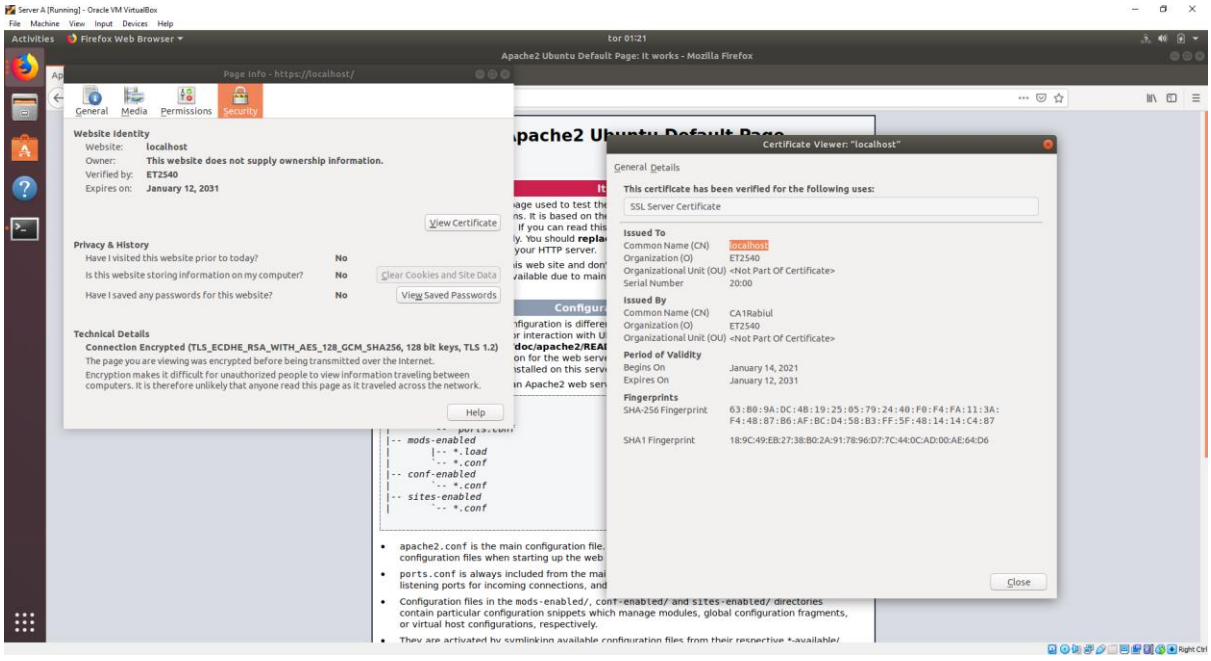


Add certificate chain to Firefox web browser.









Task 13: Create a CRL for CA1

Before that I edit the CA1 OpenSSL configuration file and add the following parameter to the serv_cert section:

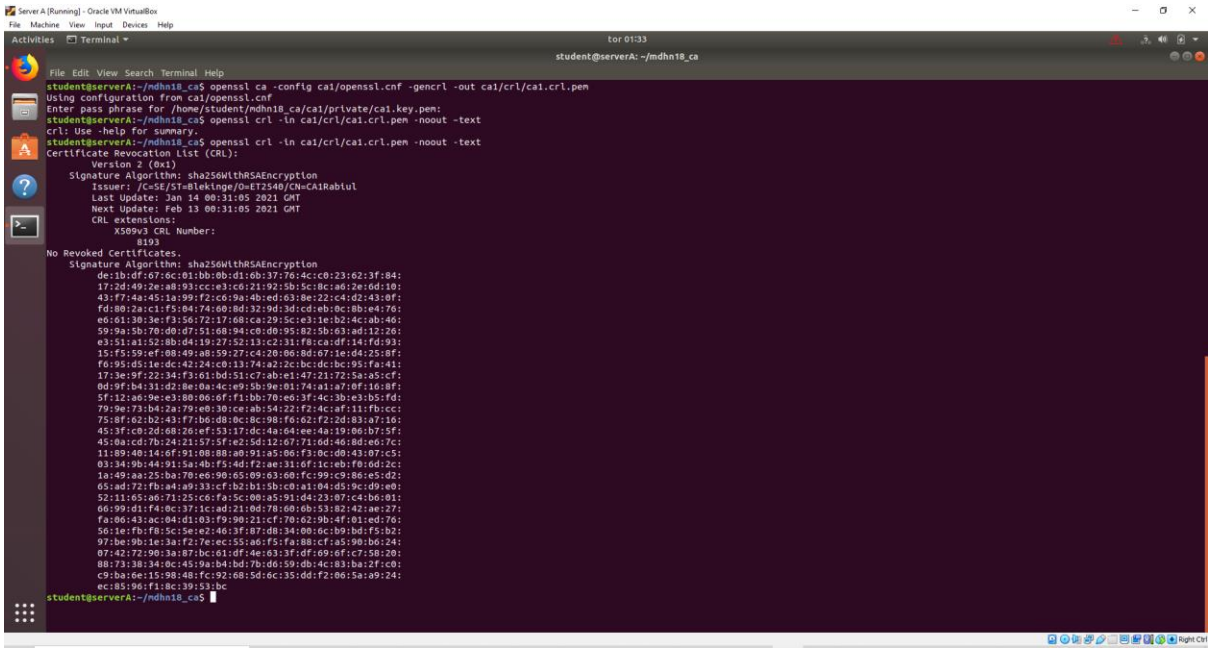
crlDistributionPoints = URI:https://localhost/ca1.crl.pem

- Following command to create a CRL for CA1

openssl ca -config ca1/openssl.cnf -gencrl -out ca1/crl/ca1.crl.pem

- Dump the contents of the CRL using the following command and include the contents in the report:

openssl crl -in ca1/crl/ca1.crl.pem -noout –text



Task 14: Revoke a certificate

I follow those steps to complete this task, and all screen short provide as proof.

Steps

1. Create an RSA private key for the server.

openssl genrsa -out ca1/private/dragos.ilie@bth.se.key.pem *2048*

2. Generate a CSR using the RSA private key from the previous step.

openssl req -config ca1/openssl.cnf -new -key ca1/private/dragos.ilie@bth.se.key.pem -out ca1/csr/dragos.ilie@bth.se.csr.pem

3. Content of the CSR.

openssl req -noout -text -in ca1/csr/dragos.ilie@bth.se.csr.pem

4. Use CA1's private key to sign the CSR and create a certificate for the server.

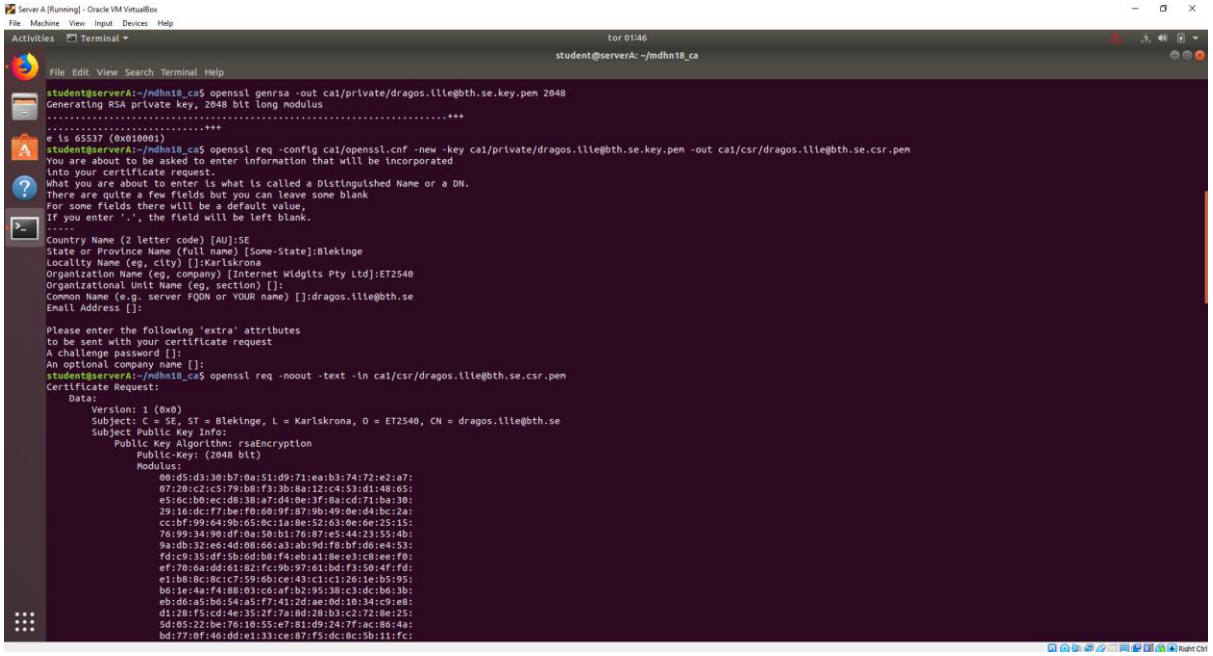
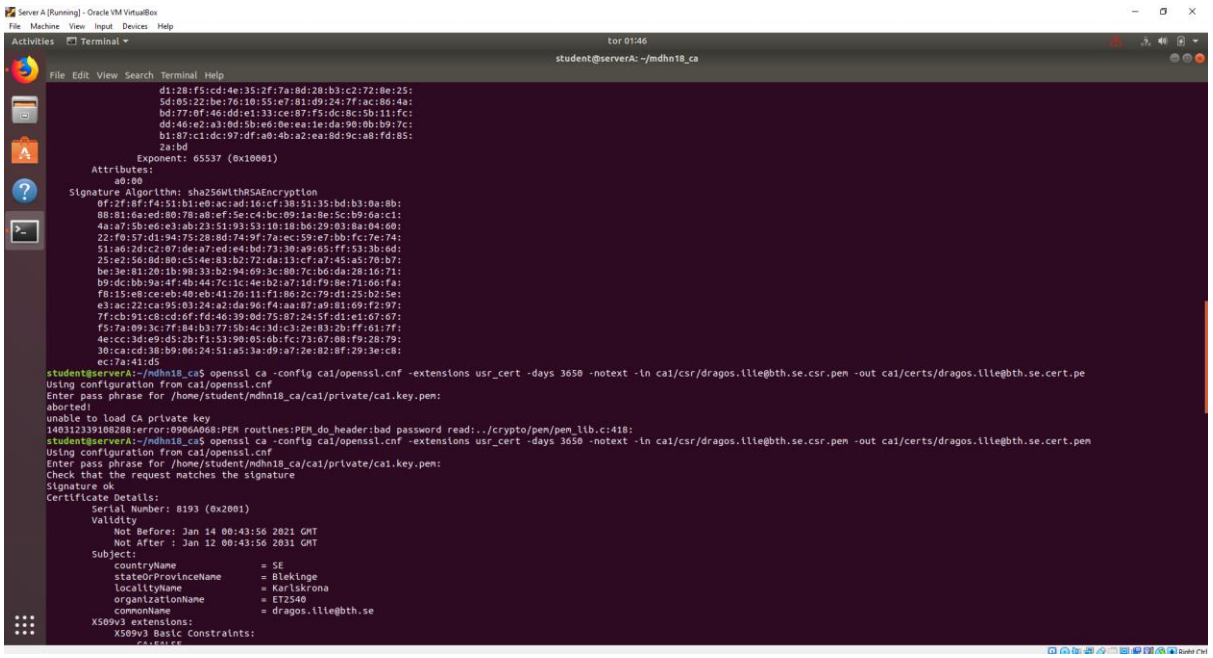
openssl ca -config ca1/openssl.cnf -extensions usr_cert -days 3650 -notext -in ca1/csr/dragos.ilie@bth.se.csr.pem -out ca1/certs/dragos.ilie@bth.se.cert.pem

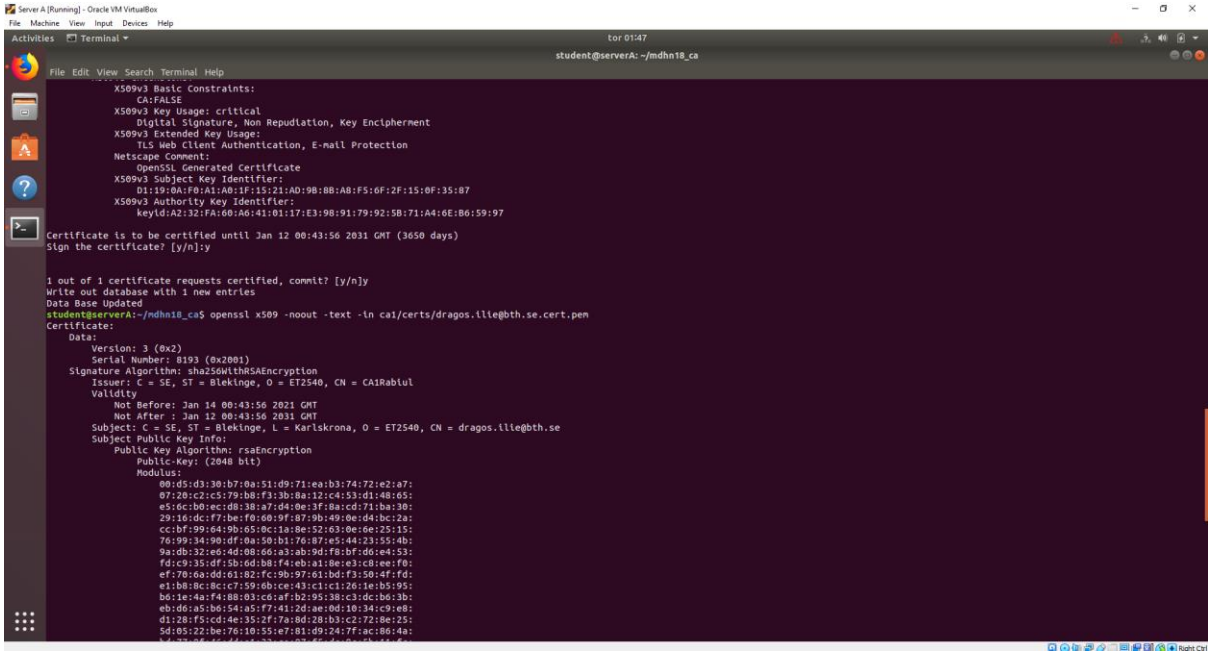
5. Content of the signed certificate.

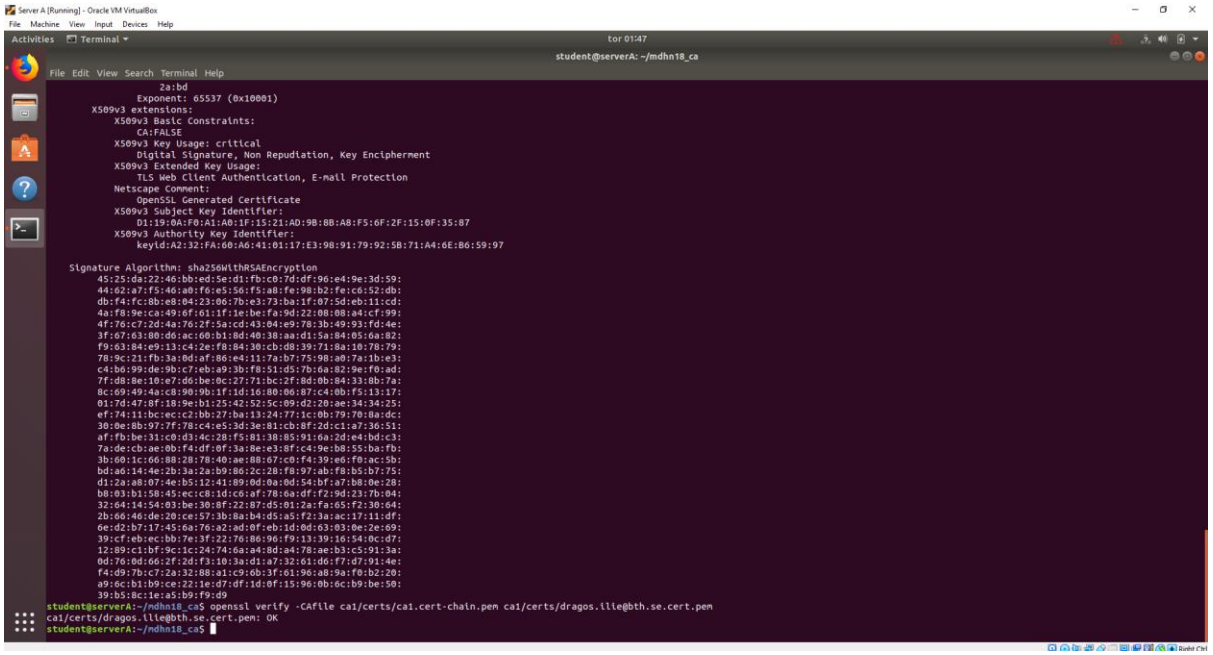
openssl x509 -noout -text -in ca1/certs/dragos.ilie@bth.se.cert.pem

6. Verify the server certificate against certificate chain.

openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/dragos.ilie@bth.se.cert.pem


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 01:46
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
student@serverA:~/mdhn18_ca$ openssl genrsa -out ca1/private/dragos.ilie@bth.se.key.pem 2048
Generating RSA private key, 2048 bit long modulus
e is 65537 (0x010001)
student@serverA:~/mdhn18_ca$ openssl req -config ca1/openssl.cnf -new -key ca1/private/dragos.ilie@bth.se.key.pem -out ca1/csr/dragos.ilie@bth.se.csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:SE
State or Province Name (full name) [Some-State]:Blekinge
Locality Name (eg, city) []:Karlskrona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ET2540
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:dragos.ilie@bth.se
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/dragos.ilie@bth.se.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = dragos.ilie@bth.se
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d5:d3:30:b7:0a:51:d9:71:ea:b3:74:72:e2:a7:
07:20:c2:c5:79:b8:f3:3b:8a:12:c4:53:d1:48:65:
e5:6c:b0:ec:d8:38:a7:d4:0e:3f:8a:cd:71:ba:30:
29:16:dc:f7:be:f0:60:9f:87:9b:49:0e:d4:bc:2a:
cc:bf:99:64:9b:65:8c:1a:8e:52:63:0e:6e:25:15:
76:99:34:90:df:0a:50:b1:76:87:e5:44:23:55:4b:
9a:db:32:e6:4d:08:66:a3:ab:9d:f8:bf:d6:e4:53:
fd:c9:35:df:5b:6d:b8:f4:eb:a1:8e:e3:c8:ee:f0:
ef:70:6a:dd:61:82:fc:9b:97:61:bd:f3:50:4f:fd:
e1:b8:8c:8c:c7:59:6b:ce:43:c1:c1:26:1e:b5:95:
b6:1e:4a:f4:88:03:c6:af:b2:95:38:c3:dc:b6:3b:
eb:d6:a5:b6:54:a5:f7:41:2d:ae:0d:10:34:c9:e8:
d1:28:f5:cd:4e:35:2f:7a:8d:28:b3:c2:72:8e:25:
5d:05:22:be:76:10:55:e7:81:d9:24:7f:ac:86:4a:
bd:77:0f:46:dd:e1:33:ce:87:f5:dc:8c:5b:11:fc:

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 01:46
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
d1:28:f5:cd:4e:35:2f:7a:8d:28:b3:c2:72:8e:25:
5d:05:22:be:76:10:55:e7:81:d9:24:7f:ac:86:4a:
bd:77:0f:46:dd:e1:33:ce:87:f5:dc:8c:5b:11:fc:
dd:46:e2:a3:0d:5b:e6:0e:ea:1e:da:90:0b:b9:7c:
b1:87:c1:dc:97:df:a0:4b:a2:ea:8d:9c:a8:fd:85:
2a:bd
Exponent: 65537 (0x10001)
Attributes:
a0:00
Signature Algorithm: sha256WithRSAEncryption
0f:2f:8f:f4:51:b1:e0:ac:ad:16:cf:38:51:35:bd:b3:0a:8b:
88:81:6a:ed:80:78:a8:ef:5e:c4:bc:09:1a:8e:5c:b9:6a:c1:
4a:a7:5b:e6:e3:ab:23:51:93:53:10:18:b6:29:03:8a:04:60:
22:f0:57:d1:94:75:28:8d:74:9f:7a:ec:59:e7:bb:fc:7e:74:
51:a6:2d:c2:07:de:a7:ed:e4:bd:73:30:a9:65:ff:53:3b:6d:
25:e2:56:8d:80:c5:4e:83:b2:72:da:13:cf:a7:45:a5:70:b7:
be:3e:81:20:1b:98:33:b2:94:69:3c:80:7c:b6:da:28:16:71:
b9:dc:bb:9a:4f:4b:44:7c:1c:4e:b2:a7:1d:f9:8e:71:66:fa:
f8:15:e8:ce:eb:40:eb:41:26:11:f1:86:2c:79:d1:25:b2:5e:
e3:ac:22:ca:95:03:24:a2:da:96:f4:aa:87:a9:81:69:f2:97:
7f:cb:91:c8:cd:6f:fd:46:39:0d:75:87:24:5f:d1:e1:67:67:
f5:7a:09:3c:7f:84:b3:77:5b:4c:3d:c3:2e:83:2b:ff:61:7f:
4e:cc:3d:e9:d5:2b:f1:53:90:05:6b:fc:73:67:08:f9:2B:79:
30:ca:cd:38:b9:06:24:51:a5:3a:d9:a7:2e:82:8f:29:3e:c8:
ec:7a:41:d5
student@serverA:~/mdhn18_ca$ openssl ca -config ca1/openssl.cnf -extensions usr_cert -days 3650 -notext -in ca1/csr/dragos.ilie@bth.se.csr.pem -out ca1/certs/dragos.ilie@bth.se.cert.pe
Using configuration from ca1/openssl.cnf
Enter pass phrase for /home/student/mdhn18_ca/ca1/private/ca1.key.pem:
aborted!
unable to load CA private key
140312339108288:error:0906A068:PEM routines:PEM_do_header:bad password read:../crypto/pem/pem_lib.c:418:
student@serverA:~/mdhn18_ca$ openssl ca -config ca1/openssl.cnf -extensions usr_cert -days 3650 -notext -in ca1/csr/dragos.ilie@bth.se.csr.pem -out ca1/certs/dragos.ilie@bth.se.cert.pem
Using configuration from ca1/openssl.cnf
Enter pass phrase for /home/student/mdhn18_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 8193 (0x2001)
Validity
Not Before: Jan 14 00:43:56 2021 GMT
Not After : Jan 12 00:43:56 2031 GMT
Subject:
countryName = SE
stateOrProvinceName = Blekinge
localityName = Karlskrona
organizationName = ET2540
commonName = dragos.ilie@bth.se
X509v3 extensions:
X509v3 Basic Constraints:

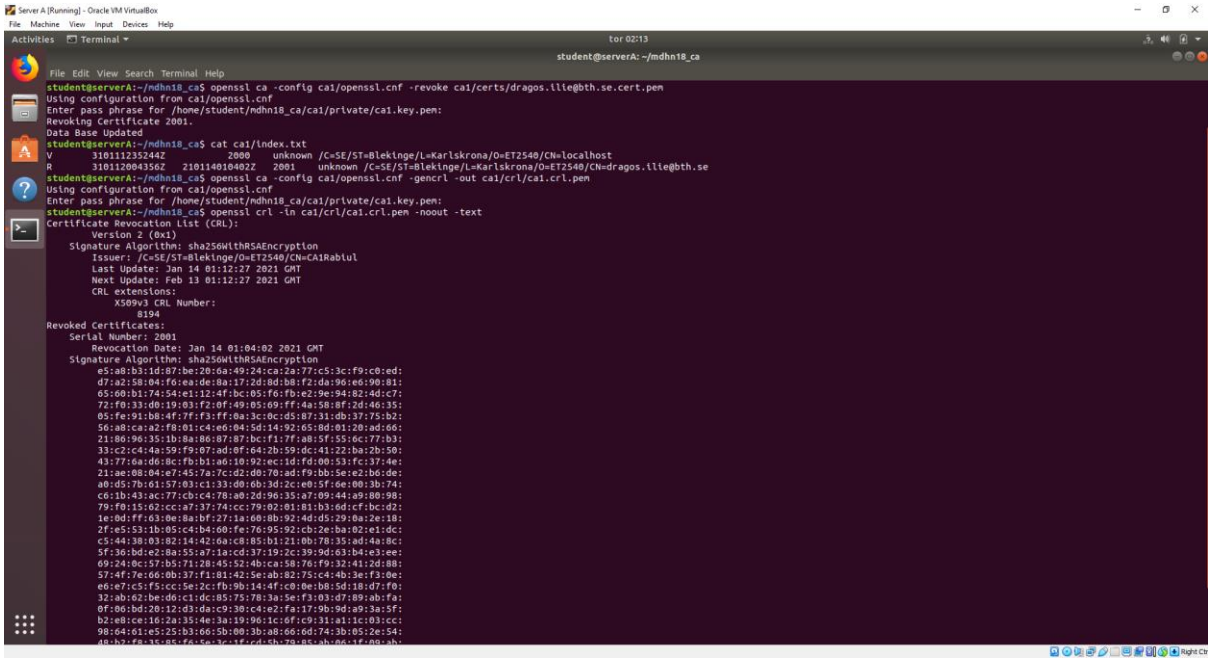

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 01:47
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
X509v3 Basic Constraints:
CA:FALSE
X509v3 Key Usage: critical
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, E-mail Protection
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
D1:19:0A:F0:A1:A6:1F:15:21:AD:9B:8B:A8:F5:6F:2F:15:0F:35:87
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
Certificate is to be certified until Jan 12 00:43:56 2031 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
student@serverA:~/mdhn18_ca$ openssl x509 -noout -text -in ca1/certs/dragos.ilie@bth.se.cert.pem
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 8193 (0x2001)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2540, CN = CA1Rabiul
Validity
Not Before: Jan 14 00:43:56 2021 GMT
Not After : Jan 12 00:43:56 2031 GMT
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = dragos.ilie@bth.se
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d5:d3:30:b7:0a:51:d9:71:ea:b3:74:72:e2:a7:
07:20:c2:c5:79:b8:f3:3b:8a:12:c4:53:d1:48:65:
e5:6c:b0:ec:d8:38:a7:d4:0e:3f:8a:cd:71:ba:30:
29:16:dc:f7:be:f0:60:9f:87:9b:49:0e:d4:bc:2a:
cc:bf:99:64:9b:65:0c:1a:8e:52:63:0e:6e:25:15:
76:99:34:90:df:0a:50:b1:76:87:e5:44:23:55:4b:
9a:db:32:e6:4d:88:66:a3:ab:9d:f8:bf:d6:e4:53:
fd:c9:35:df:5b:6d:b8:f4:eb:a1:8e:e3:c8:ee:f0:
ef:70:6a:dd:61:82:fc:9b:97:61:bd:f3:50:4f:fd:
e1:b8:8c:8c:c7:59:6b:ce:43:c1:c1:26:1e:b5:95:
b6:1e:4a:f4:88:03:c6:af:b2:95:38:c3:dc:b6:3b:
eb:d6:a5:b6:54:a5:f7:41:2d:ae:0d:10:34:c9:e8:
d1:28:f5:cd:4e:35:2f:7a:8d:28:b3:c2:72:8e:25:
5d:05:22:be:76:10:55:e7:81:d9:24:7f:ac:86:4a:


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 01:47
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
2a:bd
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Key Usage: critical
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, E-mail Protection
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
D1:19:0A:F0:A1:A6:1F:15:21:AD:9B:8B:A8:F5:6F:2F:15:0F:35:87
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
Signature Algorithm: sha256WithRSAEncryption
45:25:da:22:46:bb:ed:5e:d1:fb:c0:7d:df:96:e4:9e:3d:59:
44:62:a7:f5:46:a0:f6:e5:56:f5:a0:fe:98:b2:fe:c6:52:db:
db:f4:fc:8b:e8:04:23:06:7b:e3:73:ba:1f:07:5d:eb:11:cd:
4a:f8:9e:ca:49:6f:61:1f:1e:be:fa:9d:22:08:08:a4:cf:99:
4f:76:c7:2d:4a:76:2f:5a:cd:43:04:e9:78:3b:49:93:fd:4e:
3f:67:63:80:d6:ac:60:b1:8d:40:38:aa:d1:5a:84:05:6a:82:
f9:63:84:e9:13:c4:2e:f8:84:30:cb:d8:39:71:8a:10:78:79:
78:9c:21:fb:3a:8d:af:86:e4:11:7a:b7:75:98:a8:7a:1b:e3:
c4:b6:99:de:9b:c7:eb:a9:3b:f8:51:d5:7b:6a:82:9e:f0:ad:
7f:d8:8e:10:e7:d6:be:0c:27:71:bc:2f:8d:0b:84:33:8b:7a:
8c:69:49:4a:c8:90:9b:1f:1d:16:80:06:87:c4:0b:f5:13:17:
01:7d:47:8f:18:9e:b1:25:42:52:5c:09:d2:20:ae:34:34:25:
ef:74:11:bc:ec:c2:bb:27:ba:13:24:77:1c:0b:79:70:8a:dc:
30:8e:8b:97:7f:78:c4:e5:3d:3e:81:cb:8f:2d:c1:a7:36:51:
af:fb:be:31:c0:d3:4c:28:f5:81:38:85:91:6a:2d:e4:bd:c3:
7a:de:cb:ae:0b:f4:df:0f:3a:8e:e3:8f:c4:9e:b8:55:ba:fb:
3b:60:1c:66:88:28:78:40:ae:88:67:c0:f4:39:e6:f0:ac:5b:
bd:a6:14:4e:2b:3a:2a:b9:86:2c:28:f8:97:ab:f8:b5:b7:75:
d1:2a:a8:07:4e:b5:12:41:89:0d:0a:0d:54:bf:a7:b8:0e:28:
b8:03:b1:58:45:ec:c8:1d:c6:af:78:6a:df:f2:9d:23:7b:04:
32:64:14:54:03:be:30:8f:22:87:d5:01:2a:fa:65:f2:30:64:
2b:66:46:de:20:ce:57:3b:8a:b4:d5:a5:f2:3a:ac:17:11:df:
6e:d2:b7:17:45:6a:76:a2:ad:0f:eb:1d:0d:63:03:0e:2e:69:
39:cf:eb:ec:bb:7e:3f:22:76:86:96:f9:13:39:16:54:0c:d7:
12:89:c1:bf:9c:1c:24:74:6a:a4:8d:a4:78:ae:b3:c5:91:3a:
0d:76:0d:66:2f:2d:f3:10:3a:d1:a7:32:61:d6:f7:d7:91:4e:
f4:d9:7b:c7:2a:32:88:a1:c9:6b:3f:61:96:a8:9a:f0:b2:20:
a9:6c:b1:b9:ce:22:1e:d7:df:1d:0f:15:96:0b:6c:b9:be:50:
39:b5:8c:1e:a5:b9:f9:d9
student@serverA:~/mdhn18_ca$ openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/dragos.ilie@bth.se.cert.pem
ca1/certs/dragos.ilie@bth.se.cert.pem: OK
student@serverA:~/mdhn18_ca$

- Now, revoke the newly created certificate by entering the following command:

openssl ca -config ca1/openssl.cnf –revoke ca1/certs/dragos.ilie@bth.se.cert.pem

- Content of the file ca1/index.txt file

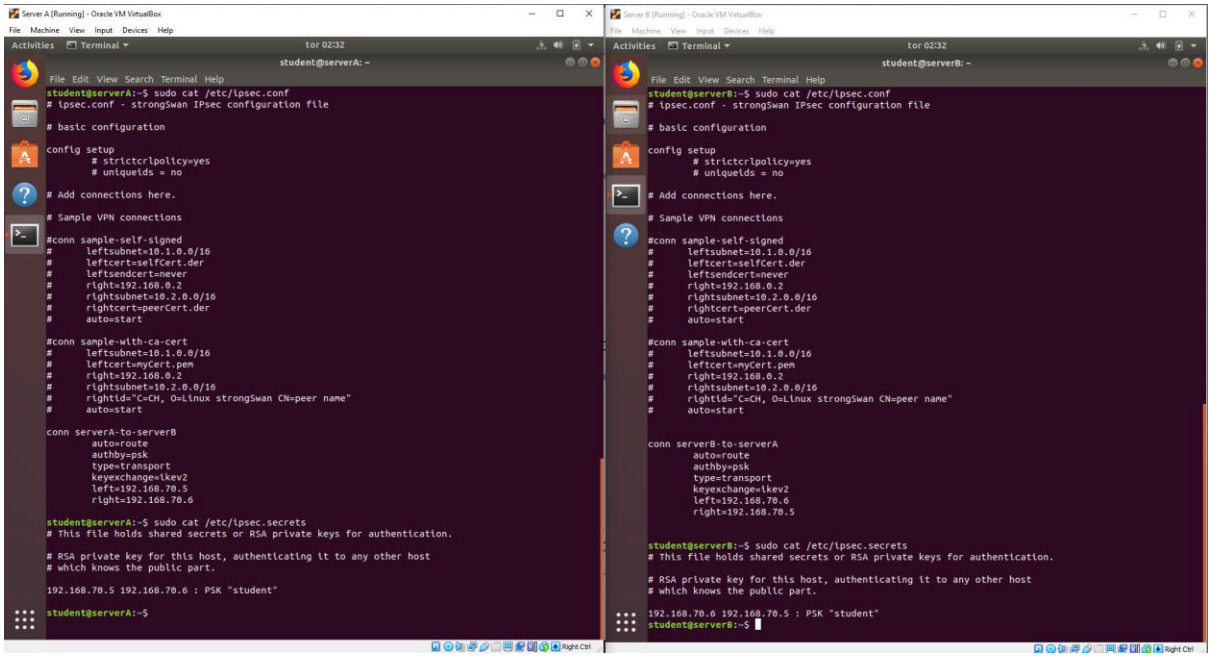


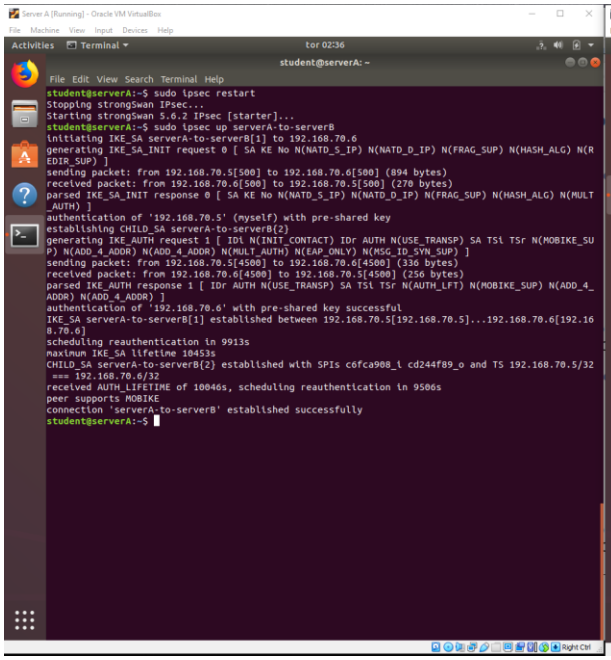
Task 15: Host-to-host transport mode VPN with PSK authentication

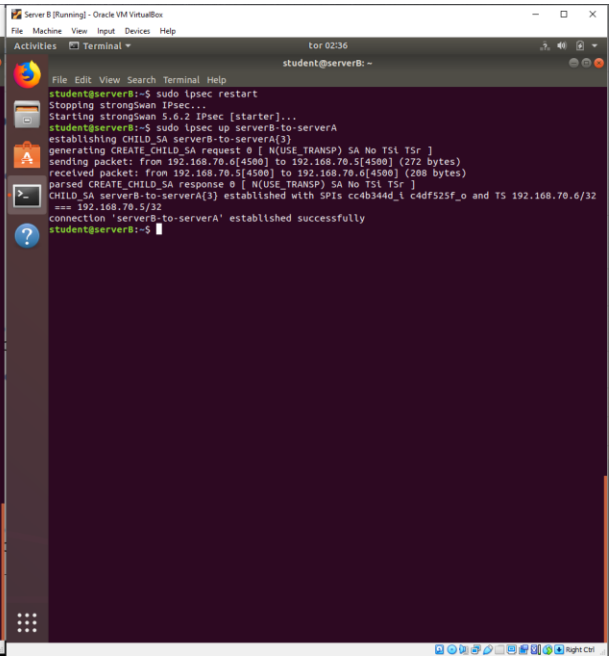
- For this task need to install by this command:

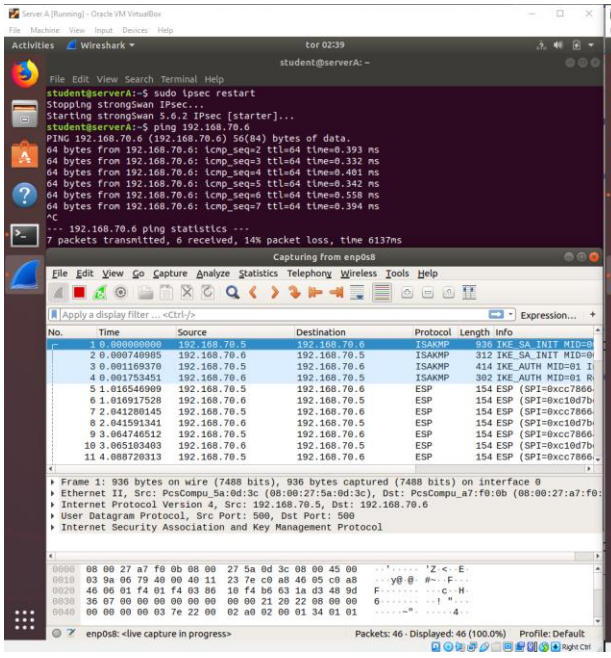
sudo apt-get install strongswan

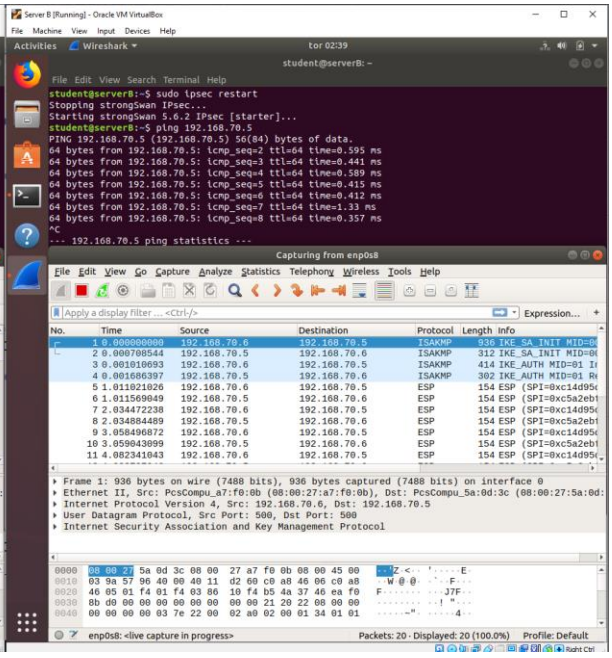
- The main configuration files for strongSwan are */etc/ipsec.conf* and */etc/ipsec.secrets*.




Server A [Running] - Oracle VM VirtualBox
Activities
Terminal
tor 02:36
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.6.2 IPsec [starter]...
student@serverA:~$ sudo ipsec up serverA-to-serverB
initiating IKE_SA serverA-to-serverB[1] to 192.168.70.6
generating IKE_SA_INIT request 0 [SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP)]
sending packet: from 192.168.70.5[500] to 192.168.70.6[500] (894 bytes)
received packet: from 192.168.70.6[500] to 192.168.70.5[500] (270 bytes)
parsed IKE_SA_INIT response 0 [SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(MULT_AUTH)]
authentication of '192.168.70.5' (myself) with pre-shared key
establishing CHILD_SA serverA-to-serverB{2}
generating IKE_AUTH request 1 [IDi N(INIT_CONTACT) IDr AUTH N(USE_TRANSP) SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP)]
sending packet: from 192.168.70.5[4500] to 192.168.70.6[4500] (336 bytes)
received packet: from 192.168.70.6[4500] to 192.168.70.5[4500] (256 bytes)
parsed IKE_AUTH response 1 [IDr AUTH N(USE_TRANSP) SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR)]
authentication of '192.168.70.6' with pre-shared key successful
IKE_SA serverA-to-serverB[1] established between 192.168.70.5[192.168.70.5]...192.168.70.6[192.168.70.6]
scheduling reauthentication in 9913s
maximum IKE_SA lifetime 10453s
CHILD_SA serverA-to-serverB{2} established with SPIs c6fca908_i cd244f89_o and TS 192.168.70.5/32 === 192.168.70.6/32
received AUTH_LIFETIME of 10046s, scheduling reauthentication in 9506s
peer supports MOBIKE
connection 'serverA-to-serverB' established successfully
student@serverA:~$

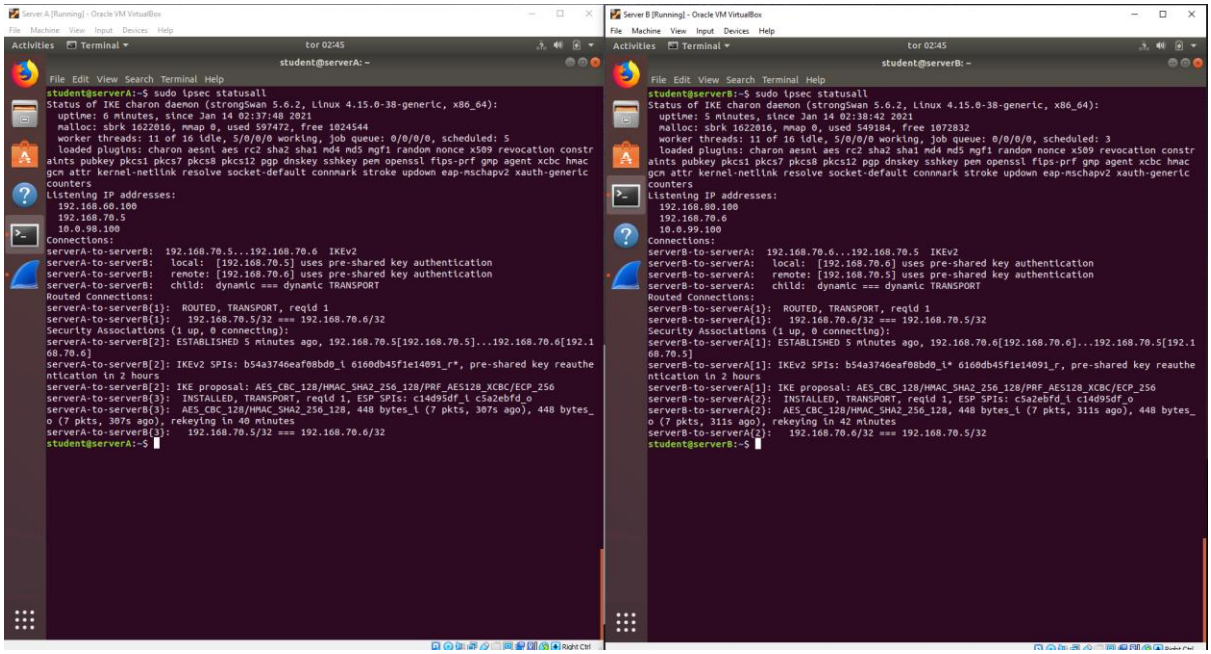

Server B [Running] - Oracle VM VirtualBox
Activities
Terminal
tor 02:36
student@serverB: ~
File Edit View Search Terminal Help
student@serverB:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.6.2 IPsec [starter]...
student@serverB:~$ sudo ipsec up serverB-to-serverA
establishing CHILD_SA serverB-to-serverA{3}
generating CREATE_CHILD_SA request 0 [N(USE_TRANSP) SA No TSi TSr]
sending packet: from 192.168.70.6[4500] to 192.168.70.5[4500] (272 bytes)
received packet: from 192.168.70.5[4500] to 192.168.70.6[4500] (208 bytes)
parsed CREATE_CHILD_SA response 0 [N(USE_TRANSP) SA No TSi TSr]
CHILD_SA serverB-to-serverA{3} established with SPIs cc4b344d_i c4df525f_o and TS 192.168.70.6/32 === 192.168.70.5/32
connection 'serverB-to-serverA' established successfully
student@serverB:~$

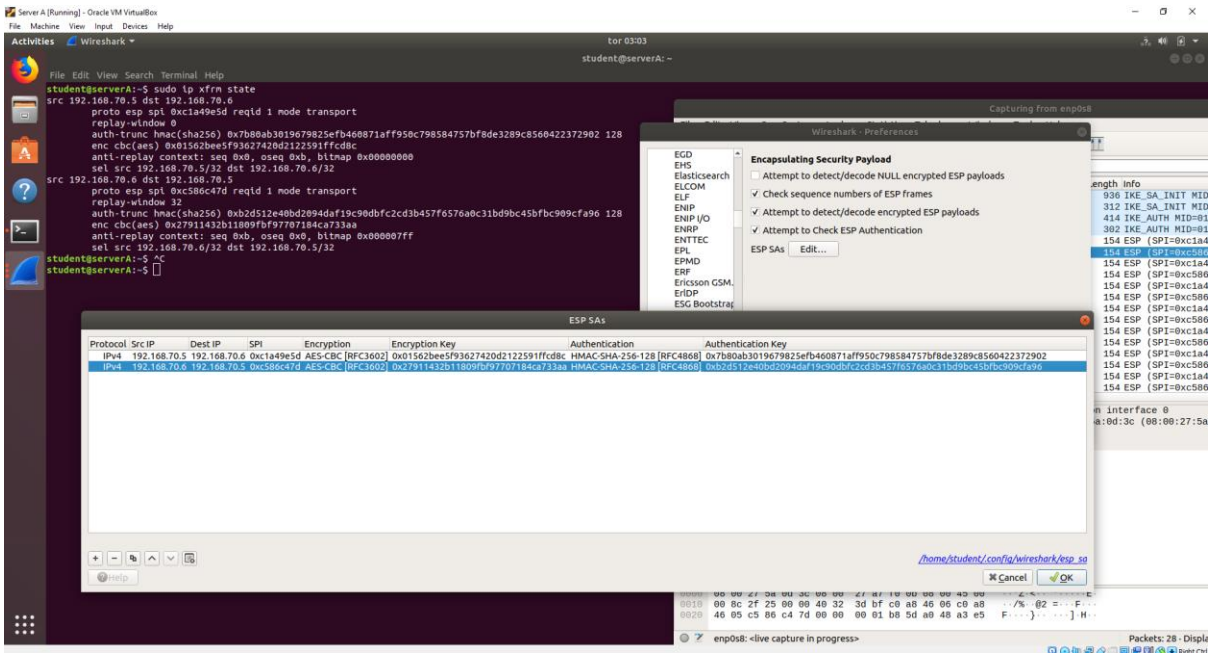

Server A [Running] - Oracle VM VirtualBox
Activities
Wireshark
tor 02:39
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.6.2 IPsec [starter]...
student@serverA:~$ ping 192.168.70.6
PING 192.168.70.6 (192.168.70.6) 56(84) bytes of data.
64 bytes from 192.168.70.6: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 192.168.70.6: icmp_seq=3 ttl=64 time=0.332 ms
64 bytes from 192.168.70.6: icmp_seq=4 ttl=64 time=0.401 ms
64 bytes from 192.168.70.6: icmp_seq=5 ttl=64 time=0.342 ms
64 bytes from 192.168.70.6: icmp_seq=6 ttl=64 time=0.558 ms
64 bytes from 192.168.70.6: icmp_seq=7 ttl=64 time=0.394 ms
^C
--- 192.168.70.6 ping statistics ---
7 packets transmitted, 6 received, 14% packet loss, time 6137ms
Capturing from enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000000 192.168.70.5 192.168.70.6 ISAKMP 936 IKE_SA_INIT MID=0
2 0.000740985 192.168.70.6 192.168.70.5 ISAKMP 312 IKE_SA_INIT MID=0
3 0.001169370 192.168.70.5 192.168.70.6 ISAKMP 414 IKE_AUTH MID=01 I
4 0.001753451 192.168.70.6 192.168.70.5 ISAKMP 302 IKE_AUTH MID=01 R
5 1.016546909 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xcc7866
6 1.016917528 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc10d7b
7 2.041280145 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xcc7866
8 2.041591341 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc10d7b
9 3.064651202 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xcc7866
10 3.065103403 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc10d7b
11 4.088720313 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xcc7866
Frame 1: 936 bytes on wire (7488 bits), 936 bytes captured (7488 bits) on interface 0
Ethernet II, Src: PcsCompu_5a:0d:3c (08:00:27:5a:0d:3c), Dst: PcsCompu_a7:f0:0b (08:00:27:a7:f0:
Internet Protocol Version 4, Src: 192.168.70.5, Dst: 192.168.70.6
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
0000 08 00 27 a7 f0 0b 08 00 27 5a 0d 3c 08 00 45 00
0010 03 9a 06 79 40 00 40 11 23 7e c0 a8 46 05 c0 a8
0020 46 06 01 f4 01 f4 03 86 10 f4 b6 63 1a d3 48 9d
0030 36 07 00 00 00 00 00 00 00 00 21 20 22 08 00 00
0040 00 00 00 00 03 7e 22 00 02 a0 02 00 01 34 01 01
enp0s8: <live capture in progress>
Packets: 46 · Displayed: 46 (100.0%)
Profile: Default

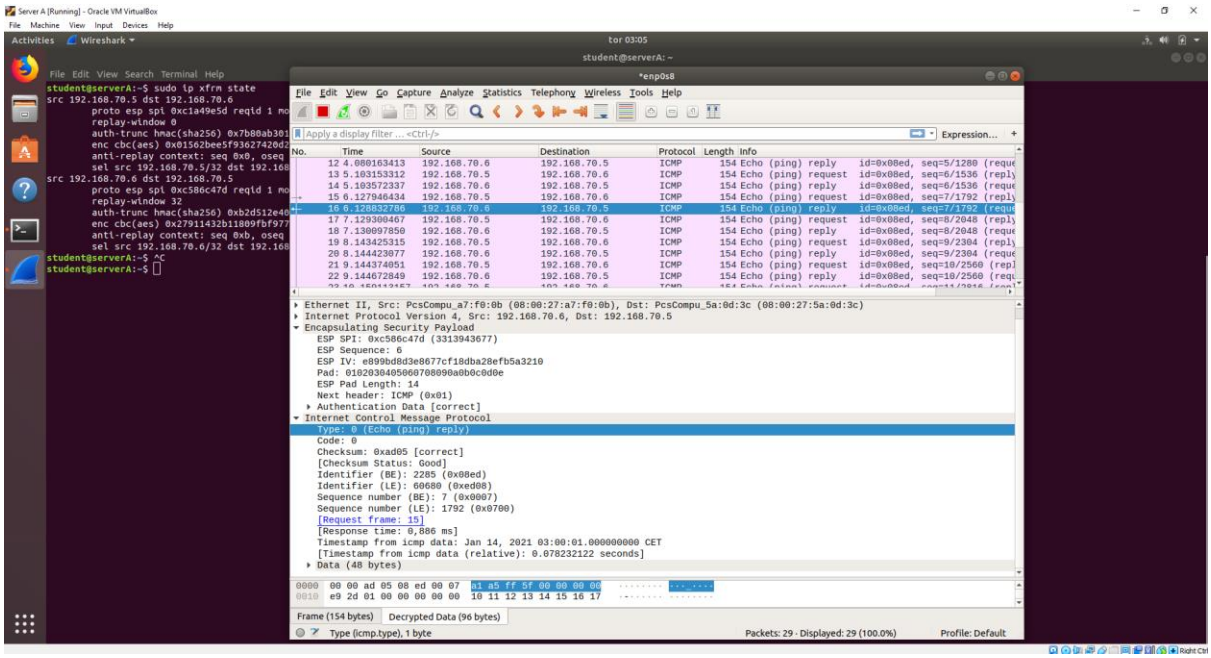

Server B [Running] - Oracle VM VirtualBox
Activities
Wireshark
tor 02:39
student@serverB: ~
File Edit View Search Terminal Help
student@serverB:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.6.2 IPsec [starter]...
student@serverB:~$ ping 192.168.70.5
PING 192.168.70.5 (192.168.70.5) 56(84) bytes of data.
64 bytes from 192.168.70.5: icmp_seq=2 ttl=64 time=0.595 ms
64 bytes from 192.168.70.5: icmp_seq=3 ttl=64 time=0.441 ms
64 bytes from 192.168.70.5: icmp_seq=4 ttl=64 time=0.589 ms
64 bytes from 192.168.70.5: icmp_seq=5 ttl=64 time=0.415 ms
64 bytes from 192.168.70.5: icmp_seq=6 ttl=64 time=0.412 ms
64 bytes from 192.168.70.5: icmp_seq=7 ttl=64 time=1.33 ms
64 bytes from 192.168.70.5: icmp_seq=8 ttl=64 time=0.357 ms
^C
--- 192.168.70.5 ping statistics ---
Capturing from enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000000 192.168.70.6 192.168.70.5 ISAKMP 936 IKE_SA_INIT MID=0
2 0.000708544 192.168.70.5 192.168.70.6 ISAKMP 312 IKE_SA_INIT MID=0
3 0.001010693 192.168.70.6 192.168.70.5 ISAKMP 414 IKE_AUTH MID=01 I
4 0.001686397 192.168.70.5 192.168.70.6 ISAKMP 302 IKE_AUTH MID=01 R
5 1.011021026 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc14d95
6 1.011569049 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xc5a2eb
7 2.034472238 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc14d95
8 2.034884489 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xc5a2eb
9 3.058496872 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc14d95
10 3.059043099 192.168.70.5 192.168.70.6 ESP 154 ESP (SPI=0xc5a2eb
11 4.082341043 192.168.70.6 192.168.70.5 ESP 154 ESP (SPI=0xc14d95
Frame 1: 936 bytes on wire (7488 bits), 936 bytes captured (7488 bits) on interface 0
Ethernet II, Src: PcsCompu_a7:f0:0b (08:00:27:a7:f0:0b), Dst: PcsCompu_5a:0d:3c (08:00:27:5a:0d:
Internet Protocol Version 4, Src: 192.168.70.6, Dst: 192.168.70.5
User Datagram Protocol, Src Port: 500, Dst Port: 500
Internet Security Association and Key Management Protocol
0000 08 00 27 5a 0d 3c 08 00 27 a7 f0 0b 08 00 45 00
0010 03 9a 57 96 40 00 40 11 d2 60 c0 a8 46 06 c0 a8
0020 46 05 01 f4 01 f4 03 86 10 f4 b5 4a 37 46 ea f0
0030 8b d0 00 00 00 00 00 00 00 00 21 20 22 08 00 00
0040 00 00 00 00 03 7e 22 00 02 a0 02 00 01 34 01 01
enp0s8: <live capture in progress>
Packets: 20 · Displayed: 20 (100.0%)
Profile: Default

Task 16: Decrypt traffic with Wireshark

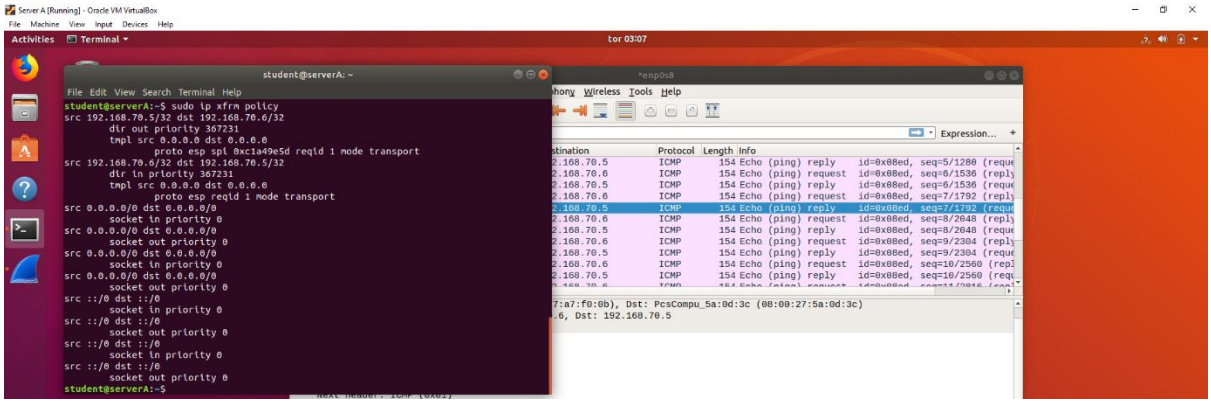
- For this task, decrypt traffic with wireshark shown below screen shorts.







Task 17: List the entries in the SPD



Description

in

It specifies the incoming SAs from ipsec.

tmpl

It defines how ipsec should encapsulate matching packets.

esp

It shows that the type of protocol was esp.

reqid

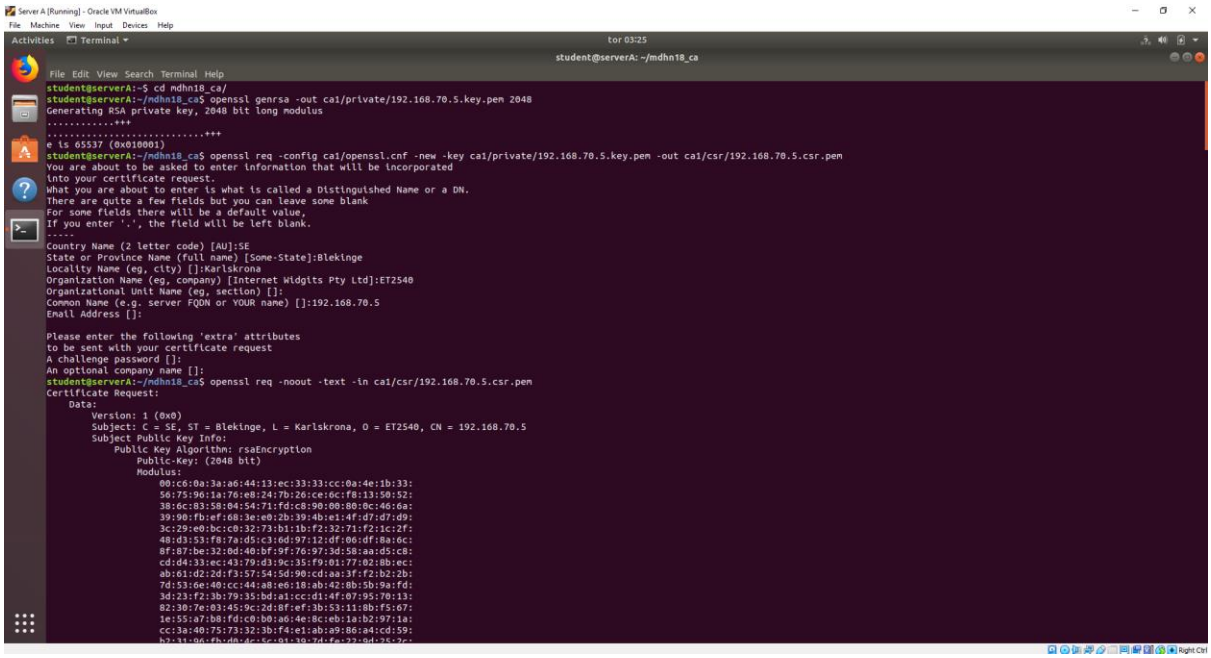
It links the Security Policy Database (SPD) and the Security Association Database (SAD).

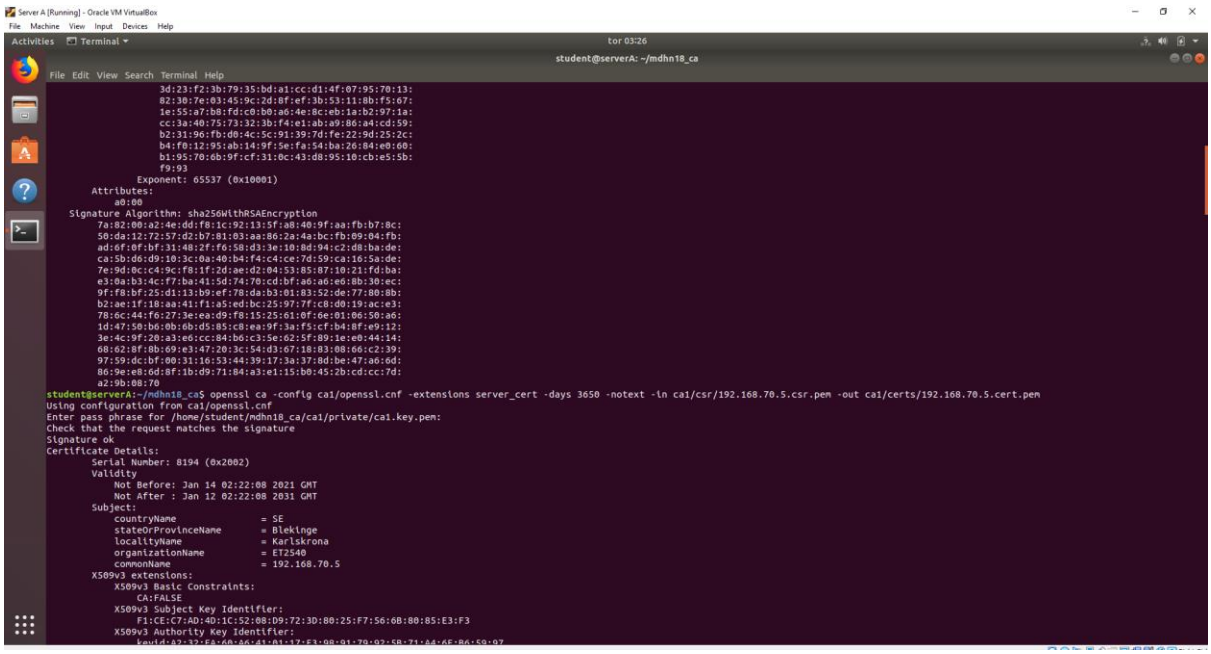
transport

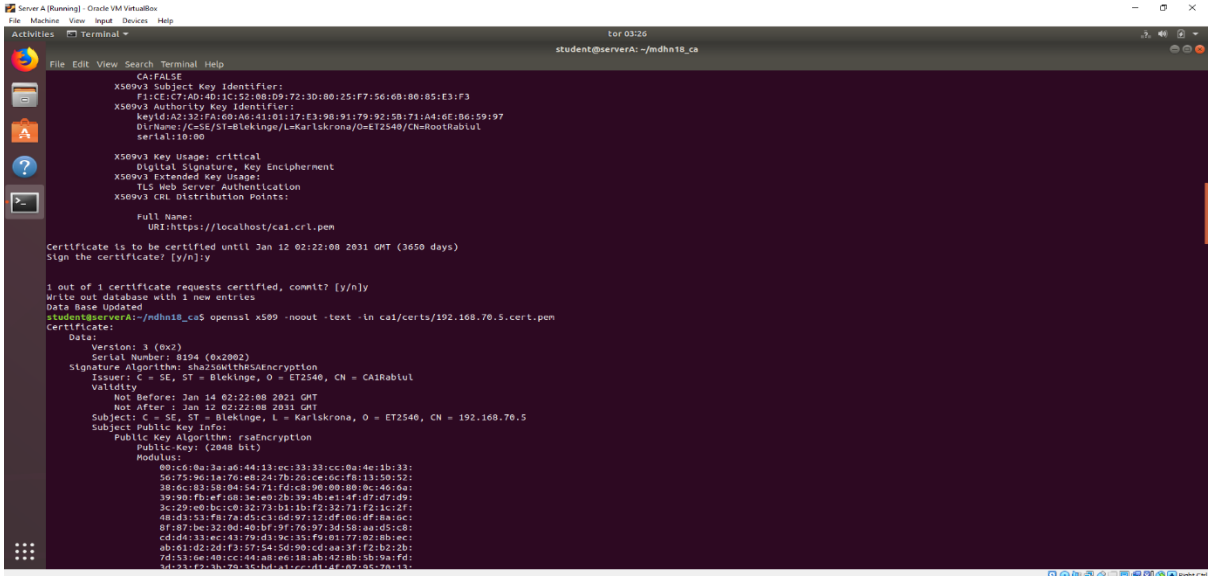
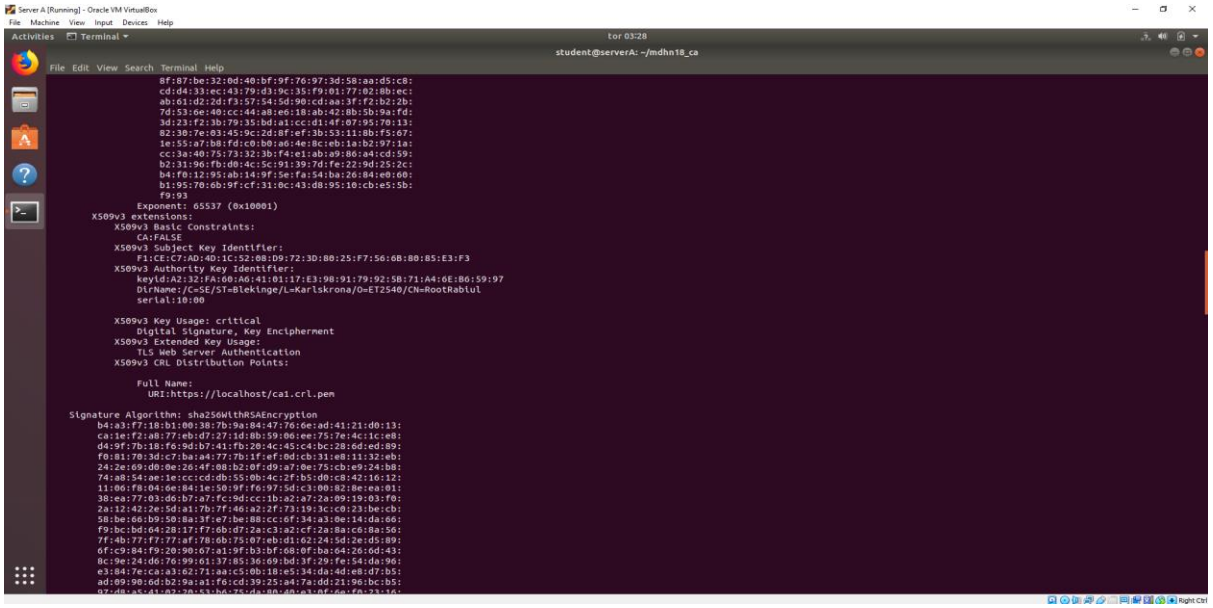
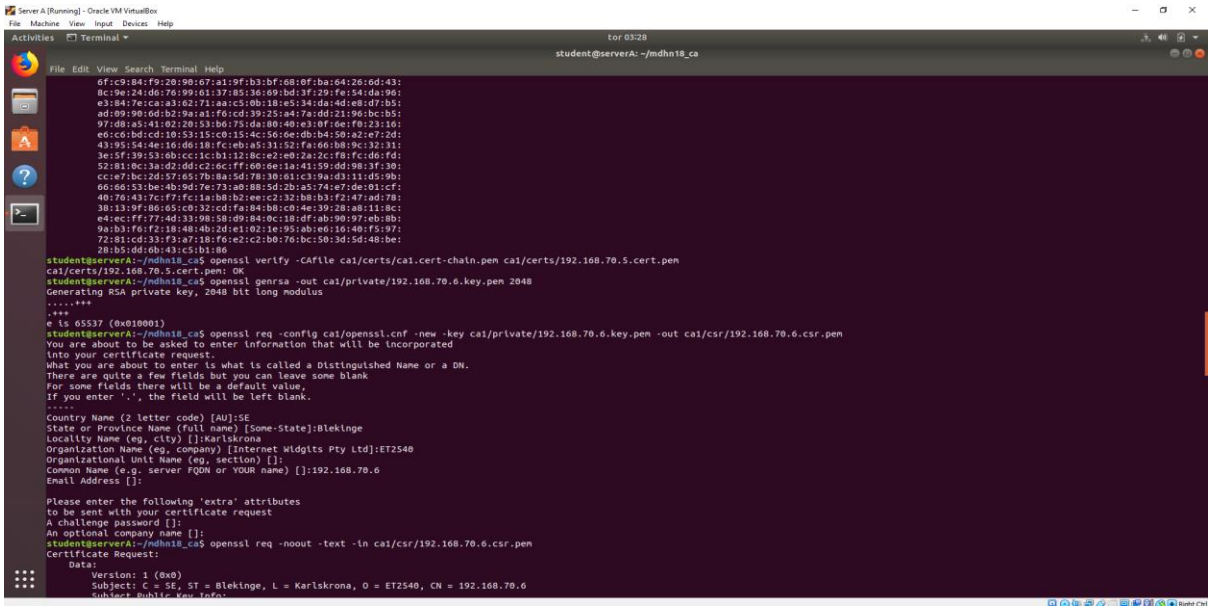
It shows that the mode of transport was transport.

Task 18: Host-to-host transport mode VPN with cert authentication

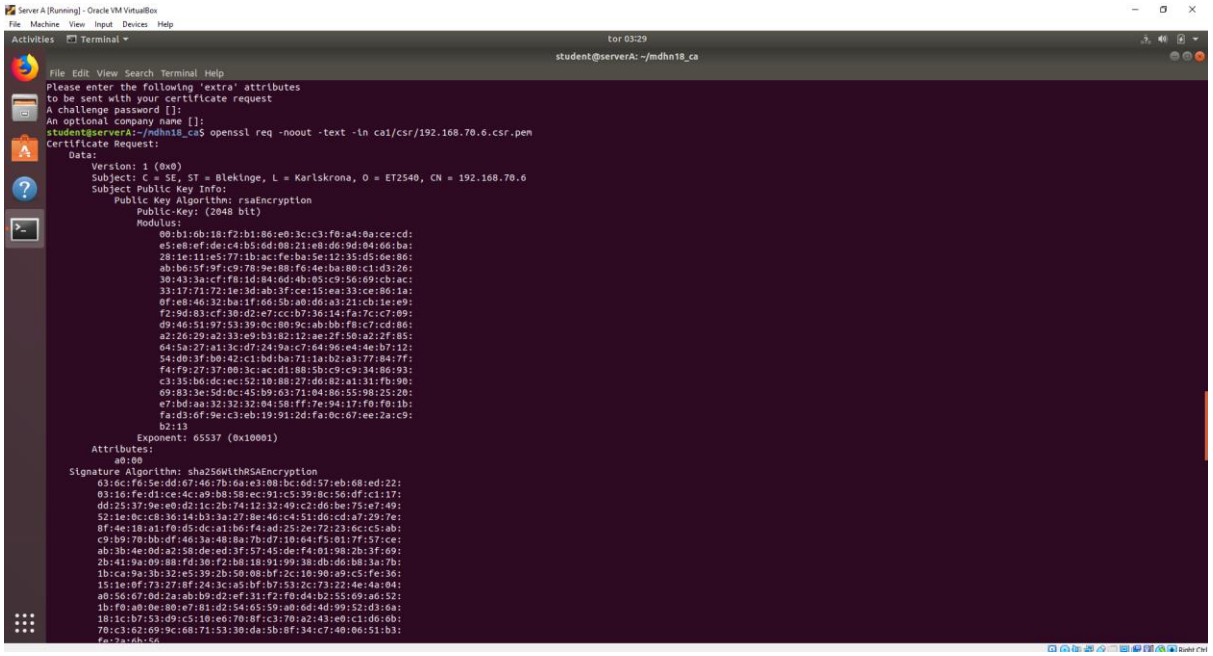
- Before Host-to-host transport mode VPN with certificate authentication we need to create key for serverA and serverB.
- All screen short are shown those process step by step.
- Lastly verified all the certificates.

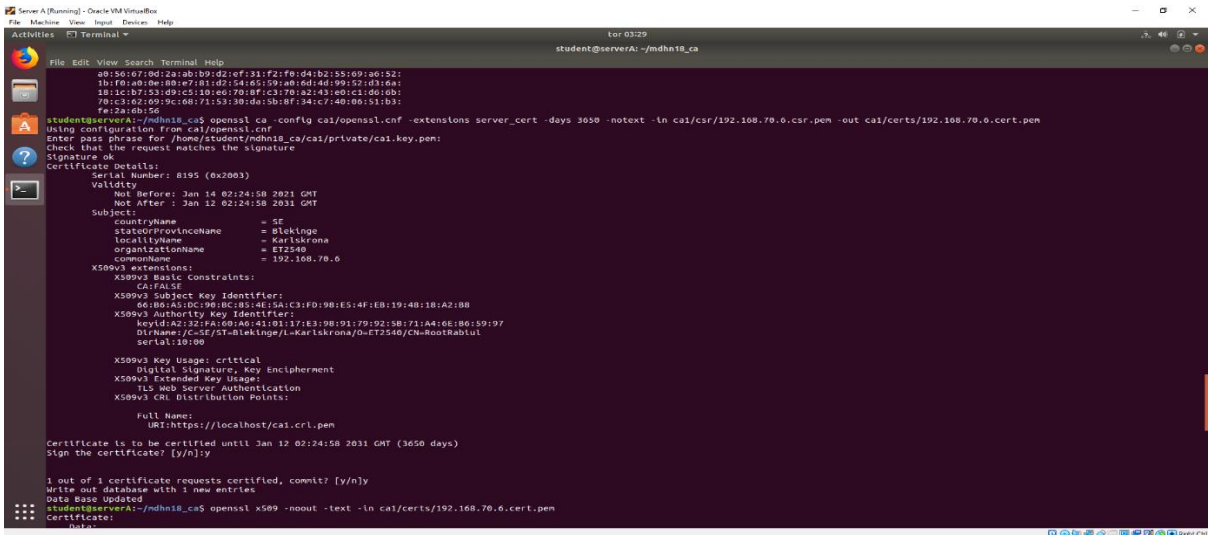


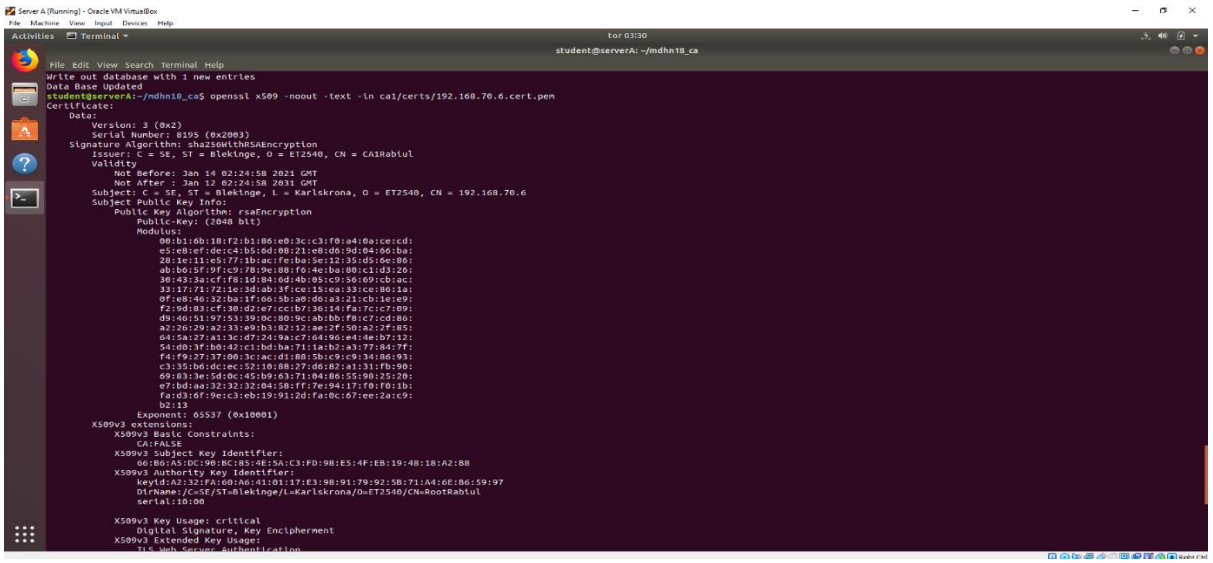


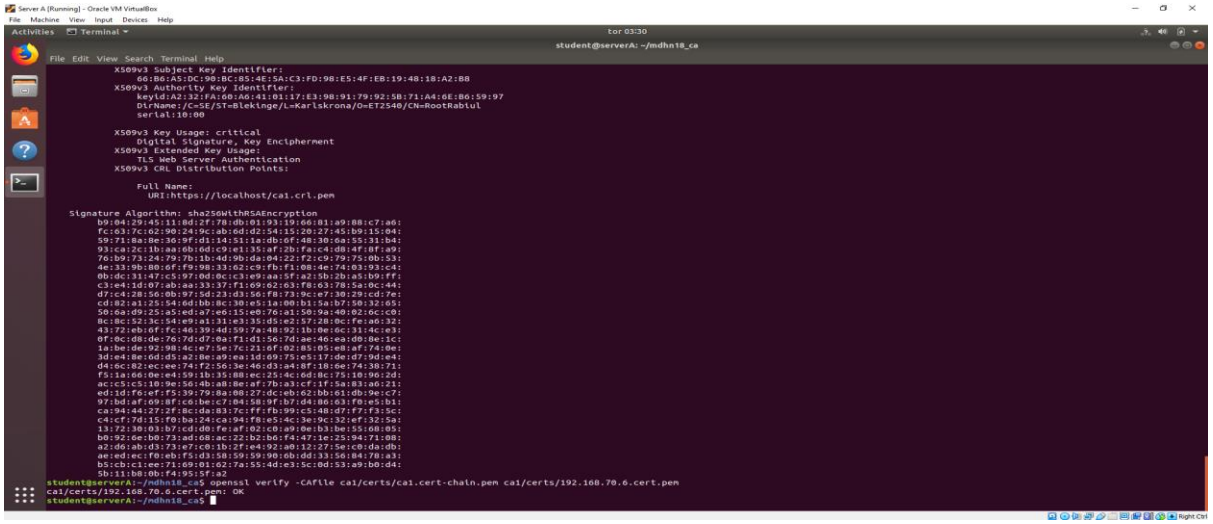

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:26
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
CA:FALSE
X509v3 Subject Key Identifier:
F1:CE:C7:AD:4D:1C:52:08:D9:72:3D:80:25:F7:56:6B:80:85:E3:F3
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/CN=RootRabiul
serial:10:00
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 CRL Distribution Points:
Full Name:
URI:https://localhost/ca1.crl.pem
Certificate is to be certified until Jan 12 02:22:08 2031 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
student@serverA:~/mdhn18_ca$ openssl x509 -noout -text -in ca1/certs/192.168.70.5.cert.pem
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 8194 (0x2002)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2540, CN = CA1Rabiul
Validity
Not Before: Jan 14 02:22:08 2021 GMT
Not After : Jan 12 02:22:08 2031 GMT
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = 192.168.70.5
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:c6:0a:3a:a6:44:13:ec:33:33:cc:0a:4e:1b:33:
56:75:96:1a:76:e8:24:7b:26:ce:6c:f8:13:50:52:
38:6c:83:58:04:54:71:fd:c8:90:00:80:0c:46:6a:
39:90:fb:ef:68:3e:e0:2b:39:4b:e1:4f:d7:d7:d9:
3c:29:e0:bc:c0:32:73:b1:1b:f2:32:71:f2:1c:2f:
4b:d3:53:f8:7a:d5:c3:6d:97:12:df:06:df:8a:6c:
8f:87:be:32:0d:40:bf:9f:76:97:3d:58:aa:d5:c8:
cd:d4:33:ec:43:79:d3:9c:35:f9:01:77:02:8b:ec:
ab:61:d2:2d:f3:57:54:5d:90:cd:aa:3f:f2:b2:2b:
7d:53:6e:40:cc:44:a8:e6:18:ab:42:8b:5b:9a:fd:

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:28
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
8f:87:be:32:0d:40:bf:9f:76:97:3d:58:aa:d5:c8:
cd:d4:33:ec:43:79:d3:9c:35:f9:01:77:02:8b:ec:
ab:61:d2:2d:f3:57:54:5d:90:cd:aa:3f:f2:b2:2b:
7d:53:6e:40:cc:44:a8:e6:18:ab:42:8b:5b:9a:fd:
3d:23:f2:3b:79:35:bd:a1:cc:d1:4f:07:95:70:13:
82:30:7e:03:45:9c:2d:8f:ef:3b:53:11:8b:f5:67:
1e:55:a7:b8:fd:c0:b0:a6:4e:8c:eb:1a:b2:97:1a:
cc:3a:40:75:73:32:3b:f4:e1:ab:a9:86:a4:cd:59:
b2:31:96:fb:d0:4c:5c:91:39:7d:fe:22:9d:25:2c:
b4:f0:12:95:ab:14:9f:5e:fa:54:ba:26:84:e0:60:
b1:95:70:6b:9f:cf:31:0c:43:d8:95:10:cb:e5:5b:
f9:93
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
F1:CE:C7:AD:4D:1C:52:08:D9:72:3D:80:25:F7:56:6B:80:85:E3:F3
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/CN=RootRabiul
serial:10:00
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 CRL Distribution Points:
Full Name:
URI:https://localhost/ca1.crl.pem
Signature Algorithm: sha256WithRSAEncryption
b4:a3:f7:18:b1:00:38:7b:9a:84:47:76:6e:ad:41:21:d0:13:
ca:1e:f2:a8:77:eb:d7:27:1d:8b:59:06:ee:75:7e:4c:1c:e8:
d4:9f:7b:18:f6:9d:b7:41:fb:20:4c:45:c4:bc:28:6d:ed:89:
f0:81:70:3d:c7:ba:a4:77:7b:1f:ef:0d:cb:31:e8:11:32:eb:
24:2e:69:d0:0e:26:4f:08:b2:0f:d9:a7:0e:75:cb:e9:24:b8:
74:a8:54:ae:1e:cc:cd:db:55:0b:4c:2f:b5:d0:c8:42:16:12:
11:06:f8:04:6e:84:1e:50:9f:f6:97:5d:c3:00:82:8e:ea:01:
38:ea:77:03:d6:b7:a7:fc:9d:cc:1b:a2:a7:2a:09:19:03:f0:
2a:12:42:2e:5d:a1:7b:7f:46:a2:2f:73:19:3c:c0:23:be:cb:
5a:be:66:b9:50:8a:3f:e7:be:88:cc:6f:34:a3:0e:14:da:66:
f9:bc:bd:64:28:17:f7:6b:d7:2a:c3:a2:cf:2a:8a:c6:8a:56:
7f:4b:77:f7:77:af:78:6b:75:07:eb:d1:62:24:5d:2e:d5:89:
6f:c9:84:f9:20:90:67:a1:9f:b3:bf:68:0f:ba:64:26:6d:43:
8c:9e:24:d6:76:99:61:37:85:36:69:bd:3f:29:fe:54:da:96:
e3:84:7e:ca:a3:62:71:aa:c5:0b:18:e5:34:da:4d:e8:d7:b5:
ad:09:90:6d:b2:9a:a1:f6:cd:39:25:a4:7a:dd:21:96:bc:b5:

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:28
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
6f:c9:84:f9:20:90:67:a1:9f:b3:bf:68:0f:ba:64:26:6d:43:
8c:9e:24:d6:76:99:61:37:85:36:69:bd:3f:29:fe:54:da:96:
e3:84:7e:ca:a3:62:71:aa:c5:0b:18:e5:34:da:4d:e8:d7:b5:
ad:09:90:6d:b2:9a:a1:f6:cd:39:25:a4:7a:dd:21:96:bc:b5:
97:d8:a5:41:02:20:53:b6:75:da:80:40:e3:0f:6e:f0:23:16:
e6:c6:bd:cd:10:53:15:c0:15:4c:56:6e:db:b4:50:a2:e7:2d:
43:95:54:4e:16:d6:18:fc:eb:a5:31:52:fa:66:b8:9c:32:31:
3e:5f:39:53:0b:cc:1c:b1:12:8c:e2:e0:2a:2c:f8:fc:d6:fd:
52:81:0c:3a:d2:dd:c2:6c:ff:60:ee:1a:41:59:dd:98:3f:30:
cc:e7:bc:2d:57:65:7b:8a:5d:78:30:61:c3:9a:d3:11:d5:9b:
66:66:53:be:4b:9d:7e:73:a0:88:5d:2b:a5:74:e7:de:01:cf:
40:76:43:7c:f7:fc:1a:b8:b2:ee:c2:32:b8:b3:f2:47:ad:78:
38:13:9f:86:65:c0:32:cd:fa:84:b8:c0:4e:39:28:a8:11:8c:
e4:ec:ff:77:4d:33:98:58:d9:84:0c:18:df:ab:90:97:eb:8b:
9a:b3:f6:f2:18:48:4b:2d:e1:02:1e:95:ab:e6:16:40:f5:97:
72:81:cd:33:f3:a7:18:f6:e2:c2:b0:76:bc:50:3d:5d:48:be:
28:b5:dd:6b:43:c5:b1:86
student@serverA:~/mdhn18_ca$ openssl verify -CAFile ca1/certs/ca1.cert-chain.pem ca1/certs/192.168.70.5.cert.pem
ca1/certs/192.168.70.5.cert.pem: OK
student@serverA:~/mdhn18_ca$ openssl genrsa -out ca1/private/192.168.70.6.key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.+++
e is 65537 (0x010001)
student@serverA:~/mdhn18_ca$ openssl req -config ca1/openssl.cnf -new -key ca1/private/192.168.70.6.key.pem -out ca1/csr/192.168.70.6.csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:SE
State or Province Name (full name) [Some-State]:Blekinge
Locality Name (eg, city) []:Karlskrona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ET2540
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.70.6
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/192.168.70.6.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = 192.168.70.6

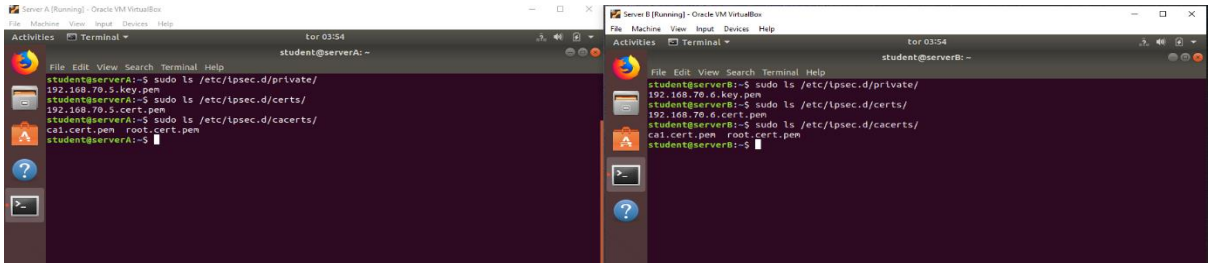

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:29
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
student@serverA:~/mdhn18_ca$ openssl req -noout -text -in ca1/csr/192.168.70.6.csr.pem
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = 192.168.70.6
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b1:6b:18:f2:b1:86:e0:3c:c3:f0:a4:0a:ce:cd:
e5:e8:ef:de:c4:b5:6d:08:21:e8:d6:9d:04:66:ba:
28:1e:11:e5:77:1b:ac:fe:ba:5e:12:35:d5:6e:86:
ab:b6:5f:9f:c9:78:9e:88:f6:4e:ba:80:c1:d3:26:
30:43:3a:cf:f8:1d:84:6d:4b:85:c9:56:69:cb:ac:
33:17:71:72:1e:3d:ab:3f:ce:15:ea:33:ce:86:1a:
0f:e8:46:32:ba:1f:66:5b:a0:d6:a3:21:cb:1e:e9:
f2:9d:83:cf:30:d2:e7:cc:b7:36:14:fa:7c:c7:09:
d9:46:51:97:53:39:0c:80:9c:ab:bb:f8:c7:cd:86:
a2:26:29:a2:33:e9:b3:82:12:ae:2f:50:a2:2f:85:
64:5a:27:a1:3c:d7:24:9a:c7:64:96:e4:4e:b7:12:
54:d0:3f:b0:42:c1:bd:ba:71:1a:b2:a3:77:84:7f:
f4:f9:27:37:00:3c:ac:d1:88:5b:c9:c9:34:86:93:
c3:35:b6:dc:ec:52:10:88:27:d6:82:a1:31:fb:90:
69:83:3e:5d:0c:45:b9:63:71:04:86:55:98:25:20:
e7:bd:aa:32:32:32:04:58:ff:7e:94:17:f0:f0:1b:
fa:d3:6f:9e:c3:eb:19:91:2d:fa:0c:67:ee:2a:c9:
b2:13
Exponent: 65537 (0x10001)
Attributes:
a0:00
Signature Algorithm: sha256WithRSAEncryption
63:6c:f6:5e:dd:67:46:7b:6a:e3:08:bc:6d:57:eb:68:ed:22:
03:16:fe:d1:ce:4c:a9:b8:58:ec:91:c5:39:8c:56:df:c1:17:
dd:25:37:9e:e0:d3:1c:2b:74:12:32:49:c2:d6:be:75:e7:49:
52:1e:0c:c8:36:14:b3:3a:27:8e:46:c4:51:d6:cd:a7:29:7e:
8f:4e:18:a1:f0:d5:dc:a1:b6:f4:ad:25:2e:72:23:6c:c5:ab:
c9:b9:70:bb:df:46:3a:48:8a:7b:d7:10:64:f5:01:7f:57:ce:
ab:3b:4e:0d:a2:58:de:ed:3f:57:45:de:f4:01:98:2b:3f:69:
2b:41:9a:09:88:fd:30:f2:b8:18:91:99:38:db:d6:b8:3a:7b:
1b:ca:9a:3b:32:e5:39:2b:50:08:bf:2c:10:90:a9:c5:fe:36:
15:1e:0f:73:27:8f:24:3c:a5:bf:b7:53:2c:73:22:4e:4a:04:
a0:56:67:0d:2a:ab:b9:d2:ef:31:f2:f0:d4:b2:55:69:a6:52:
1b:f0:a0:0e:80:e7:81:d2:54:65:59:a0:6d:4d:99:52:d3:6a:
18:1c:b7:53:d9:c5:10:e6:70:8f:c3:70:a2:43:e0:c1:d6:6b:
70:c3:62:69:9c:68:71:53:30:da:5b:8f:34:c7:40:06:51:b3:
fe:2a:6b:56


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:29
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
a0:56:67:0d:2a:ab:b9:d2:ef:31:f2:f0:d4:b2:55:69:a6:52:
1b:f0:a0:0e:80:e7:81:d2:54:65:59:a0:6d:4d:99:52:d3:6a:
18:1c:b7:53:d9:c5:10:e6:70:8f:c3:70:a2:43:e0:c1:d6:6b:
70:c3:62:69:9c:68:71:53:30:da:5b:8f:34:c7:40:06:51:b3:
fe:2a:6b:56
student@serverA:~/mdhn18_ca$ openssl ca -config ca1/openssl.cnf -extensions server_cert -days 3650 -notext -in ca1/csr/192.168.70.6.csr.pem -out ca1/certs/192.168.70.6.cert.pem
Using configuration from ca1/openssl.cnf
Enter pass phrase for /home/student/mdhn18_ca/ca1/private/ca1.key.pen:
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 8195 (0x2003)
Validity
Not Before: Jan 14 02:24:58 2021 GMT
Not After : Jan 12 02:24:58 2031 GMT
Subject:
countryName = SE
stateOrProvinceName = Blekinge
localityName = Karlskrona
organizationName = ET2540
commonName = 192.168.70.6
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
66:B6:A5:DC:90:BC:85:4E:5A:C3:FD:98:E5:4F:EB:19:48:18:A2:88
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/CN=RootRablul
serial:10:00
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 CRL Distribution Points:
Full Name:
URI:https://localhost/ca1.crl.pem
Certificate is to be certified until Jan 12 02:24:58 2031 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
student@serverA:~/mdhn18_ca$ openssl x509 -noout -text -in ca1/certs/192.168.70.6.cert.pem
Certificate:
Data:

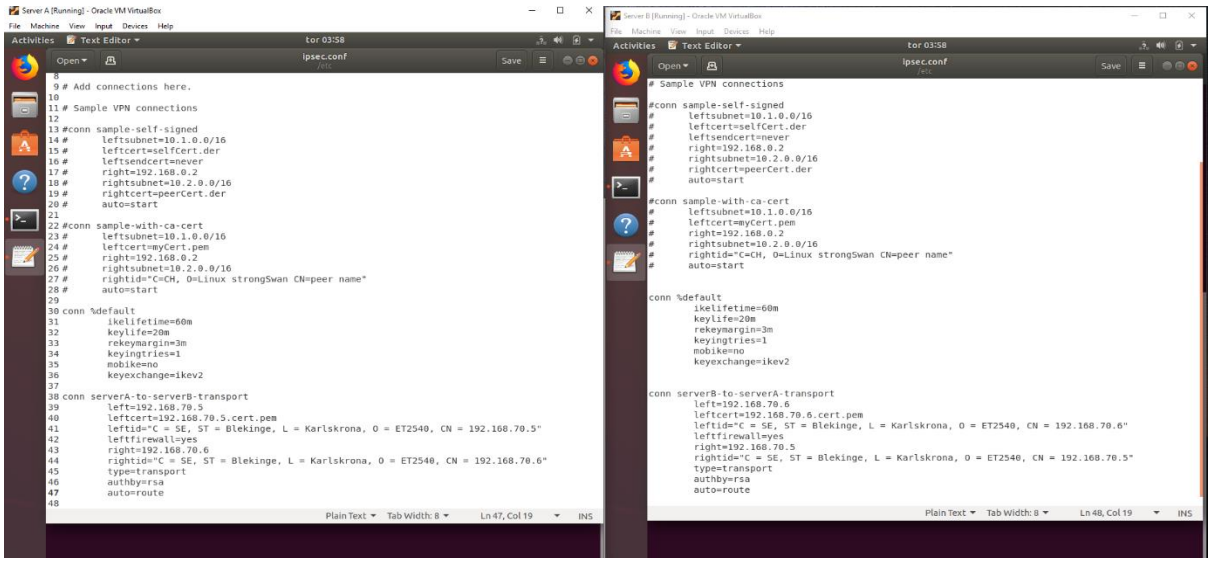

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 03:30
student@serverA: ~/mdhn18_ca
File Edit View Search Terminal Help
Write out database with 1 new entries
Data Base Updated
student@serverA:~/mdhn18_ca$ openssl x509 -noout -text -in ca1/certs/192.168.70.6.cert.pem
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 8195 (0x2003)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2540, CN = CA1Rablul
Validity
Not Before: Jan 14 02:24:58 2021 GMT
Not After : Jan 12 02:24:58 2031 GMT
Subject: C = SE, ST = Blekinge, L = Karlskrona, O = ET2540, CN = 192.168.70.6
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b1:6b:18:f2:b1:86:e0:3c:c3:f0:a4:0a:ce:cd:
e5:e8:ef:de:c4:b5:6d:08:21:e8:d6:9d:04:66:ba:
28:1e:11:e5:77:1b:ac:fe:ba:5e:12:35:d5:6e:86:
ab:b6:5f:9f:c9:78:9e:88:f6:4e:ba:80:c1:d3:26:
30:43:3a:cf:f8:1d:84:6d:4b:85:c9:56:69:cb:ac:
33:17:71:72:1e:3d:ab:3f:ce:15:ea:33:ce:86:1a:
0f:e8:46:32:ba:1f:66:5b:a0:d6:a3:21:cb:1e:e9:
f2:9d:83:cf:30:d2:e7:cc:b7:36:14:fa:7c:c7:09:
d9:46:51:97:53:39:0c:80:9c:ab:bb:f8:c7:cd:86:
a2:26:29:a2:33:e9:b3:82:12:ae:2f:50:a2:2f:85:
64:5a:27:a1:3c:d7:24:9a:c7:64:96:e4:4e:b7:12:
54:d0:3f:b0:42:c1:bd:ba:71:1a:b2:a3:77:84:7f:
f4:f9:27:37:00:3c:ac:d1:88:5b:c9:c9:34:86:93:
c3:35:b6:dc:ec:52:10:88:27:d6:82:a1:31:fb:90:
69:83:3e:5d:0c:45:b9:63:71:04:86:55:98:25:20:
e7:bd:aa:32:32:32:04:58:ff:7e:94:17:f0:f0:1b:
fa:d3:6f:9e:c3:eb:19:91:2d:fa:0c:67:ee:2a:c9:
b2:13
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
66:B6:A5:DC:90:BC:85:4E:5A:C3:FD:98:E5:4F:EB:19:48:18:A2:88
X509v3 Authority Key Identifier:
keyid:A2:32:FA:60:A6:41:01:17:E3:98:91:79:92:5B:71:A4:6E:B6:59:97
DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/CN=RootRablul
serial:10:00
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication

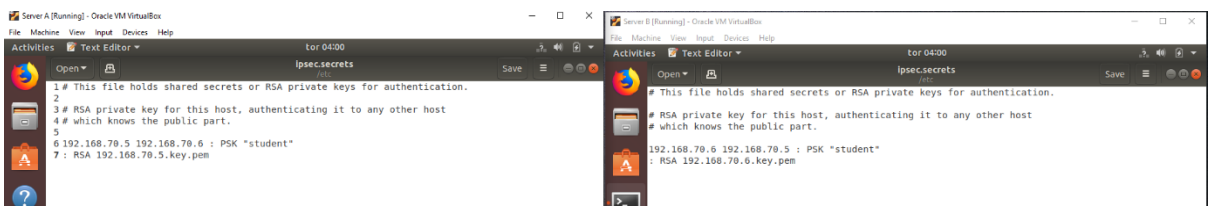


- Distribute the certificate exact location in serverA and server.

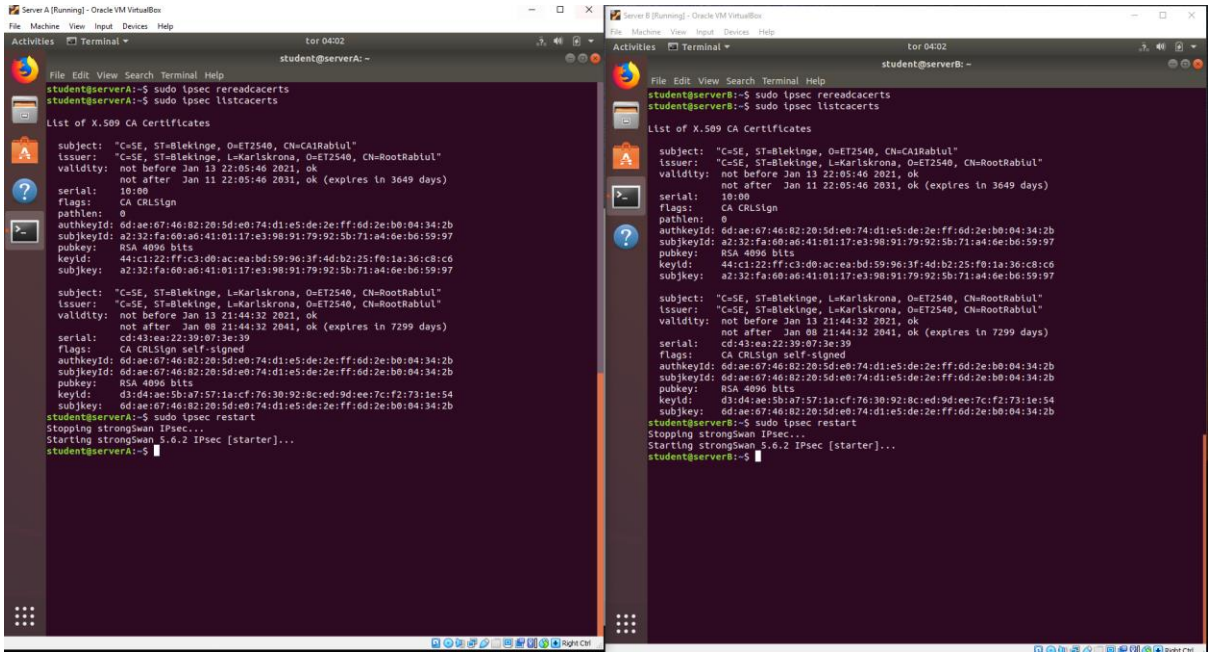


- Edit the ipsec.conf and ipsec.secrets as shown in below:

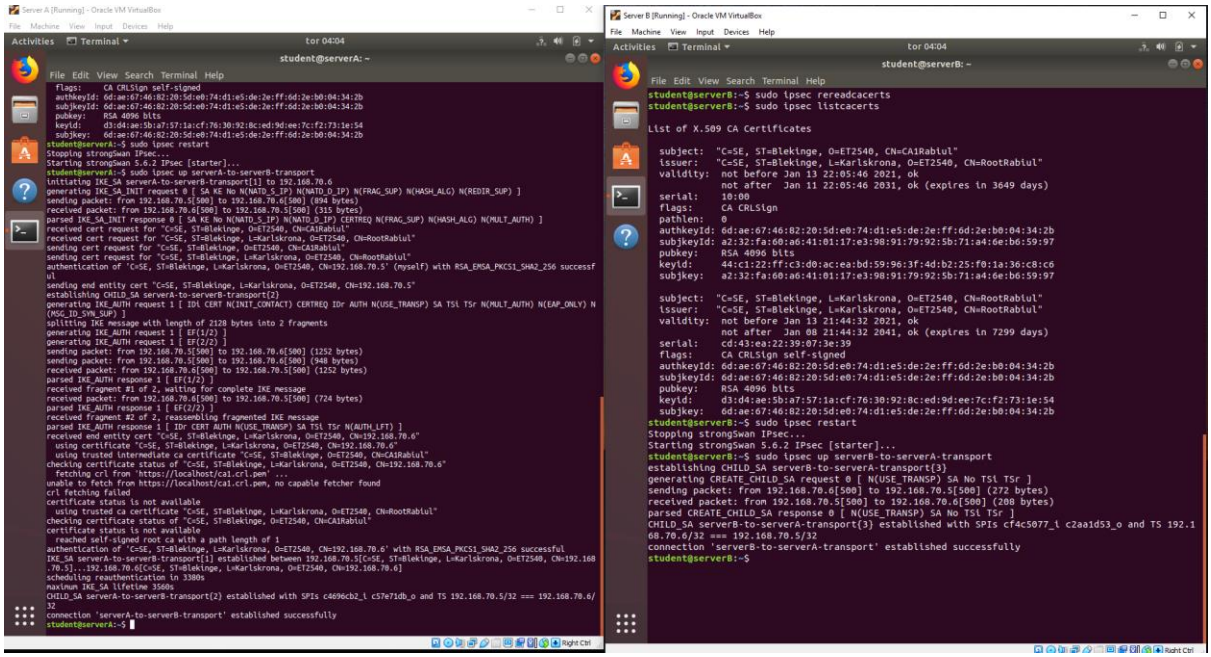


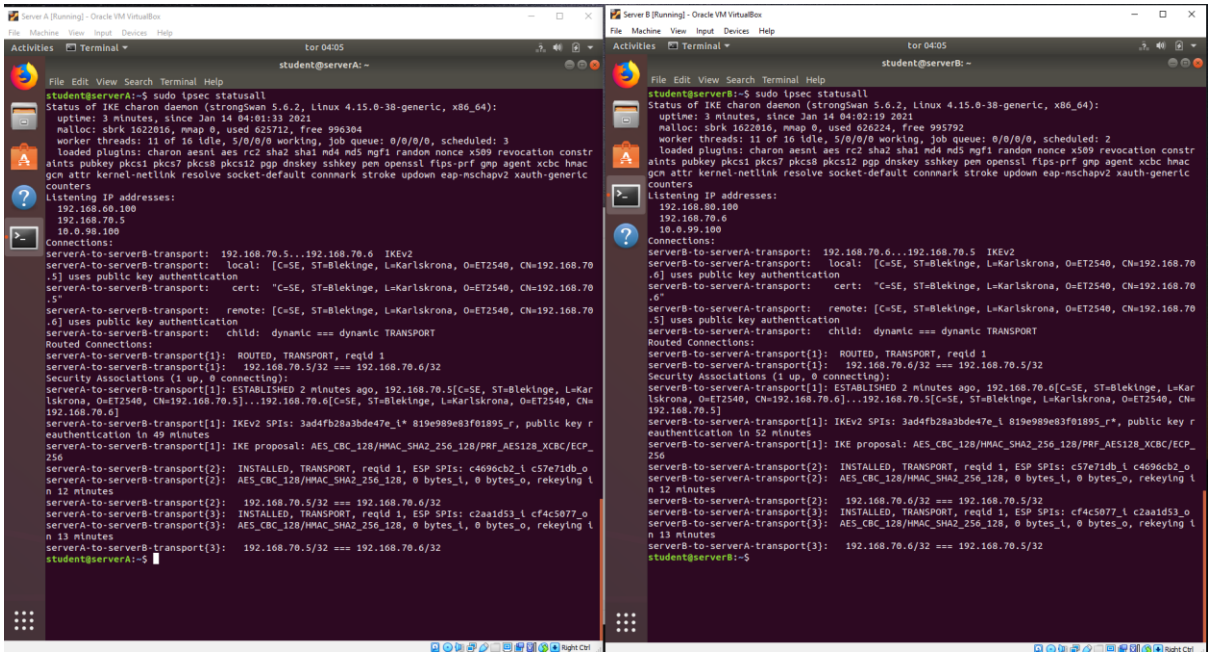


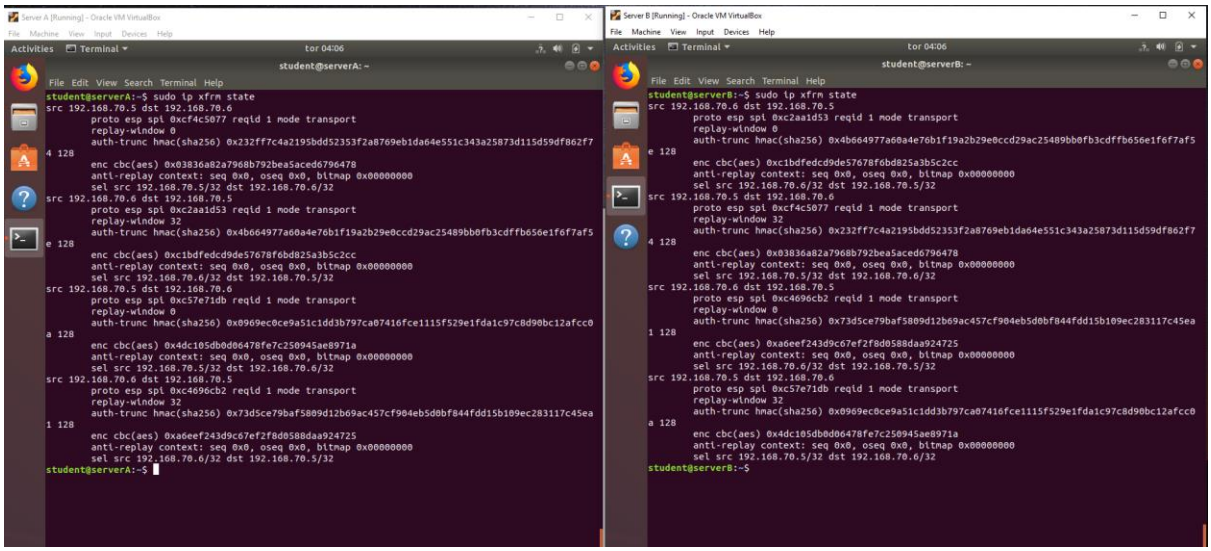
- Here is ipsec listcacerts

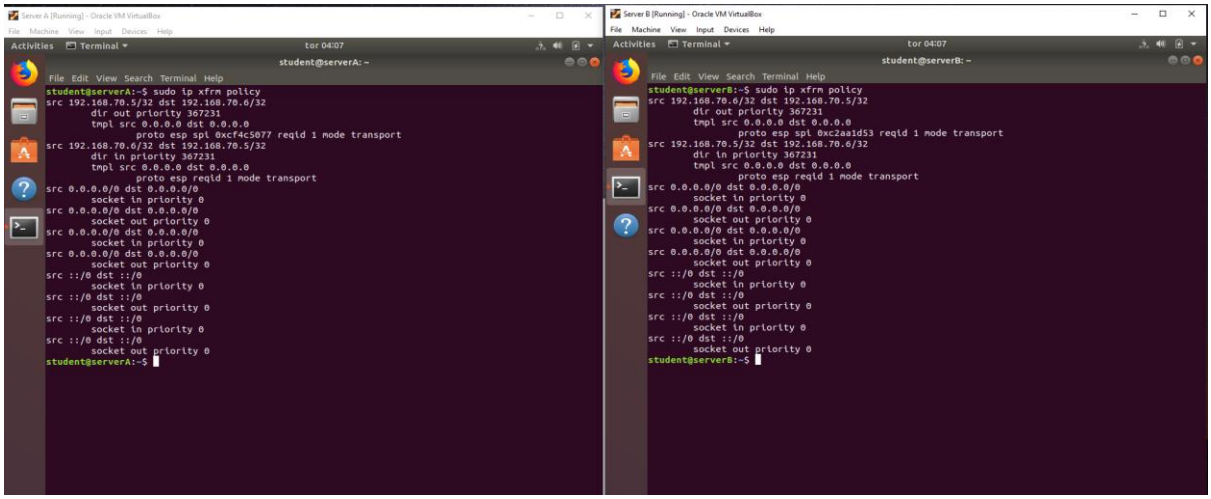


- Host-to-host transport mode VPN with cert authentication successfully established.

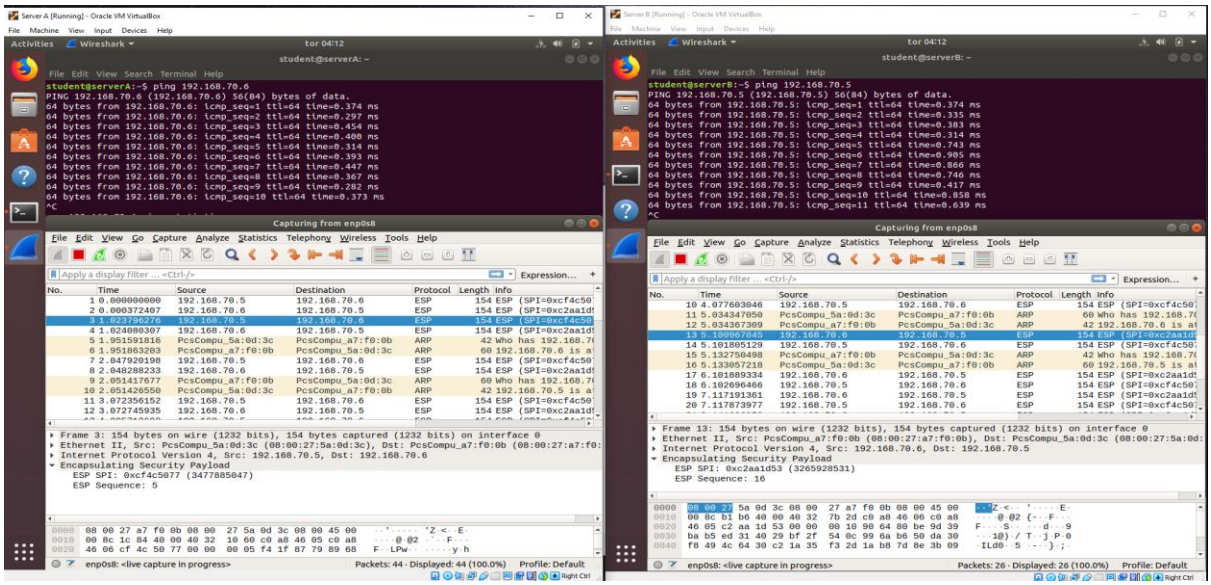



Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:05
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.6.2, Linux 4.15.0-38-generic, x86_64):
uptime: 3 minutes, since Jan 14 04:01:33 2021
malloc: sbrk 1622016, mmap 0, used 625712, free 996304
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
loaded plugins: charon aesni aes rc2 sha2 sha1 md4 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gcm attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-generic counters
Listening IP addresses:
192.168.60.100
192.168.70.5
10.0.98.100
Connections:
serverA-to-serverB-transport: 192.168.70.5...192.168.70.6 IKEv2
serverA-to-serverB-transport: local: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.5] uses public key authentication
serverA-to-serverB-transport: cert: "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.5"
serverA-to-serverB-transport: remote: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.6] uses public key authentication
serverA-to-serverB-transport: child: dynamic === dynamic TRANSPORT
Routed Connections:
serverA-to-serverB-transport[1]: ROUTED, TRANSPORT, reqid 1
serverA-to-serverB-transport[1]: 192.168.70.5/32 === 192.168.70.6/32
Security Associations (1 up, 0 connecting):
serverA-to-serverB-transport[1]: ESTABLISHED 2 minutes ago, 192.168.70.5[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.5]...192.168.70.6[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.6]
serverA-to-serverB-transport[1]: IKEv2 SPIs: 3ad4fb28a3bde47e_i* 819e989e83f01895_r, public key reauthentication in 49 minutes
serverA-to-serverB-transport[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
serverA-to-serverB-transport[2]: INSTALLED, TRANSPORT, reqid 1, ESP SPIs: c4696cb2_i c57e71db_o
serverA-to-serverB-transport[2]: AES_CBC_128/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 12 minutes
serverA-to-serverB-transport[2]: 192.168.70.5/32 === 192.168.70.6/32
serverA-to-serverB-transport[3]: INSTALLED, TRANSPORT, reqid 1, ESP SPIs: c2aa1d53_i cf4c5077_o
serverA-to-serverB-transport[3]: AES_CBC_128/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 13 minutes
serverA-to-serverB-transport[3]: 192.168.70.5/32 === 192.168.70.6/32
student@serverA:~$
Server B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:05
student@serverB: ~
File Edit View Search Terminal Help
student@serverB:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.6.2, Linux 4.15.0-38-generic, x86_64):
uptime: 3 minutes, since Jan 14 04:02:19 2021
malloc: sbrk 1622016, mmap 0, used 626224, free 995792
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
loaded plugins: charon aesni aes rc2 sha2 sha1 md4 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gcm attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-generic counters
Listening IP addresses:
192.168.80.100
192.168.70.6
10.0.99.100
Connections:
serverB-to-serverA-transport: 192.168.70.6...192.168.70.5 IKEv2
serverB-to-serverA-transport: local: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.6] uses public key authentication
serverB-to-serverA-transport: cert: "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.6"
serverB-to-serverA-transport: remote: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.5] uses public key authentication
serverB-to-serverA-transport: child: dynamic === dynamic TRANSPORT
Routed Connections:
serverB-to-serverA-transport[1]: ROUTED, TRANSPORT, reqid 1
serverB-to-serverA-transport[1]: 192.168.70.6/32 === 192.168.70.5/32
Security Associations (1 up, 0 connecting):
serverB-to-serverA-transport[1]: ESTABLISHED 2 minutes ago, 192.168.70.6[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.6]...192.168.70.5[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, CN=192.168.70.5]
serverB-to-serverA-transport[1]: IKEv2 SPIs: 3ad4fb28a3bde47e_i 819e989e83f01895_r*, public key reauthentication in 52 minutes
serverB-to-serverA-transport[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
serverB-to-serverA-transport[2]: INSTALLED, TRANSPORT, reqid 1, ESP SPIs: c57e71db_i c4696cb2_o
serverB-to-serverA-transport[2]: AES_CBC_128/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 12 minutes
serverB-to-serverA-transport[2]: 192.168.70.6/32 === 192.168.70.5/32
serverB-to-serverA-transport[3]: INSTALLED, TRANSPORT, reqid 1, ESP SPIs: cf4c5077_i c2aa1d53_o
serverB-to-serverA-transport[3]: AES_CBC_128/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 13 minutes
serverB-to-serverA-transport[3]: 192.168.70.6/32 === 192.168.70.5/32
student@serverB:~$


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:06
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ sudo ip xfrm state
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xcf4c5077 reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0x232ff7c4a2195bdd52353f2a8769eb1da64e551c343a25873d115d59df862f74 128
enc cbc(aes) 0x03836a82a7968b792bea5aced6796478
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.5/32 dst 192.168.70.6/32
src 192.168.70.6 dst 192.168.70.5
proto esp spi 0xc2aa1d53 reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0x4b664977a60a4e76b1f19a2b29e0ccd29ac25489bb0fb3cdffb656e1f6f7af5e 128
enc cbc(aes) 0xc1bdfedcd9de57678f6bd825a3b5c2cc
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.6/32 dst 192.168.70.5/32
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xc57e71db reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0x0969ec0ce9a51c1dd3b797ca07416fce1115f529e1fda1c97c8d98bc12afcc0a 128
enc cbc(aes) 0x4dc105db0d06478fe7c250945ae8971a
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.5/32 dst 192.168.70.6/32
src 192.168.70.6 dst 192.168.70.5
proto esp spi 0xc4696cb2 reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0x73d5ce79baf5809d12b69ac457cf904eb5d0bf844fdd15b109ec283117c45ea1 128
enc cbc(aes) 0xa6eef243d9c67ef2f8d0588daa924725
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.6/32 dst 192.168.70.5/32
student@serverA:~$
Server B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:06
student@serverB: ~
File Edit View Search Terminal Help
student@serverB:~$ sudo ip xfrm state
src 192.168.70.6 dst 192.168.70.5
proto esp spi 0xc2aa1d53 reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0x4b664977a60a4e76b1f19a2b29e0ccd29ac25489bb0fb3cdffb656e1f6f7af5e 128
enc cbc(aes) 0xc1bdfedcd9de57678f6bd825a3b5c2cc
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.6/32 dst 192.168.70.5/32
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xcf4c5077 reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0x232ff7c4a2195bdd52353f2a8769eb1da64e551c343a25873d115d59df862f74 128
enc cbc(aes) 0x03836a82a7968b792bea5aced6796478
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.5/32 dst 192.168.70.6/32
src 192.168.70.6 dst 192.168.70.5
proto esp spi 0xc4696cb2 reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0x73d5ce79baf5809d12b69ac457cf904eb5d0bf844fdd15b109ec283117c45ea1 128
enc cbc(aes) 0xa6eef243d9c67ef2f8d0588daa924725
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.6/32 dst 192.168.70.5/32
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xc57e71db reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0x0969ec0ce9a51c1dd3b797ca07416fce1115f529e1fda1c97c8d98bc12afcc0a 128
enc cbc(aes) 0x4dc105db0d06478fe7c250945ae8971a
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
sel src 192.168.70.5/32 dst 192.168.70.6/32
student@serverB:~$

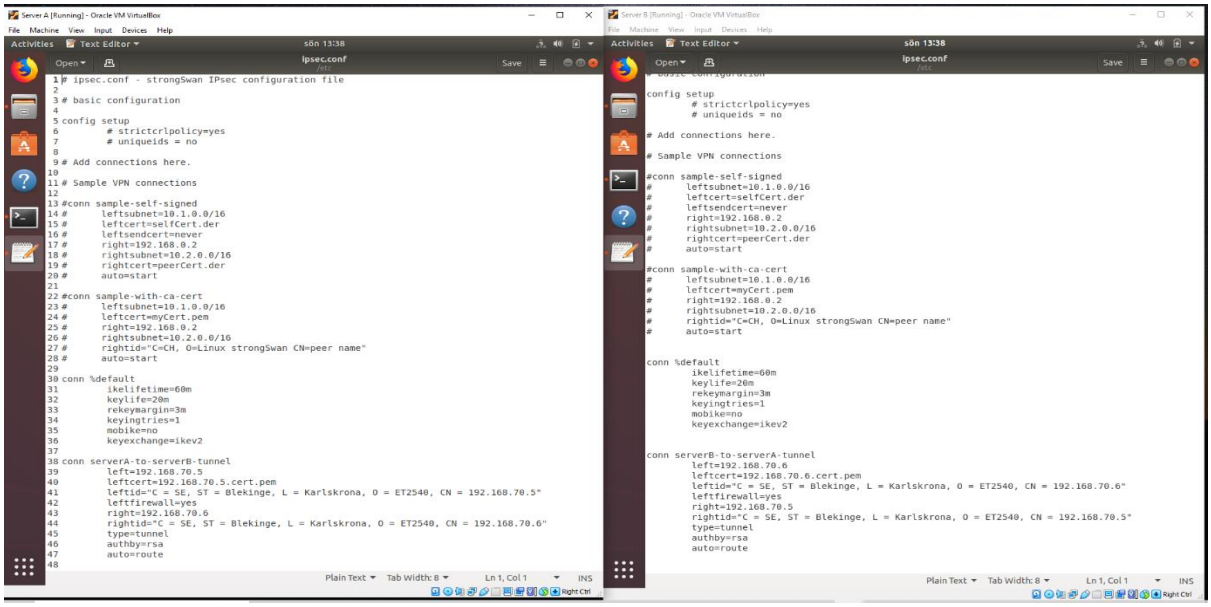

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:07
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ sudo ip xfrm policy
src 192.168.70.5/32 dst 192.168.70.6/32
dir out priority 367231
tmpl src 0.0.0.0 dst 0.0.0.0
proto esp spi 0xcf4c5077 reqid 1 mode transport
src 192.168.70.6/32 dst 192.168.70.5/32
dir in priority 367231
tmpl src 0.0.0.0 dst 0.0.0.0
proto esp reqid 1 mode transport
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
student@serverA:~$
Server B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal tor 04:07
student@serverB: ~
File Edit View Search Terminal Help
student@serverB:~$ sudo ip xfrm policy
src 192.168.70.6/32 dst 192.168.70.5/32
dir out priority 367231
tmpl src 0.0.0.0 dst 0.0.0.0
proto esp spi 0xc2aa1d53 reqid 1 mode transport
src 192.168.70.5/32 dst 192.168.70.6/32
dir in priority 367231
tmpl src 0.0.0.0 dst 0.0.0.0
proto esp reqid 1 mode transport
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
student@serverB:~$

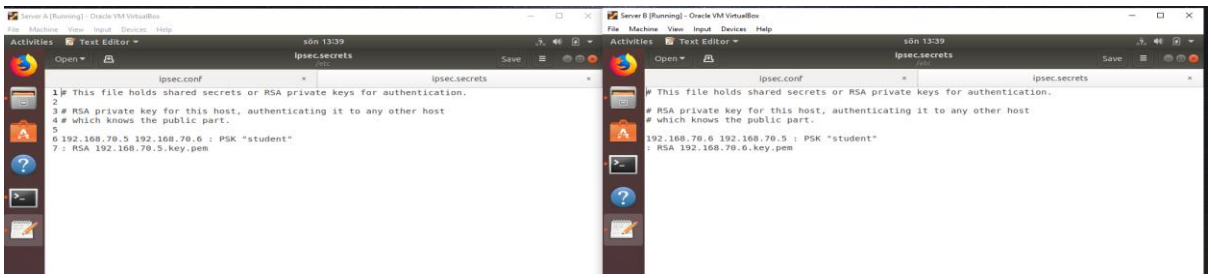
- Successfully ping serverA to server and server to serverA with wireshark traffic.



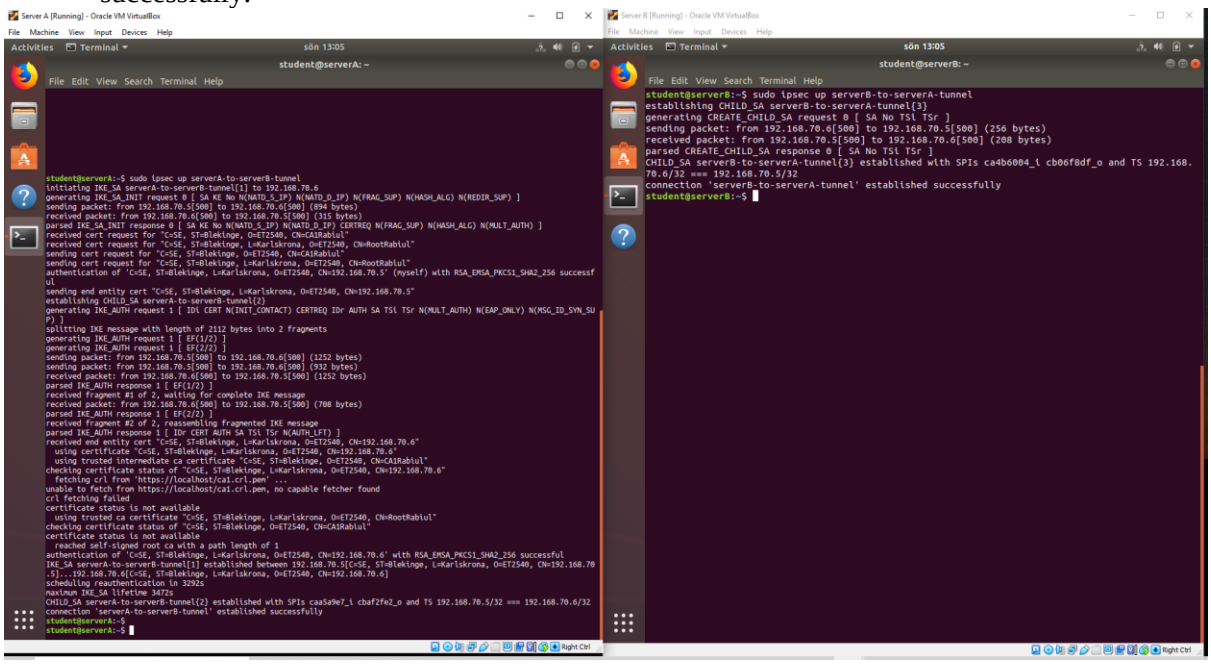
Task 19: Tunnel mode VPN with cert authentication between Server A and Server B

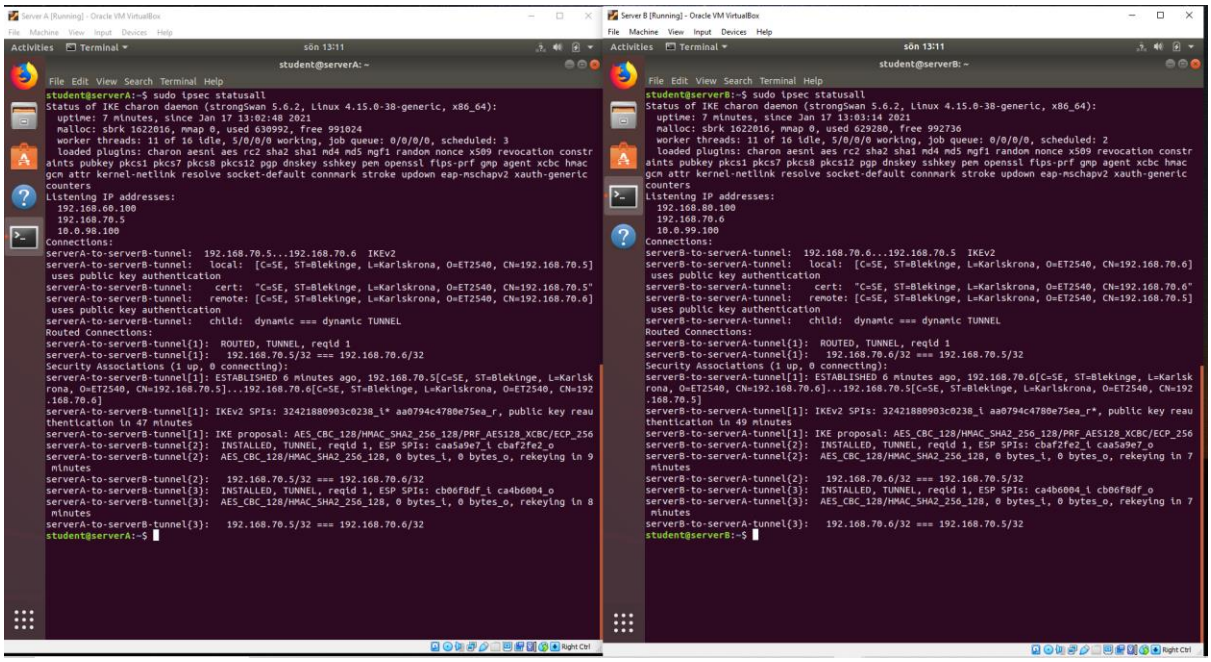
- For this task I edit again ipsec.conf and ipsec.secrets.

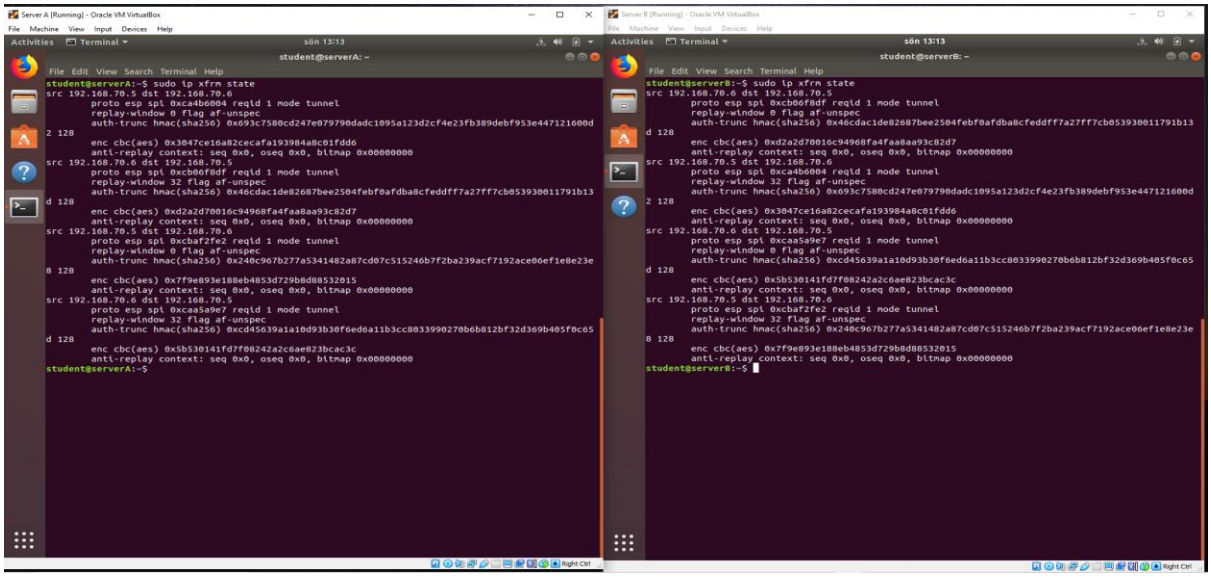


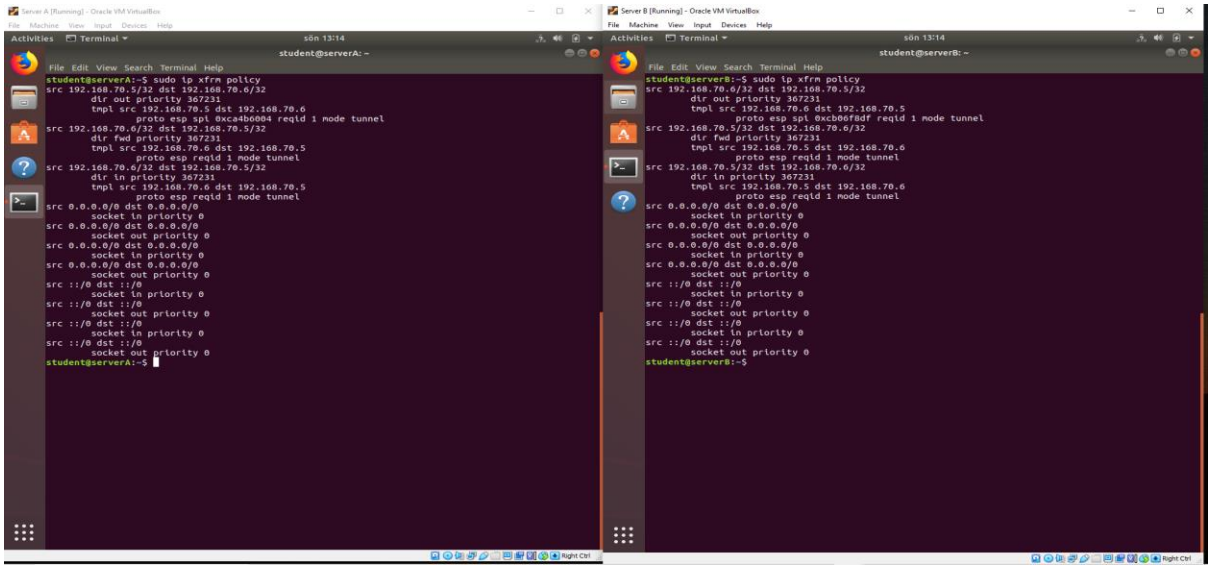


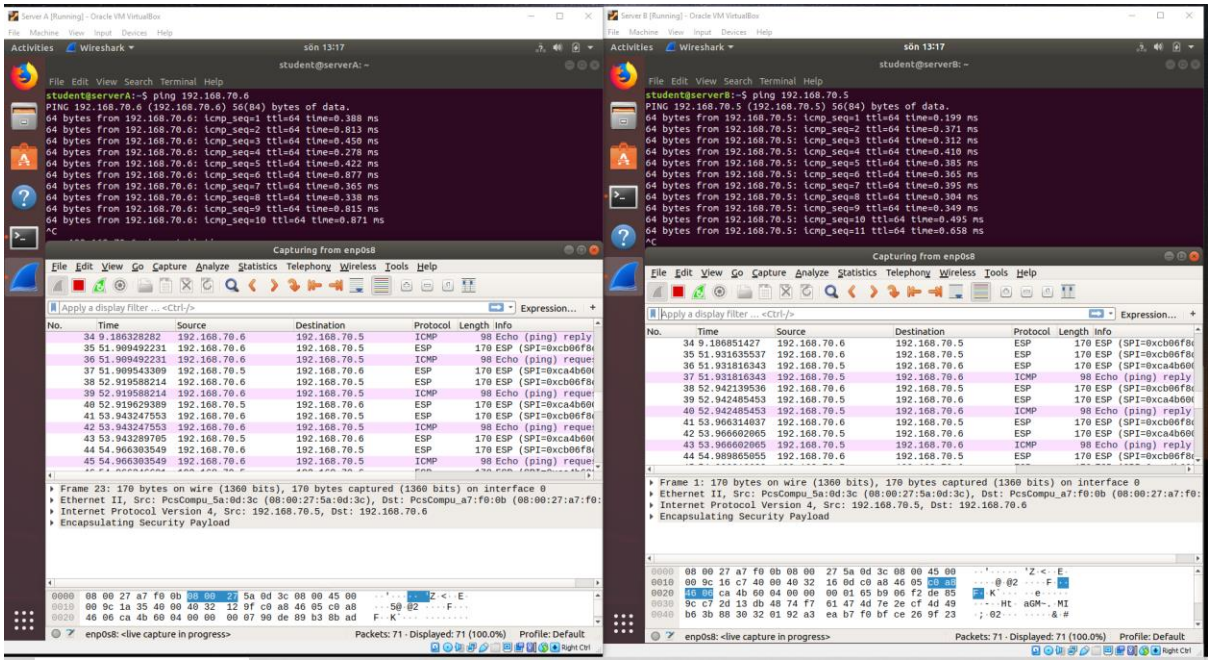
- Tunnel mode VPN with cert authentication between Server A and Server B established successfully.



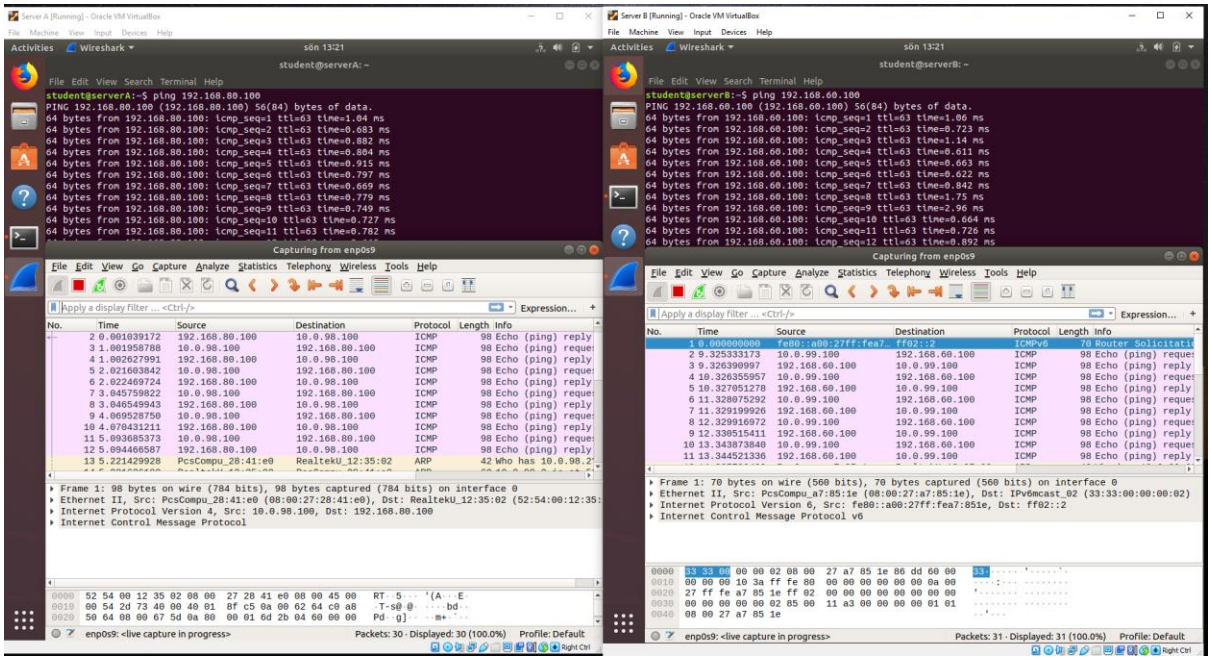



Server A [Running] - Oracle VM VirtualBox
Server B [Running] - Oracle VM VirtualBox
student@serverA: ~
student@serverB: ~


Server A [Running] - Oracle VM VirtualBox
Server B [Running] - Oracle VM VirtualBox
student@serverA: ~
student@serverB: ~

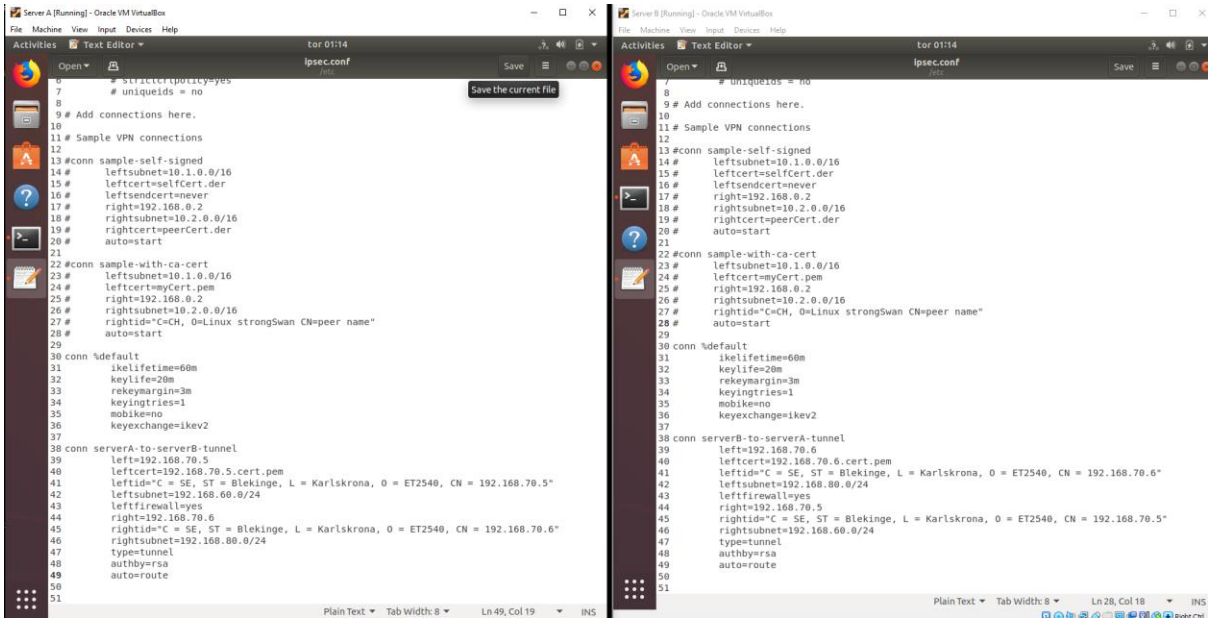

Server A [Running] - Oracle VM VirtualBox
Server B [Running] - Oracle VM VirtualBox
Capturing from enp0s8
Capturing from enp0s8

- Tested by pinging the private addresses of Server A (192.168.60.100) and Server B (192.168.80.100), respectively.

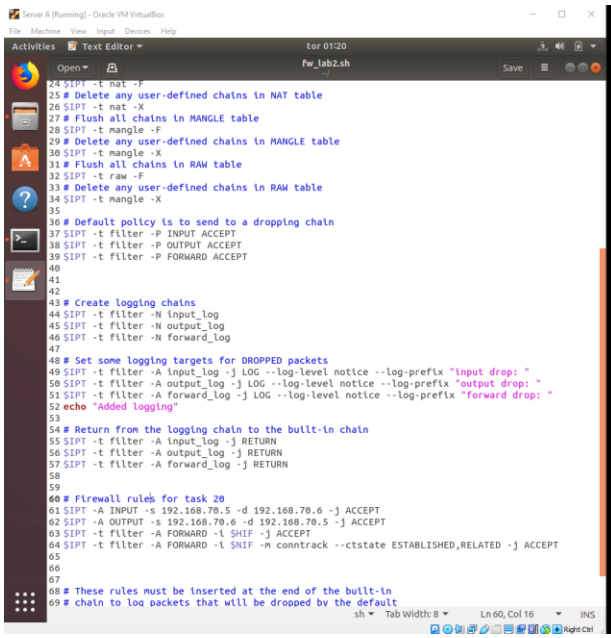


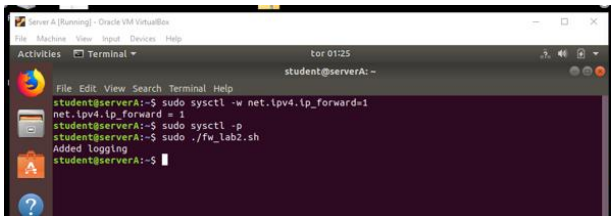
Task 20: Tunnel mode VPN with IP forwarding for client A and client B

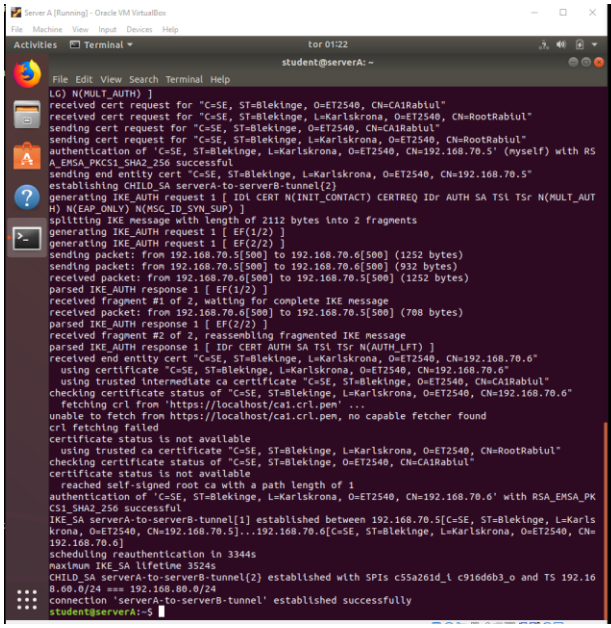
- For this task I edit ipsec.conf.



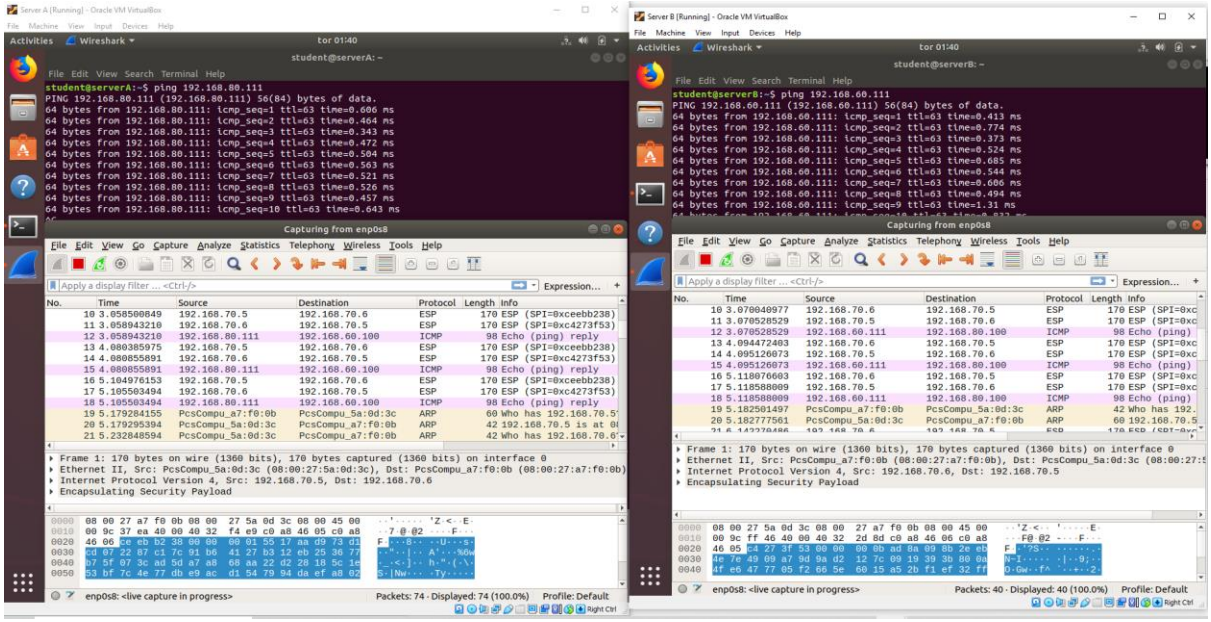
- Main part of this task added firewall rules under the section “# Firewall rules for task 20”



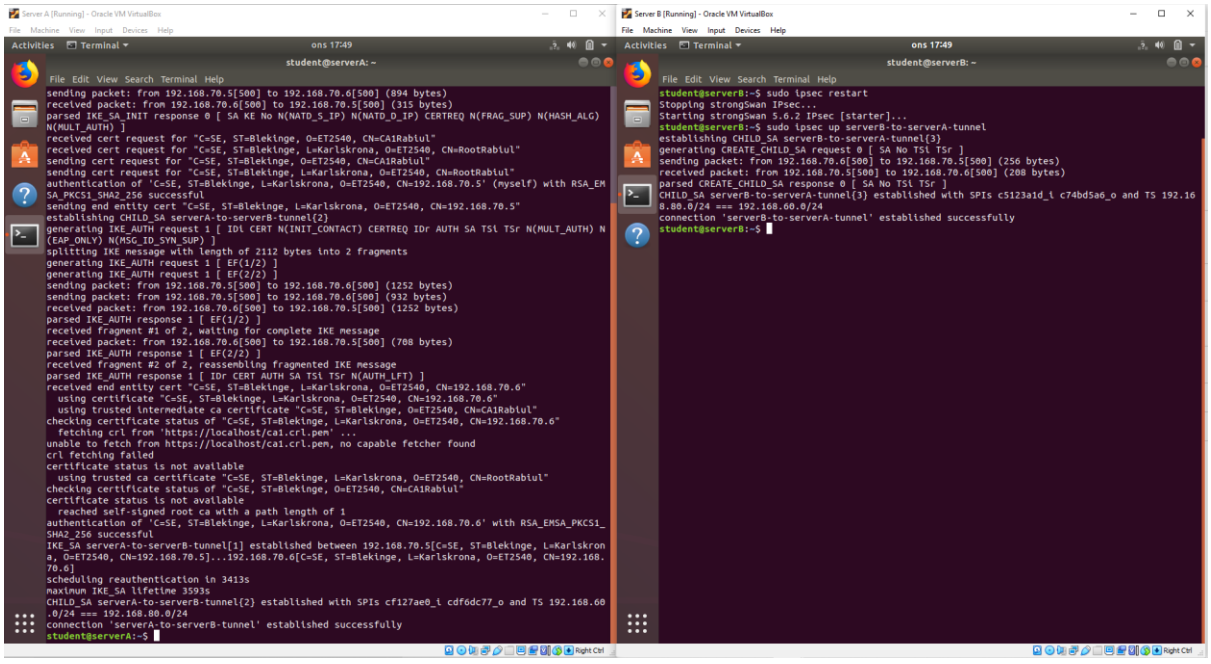




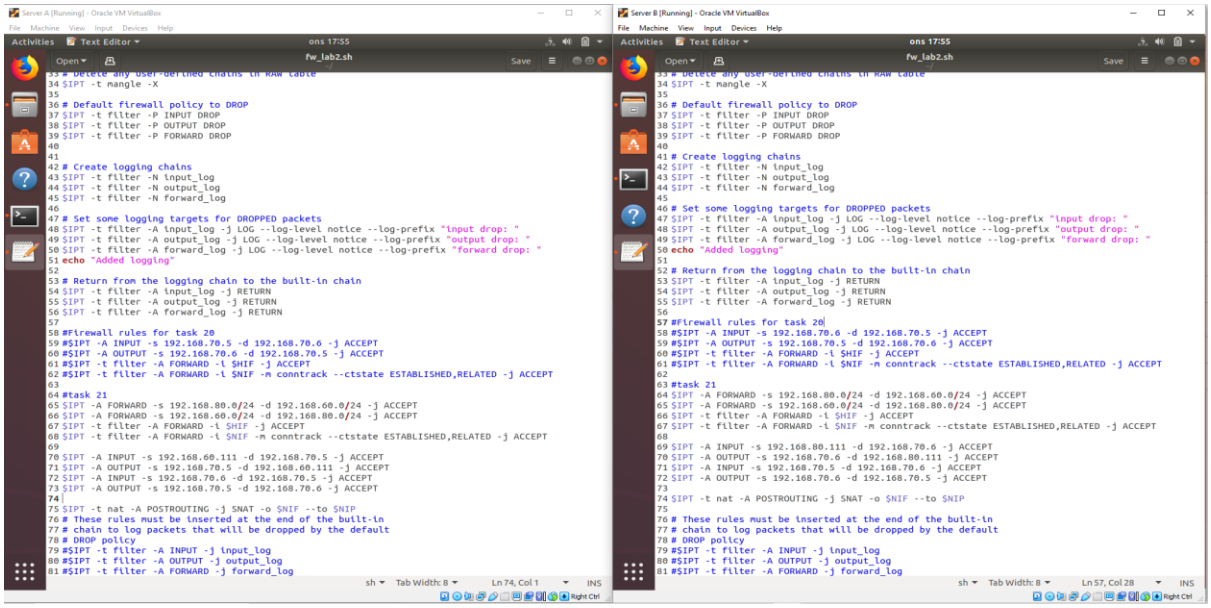
- As a evidence for this task ping from serverA to clientB and from serverB to clientA which is shown below screen short with wireshark traffic.



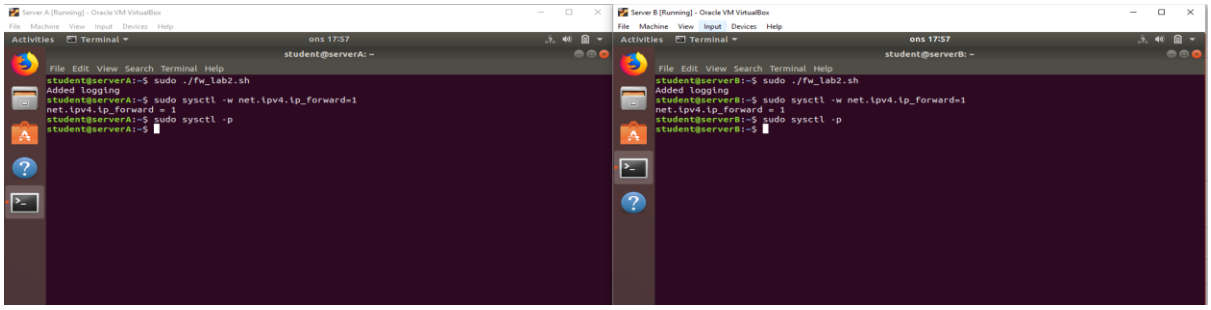
Task 21: Site A to Site B VPN with default DROP firewall rules



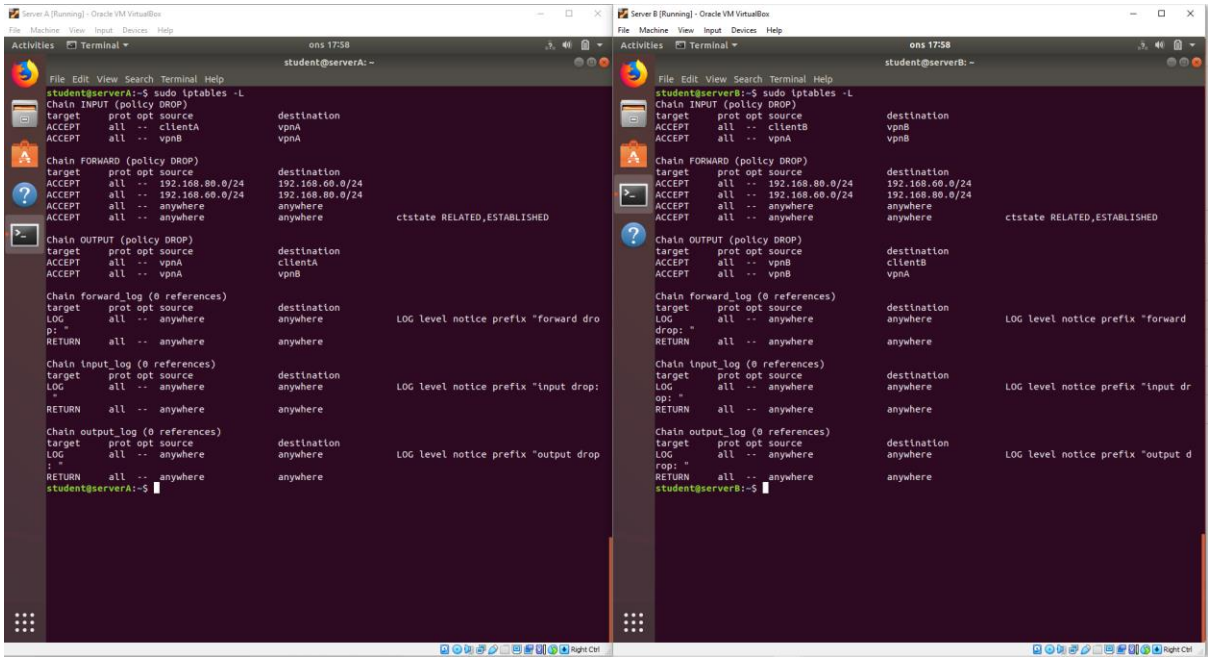
- I did not make any changes to ipsec.conf files. I just restarted strongSwan and enabled tunnel mode. Then I set the default firewall policies to DROP for both Server A and Server B. After that, I added the firewall rules for Server A and Server B mentioned in the screenshot below:



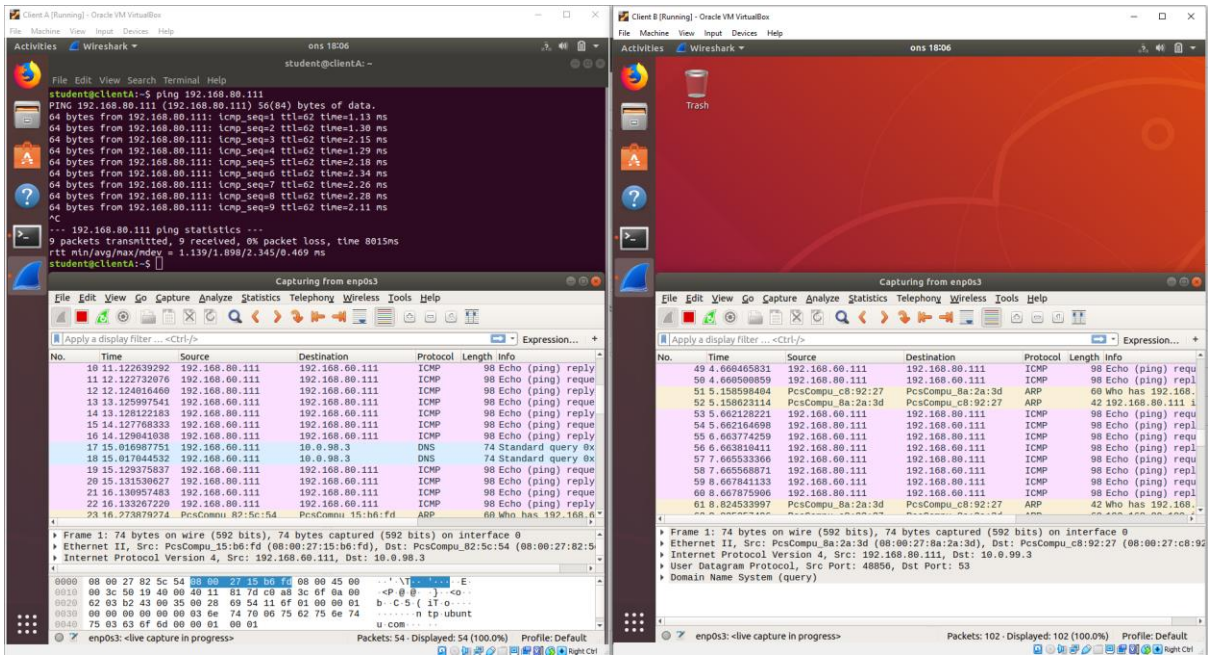
- Then I ran the firewall scripts and enabled forwarding.



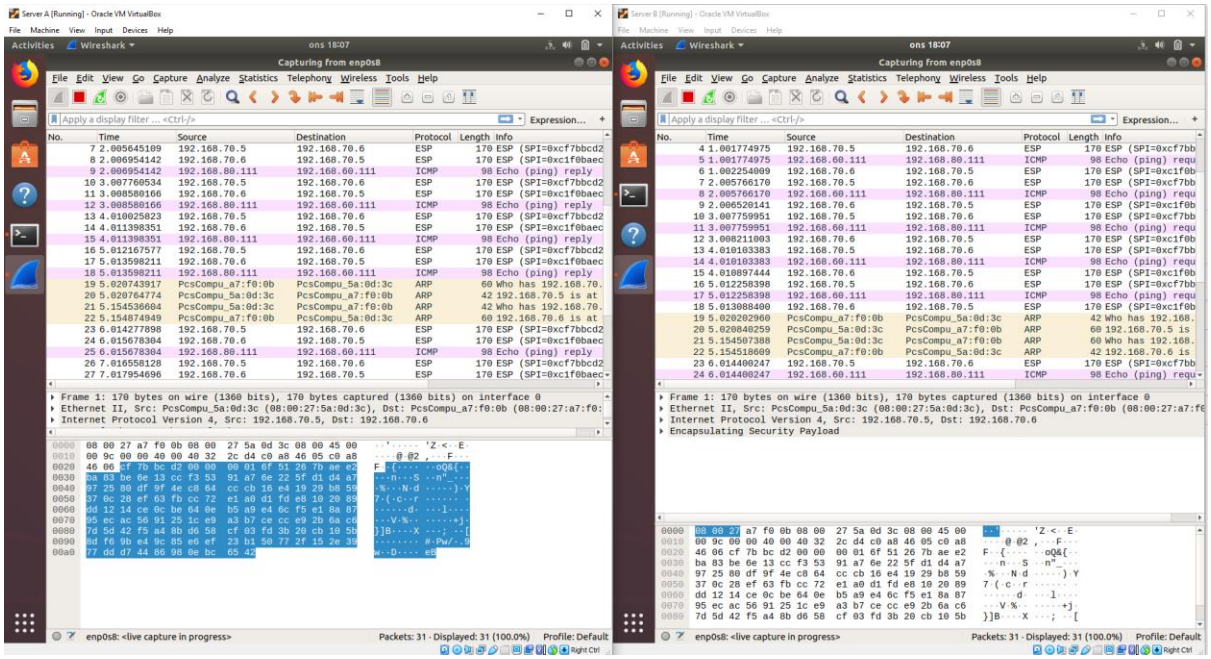
- The following is the iptables state after running the firewall script from server A and serverB.



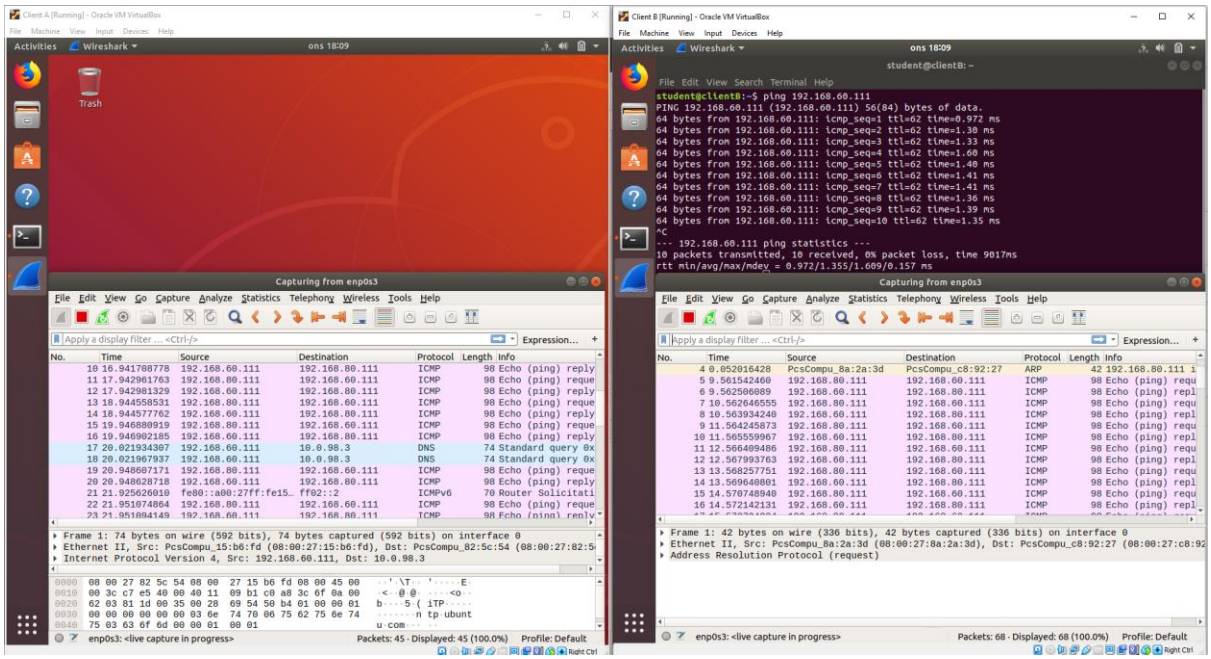
- Then I ping from Client A to Client B and it was working.



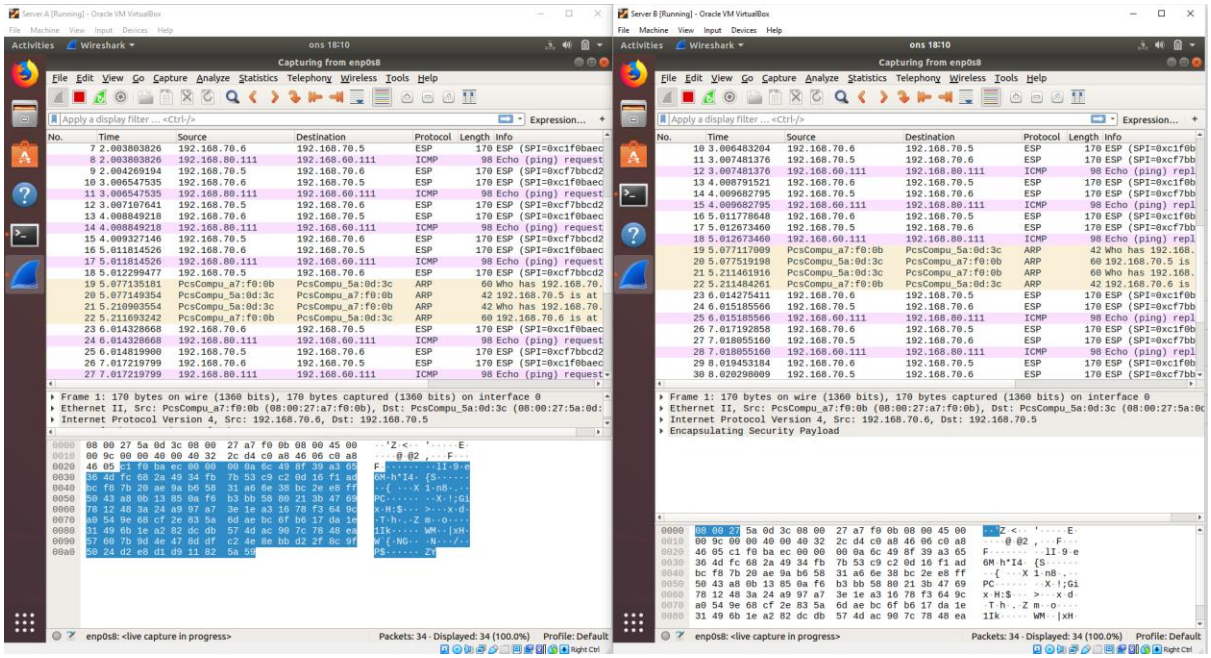
- At the same time, I observed the traffic in Server A and Server B. I found the traffic is going on over enp0s8 interface and it was clear that the traffic was going over the IPsec tunnel.



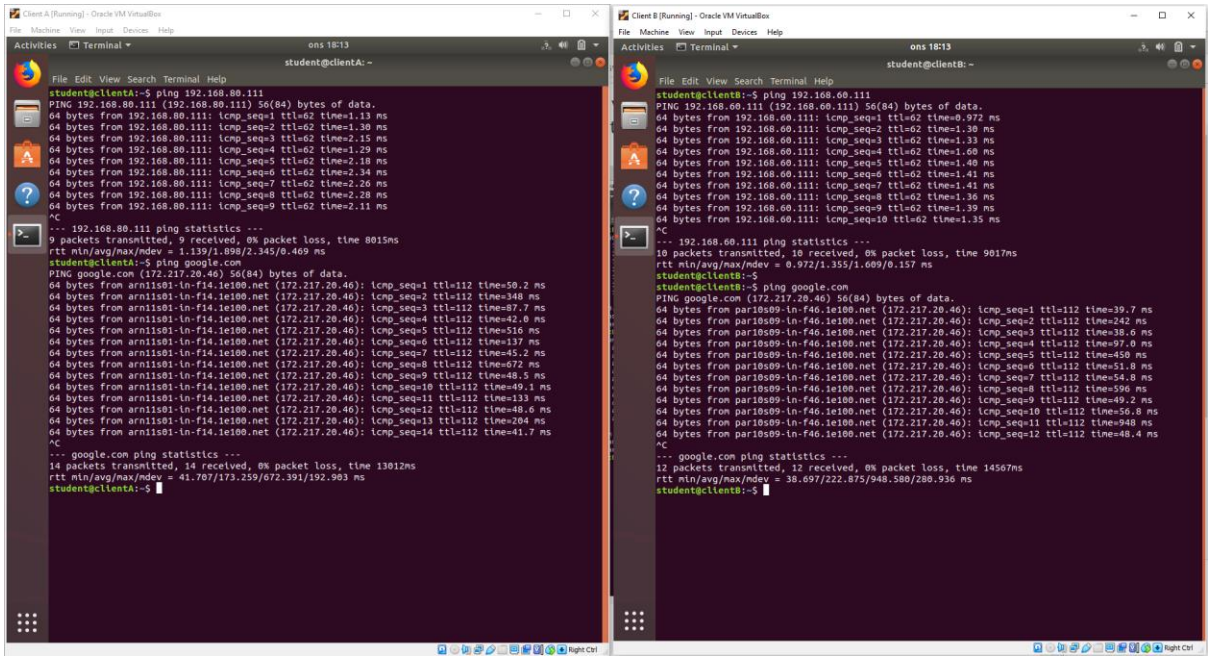
- Next, I ping from Client B to Client A and it was working.



- At the same time, I again observed the traffic in Server A and Server B. I found the traffic is going on over enp0s8 interface and it was clear that the traffic was going over the IPsec tunnel.



- Then I ping to Google from both Client A and Client B. It worked in both cases. Client A's traffic was forwarded by Server A to the NAT and this is how it accessed the internet. Same goes for Client B's to Server B.



Reference:

1. Lab 1.2: Certificates and VPN instructions
2. https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.13/gtps7/cfgcert.html
3. http://rcardon.free.fr/websign/download/api-x509ext/be/cardon/asn1/x509/extensions/KeyUsage.html
4. https://www.phildev.net/ssl/opensslconf.html
5. https://www.ibm.com/support/knowledgecenter/sl/SSKTMJ_9.0.1/admin/conf_keyusageextensions andextendedkeyusage_r.html
6. http://javadoc.iaik.tugraz.at/iaik_jce/old/iaik/x509/extensions/ExtendedKeyUsage.html
7. https://security.stackexchange.com/questions/84816/what-exactly-is-meant-by-serverauthandclientauth
8. https://jamielinux.com/docs/openssl-certificate-authority/create-the-root-pair.html
9. https://www.phildev.net/ssl/opensslconf.html
10. https://linux.die.net/man/1/openssl
11. https://www.openssl.org/docs/manmaster/man1/smime.html
13. http://www.firewall.cx/networking-topics/protocols/870-ipsec-modes.html
14. https://www.gypthecat.com/ipsec-vpn-host-to-host-on-ubuntu-14-04-with-strongswan?fbclid=IwAR1ZUnRqyNBr6ggXWzGOzH%20UL_k5LWZyKSxqhhKGcW0eZN4zSL4INdCvK00